ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN CLUSTER DE ALTA DISPONIBILIDAD PARA VIRTUALIZAR LOS SERVIDORES DE ADQUISICIÓN Y PROCESAMIENTO DE DATOS DEL INSTITUTO GEOFÍSICO

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

ACERO QUILUMBAQUÍN WILSON ARMANDO
wilson.acero@epn.edu.ec

DIRECTOR: JORGE ARTURO AGUILAR JARAMILLO
jorge.aguilar@epn.edu.ec

CODIRECTOR: RAÚL DAVID MEJÍA NAVARRETE
david.mejia@epn.edu.ec

Quito, Enero 2016

DECLARACIÓN

Yo , Wilson Armando Acero Quilumbaquín, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

Wilson A. Acero Quilumbaquín

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Wilson Armando Acero Quilumbaquín, bajo mi supervisión.

Fis. Jorge Aguilar, MSc.	**Ing. David Mejía, MSc.**
DIRECTOR DEL PROYECTO	**CODIRECTOR DEL PROYECTO**

AGRADECIMIENTOS

Agradezco a todos los profesores de la Facultad que me transfirieron su conocimiento durante mi vida académica.

Un sincero agradecimiento a mi codirector de proyecto, Máster David Mejía por su paciencia, su tiempo y su valiosa ayuda y orientación en la realización de este Proyecto.

Agradezco también al personal del Instituto Geofísico y en especial a mi director de Proyecto, Físico Jorge Aguilar por su paciencia y confianza en la realización del Proyecto.

Wilson Armando Acero Quilumbaquín

DEDICATORIA

A mis sobrinos, Alejandro y Paola, a mi hermana, a mis padres, a mis amigos y a las personas que confiaron en mí y que de alguna forma han contribuido a concluir el presente Proyecto.

Wilson Armando Acero Quilumbaquín

CONTENIDO

DECLARACIÓN ..... **I**

CERTIFICACIÓN ..... **II**

AGRADECIMIENTOS ..... **III**

DEDICATORIA ..... **IV**

CONTENIDO ..... **V**

LISTA DE TABLAS ..... **XIV**

LISTA DE FIGURAS ..... **XVI**

LISTA DE COMANDOS ..... **XIX**

LISTA DE ECUACIONES ..... **XXIV**

RESUMEN ..... **XXV**

PRESENTACION ..... **XXVII**

CAPÍTULO I **1**

1. FUNDAMENTOS TEÓRICOS **1**

1.1 INTRODUCCIÓN 1

1.2 ALTA DISPONIBILIDAD 1

1.2.1 DISPONIBILIDAD 2

1.2.2 MODELO DE LOS NUEVES 3

1.2.3 TIEMPO DE CAÍDA 3

1.2.4 CÁLCULO DE DISPONIBILIDAD 3

1.2.5 ALTA DISPONIBILIDAD 3

1.3 CLUSTER 4
1.3.1 CLUSTER DE ALTA DISPONIBILIDAD 4
1.3.2 COMPONENTES LÓGICOS DE UN CLUSTER DE ALTA DISPONIBILIDAD 6
1.3.2.1 Almacenamiento compartido 6
1.3.2.2 Software de administración del cluster 10
1.3.2.3 Software de administración de recursos del cluster 11
1.3.2.4 Agente de recursos 11
1.3.3 COMPONENTES FÍSICOS DEL CLUSTER 13
1.3.3.1 Nodos 13
1.3.3.2 Red de comunicaciones del cluster 13
1.3.3.3 Red pública 14
1.3.3.4 Sistema de almacenamiento compartido 14

1.4 SOLUCIONES DE ALTA DISPONIBILIDAD DE CÓDIGO ABIERTO 14
1.4.1 RED HAT HIGH AVAILABILITY 14
1.4.1.1 Almacenamiento compartido 14
1.4.1.2 Software de administración del cluster 16
1.4.1.3 Software de administración de recursos 19
1.4.1.4 Agentes de recursos 20
1.4.2 SUSE LINUX ENTERPRISE SERVER HIGH AVAILABILITY (SLES) 21
1.4.2.1 Almacenamiento compartido 22
1.4.2.2 Software de administración del cluster 22
1.4.2.3 Software de administración de recursos 23
1.4.2.4 Agente de recursos 23
1.4.3 ORACLE CLUSTERWARE 24
1.4.3.1 Almacenamiento compartido 25
1.4.3.2 Software de Administración del cluster 26
1.4.3.3 Software de Administración de recursos 26

1.4.3.4 Agente de recursos 26
1.4.4 COMPARACIÓN DE LAS SOLUCIONES PRESENTADAS 27

1.5 REVISIÓN DEL SOFTWARE DE CLUSTERING ELEGIDO 30
1.5.1 CONCEPTOS EMPLEADOS EN PACEMAKER Y COROSYNC 30
1.5.1.1 Split-brain 30
1.5.1.2 Quórum 30
1.5.1.3 Virtual Synchrony 32
1.5.1.4 Totem 32
1.5.1.5 Fencing 32
1.5.2 COROSYNC 33
1.5.2.1 Características 33
1.5.3 PACEMAKER 34
1.5.3.1 Características 34
1.5.3.2 Componentes 34
1.5.3.3 Tipos de cluster con Pacemaker 37
1.5.4 AGENTE DE RECURSOS 38
1.5.5 RECURSOS 39
1.5.5.1 Propiedades generales de un recurso 39
1.5.5.2 Propiedades específicas de un recurso 40
1.5.5.3 Opciones de recursos 40
1.5.5.4 Operaciones de monitorización de un recurso 42
1.5.5.5 Restricciones de recursos 43
1.5.5.6 Tipos de recursos 44
1.5.6 PACEMAKER REMOTE 46
1.5.7 PROPIEDADES DEL CLUSTER 47
1.5.7.1 No-quorum-policy 48
1.5.7.2 Symmetric-cluster 48
1.5.7.3 Stonith-enabled 48
1.5.7.4 Stonith-action 49
1.5.8 HERRAMIENTAS DE CONFIGURACIÓN DE PACEMAKER: PCS 49

1.6 TECNOLOGÍAS DE ALMACENAMIENTO 49
1.6.1 RAID (REDUNDANT ARRAY OF INDEPENDENT DISK) 49
1.6.1.1 RAID implementado mediante software 50
1.6.1.2 RAID implementado mediante hardware 50
1.6.1.3 Niveles de RAID 50
1.6.2 DRBD 54
1.6.2.1 Características 54
1.6.2.2 Funcionamiento 54
1.6.2.3 Tipos de sincronización de datos en DRBD 55
1.6.3 ISCSI 56
1.6.3.1 Target iSCSI 56
1.6.3.2 Initiator iSCSI 58
1.6.4 SISTEMA DE ARCHIVOS PARA CLUSTER 59
1.6.4.1 OCFS2 [33] 59
1.6.4.2 GFS2 [34] 60
1.6.5 CLVM 61

1.7 VIRTUALIZACIÓN 62
1.7.1 VIRTUALIZACIÓN DE RED 62
1.7.2 VIRTUALIZACIÓN DE ALMACENAMIENTO 62
1.7.3 VIRTUALIZACIÓN DE SERVIDORES 63
1.7.3.1 Hipervisor 64
1.7.3.2 Virtualización con hipervisor de tipo 2 o virtualización de sistema operativo 64
1.7.3.3 Virtualización con hipervisor de tipo 1 65
1.7.4 SOLUCIONES DE VIRTUALIZACIÓN DE CÓDIGO ABIERTO 66
1.7.4.1 KVM 66
1.7.4.2 XEN 68
1.7.4.3 QEMU 70
1.7.5 COMPARACIÓN DE LAS SOLUCIONES PRESENTADAS 71

1.8 REVISIÓN DEL SOFTWARE DE VIRTUALIZACIÓN SELECCIONADA..... 72
1.8.1 LIBVIRT 72
1.8.2 VIRSH 74
1.8.2.1 Virt-install 74
1.8.3 RED VIRTUAL 75
1.8.3.1 Modo NAT 75
1.8.3.2 Modo Bridge 76
1.8.3.3 Modo Aislado 76
1.8.4 OPERACIONES CON MÁQUINAS VIRTUALES 76
1.8.4.1 Creación de un servidor virtual 76
1.8.4.2 Administración del servidor virtual 76
1.8.4.3 Migración de un servidor virtual 79

CAPÍTULO II 80

2. SITUACIÓN ACTUAL Y ANÁLISIS DE REQUERIMIENTOS 80

2.1 INSTITUTO GEOFÍSICO DE LA ESCUELA POLITÉCNICA NACIONAL..... 81

2.2 TÉRMINOS UTILIZADOS EN SISMOLOGÍA 81

2.3 SISTEMAS DE ADQUISICIÓN Y/O PROCESAMIENTO DEL INSTITUTO GEOFÍSICO 82
2.3.1 SEISCOMP3 83
2.3.1.1 Introducción 83
2.3.1.2 Características 83
2.3.1.3 Módulos de comunicación 84
2.3.1.4 Módulos de adquisición y almacenamiento 85
2.3.1.5 Módulos de procesamiento 87
2.3.1.6 Módulos de diseminación 90
2.3.1.7 Módulos para análisis 91
2.3.1.8 Base de datos 94

2.3.2 EARTHWORM [55][56] 94
2.3.2.1 Introducción 94
2.3.2.2 Características 94
2.3.2.3 Módulos de administración 95
2.3.2.4 Módulos de comunicación 96
2.3.2.5 Módulos para adquisición de datos 96
2.3.2.6 Módulos de procesamiento 97
2.3.2.7 Módulos de visualización y almacenamiento 97
2.3.3 SHAKEMAP 98
2.3.3.1 Introducción 98
2.3.3.2 Características 99
2.3.3.3 Componentes y funcionamiento de ShakeMap 100

2.4 ANÁLISIS DE REQUERIMIENTOS 102
2.4.1 SEISCOMP3 102
2.4.1.1 Estimación de uso del hardware del servidor SeisComP3 103
2.4.1.2 Módulos a administrar 107
2.4.2 EARTHWORM 108
2.4.2.1 Estimación del uso del hardware del servidor físico Earthworm 108
2.4.2.2 Módulos a administrar 111
2.4.3 SHAKEMAP 112
2.4.3.1 Módulos a administrar 112
2.4.4 REQUERIMIENTOS PARA EL CLUSTER DE ALTA DISPONIBILIDAD 113
2.4.4.1 Tamaño del almacenamiento compartido 113
2.4.4.2 Requerimientos físicos de los nodos del cluster 115
2.4.5 REQUISITOS ADICIONALES DE LOS NODOS DEL CLUSTER 117
2.4.5.1 Tamaño del disco duro local 117
2.4.5.2 Características de procesador 117
2.4.5.3 Características de red 117

CAPÍTULO III 119

3. DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN, PRUEBAS, RESULTADOS Y COSTOS 119

3.1 INTRODUCCIÓN..... 119

3.2 DISEÑO DEL CLUSTER DE ALTA DISPONIBILIDAD..... 119
3.2.1 ELECCIÓN DEL TIPO DE CLUSTER..... 119
3.2.2 ELECCIÓN DEL SISTEMA OPERATIVO 120
3.2.3 DISEÑO DE LA RED PARA EL CLUSTER..... 121
3.2.4 ELECCIÓN DEL DISPOSITIVO DE FENCING..... 122
3.2.5 ELECCIÓN DEL SISTEMA DE ARCHIVOS DEL ALMACENAMIENTO COMPARTIDO 123
3.2.6 CARACTERÍSTICAS DE LOS NODOS DEL CLUSTER..... 123

3.3 DISEÑO DEL ALMACENAMIENTO COMPARTIDO 124
3.3.1 ELECCIÓN DE LA TECNOLOGÍA DE REPLICACIÓN..... 125
3.3.2 DETALLES DE LA CAPA FÍSICA 127
3.3.3 ELECCIÓN DEL TIPO DE REDUNDANCIA DE CAPA FÍSICA 127
3.3.4 DETALLES DE LA TECNOLOGÍA DE ACCESO..... 129
3.3.5 DETALLES DEL SISTEMA DE ARCHIVOS SELECCIONADO 129
3.3.6 DISEÑO DEL CLUSTER DE ALMACENAMIENTO 130
3.3.6.1 Elección del tipo de cluster 130
3.3.6.2 Requisitos de los nodos del cluster de almacenamiento 131

3.4 IMPLEMENTACIÓN DEL CLUSTER DE ALMACENAMIENTO 131
3.4.1 INSTALACIÓN DEL SISTEMA OPERATIVO..... 132
3.4.2 INSTALACIÓN DE NTP 133
3.4.3 CONFIGURACIÓN DE SEGURIDAD 133
3.4.4 CONFIGURACIÓN DE LAS INTERFACES DE RED..... 134
3.4.5 INSTALACIÓN DEL SOFTWARE DEL CLUSTER 137
3.4.6 CONFIGURACIÓN DEL CLUSTER DE ALMACENAMIENTO 141
3.4.6.1 Agregar un dispositivo de fencing al cluster..... 141

3.4.6.2 Activar el encendido automático en los nodos 145
3.4.7 CREACIÓN DEL RAID 1 146
3.4.7.1 Creación de particiones 148
3.4.7.2 Creación del RAID 149
3.4.8 CREACIÓN DEL DISPOSITIVO DRBD 151
3.4.8.1 Diseño del dispositivo DRBD 151
3.4.8.2 Instalación 153
3.4.8.3 Configuración 154
3.4.8.4 Creación del dispositivo DRBD 156
3.4.8.5 Agregar el dispositivo DRBD al cluster 158
3.4.9 CREACIÓN DEL TARGET ISCSI 159
3.4.9.1 Instalación del software targetcli 159
3.4.9.2 Creación del target iSCSI 160

3.5 IMPLEMENTACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD 162
3.5.1 CONFIGURACIÓN DE LA RED PARA LAS MÁQUINAS VIRTUALES .. 163
3.5.2 CONFIGURACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD 164
3.5.2.1 Interfaz web de pcs 165
3.5.3 AGREGAR EL ALMACENAMIENTO COMPARTIDO COMO UN RECURSO 167
3.5.3.1 Instalación de software para el sistema de archivos tipo cluster 168
3.5.4 CREAR EL SISTEMA DE ARCHIVOS GFS2 172
3.5.5 INSTALACIÓN DE LA PLATAFORMA DE VIRTUALIZACIÓN KVM 174
3.5.6 CONFIGURACIÓN DE LA MIGRACIÓN EN CALIENTE 175
3.5.7 CREACIÓN DE LAS MÁQUINAS VIRTUALES 175
3.5.7.1 Servidor virtual seisc 176
3.5.7.2 Servidor virtual earth 181
3.5.7.3 Servidor virtual shake 182
3.5.8 INSTALACIÓN DE LOS SISTEMAS DE ADQUISICIÓN COMO RECURSOS CON ALTA DISPONIBILIDAD 183

3.6 PRUEBAS 188
3.6.1 PRUEBA DEL ALMACENAMIENTO COMPARTIDO 188
3.6.1.1 Prueba del disco RAID 1 implementado por software..... 188
3.6.1.2 Pruebas del dispositivo DRBD y del target iSCSI 191
3.6.1.3 Pruebas de escritura..... 195
3.6.2 PRUEBAS DEL CLUSTER DE ALTA DISPONIBILIDAD..... 196
3.6.2.1 Prueba de migración en caliente..... 196
3.6.2.2 Prueba de recuperación ante el fallo de un nodo físico 198
3.6.2.3 Prueba de alta disponibilidad de los sistemas de adquisición y procesamiento 201
3.6.3 PRUEBAS DE LA RED DEL CLUSTER 205

3.7 COSTOS REFERENCIALES..... 207

4. CONCLUSIONES Y RECOMENDACIONES..... 209

4.1 CONCLUSIONES..... 209

4.2 RECOMENDACIONES 213

REFERENCIAS BIBLIOGRÁFICAS..... 217

ANEXOS 226

LISTA DE TABLAS

Tabla 1.1. Valores de disponibilidad y su equivalencia en tiempo 2
Tabla 1.2. Modelo de cuatro capas para la solución de alta disponibilidad de Red Hat 15
Tabla 1.3. Modelo de cuatro capas para la solución de alta disponibilidad de SLES 21
Tabla 1.4. Modelo de cuatro capas para la solución de alta disponibilidad de Oracle 24
Tabla 1.5. Comparación de las soluciones de *clustering* 28
Tabla 1.6. Capas del software de clustering y alta disponibilidad seleccionado 29
Tabla 1.7. Propiedades de un recurso 39
Tabla 1.8. Opciones de un recurso 41
Tabla 1.9. Propiedades de una operación de monitoreo 43
Tabla 1.10. Propiedades de un cluster Pacemaker 47
Tabla 1.11. Comparación de las soluciones de virtualización 73
Tabla 1.12. Opciones principales del comando virt-install 77
Tabla 2.1. Información del hardware del servidor SeisComP3 103
Tabla 2.2. Requerimientos del hardware virtualizado para el sistema SeisComP3. 107
Tabla 2.3. Componentes del sistema SeisComP3 a administrar 107
Tabla 2.4. Información del hardware del servidor Earthworm 108
Tabla 2.5. Requerimientos del hardware virtualizado para el sistema Earthworm .. 111
Tabla 2.6. Componentes de Earthworm a monitorizar 111
Tabla 2.7. Información del hardware del servidor ShakeMap 112
Tabla 2.8. Componentes de ShakeMap a monitorizar 113
Tabla 2.9. Espacio requerido por el sistema SeisComP3 114
Tabla 2.10. Espacio requerido para el sistema Earthworm 115
Tabla 2.11. Memoria RAM requerida para un nodo del cluster 116
Tabla 2.12. Número de procesadores necesarios 116
Tabla 2.13. Requisitos físicos de un nodo del cluster 118
Tabla 3.1. Tiempo de soporte de sistemas operativos Linux 120
Tabla 3.2. Características de los servidores físicos 124

Tabla 3.3. Capas de un almacenamiento compartido replicado con alta disponibilidad 125
Tabla 3.4. Parámetros del sistema de archivos GFS2 130
Tabla 3.5. Datos de configuración para el dispositivo de NPS-115..... 144
Tabla 3.6. Propiedades del target iSCSI 160
Tabla 3.7. Información del cluster clusterwa 162
Tabla 3.8. Propiedades del servidor virtual seisc 178
Tabla 3.9. Velocidades de escritura de los niveles del almacenamiento compartido 195
Tabla 3.10. Costos referenciales del proyecto 207

LISTA DE FIGURAS

Figura 1.1. Cluster de alta disponibilidad básico 5
Figura 1.2. Componentes lógicos de un cluster de alta disponibilidad 7
Figura 1.3. Cluster con almacenamiento del tipo SIS..... 9
Figura 1.4. Cluster de alta disponibilidad con almacenamiento replicado 11
Figura 1.5. Capa de almacenamiento de un cluster Red Hat 17
Figura 1.6. Software de Administración del Cluster 19
Figura 1.7. Dominios de *failover* de RGManager 20
Figura 1.8. Ejemplo de recurso de tipo servicio web 21
Figura 1.9. *Stack* del cluster de alta disponibilidad desarrollado por Oracle 25
Figura 1.10. Funcionamiento de ASM 25
Figura 1.11. Arquitectura de Pacemaker 35
Figura 1.12. Archivo CIB de un cluster de prueba..... 37
Figura 1.13. Diagrama de RAID 1+0 53
Figura 1.14. Formato para crear un nombre IQN 58
Figura 1.15. Esquema de virtualización de almacenamiento 63
Figura 1.16. Hipervisor de tipo 2 64
Figura 1.17. Hipervisor de tipo 1 65
Figura 1.18. Arquitectura de KVM 67
Figura 1.19. Componentes del hipervisor Xen 69
Figura 1.20. Funcionamiento Básico de QEMU 71
Figura 1.21. Capas de un servidor virtual KVM 75
Figura 2.1. Solicitud de conexión a los grupos de SeisComP3 86
Figura 2.2. Módulos de adquisición y almacenamiento 88
Figura 2.3. Funcionamiento de los módulos de procesamiento de SeisComP3 90
Figura 2.4. Diagrama del funcionamiento del módulo GDS 92
Figura 2.5. Módulo de visualización de formas de onda Scrttv 92
Figura 2.6. Ventana principal del módulo scolv 93
Figura 2.7. Esquema del funcionamiento de Earthworm 95
Figura 2.8. ShakeMap del sismo ocurrido en Quito el 2014-08-12..... 99

Figura 2.9. Uso del CPU del servidor SeisComP3 104
Figura 2.10. Uso de memoria RAM en el servidor SeisComP3 primario 105
Figura 2.11. Uso de la interfaz de red eth0 del servidor SeisComP3 105
Figura 2.12. Utilización del disco en el sistema SeisComP3 106
Figura 2.13. Utilización del CPU en el sistema Earthworm 109
Figura 2.14. Utilización de la memoria RAM en el servidor Earthworm..... 109
Figura 2.15. Utilización de la interfaz de Ethernet en el servidor Earthworm 110
Figura 2.16. Utilización del disco duro en el servidor Earthworm 110
Figura 3.1. Diagrama de red para cluster de alta disponibilidad..... 121
Figura 3.2. Comparación del desempeño de escritura de niveles de RAID 128
Figura 3.3. Comparación del desempeño de lectura de niveles de RAID 129
Figura 3.4. Diagrama del cluster de almacenamiento a implementar..... 132
Figura 3.5. Conexión inicial del dispositivo NPS-115 143
Figura 3.6. Ingresar al portal de configuración del cluster 165
Figura 3.7. Sección de administración de cluster 165
Figura 3.8. Agregar un cluster para administrar 166
Figura 3.9. Cluster clusterwa agregado al portal de administración 166
Figura 3.10. Sección de configuración del cluster 167
Figura 3.11 Conexión al servidor VNC del nodo1 177
Figura 3.12 Estado de los nodos físicos y nodos remotos del cluster 187
Figura 3.13 Estado de los recursos que el cluster ejecuta 188
Figura 3.14. Detección del error y cambio de rol DRBD secundario a rol DRBD primario 192
Figura 3.15. Estado del cluster de almacenamiento luego del fallo del nodo iscs1. 192
Figura 3.16. Detección del error y recuperación de la LUN iSCSI en los nodos del cluster de alta disponibilidad 193
Figura 3.17. Prueba de conectividad con el comando ping al servidor virtual earth 193
Figura 3.18. Detección de una situación split-brain..... 194
Figura 3.19. Resultado de la recuperación del split-brain 195
Figura 3.20. Proceso de migración de la máquina virtual earth..... 198
Figura 3.21. Test de conectividad con el servidor seisc 199

Figura 3.22 Proceso de fencing del nodo3 200
Figura 3.23 Recuperación del servidor virtual shake luego del fallo del servidor físico nodo3 200
Figura 3.24 Test de conectividad durante la recuperación de shake 201
Figura 3.25 Proceso de recuperación de los servicios mysql y web en el servidor virtual shake 201
Figura 3.26 Proceso de migración del recurso ipsc3..... 202
Figura 3.27 Reconexión del módulo scrttv 203
Figura 3.28 Funcionamiento ininterrumpido de scrttv durante la migración de Seedlink 203
Figura 3.29 Proceso de migración del recurso ipew..... 204
Figura 3.30 Funcionamiento ininterrumpido de Swarm durante la prueba de alta disponibilidad del sistema Earthworm 204
Figura 3.31 Resultado del test de redundancia del conmutador en el nodo1..... 205
Figura 3.32 Resultado del test de redundancia del conmutador en el nodo2..... 205
Figura 3.33 Resultado del test de redundancia del conmutador en el nodo3..... 205
Figura 3.34 Prueba de conectividad con el comando ping al servidor nodo1 206
Figura 3.35 Resultado del test realizado a la interfaz enlazada bond0 206
Figura 3.36 Cambio de estado de la interfaz de enlazada bond0 207

LISTA DE LÍNEA DE COMANDOS

Línea de Comandos 1.1. Comandos para ver los estándares y proveedores disponibles 40
Línea de Comandos 1.2. Descripción de las propiedades específicas del recurso IP 41
Línea de Comandos 1.3. Identificador IQN de un nodo iSCSI initiator 59
Línea de Comandos 1.4. Creación de una máquina virtual empleando virt-install 74
Línea de Comandos 1.5. Presentar las máquinas virtuales en ejecución 77
Línea de Comandos 1.6. Encender un servidor virtual 78
Línea de Comandos 1.7. Encender un servidor virtual a partir de un archivo XML ... 78
Línea de Comandos 1.8. Apagar un servidor virtual 78
Línea de Comandos 1.9. Forzar el apagado de un servidor virtual 78
Línea de Comandos 1.10. Comando para migrar en caliente una máquina virtual ... 79
Línea de Comandos 3.1. Instalar y activar el servicio NTP 133
Línea de Comandos 3.2. Deshabilitar el cortafuegos incluido en CentOS 7 134
Línea de Comandos 3.3. Comando para ver las interfaces disponibles en el servidor iscs1 134
Línea de Comandos 3.4. Creación de los archivos de configuración de las interfaces de red 135
Línea de Comandos 3.5. Creación del archivo de configuración de la interfaz de red bond0 135
Línea de Comandos 3.6. Comando para crear y cargar la interfaz bond0 137
Línea de Comandos 3.7. Comando para revisar el funcionamiento de la interfaz bond0 137
Línea de Comandos 3.8. Instalación del software requerido para el cluster de alta disponibilidad..... 137
Línea de Comandos 3.9. Habilitar el programa pcs como un servicio..... 138
Línea de Comandos 3.10. Cambio de clave del usuario hacluster..... 138
Línea de Comandos 3.11. Autenticación del usuario hacluster en los nodos del cluster de almacenamiento 138

Línea de Comandos 3.12. Creación del cluster “clusteralmawa” formado por iscs1 y iscs2 139
Línea de Comandos 3.13. Comando para iniciar el cluster 141
Línea de Comandos 3.14. Comando para arrancar el cluster junto con el sistema operativo..... 141
Línea de Comandos 3.15. Comando para verificar el estado del cluster 142
Línea de Comandos 3.16. Configuración de la clave de acceso al dispositivo NPS-115 144
Línea de Comandos 3.17. Configuración de la interfaz de red del dispositivo NPS-115 145
Línea de Comandos 3.18. Instalación del agente para el dispositivo NPS-115 145
Línea de Comandos 3.19. Agregar el dispositivo NPS-115 al cluster 146
Línea de Comandos 3.20. Comando para probar el funcionamiento del fencing 146
Línea de Comandos 3.21. Comando para verificar el soporte para el módulo de RAID 1 147
Línea de Comandos 3.22. Comando para presentar la información de los discos disponibles en cada nodo..... 147
Línea de Comandos 3.23. Comando para cargar el módulo md en el kernel 148
Línea de Comandos 3.24. Procedimiento para crear la partición para el disco RAID 1 148
Línea de Comandos 3.25. Creación del disco RAID 1 149
Línea de Comandos 3.26. Verificación del disco RAID 150
Línea de Comandos 3.27. Test para determinar la velocidad de escritura en el disco 151
Línea de Comandos 3.28. Test para determinar la velocidad de la red 152
Línea de Comandos 3.29. Agregar el repositorio para instalar DRBD 153
Línea de Comandos 3.30. Instalar el software DRBD 154
Línea de Comandos 3.31. Comando para verificar la configuración del dispositivo drbd0 156
Línea de Comandos 3.32. Comandos para crear el dispositivo drbd0 156

Línea de Comandos 3.33. Comando para monitorizar el estado del dispositivo drbd0 157
Línea de Comandos 3.34. Comando para sincronizar los datos entre los nodos ... 157
Línea de Comandos 3.35. Sincronización de los nodos DRBD..... 157
Línea de Comandos 3.36. Agregar al recurso drbdwa al cluster..... 158
Línea de Comandos 3.37. Comando para ver el estado del recurso drbd_wa..... 159
Línea de Comandos 3.38. Comando para instalar el software del target iSCSI..... 159
Línea de Comandos 3.39. Comando para agregar la dirección IP como recurso del cluster..... 160
Línea de Comandos 3.40. Comando para agregar el recurso iscsiIP al grupo grupiSCSI..... 160
Línea de Comandos 3.41. Agregar una restricción de colocación al grupo de recursos grupiSCSI 161
Línea de Comandos 3.42. Comando para crear el recurso target iSCSI en el grupo grupiSCSI..... 161
Línea de Comandos 3.43. Comando para crear el recurso iscsLUN y agregarlo al grupo grupiSCSI..... 161
Línea de Comandos 3.44. Comando para crear la restricción de orden para el grupo grupiSCSI y el recurso masterdrbd 162
Línea de Comandos 3.45. Creación del archivo de configuración de la interfaz enlazada bond1..... 163
Línea de Comandos 3.46. Creación del archivo de configuración de la interfaz enlazada br0..... 164
Línea de Comandos 3.47. Comando para instalar el software del initiator iSCSI ... 167
Línea de Comandos 3.48. Comando para agregar la LUN iSCSI como un recurso 168
Línea de Comandos 3.49. Instalación del software necesario para GFS2..... 168
Línea de Comandos 3.50. Comando para crear el recurso dlmwa 169
Línea de Comandos 3.51. Comando para agregar las restricciones para el recurso dlmwa..... 169
Línea de Comandos 3.52. Comando para habilitar el modo cluster de LVM 169
Línea de Comandos 3.53. Comando para crear el recurso clvmwa..... 169

Línea de Comandos 3.54. Comando para agregar las restricciones para el recurso clvmwa 170
Línea de Comandos 3.55. Comando para crear una partición en la LUN iSCSI (/dev/sdb) 170
Línea de Comandos 3.56. Comando para reiniciar el recurso iscsiwa 170
Línea de Comandos 3.57. Comando para crear el volumen físico empleando la LUN iSCSI 171
Línea de Comandos 3.58. Comando para mostrar los volúmenes físicos disponibles 171
Línea de Comandos 3.59. Comando para crear el grupo de volúmenes vgiscsiwa 171
Línea de Comandos 3.60. Comando para crear el volumen lógico lviscsiwa 171
Línea de Comandos 3.61. Comando para dar formato al volumen lógico lviscsiwa con el sistema de archivos GFS2..... 172
Línea de Comandos 3.62. Comando para crear el recurso gfs2wa 173
Línea de Comandos 3.63. Comandos para agregar las restricciones de orden y colocación del recurso gfs2wa 173
Línea de Comandos 3.64. Comando para configurar la propiedad del cluster no-quorum-policy..... 173
Línea de Comandos 3.65. Verificar que los nodos soportan la virtualización asistida por hardware 174
Línea de Comandos 3.66. Verificar que el kernel soporta la virtualización con KVM 174
Línea de Comandos 3.67. Instalar el software de virtualización necesario 174
Línea de Comandos 3.68. Comandos para compartir las llaves SSH..... 175
Línea de Comandos 3.69. Comando para crear el servidor virtual seisc 176
Línea de Comandos 3.70. Comando para agregar la dirección IP del servidor virtual seisc a la lista de /etc/hosts..... 178
Línea de Comandos 3.71. Comando para crear el servidor virtual seisc 178
Línea de Comandos 3.72. Comando para crear los discos virtuales seisc2 y seisc3 178
Línea de Comandos 3.73. Generación y copia de la clave de autenticación 180

Línea de Comandos 3.74. Instalación del software de administración de Pacemaker remoto 180
Línea de Comandos 3.75. Instalación del software de administración de Pacemaker remoto 181
Línea de Comandos 3.76. Creación del servidor virtual Earthworm..... 182
Línea de Comandos 3.77. Creación del servidor recurso vmearth..... 182
Línea de Comandos 3.78. Creación del servidor virtual shake 183
Línea de Comandos 3.79. Creación y configuración del recurso ipsc3..... 184
Línea de Comandos 3.80. Creación del recurso spread 185
Línea de Comandos 3.81. Creación del recurso scmaster..... 185
Línea de Comandos 3.82. Creación y configuración del recurso ipew..... 186
Línea de Comandos 3.83. Creación y configuración de los recursos del sistema ShakeMap 187
Línea de Comandos 3.84. Comando para simular el fallo en un disco del arreglo RAID1..... 189
Línea de Comandos 3.85. Estado del disco RAID1 detectado por el sistema operativo 189
Línea de Comandos 3.86. Estado del dispositivo DRBD 189
Línea de Comandos 3.87. Estado del disco RAID 1 190
Línea de Comandos 3.88. Remover y agregar el disco /dev/sdc1 al RAID 1..... 190
Línea de Comandos 3.89. Procedimiento para la recuperación de una situación DRBD split-brain..... 194
Línea de Comandos 3.90. Prueba de escritura 196
Línea de Comandos 3.91. Comando para mover el servidor virtual del nodo actual al nodo2 197
Línea de Comandos 3.92. Comando para colocar al nodo1 en modo de reposo y migrar todos sus recursos 198
Línea de Comandos 3.93. Comando para detener el servidor virtual seisc 202
Línea de Comandos 3.94. Comando para detener el servidor virtual earth 204

LISTA DE ECUACIONES

Ecuación 1.1. Disponibilidad de un servicio 3
Ecuación 1.2. Tamaño del arreglo en RAID 4 52
Ecuación 1.3. Tamaño del arreglo en RAID 6 52
Ecuación 1.4. Tamaño del arreglo en RAID 1+0 53
Ecuación 2.1. Determinar la utilización real de memoria RAM 104

RESUMEN

El objetivo del presente Proyecto de Titulación es diseñar e implementar un *cluster* de alta disponibilidad para los sistemas de adquisición y procesamiento del Instituto Geofísico de la Escuela Politécnica Nacional.

En el primer capítulo se revisan conceptos básicos sobre alta disponibilidad, *clusters* de alta disponibilidad, y los componentes que conforman un *cluster*, para luego revisar y comparar tres soluciones para *clusters* de alta disponibilidad de código abierto, a fin de elegir la que se utilizará, para posteriormente revisar más a fondo la solución elegida. Se revisan también tecnologías de almacenamiento utilizadas en los *clusters* de alta disponibilidad. Por último se comparan tres soluciones de virtualización de código abierto, y se elige la que se empleará en el presente Proyecto.

En el segundo capítulo se realiza un análisis de la situación actual de los sistemas de adquisición y procesamiento del Instituto Geofísico a los que se brindará alta disponibilidad, se presentan las características principales de estos sistemas y los módulos que conforman y cuales serán administrados por el software del *cluster* de alta disponibilidad. Se determina también los requerimientos de hardware de estos sistemas para poder realizar el dimensionamiento de los nodos que serán parte del *cluster*.

En el tercer capítulo se presentan el diseño y la implementación del *cluster*, en primer lugar se elige el tipo de *cluster* de alta disponibilidad, la red que el *cluster* emplea, el tipo de almacenamiento compartido, etc.. Luego de esto se presentan las implementaciones así como las pruebas correspondientes.

En el cuarto capítulo se presentan las conclusiones y recomendaciones resultado del Proyecto de Titulación.

Por último en los Anexos se encuentran los procedimientos para la instalación y configuración de los sistemas de adquisición y procesamiento de datos a los que se ha brindado alta disponibilidad.

PRESENTACIÓN

Diariamente se utilizan los servicios que se brindan mediante sistemas informáticos de diferentes instituciones y empresas, y se espera poder usar estos servicios a cualquier hora del día al momento que se los necesite. A fin de poder asegurar la continuidad en el funcionamiento de un sistema, el mismo debe estar diseñado con alta disponibilidad.

Un *cluster* de alta disponibilidad es una de las soluciones mediante la cual es posible garantizar que un sistema siga funcionando aunque un fallo de hardware o software suceda. Este tipo de *cluster* debe tener un diseño que elimine o reduzca los puntos de falla, lo que se consigue en la mayoría de los casos mediante redundancia de hardware, para lo que se utilizan tecnologías tales como discos RAID, replicación de datos, interfaces de red enlazadas, etc.

Otra de las tecnologías que un *cluster* de alta disponibilidad puede utilizar es la virtualización, con lo que el *cluster* monitoriza y controla máquinas virtuales y los recursos que se ejecutan en ellas.

En este Proyecto de Titulación se presenta el diseño e implementación de un *cluster* de alta disponibilidad empleando software libre y con componentes de bajo costo.

CAPÍTULO I

1. FUNDAMENTOS TEÓRICOS

1.1 INTRODUCCIÓN

Existen sistemas y servicios informáticos que deben estar disponibles de forma ininterrumpida, dada la criticidad e importancia que tienen. Con este objetivo se han desarrollado soluciones tecnológicas cuyo objetivo es disminuir o anular el tiempo de caída de esos sistemas ante fallas de cualquier tipo.

Este Proyecto de Titulación presenta una solución basada en el uso de la tecnología de *clustering* y la virtualización, para brindar alta disponibilidad a los sistemas de adquisición y procesamiento del Instituto Geofísico de la Escuela Politécnica Nacional.

En este capítulo se revisará en primer lugar conceptos de alta disponibilidad, a continuación se presentarán conceptos de *cluster*, para luego comparar tres soluciones de *clustering* de alta disponibilidad a fin de decidir la que se utilizará, para luego continuar con una revisión más detallada de la solución de alta disponibilidad elegida.

Luego de esto se revisarán conceptos relacionados con el almacenamiento que un *cluster* de alta disponibilidad necesita.

Por último se revisarán conceptos y tecnologías de virtualización, a fin de comparar tres soluciones de virtualización de código abierto y decidir la que se empleará en el presente proyecto.

1.2 ALTA DISPONIBILIDAD

Se entiende como disponibilidad de un sistema al tiempo durante el que dicho sistema trabaja de forma normal y puede brindar servicio a sus usuarios [1].

Para medir el nivel de disponibilidad de un sistema existen ciertos parámetros que se revisarán a continuación y que son necesarios para entender mejor la disponibilidad.

1.2.1 DISPONIBILIDAD

Se expresa la disponibilidad como un porcentaje, que permite determinar el tiempo en el que el sistema funcionará correctamente, durante un período de tiempo determinado. En la Tabla 1.1 se presentan algunos valores de disponibilidad y su equivalente en horas de funcionamiento.

Disponibilidad	Porcentaje de caída	Tiempo de caída por año	Tiempo de caída por semana
98%	2%	7,3 días	3,3 horas
99%	1%	3,65 días	1,6 horas
99,8%	0,2%	17,5 horas	20 min
99,9%	0,1%	8,7 horas	10 min
99,99%	0,01%	52,5 min	1 min
99,999%	0,001%	5,2 min	6 segundos

Tabla 1.1. Valores de disponibilidad y su equivalencia en tiempo [3]

A medida que aumenta el porcentaje de disponibilidad lo hace también el costo que el sistema tendrá.

Para muchas aplicaciones un porcentaje de disponibilidad del 99% es adecuado, es decir existen sistemas que pueden tolerar una caída de aproximadamente dos horas cada semana, lo que depende de a qué hora del día suceda esa caída.

Si la caída sucede en la madrugada de un domingo no genera los mismos inconvenientes que si sucede en un día laborable a la hora en que todos los usuarios se conectan.

1.2.2 MODELO DE LOS NUEVES

El expresar la disponibilidad usando porcentajes, como 99,99%, se conoce como el modelo de los nueves y debería utilizarse solo para propósitos teóricos, ya que no es posible modelar todos los componentes o sistemas involucrados en la entrega de un recurso o servicio.

1.2.3 TIEMPO DE CAÍDA

Es el tiempo durante el cual el usuario no puede acceder al servicio que el sistema brinda, debido a un fallo del sistema o uno de sus componentes.

1.2.4 CÁLCULO DE DISPONIBILIDAD

Para determinar la disponibilidad de un sistema se utiliza la Ecuación 1.1.

```
A = [MTBF / (MTBF + MTTR)]*100
```

Ecuación 1.1. Disponibilidad de un servicio [2]

En la Ecuación 1.1 `A` es el grado de disponibilidad expresada en porcentaje, `MTBF` (*Mean Time Between Failures*) es el tiempo medio entre fallas mientras que `MTTR` (*Maximum Time To Restore*) es el tiempo máximo que tomaría reparar o resolver un error en particular.

Por ejemplo si se tiene un sistema que tiene un tiempo medio entre fallas igual a 50.000 horas y un tiempo de reparación promedio de 5 horas, la disponibilidad del sistema será 99,99%, mejorar ese porcentaje a 99,998% implicaría disminuir el tiempo de caída a menos de 10 minutos al año.

1.2.5 ALTA DISPONIBILIDAD

Que un sistema esté disponible significa que el usuario puede acceder a los servicios que el sistema brinda, dentro del período de tiempo que se supone que el sistema esté en funcionamiento.

Alta disponibilidad implica maximizar el tiempo de funcionamiento del sistema, o lo que es lo mismo disminuir el tiempo de caída y el tiempo de recuperación, mediante técnicas de software, redundancia de hardware, redundancia a nivel de red, planes de recuperación ante desastres, etc.

Un sistema de cómputo con alta disponibilidad es aquel que ha sido diseñado para minimizar los tiempos de caída en los servicios que el sistema brinda mediante la reducción de puntos de falla y el control de los posibles errores que puedan presentarse, un punto de falla puede ser cualquier parte que forma al sistema de cómputo, como por ejemplo un disco duro, un punto de red, un conmutador, etc.

Es posible diseñar e implementar un sistema de cómputo con alta disponibilidad empleando tecnologías de *clustering*, que se revisan en la siguiente sección.

1.3 CLUSTER

La tecnología de *clustering* consiste en que dos o más computadores trabajen de manera coordinada para ofrecer escalabilidad, alta disponibilidad o alto desempeño, y en caso de fallas la carga de trabajo pueda distribuirse entre los computadores que forman parte del *cluster* [3].

1.3.1 CLUSTER DE ALTA DISPONIBILIDAD

Un *cluster* de alta disponibilidad es un conjunto de nodos que se comunican entre sí y trabajan con un objetivo común que consiste en ofrecer acceso ininterrumpido a datos o servicios.

Si uno de los servidores que forma parte del *cluster* pierde conexión con la red o experimenta un fallo de hardware e incluso si el error se presenta en la aplicación que brinda el servicio el *cluster* debe ser capaz de continuar con su funcionamiento con un tiempo de caída mínimo.

Un *cluster* de alta disponibilidad básico puede verse en la Figura 1.1, el mismo consta de dos nodos o servidores que comparten un almacenamiento común, en donde se guardan los datos que los clientes necesitan.

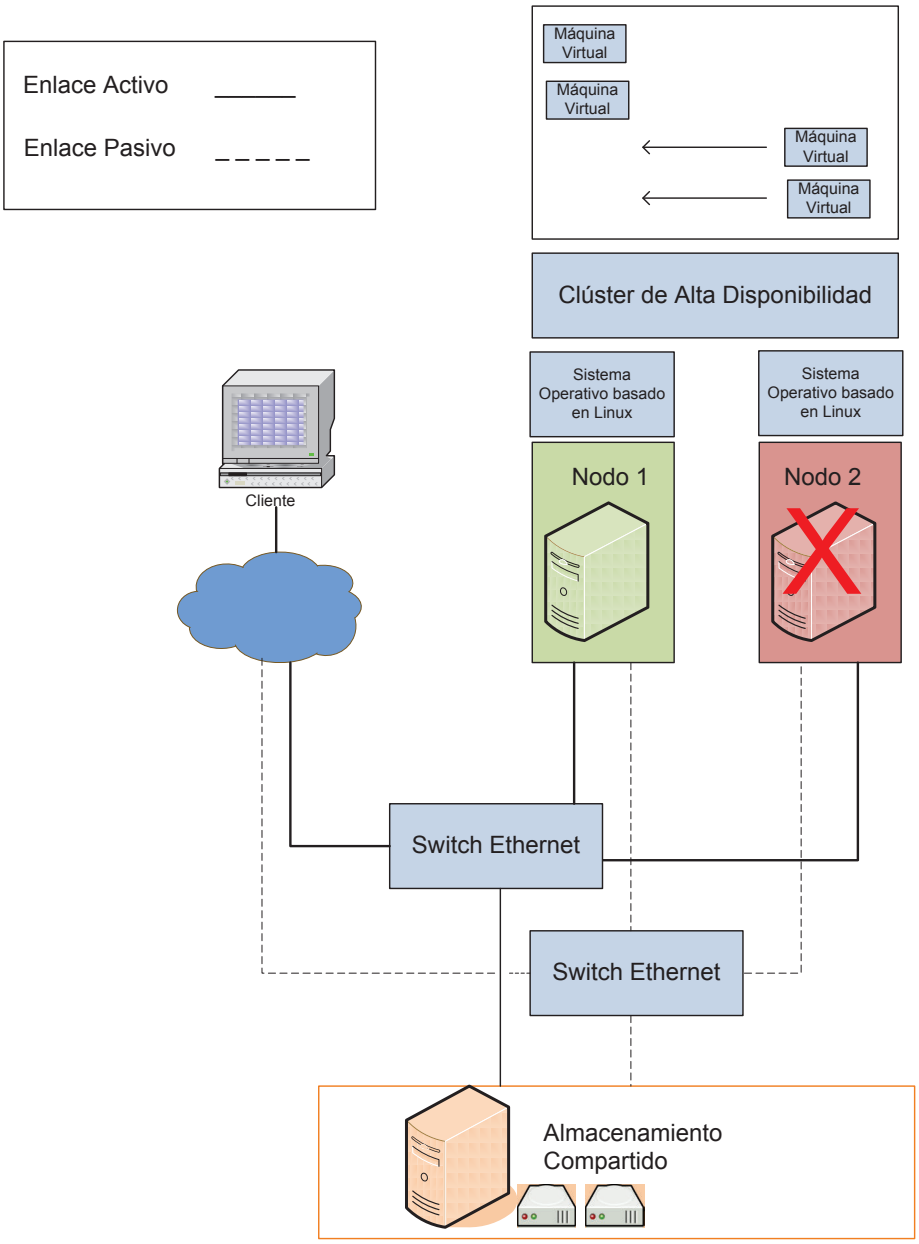

Figura 1.1. Cluster de alta disponibilidad básico [5]

Si uno de los nodos no pudiera acceder a los datos, el otro tomaría su lugar respondiendo a las peticiones de los clientes.

1.3.2 COMPONENTES LÓGICOS DE UN CLUSTER DE ALTA DISPONIBILIDAD

A nivel lógico un *cluster* de alta disponibilidad puede dividirse en 4 capas [4]:

1. Almacenamiento compartido del *cluster*
2. Software de administración del *cluster*
3. Software de administración de recursos del *cluster*
4. Agentes de recursos

En la Figura 1.2 se presenta como ejemplo el modelo de cuatro capas que implementa la solución de alta disponibilidad de SUSE Linux Enterprise Server (SLES)[1], en la cual se puede apreciar las herramientas que se emplean en cada una de las capas y que se revisarán en esta sección.

1.3.2.1 Almacenamiento compartido

En un *cluster* de alta disponibilidad es necesario que cada nodo del *cluster* tenga acceso a un almacenamiento común, para que si falla un nodo que ejecuta alguna aplicación, la misma pueda levantarse en otro nodo del *cluster* y tenga acceso a los mismos datos.

Así, por ejemplo en un *cluster* con tres nodos denominados A, B, C; si falla el nodo A en el que se ejecutaba la aplicación MySQL[2], la aplicación puede levantarse en el nodo B, siempre y cuando los archivos que forman la base de datos se encuentren en el almacenamiento compartido.

Existen dos tipos de almacenamiento que se emplean en un *cluster* [7]:

---

[1] SLES: es un sistema operativo de código abierto con licenciamiento basado en Linux, utilizado principalmente en servidores. Para más información se recomienda revisar la referencia [76]

[2] MySQL: es un servidor de base de datos de código abierto que puede ejecutarse en sistemas operativos Linux o Windows.

1.3.2.1.1 Single-Instance Storage (SIS)

En este tipo de almacenamiento el *cluster* almacena todos sus datos en una instancia centralizada, como por ejemplo una SAN[3].

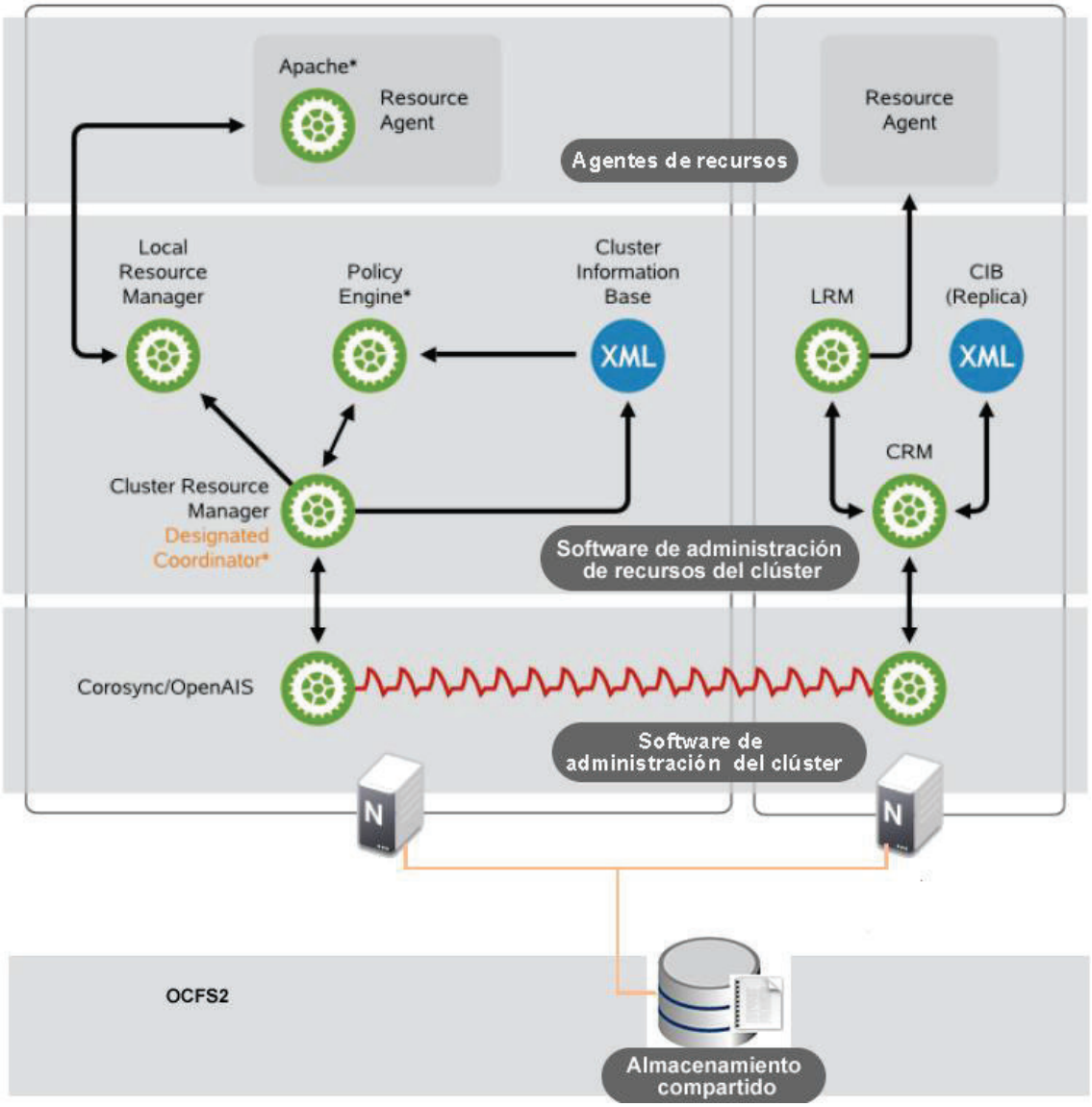

Figura 1.2. Componentes lógicos de un cluster de alta disponibilidad [6]

[3] *Storage Area Network*: es un sistema de almacenamiento con una red dedicada que provee acceso a un dispositivo de bloque.

El acceso al almacenamiento puede ser activo/pasivo es decir un nodo a la vez o activo/activo en donde múltiples nodos acceden simultáneamente.

Si se emplea la configuración activa/activa es necesario utilizar un sistema de archivos del tipo GFS2[4] u OCFS2[5].

Cualquiera sea el método de acceso empleado en SIS, es necesario que exista un mecanismo para coordinar el acceso de los nodos al almacenamiento compartido; Linux emplea DLM[6] a nivel de kernel[7] para realizar esa tarea.

La implementación de SIS es sencilla pero tiene un grave inconveniente, ya que si por algún motivo la SAN falla o se vuelve inaccesible, se perderán los datos que necesitan las aplicaciones que se ejecutan en el *cluster*, y no estarán disponibles para el usuario.

En la Figura 1.3 se presenta un *cluster* con almacenamiento del tipo SIS, si por algún motivo los enlaces o el conmutador fallaran, de nada serviría tener redundancia en los servidores físicos, ya que estos no podrían acceder a los datos que se encuentran almacenados en la SAN, con lo que se perdería la alta disponibilidad.

1.3.2.1.2 Almacenamiento Replicado

Permite mediante hardware o software que los datos escritos en un almacenamiento primario se repliquen de manera síncrona[8] o asíncrona[9] en un almacenamiento secundario.

---

[4] *Global File System* 2: es un sistema de archivos que permite el acceso coordinado de N nodos a un dispositivo de bloque.

[5] *Oracle Cluster File System* 2: es un sistema de archivos libre y de código abierto para *cluster*.

[6] *Distributed Lock Manager*: es una aplicación que permite sincronizar el acceso a un recurso compartido.

[7] El kernel es el núcleo de un sistema operativo y está encargado de ejecutar instrucciones en el CPU.

[8] La replicación síncrona escribe al mismo tiempo en el almacenamiento local y secundario.

[9] La replicación asíncrona escribe primero en el almacenamiento local y luego en el secundario.

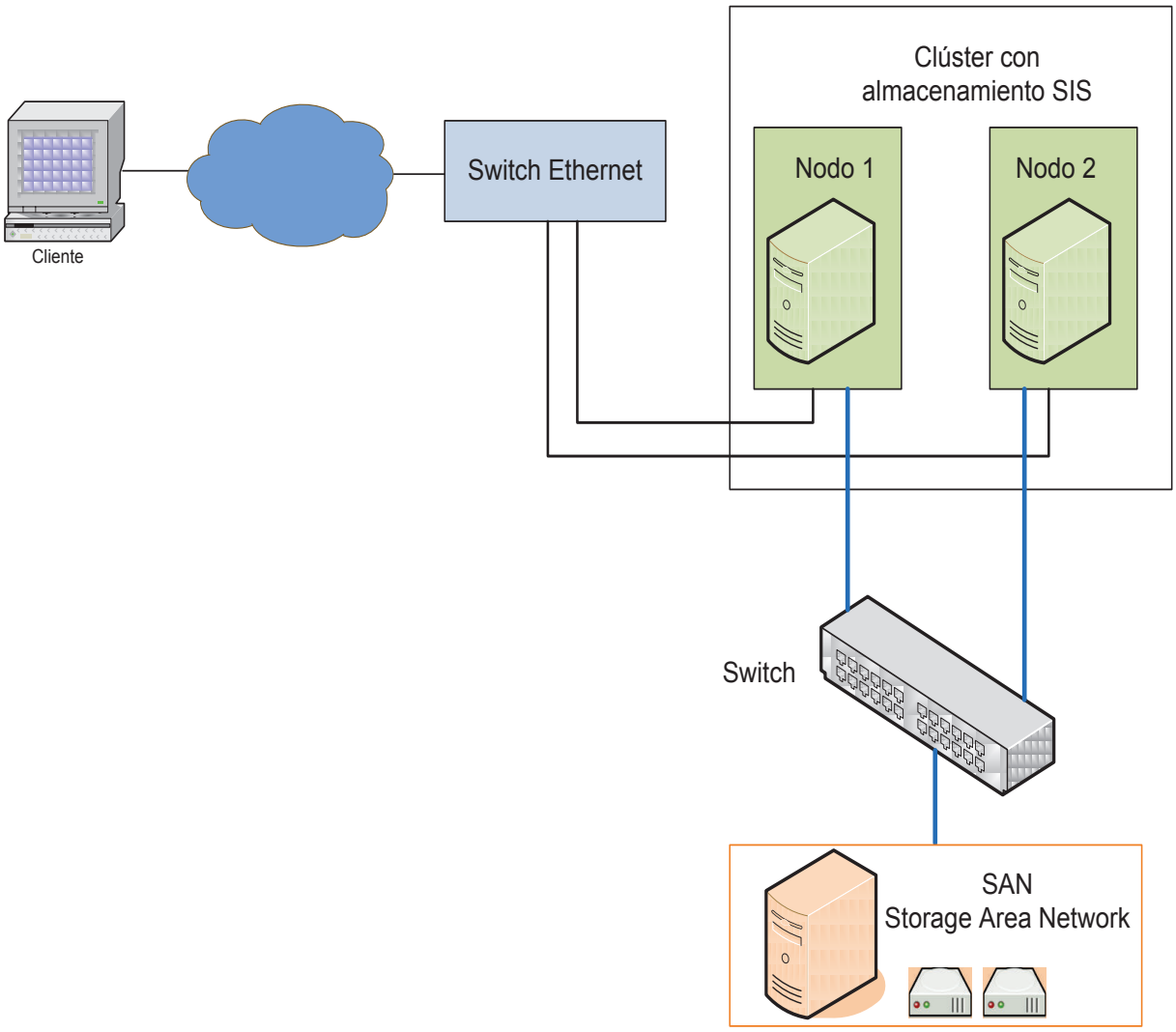

Figura 1.3. Cluster con almacenamiento del tipo SIS

Existen varias opciones para implementar la replicación de datos vía software. La forma estándar en Linux se denomina DRBD (*Distributed Replicated Block Device*) [7], [8].

DRBD consiste de un módulo de kernel y aplicaciones para configurar y administrar un dispositivo de bloque denominado `/dev/drbd0`, cuya información se sincroniza entre dos servidores vía red.

En la Figura 1.4 se indica el esquema de un *cluster* con almacenamiento replicado del tipo DRBD, si el servidor DRBD principal falla, el servidor secundario toma su

lugar, y los nodos del *cluster* no pierden conectividad con el almacenamiento compartido.

Las soluciones de replicación mediante hardware dependen de la empresa que suministre el almacenamiento, por ejemplo IBM ofrece la solución PPRC (*Peer to Peer Remote Copy*) que funciona solamente con los sistemas de almacenamiento de IBM, como IBM *System Storage DS8000* o IBM *System Storewize*.

Por lo general la replicación por hardware tiene costos bastante elevados.

1.3.2.2 Software de administración del cluster

Permite que un grupo de computadoras o nodos funcionen como un *cluster*. Las tareas del software de administración del *cluster* son enviar mensajes entre los nodos que son parte del *cluster*, determinar si el número de nodos de los que dispone el *cluster* es suficiente para funcionar (*quorum*) y determinar si un nodo pertenece o no al *cluster* (*membership*).

En los *cluster* de HA basados en Linux se utilizaba Heartbeat[10] para realizar estas tareas, sin embargo en los últimos años en la mayoría de distribuciones se emplea Corosync, el cual es un software que implementa el protocolo Totem[11] de membresía y envío de mensajes ordenados, Corosync además cifra los mensajes y autentica a los nodos del *cluster*, lo que permite el envío de mensajes de forma segura y ordenada utilizando UDP/IP[12] sobre Ethernet.

Al software de administración del *cluster* se le conoce también como *cluster stack*.

---

[10] Heartbeat: es un demonio que implementaba comunicaciones y pertenencia en un *cluster* Linux.

[11] Totem: es un protocolo de comunicaciones para enviar mensajes de forma segura y confiable. Para más información se recomienda revisar la referencia [79]

[12] User Datagram Protocol: es un protocolo de la capa de transporte no confiable , con envío de paquetes sin control de orden, de entrega o duplicación.

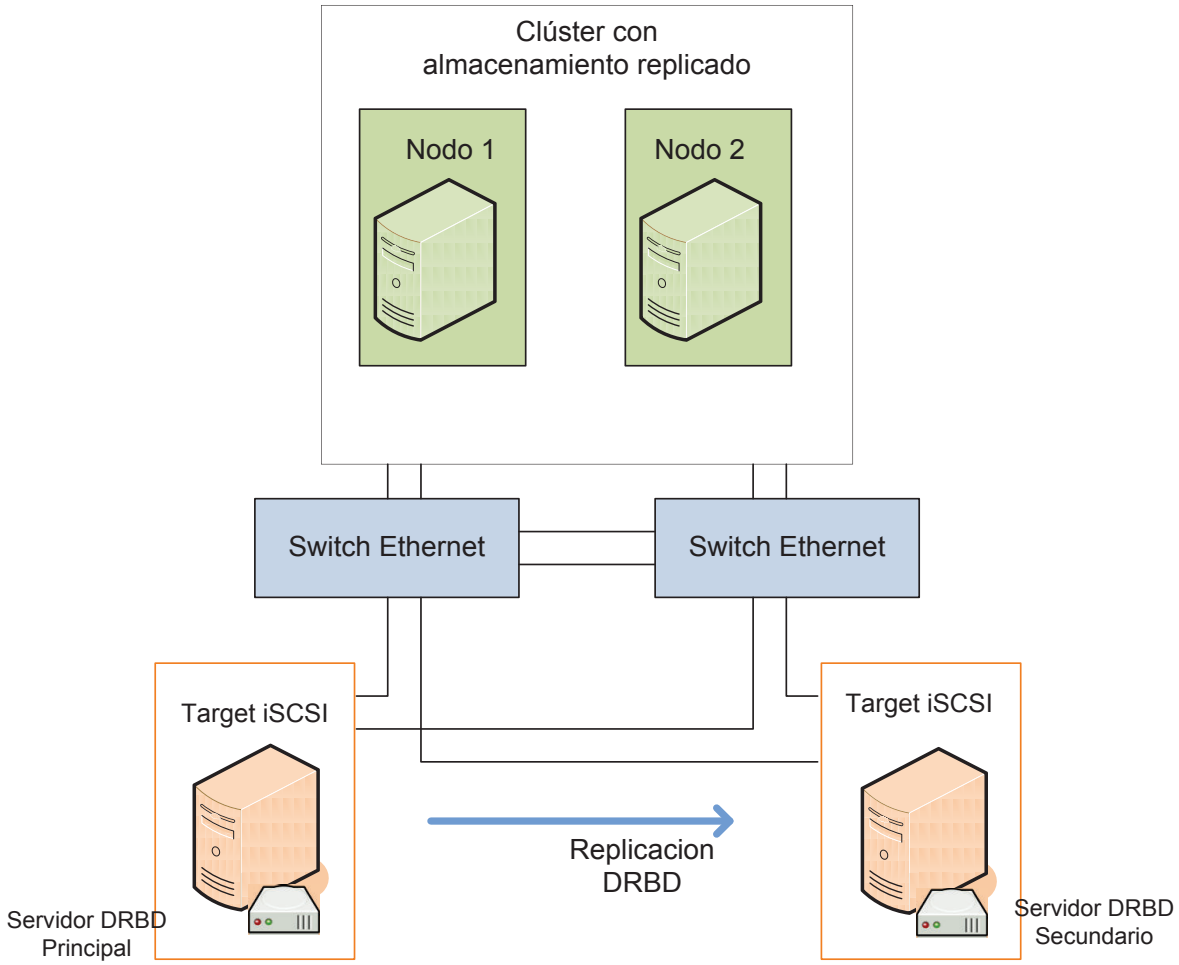

Figura 1.4. Cluster de alta disponibilidad con almacenamiento replicado

1.3.2.3 Software de administración de recursos del cluster

Como su nombre lo indica es el software encargado de activar, detener y monitorizar el estado de los servicios que un clúster ejecuta. En base al estado de los nodos del clúster y políticas y reglas configuradas previamente, el software de administración de recursos toma decisiones para garantizar el funcionamiento continuo de los servicios. El administrador de recursos por defecto en Linux es Pacemaker[13],,

[13] Pacemaker: es un administrador de recursos *open source* que permite crear recursos con alta disponibilidad.

aunque distribuciones como Red Hat[14] y CentOS[15] pueden utilizar el administrador RGManager[16].

1.3.2.4 Agente de recursos

Es un *script* o programa que actúa como interfaz entre el recurso y el administrador de recursos.

Pueden escribirse en cualquier lenguaje de programación, pero lo común es que sean *scripts* que se ejecutan en la consola de Linux, y que permitan realizar operaciones como *start, stop, status, validate-all* sobre un recurso, servicio o programa:

- *Start*: esta operación inicia el servicio.
- *Stop*: detiene el servicio o programa.
- *Status*: retorna el estado del servicio.
- *Validate-all*: Valida los parámetros con los que se configuró el recurso.

El administrador de recursos Pacemaker es compatible con varias clases de agentes:

- *LSB Resource Agents* (Linux *Standard Base*) son los *scripts* que se encuentran en la carpeta `/etc/init.d/`, se crean por defecto al instalar un programa que funciona como un servicio, tal como Apache[17], SSH[18], MySQL, etc. Varían dependiendo de la versión de Linux.
- *OCF Resource Agents* (Oracle *Cluster Framework*) son *scripts* basados en LSB que tienen características adicionales.

---

[14] Red Hat: es un sistema operativo de código abierto con licenciamiento basado en Linux.

[15] CentOS: es un sistema operativo gratuito, libre y de código abierto basado en Linux.

[16] RGManager: es un administrador de recursos desarrollado por Red Hat en el año 2005.

[17] Apache: es el servidor web más utilizado en sistemas operativos Linux y Windows.

[18] Secure Shell: es un protocolo de comunicaciones encriptado para acceder remotamente a un computador.

- Systemd son los *scripts* de inicio que utiliza el programa `systemd`, el cual es el encargado de arrancar un programa como un servicio del sistema, como por ejemplo, programas de correo, firewall, etc.

1.3.3 COMPONENTES FÍSICOS DEL CLUSTER

1.3.3.1 Nodos

Deben cumplir ciertos requisitos de RAM y procesador, en particular para el caso de la virtualización el procesador debe soportarla. En el caso de procesadores Intel esta característica se indica con la bandera VT-x, mientras que en los procesadores AMD se indica con la bandera AMD-V.

A fin de tener redundancia a nivel de red se recomienda que los nodos cuenten con al menos dos tarjetas Ethernet, así el nodo podrá continuar comunicándose con el resto del *cluster* si una de ellas falla.

Si para el almacenamiento compartido se emplea una SAN los nodos deben tener tarjetas de conexión de fibra, o Gigabit Ethernet si el almacenamiento compartido es mediante un servidor iSCSI [19].

1.3.3.2 Red de comunicaciones del cluster

Es la red que los nodos utilizan para enviar los mensajes de quórum, pertenencia, administración de recursos, etc.

Es recomendable implementar redundancia en la red de comunicaciones ya sea mediante el enlazado de interfaces o *network bonding*[20], o definiendo en el software de administrador del *cluster* un canal de comunicación de respaldo. Sin embargo, la misma red que se utiliza para las comunicaciones del *cluster* puede emplearse para

---

[19] *Internet Small Computer System Interface*: es un protocolo que permite la escritura remota en un disco duro.

[20] *Network bonding*: es una tecnología que permite enlazar dos interfaces de red de tal forma que funcionen como una sola, para más información se recomienda revisar la referencia [73]

las comunicaciones con los clientes (red pública) e incluso para las comunicaciones con el almacenamiento compartido.

1.3.3.3 Red pública

Es la red a la que los clientes se conectan para requerir los servicios de alta disponibilidad que el *cluster* brinda.

1.3.3.4 Sistema de almacenamiento compartido

Se refiere a la capa física sobre la que se implementa la capa de almacenamiento lógico, puede tratarse de un servidor iSCSI, un almacenamiento SAN con conexión de fibra o inclusive un *cluster* iSCSI con alta disponibilidad.

1.4 SOLUCIONES DE ALTA DISPONIBILIDAD DE CÓDIGO ABIERTO

A continuación se presentan tres tecnologías de *clustering* de alta disponibilidad de código abierto que posteriormente se compararán entre sí para definir la que se empleará en este Proyecto de Titulación. Es necesario aclarar que aunque las soluciones a analizar son de código abierto, tienen un costo de suscripción que permite acceder a horas de soporte y actualizaciones del software.

1.4.1 RED HAT HIGH AVAILABILITY [9]

Los componentes de la solución de *clustering* de alta disponibilidad que ofrece Red Hat 6, de acuerdo al modelo de cuatro capas presentado en la Sección 1.3.2 se presenta en la Tabla 1.2.

1.4.1.1 Almacenamiento compartido

1.4.1.1.1 Red Hat GFS

Es un sistema de archivos que permite a los nodos de un *cluster* acceder coordinadamente a un dispositivo de almacenamiento compartido como por ejemplo

una SAN, como si se tratara de un dispositivo local. GFS puede usarse con iSCSI o GNBD para ofrecer un sistema de almacenamiento compartido de bajo costo.

Capa	Componentes
Agente de recursos	Agentes de Recursos del tipo LSB
Software de administración de recursos del cluster	RGManager
Software de administración del cluster	CMAN21 + Corosync DLM (Data Lock Manager) Fencing22 CCS23 (Cluster Configuration System)
Almacenamiento Compartido	CLVM24 (Cluster Logical Volume Manager) Red Hat GFS25 (Global File System) GNBD26 (Global Network Block Device)

Tabla 1.2. Modelo de cuatro capas para la solución de alta disponibilidad de Red Hat

1.4.1.1.2 CLVM

Es una versión para *cluster* de LVM[27], que permite que los volúmenes lógicos estén disponibles para todos los nodos que forman parte del *cluster*.

---

[21] Cluster MANager: es un módulo de kernel utilizado anteriormente en *cluster* Linux.

[22] *Fencing*: es el procedimiento para aislar un nodo en un *cluster* de computación y así proteger los recursos compartidos del *cluster*.

[23] CCS: es el software encargado de que el archivo de configuración del *cluster* se sincronice en todos los nodos del *cluster*.

[24] CLVM: es la versión clusterizada del sistema de archivos LVM, para más información se recomienda revisar la referencia [77].

[25] GFS: es la primera versión del sistema de archivos GFS2.

[26] GNBD: es una tecnología similar a iSCSI, que funciona solamente con sistemas de archivos GFS, para mayor información sobre GNBD se recomienda revisar la referencia [10]

[27] *Logical Volume Mananager*: permite la creación de volúmenes lógicos formados por uno o más discos o particiones físicas.

1.4.1.1.3 GNBD

Es un componente auxiliar de GFS que permite acceder a dispositivos de almacenamiento a nivel de bloque[28] vía TCP/IP.

Tiene dos componentes un cliente GNBD y un servidor GNBD, el servidor pone a disposición del cliente un dispositivo de bloque[29], sobre el cual el cliente GNBD crea un sistema de archivos GFS. GNBD tiene como limitante que puede emplearse solamente con el sistema de archivos GFS.

En la Figura 1.5 se puede observar un esquema con los componentes del almacenamiento utilizado por la solución de Red Hat.

1.4.1.2 Software de administración del cluster

El software de administración de un *cluster* Red Hat está compuesto por un módulo del kernel llamado CMAN y el sistema de mensajes Corosync. Juntos se encargan de manejar el envío de mensajes entre nodos, la pertenencia del nodo al *cluster*, verificar la suficiencia de nodos para funcionar y también la notificación de eventos y cambios en el *cluster*.

Si un nodo no transmite mensajes durante una determinada cantidad de tiempo, CMAN remueve el nodo del *cluster* y comunica al resto de componentes que el nodo ya no es un miembro, lo que hace que el resto de componentes determinen la acción a tomar, como por ejemplo aislar el nodo. De igual forma si se agrega un nodo al *cluster* se avisa al resto de componentes para permitir al nuevo nodo acceder al almacenamiento compartido, modificar el quórum, etc.

---

[28] *Block Level Storage*: es el tipo de almacenamiento que una SAN pone a disposición del cliente, es decir un disco sin formato con un tamaño fijo, que el cliente utiliza como si fuese un disco local.

[29] Dispositivo de bloque: son dispositivo que permite lectura y/o escritura en bloques de datos, como por ejemplo discos duros, CD-ROM, etc.

1.4.1.2.1 Administrador de bloqueo

Se utiliza DLM el cual permite que los componentes del *cluster* accedan ordenadamente a los recursos compartidos del *cluster*. DLM se ejecuta en cada uno de los nodos que forman el *cluster*.

1.4.1.2.2 Fencing

Consiste en desconectar o aislar a un nodo del *cluster*, para impedir que acceda al almacenamiento compartido a fin de evitar posibles inconsistencias en los datos; en Red Hat esta tarea la cumple el demonio `fenced`.

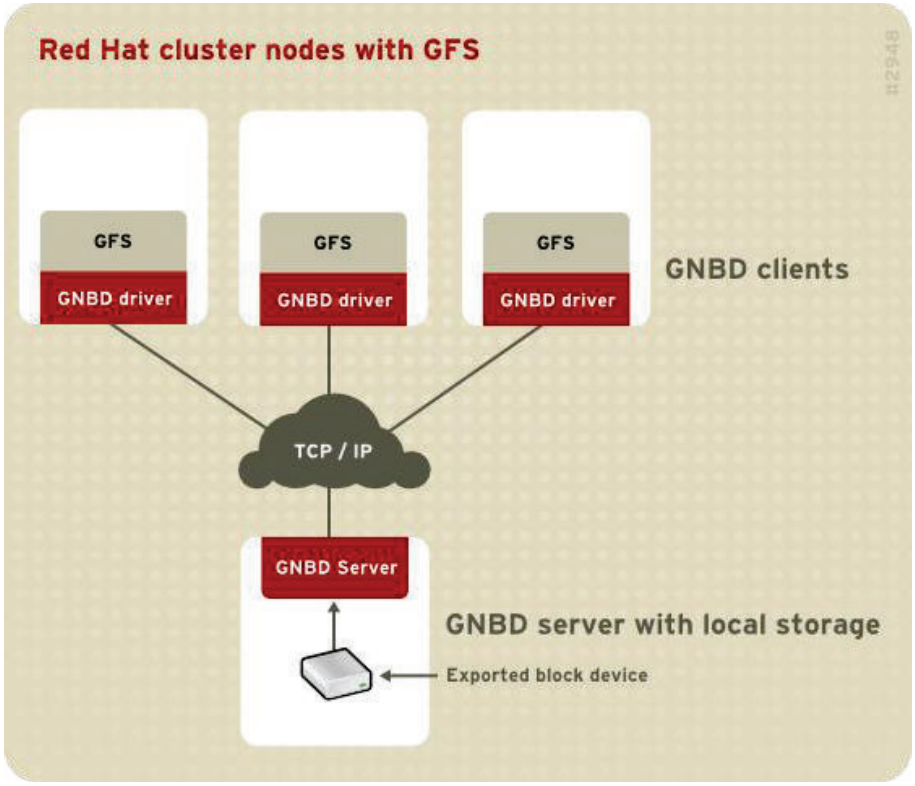

Figura 1.5. Capa de almacenamiento de un cluster Red Hat [10]

Cuando un nodo falla CMAN alerta a `fenced` para que aísle al nodo, mientras tanto para garantizar la integridad de los datos DLM y GFS detienen toda actividad hasta que el demonio `fenced` les confirme que el nodo ha sido aislado.

1.4.1.2.3 Administrador de configuración

Conocido como CCS, se encarga de compartir y actualizar el archivo de configuración del *cluster* con otros componentes del mismo, al igual que DLM se ejecuta en cada uno de los nodos que forman parte del *cluster*

El archivo de configuración del *cluster* es un archivo XML[30] con la siguiente información:

Nombre del *cluster*: indica el nombre del *cluster*, versión del archivo de configuración, información de *fencing*, etc.

- Cluster: indica el nombre del nodo y su ID, número de votos para el quórum y métodos de *fencing* que puede emplearse con el nodo.

- Dispositivo de *fencing*: define los dispositivos para aislar un nodo en caso de error. Los parámetros varían de acuerdo al tipo de dispositivo, por ejemplo en un *BladeCenter IBM*[31] sería necesaria la dirección IP del dispositivo, el usuario y la clave.

- Recursos administrados: contiene información de los recursos con alta disponibilidad, por ejemplo una dirección IP virtual, un sistema de archivos, etc.

En la Figura 1.6 puede verse un esquema de como el sistema de archivos GFS necesita de CMAN y DLM para acceder al almacenamiento compartido.

---

[30] *Extensible Markup Language*: es un lenguaje de etiquetas que permite estructurar información en un documento.

[31] BladeCenter IBM es la línea de servidores y chasis que integra servidores, almacenamiento y dispositivos de red.

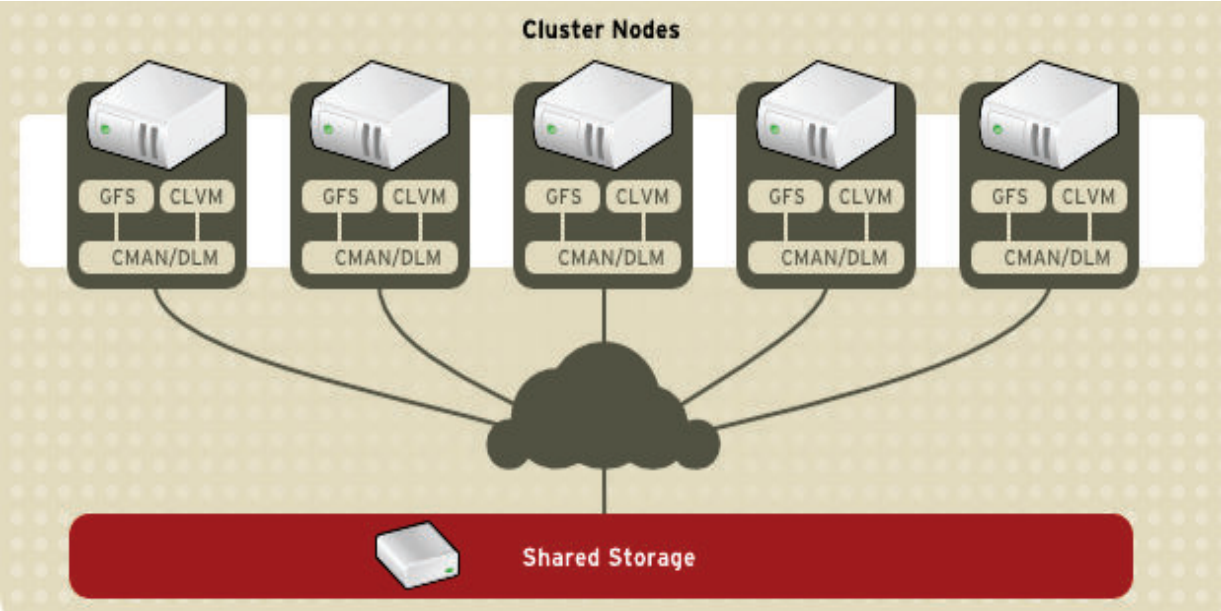

Figura 1.6. Software de Administración del Cluster [11]

1.4.1.3 Software de administración de recursos [9]

El administrador de recursos de la solución de Red Hat es RGManager y permite crear, configurar y administrar servicios o recursos que se desea que tengan alta disponibilidad.

Si un nodo que ejecutaba un servicio sufre un fallo RGManager reubicará ese servicio en otro nodo con una mínima interrupción. Este proceso se denomina *failover*[32].

RGManager crea "dominios de *failover*" o subgrupos formados por dos o más nodos del *cluster*, que tienen asociado un recurso al que garantizan alta disponibilidad, como se puede observar en la Figura 1.7.

[32] *Failover*: consiste en activar un recurso de respaldo o *backup* si el recurso principal falla, un recurso puede ser un enlace de red, un disco duro, un servidor, etc. Este procedimiento se conoce como conmutación por error.

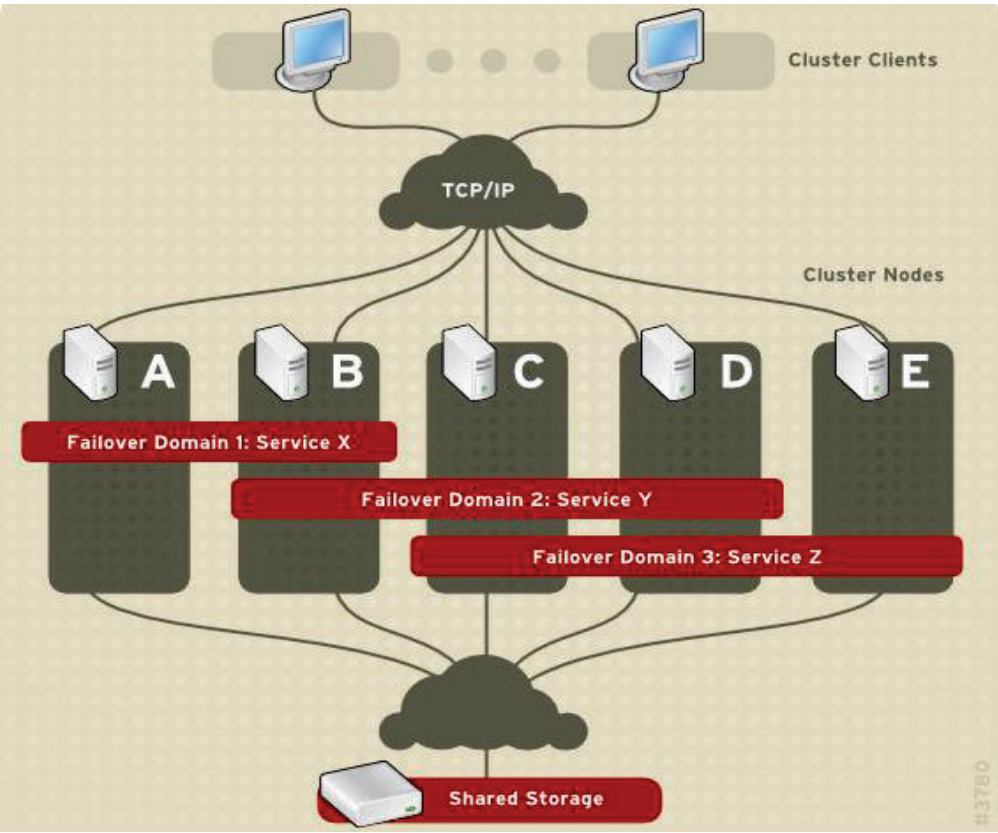

Figura 1.7. Dominios de *failover* de RGManager [11]

1.4.1.4 Agentes de recursos

La solución de Red Hat utiliza principalmente los *scripts* LSB que el demonio `init`[33] emplea para arrancar los servicios, es decir los *scripts* que se encuentran en `/etc/init.d/`.

En la Figura 1.8 se puede observar un ejemplo de un recurso del tipo web, el agente que se utiliza es el *script* `/etc/init.d/httpd`, al recurso se le ha asignado el nombre `content webserver`, tiene un recurso del tipo dirección IP asociado y un sistema de archivos que contiene documentos HTML[34] correspondientes a la página web.

[33] Init: en los sistemas operativos Linux es el primer proceso que inicia al arrancar el computador.

[34] HTML: es un lenguaje de programación empleado para la elaboración de páginas web.

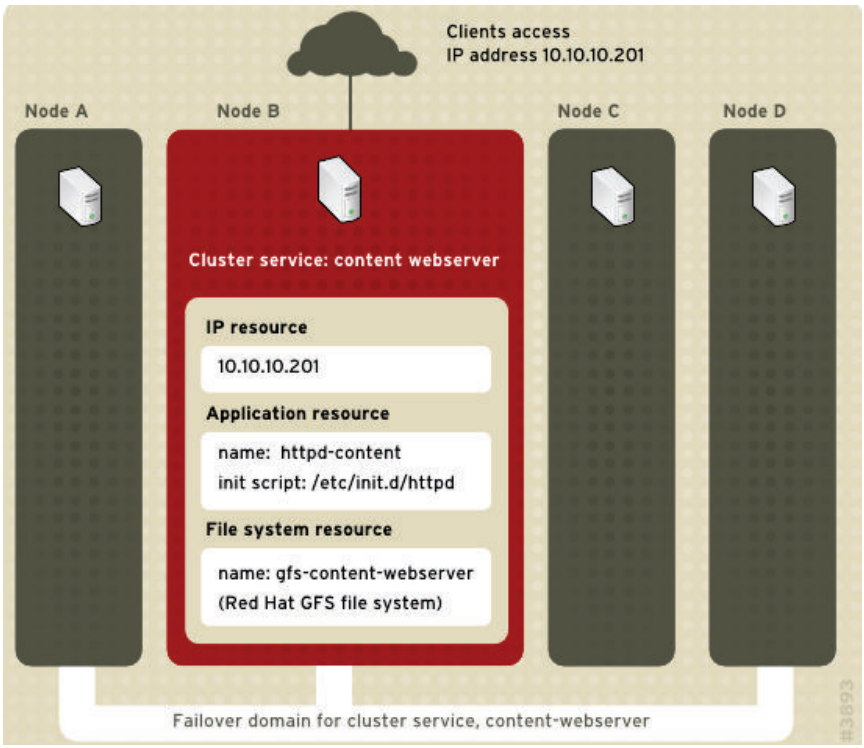

Figura 1.8. Ejemplo de recurso de tipo servicio web [11]

1.4.2 SUSE LINUX ENTERPRISE SERVER HIGH AVAILABILITY (SLES) [6]

El modelo de capas de la solución que ofrece el *plugin* de alta disponibilidad de SUSE Linux Enterprise Server se indica en la Tabla 1.3.

Capa	Componentes
Agente de recursos	Agentes de recursos Pacemaker Agentes de recursos OCF Agentes del tipo Heartbeat
Software de administración de recursos del cluster	Pacemaker
Software de administración del cluster	Corosync/OpenAIS
Almacenamiento compartido	OCFS2/GFS2

Tabla 1.3. Modelo de cuatro capas para la solución de alta disponibilidad de SLES

1.4.2.1 Almacenamiento compartido

OCFS2 es un sistema de archivos para *cluster* con licencia *open source*, desarrollado por Oracle que permite que todos los nodos del *cluster* lean y escriban de forma coordinada en el almacenamiento compartido. El acceso de los nodos al almacenamiento se coordina empleando DLM.

1.4.2.2 Software de administración del cluster

La capa de administración del *cluster* utiliza Corosync y OpenAIS.

Corosync es un sistema de mensajes con características que permiten brindar alta disponibilidad a aplicaciones, en los *cluster* Linux es el sistema de mensajes que se utiliza por defecto [12].

Implementa el protocolo Totem, un algoritmo que tiene aproximadamente 20 años de desarrollo que ofrece una forma segura y confiable de enviar mensajes entre los nodos del *cluster* empleando multicast[35] UDP, pero puede emplear también *unicast*[36] o *broadcast*[37] y el medio de transmisión puede ser Ethernet o Infiniband[38].

OpenAIS es una implementación de AIS[39] (*Aplication Interface Specification*). Dispone de una API[40] y un conjunto de políticas para desarrollar aplicaciones que continúen funcionando en caso de fallas [13].

---

[35] *Multicast*: es un método de distribuir mensajes de uno a varios receptores o de varios a varios receptores en una sola transmisión.

[36] *Unicast*: es un método de comunicación y envío de mensajes de uno a uno, por ejemplo entre un cliente y un servidor.

[37] *Broadcast*: es un método para transmitir mensajes a todos los dispositivos de una red o miembros del dominio broadcast.

[38] Infiniband: es una red de comunicaciones usada para computación de alto desempeño, gracias a su alto rendimiento y baja latencia.

[39] AIS: es un conjunto de especificaciones para desarrollo de software de alta disponibilidad, para consultar las especificaciones de AIS se recomienda revisar la referencia [80].

A pesar que el proyecto OpenAIS está discontinuado, es un requisito para utilizar el sistema de archivos OCFS2.

1.4.2.3 Software de administración de recursos

Pacemaker es el software encargado de que los servicios que el *cluster* brinda tengan alta disponibilidad, es capaz de detectar y corregir fallas a nivel de aplicación o de los nodos del *cluster*, en caso detectar que una aplicación falló, tratará de recuperar la aplicación automáticamente.

Por ejemplo si el recurso es una página web y el nodo en el que se ejecutaba el programa Apache falla, Pacemaker tratará de reiniciar el nodo, si no lo consigue levantará Apache en otro de los nodos del *cluster*.

El proyecto Pacemaker inició en el año 2004 como una colaboración entre Red Hat y Novell y ha recibido apoyo de toda la comunidad de Linux, y es utilizado para aplicaciones tan delicadas como programas de control de tráfico aéreo[41].

1.4.2.4 Agente de recursos

Como se mencionó en la Sección 0, Pacemaker puede comunicarse con varios tipos de agentes OCF, LSB y Systemd.

Dispone de una gran variedad de agentes para dar alta disponibilidad a diferentes recursos, por ejemplo el agente Apache para servicios web, IPaddr2 para direcciones IP, iscsiTarget para un servidor iSCSI, etc. Los *scripts* de los agentes de recursos se encuentran en la carpeta `/usr/lib/ocf/resource.d/`.

En la Figura 1.2 se indica un diagrama con los componentes de esta solución.

---

[40] *Application Programming Interface*: es una capa de abstracción de las funciones y procedimientos que un programa ofrece.

[41] Pacemaker se emplea en el sistema de control de tráfico alemán, para más información se recomienda revisar la referencia [89].

1.4.3 ORACLE CLUSTERWARE [14], [15]

La solución que ofrece Oracle es diferente a las presentadas anteriormente. En un inicio la solución de Oracle se orientaba a brindar alta disponibilidad solamente a bases de datos, sin embargo a partir de las últimas versiones esta solución puede usarse para cualquier servicio.

La Tabla 1.4 presenta los componentes de la solución Oracle Clusterware según el modelo de 4 capas.

Capa	Componentes
Agente de recursos	*Stack* OHAS
Software de administración de recursos	*Stack* CRS: crsd.bin (demonio *Cluster Ready Services*)
Software de administración del cluster	*Stack* CRS: ocssd.bin (demonio Oracle *Cluster Synchronization Services*) cssdagent (demonio encargado del *fencing*) ons (Oracle *Notification Service*)
Almacenamiento	Oracle ACFS (*Automatic Storage Management Cluster File System*)

Tabla 1.4. Modelo de cuatro capas para la solución de alta disponibilidad de Oracle

La solución se denomina Oracle Clusterware y a partir de la versión 11gR2, se divide en dos *stacks* o grupos, CRS (*Cluster Ready Services Stack*) y OHAS (Oracle *High Availability Service Stack*).

El *stack* CRS contiene módulos que cumplen tareas de la capa de administración de *cluster* , administración de recursos e incluso de la capa de agentes de recursos, mientras que los programas del *stack* OHAS cumplen tareas propias de la capa de agente de recursos. En la Figura 1.9 se presentan los componentes o módulos que forman el *stack* CRS y OHAS.

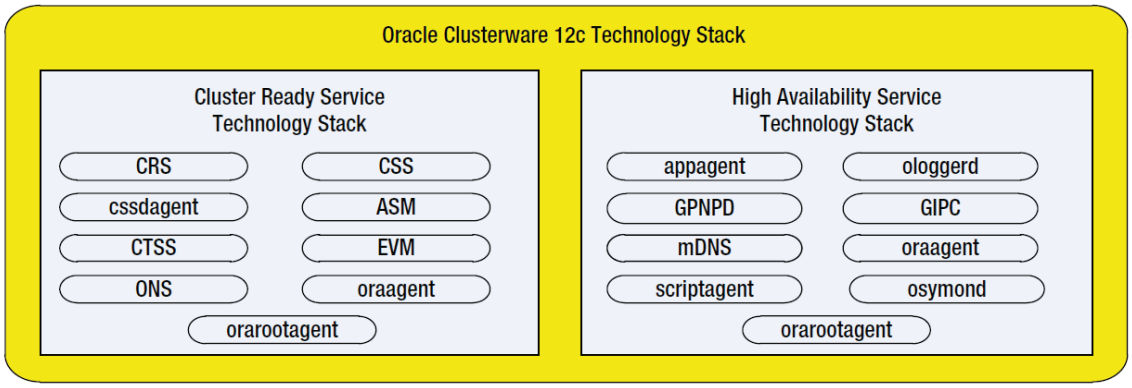

Figura 1.9. *Stack* del cluster de alta disponibilidad desarrollado por Oracle [15]

1.4.3.1 Almacenamiento compartido

Para el almacenamiento que la solución emplea, Oracle desarrolló un sistema de archivos llamado ACFS (*Automatic Storage Management Cluster File System*) que está basado en el sistema de archivos ASM (*Automatic Storage Management*) [16].

ASM es también un administrador de volúmenes similar a LVM, en la Figura 1.10 puede verse su funcionamiento, en la parte inferior están los dispositivos de bloque, que pueden ser discos duros o particiones, con esas particiones se crean discos ASM que se agrupan para formar un volumen en el que pueden escribir los nodos del *cluster*.

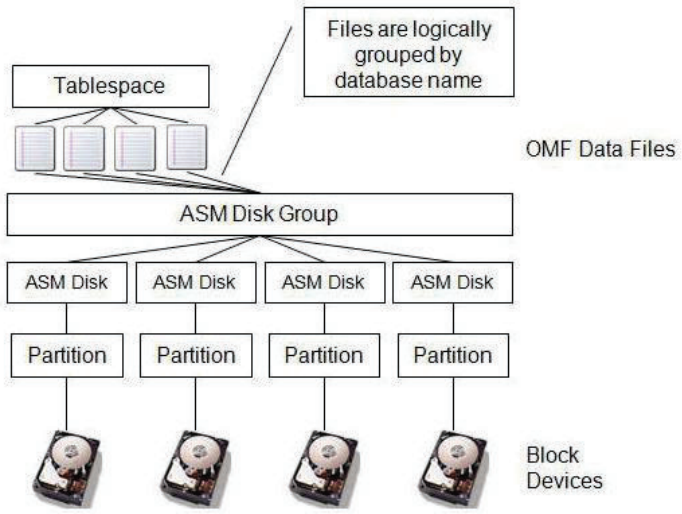

Figura 1.10. Funcionamiento de ASM [17]

ASM permitía almacenar solamente archivos de bases de datos Oracle, ACFS extiende las características de ASM para que pueda guardar cualquier tipo de archivo, como por ejemplo texto, imágenes, archivos ejecutables, etc.

1.4.3.2 Software de Administración del cluster

- `ocsdd.bin`: es el programa encargado de monitorear los nodos que pertenecen al *cluster*.
- `ons`: es el módulo encargado del envío y notificación de eventos y en caso de que un nodo falle informa al módulo `cssdagent`
- `cssdagent`: es el programa encargado de aislar a un nodo del *cluster* y del almacenamiento compartido en caso de que el nodo falle.

1.4.3.3 Software de Administración de recursos

`crsd.bin` es el módulo encargado de gestionar el inicio, interrupción y monitorización de los recursos que tienen alta disponibilidad.

La información de los recursos se almacena en un registro llamado OCR (Oracle *Cluster Registry*), del que el *cluster* tiene varias copias en los nodos que forman el *cluster*.

1.4.3.4 Agente de recursos

Esta capa está a cargo del *stack* OHAS (Oracle *High Availabity Services*) y está compuesta por algunos módulos:

- `Appagent`: es el agente que brinda alta disponibilidad para recursos del tipo aplicación.
- `Scriptagent`: brinda alta disponibilidad a recursos que no sean del tipo aplicación.

- `oraagent` (Oracle *Agent*): este módulo permite interactuar con recursos complejos

- `orarootagent` (Oracle *Root Agent*): es un programa similar a `oraagent`, que administra recursos que solo `root` puede, tal como elementos de red o direcciones IP virtuales.

1.4.4 COMPARACIÓN DE LAS SOLUCIONES PRESENTADAS

Se ha descartado utilizar la solución de Oracle porque su arquitectura es demasiado compleja y su sistema de archivos no soporta almacenar archivos de discos de máquinas virtuales. La opción de Red Hat es bastante estable y documentada, sin embargo a partir de la versión 7 de Red Hat, la empresa planea utilizar Pacemaker como administrador de recursos y discontinuar RGManager, el cual tendrá soporte hasta el año 2020.

La combinación de Pacemaker y Corosync que ofrece la solución de SLES *High Availability* tiene algunas características que las otras soluciones no ofrecen tal como la posibilidad de clonar recursos, lo que permite realizar balanceo de carga; recursos multi-estado, lo que permitiría administrar un almacenamiento del tipo DRBD y la característica más importante es la posibilidad de monitorizar recursos dentro de máquinas virtuales, característica que se conoce como Pacemaker Remote, lo que permitiría al *cluster* monitorizar el estado de los sistemas de adquisición y procesamiento que se instalarán en los servidores virtuales. Esta solución es al momento la que más se utiliza en sistemas Linux, tales como Debian[42], Ubuntu[43], OpenSUSE[44], etc. y la que se utilizará en las versiones 7 de los sistemas operativos basados en Red Hat, tal como CentOS y Scientific Linux. La comparación de estas características se indican en la Tabla 1.5.

---

[42] Debian: es un sistema operativo para servidores, libre y de código abierto basado en Linux.

[43] Ubuntu: es un sistema operativo de escritorio, libre y de código abierto basado en Linux.

[44] OpenSUSE: es un sistema operativo de escritorio basado en SUSE Linux Enterprise Server.

Plataforma de *clustering*	Red Hat High Availability	SLES High Availability	Oracle Clusterware
Arquitectura	RGManager + Corosync	Pacemaker + Corosync	OHAS + CRS
Estado del proyecto	Migró de RGManager a Pacemaker	Activo	Activo
Licenciamiento	GPLV2[45]	GPLV2	GPL[46]
Soporte	Red Hat	SLES	Oracle
Sistema Operativo	Red Hat 6, CentOS 6, SL[47] 6	SLES, Linux Red Hat 7, CentOS 7	AIX[48], Solaris, Linux, Windows 2008
Costo[49]	$399 pr año (2 *socket*) [18]	$699 por año (2 *socket*) [19]	$499 por año (2 *socket*) [20]
Almacenamiento Compartido	GFS2	OCFS2, GFS2	Oracle ACFS
Máximo número de nodos [21]	16 - 32	16 - 32	100 [22]
Interfaz de configuración	Línea de comandos, interfaz web	Línea de comandos, interfaz web	Línea de comandos
Requiere permisos de root	Obligatorio	No necesario	Necesario
Clonación de recursos	No completamente	Sí	No
Recursos de estado múltiple	No soporta	Sí soporta	No soporta
Nodos Remotos	No soporta	Sí soporta	No soporta

Tabla 1.5. Comparación de las soluciones de *clustering*

[45] GNU *General Public License Version* 2: es la licencia de software libre más empleada.

[46] GPL: es la primera versión de la licencia General Public Licence.

[47] Scientific Linux: es un sistema operativo libre y de código abierto basado en Red Hat Linux.

[48] AIX: *Advanced Interactive eXecutive* es un sistema operativo desarrollado y comercializado por IBM.

[49] Los costos mostrados corresponden a horas de soporte, mantenimiento y actualizaciones de software.

De las soluciones presentadas el modelo utilizado por SLES es el que mejores características ofrece, pero como se indica en la Tabla 1.5, para acceder a la solución de SLES es necesario comprar una licencia que brinda soporte, actualizaciones y mantenimiento del software.

Ya que no se considera necesario adquirir tal licenciamiento se elige utilizar el modelo de la solución SLES, es decir Pacemaker y Corosync como software del clúster, pero utilizando un sistema operativo Linux sin licenciamiento, tal como Debian, OpenSUSE, etc,.

La elección del sistema operativo se decidirá en la sección correspondiente al diseño del *clúster*.

Por lo tanto el modelo de la solución del *cluster* de alta disponibilidad que se pretende implementar en este Proyecto de Titulación es el que se indica en la Tabla 1.6.

Capa	Componentes
Agente de recursos	Agentes Pacemaker Agentes Heartbeat
Administración de recursos	Pacemaker
Administración del cluster	Corosync
Almacenamiento compartido	OCFS2/GFS2

Tabla 1.6. Capas del software de clustering y alta disponibilidad seleccionado

En la capa de almacenamiento compartido puede utilizarse el sistema de archivos OCFS2 o GFS2, la elección dependerá de los requerimientos que tenga el *cluster*.

1.5 REVISIÓN DEL SOFTWARE DE CLUSTERING ELEGIDO

1.5.1 CONCEPTOS EMPLEADOS EN PACEMAKER Y COROSYNC [23]

1.5.1.1 Split-brain

Es una situación que ocurre debido a fallas en el sistema de comunicación entre los nodos de un *cluster* y consiste en que diferentes nodos tratan de iniciar una aplicación al mismo tiempo.

Por ejemplo en un *cluster* con tres nodos denominados A, B y C, si cada uno monitoriza el estado de los otros dos nodos vía red el problema es que no es posible distinguir un fallo de un nodo de una posible falla de la red.

Si por ejemplo, la tarjeta de red del nodo C falla, el *cluster* se dividiría en dos (*split*), una parte formada por los nodos A y B y otra parte por el nodo C. Ninguna de las partes tiene conocimiento de la otra, y asume que la otra es la que ha fallado, por lo que cada sección del *cluster* intentará levantar los servicios de alta disponibilidad, poniendo en riesgo el almacenamiento compartido.

Para evitar esta situación el *cluster* utiliza un quórum y un sistema de aislamiento del nodo o *fencing*.

1.5.1.2 Quórum

Se define como el mínimo número de servidores necesarios para que los servicios que el *cluster* brinda entren en funcionamiento, se emplea para evitar situaciones de *split-brain*.

El algoritmo que usa Corosync para implementar el quórum es el de mayoría simple, y consiste en que más de la mitad de los nodos deben estar en funcionamiento y poder comunicarse entre sí para que los servicios de alta disponibilidad estén activos.

Se emplea un quórum a fin de que si el *cluster* se divide en dos o más partes, la parte que tenga la mayoría del quórum pueda iniciar los servicios del *cluster* sabiendo que ningún otro grupo de nodos tratará de hacer lo mismo.

Por ejemplo si se tuviera un *cluster* de cuatro nodos cado uno con un voto, los votos que se espera recibir son cuatro, con el algoritmo de mayoría simple la mayoría es igual a la mitad de votos totales más uno, es decir tres votos.

Si existiera un fallo en el sistema de red y uno de los nodos se desconectara del resto del *cluster*, se produciría una situación de s*plit-brain* y se tendrían dos *cluster* uno con tres nodos (grupo X) y otro con uno (grupo Y), claramente el grupo X tendrá mayoría mientras que el grupo Y, perderá el quórum y apagará los servicios de alta disponibilidad que ejecutaba.

Una vez que el *cluster* X gana el quórum, la configuración del *cluster* cambia y procede a aislar al nodo del *cluster* Y, una vez se completa el proceso de aislamiento del nodo, el grupo X podrá acceder al sistema de archivos compartidos.

Un empate de 50% no sería suficiente para ganar el quórum, si en el ejemplo anterior el grupo X y Y tuvieran dos nodos cada uno, no habría forma segura de proceder, ya que ambos grupos tratarían de levantar los servicios, poniendo en riesgo el sistema de archivos.

En un *cluster* formado solo por dos nodos por lo general se deshabilita el quórum, ya que siempre se tendrá un empate en la votación, y se confía exclusivamente en el sistema de *fencing* para que apague al nodo con fallas. Este tipo de *cluster* se conoce como *cluster* de alta disponibilidad activo/pasivo.

Otra opción cuando se tiene un *cluster* de solo dos nodos es utilizar un quórum de disco o qdisk, que consiste en que cada nodo escriba de forma periódica en un disco compartido verificando así que se encuentra activo, sin embargo se recomienda emplear este sistema solamente si el almacenamiento compartido utiliza un sistema independiente del sistema de red, como por ejemplo una SAN con enlaces de fibra.

1.5.1.3 Virtual Synchrony

Es necesario que las operaciones del *cluster* ocurran en un determinado orden en cada uno de los nodos, este concepto se conoce como *virtual synchrony* (sincronización virtual) y Corosync lo implementa utilizando grupos de mensajes.

1.5.1.4 Totem

Corosync realiza la implementación del protocolo tótem, el protocolo define la forma en que los mensajes se envían. Existe un *token* que se pasa entre los nodos, solo el nodo que posea el *token* puede enviar mensajes, los que no lo posean, guardarán el mensaje que deseen enviar en memoria hasta que puedan acceder al *token*. El protocolo soporta RRP[50] (*Redundant Ring Protocol*), lo que permite utilizar una red adicional para el envío de mensajes, red que se utiliza como respaldo.

1.5.1.5 Fencing

Consiste en aislar un nodo del *cluster* a fin de proteger el almacenamiento compartido y los datos que contiene.

La detección de un nodo que ha dejado de responder se realiza mediante el parámetro "*totem token timeout*" (el valor por defecto es 238 ms), es decir, si no se recibe un *token* de un nodo después de ese periodo de tiempo se asume que el *token* se perdió y se envía uno nuevo, después de cierto número de *token* sin respuesta el *cluster* declara al nodo muerto, mientras que el resto de nodos reconfiguran al *cluster* y aíslan al nodo defectuoso.

El proceso encargado de aislar al nodo se llama *fenced*, el cual revisa la configuración del *cluster* en búsqueda de un dispositivo que permita apagar el nodo o por lo menos impedir su acceso al almacenamiento compartido.

---

[50] RRP: es un protocolo que permite a Corosync utilizar varios enlaces de red para el intercambio de mensajes garantizando su envío siempre que una red esté activa, para más información se recomienda revisar la referencia [101]

Los dispositivos usados para el aislamiento de un nodo pueden clasificarse en:

- PDU (*Power Distribution Units*): son dispositivos con varias tomas de alimentación que permiten administrar el suplemento de energía a un equipo vía web, SNMP[51], Telnet[52], etc.

- Dispositivos de control *Blade*: son sistemas instalados en el centro de datos que permiten apagar o encender de forma remota las cuchillas o servidores.

- *Lights-out devices*: son tarjetas que se instalan en cada servidor y que mediante red permiten administrar el encendido y apagado del nodo de forma remota.

1.5.2 COROSYNC

Es el núcleo del *cluster*, el resto de componentes interactúan entre sí por medio de este componente.

1.5.2.1 Características

- Diagnóstico y análisis de fallas

- Soporte para redes Ethernet e Infiniband

- Soporte para IPv4 e IPv6

- Soporte para autenticación y cifrado

- Soporte para envío de mensajes

- Soporte para interfaces redundantes

---

[51] *Simple Network Management Protocol* permite administrar dispositivos tales como enrutadores, conmutadores, impresoras, etc. en una red IP.

[52] Telnet: es un protocolo de red que permite acceder a una consola de comandos en un computador remoto.

- Envío de mensajes *multicast* y *unicast*

1.5.3 PACEMAKER

Pacemaker es un software de administración de recursos que mediante algoritmos y haciendo uso de un sistema de mensajes (Corosync o Heartbeat) brinda alta disponibilidad a los servicios o recursos del *cluster*.

1.5.3.1 Características

- Detección y recuperación de fallas a nivel de nodo o servicio.
- No necesita la utilización de un tipo especial de almacenamiento, inclusive puede funcionar sin un sistema de almacenamiento compartido.
- No requiere que un recurso tenga características especiales, si el servicio puede manejarse usando *scripts* puede ser administrado por un *cluster*.
- Soporta dispositivos *fencing* para asegurar la integridad de los datos.
- Puede emplearse para *clusters* grandes y pequeños.
- La configuración se replica de manera automática desde cualquier nodo.
- Permite realizar clonación de servicios, a fin de que el mismo esté activo en múltiples nodos.

1.5.3.2 Componentes

En la Figura 1.11 se indican los módulos que forman parte de Pacemaker, los rectángulos en color azul representan los componentes del proyecto Pacemaker, el rectángulo en color verde representa al demonio `lrmd`[53], el cual es un componentes

[53] lrmd: es el demonio que administra los recursos de cada nodo o recursos locales.

del proyecto Linux *High Availability*[54], el rectángulo rojo representa al software de comunicaciones Corosync. En la siguiente sección se detalla cada uno de estos componentes.

1.5.3.2.1 Local Resource Manager Daemon (lrmd)

Se ejecuta en cada nodo y se comunica con los agentes de recursos (*scripts*) del nodo para iniciar/detener servicios o monitorizarlos. Sirve como una interfaz para los diferentes tipos de agentes con los que Pacemaker es compatible.

1.5.3.2.2 Cluster Resource Management Daemon (crmd)

Su trabajo es servir de intermediario para ejecutar tareas de monitorización, arranque/parada de los recursos del *cluster*; puede describirse como un agente de mensajes entre *Policy Engine* (PE) y `lrmd`.

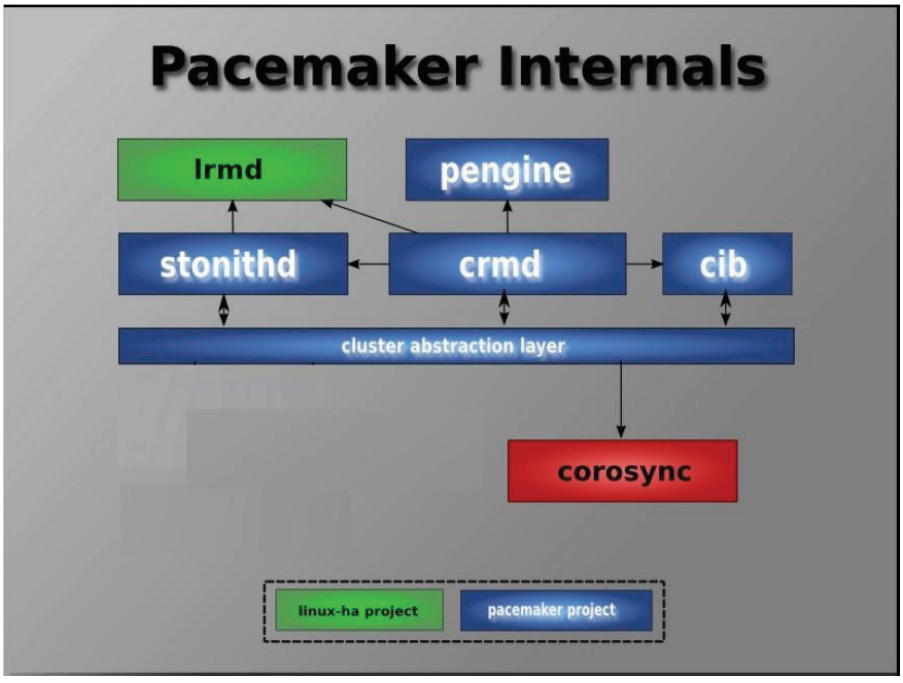

Figura 1.11. Arquitectura de Pacemaker [24]

[54] Linux *High Availability*: es el proyecto que creó el sistema de comunicación Heartbeat que se usaba en *clusters* Linux, para más información puede revisar [81].

Si *Policy Engine* decide que un recurso debe detenerse, envía un mensaje a `crmd`, el cual lo comunica a su vez a `lrmd`, el cual emplea el agente de recursos adecuado para que detenga el servicio.

Se encarga también de mantener actualizada la base de información del *cluster* (CIB). Así por ejemplo si PE desea activar una dirección IP, envía un mensaje a `crmd`, el que a su vez envía esta solicitud al `lrmd` de uno de los nodos del *cluster*, `lrmd` usa el agente IPaddr2 para cumplir con la tarea solicitada. Mientras tanto el demonio `crmd` habrá modificado el archivo CIB para que refleje el cambio realizado.

1.5.3.2.3 Cluster Information Base (CIB)

Es un archivo XML con la configuración y estado del *cluster*. Contiene toda la información del *cluster*, nodos, recursos, etc.

Se designa a un nodo para que contenga el archivo XML principal mientras que el resto de nodos tiene una copia, el nodo designado se conoce como *Designated Coordinator*, y como su nombre lo indica es el nodo encargado de coordinar las tareas del *cluster*, tal como desconectar un nodo, agregar o quitar recursos, etc.

En la Figura 1.12 se presenta una parte de un CIB, en las primeras versiones de Pacemaker era necesario crear y editar este archivo manualmente cada vez que se realizaba una modificación en el *cluster*, mientras que en las últimas versiones se dispone de la consola de comandos `crm`[55] o `pcs`[56] que mediante sencillos comandos modifican el archivo CIB, facilitando tareas como agregar recursos, modificarlos, agregar nodos, etc.

---

[55] crm: es una línea de comandos que permite configurar los recursos de un *cluster* Pacemaker. Para más información se recomienda revisar la referencia [82].

[56] pcs: Pacemaker *Configuration System* es la nueva consola de comandos usada para configurar un *cluster* de alta disponibilidad Pacemaker.

1.5.3.2.4 Policy Engine (PE)

Revisa el contenido de CIB y determina las acciones que hay que realizar para que el *cluster* alcance el estado que CIB describe. PE siempre se ejecuta en el nodo que actúa como *Designated Coordinator*.

```
<cib epoch="105" num_updates="35" admin_epoch="0" validate-with="pacemaker-1.2" cib-last-written="Thu Aug  7 14:47:53 2014" update-origin=
"nodo2" update-client="cibadmin" crm_feature_set="3.0.8" have-quorum="1" dc-uuid="1">
  <configuration>
    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.1.11-1.fc20-9d39a6b"/>
        <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-infrastructure" value="corosync"/>
        <nvpair id="cib-bootstrap-options-last-lrm-refresh" name="last-lrm-refresh" value="1403986928"/>
        <nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="false"/>
      </cluster_property_set>
    </crm_config>
    <nodes>
      <node id="1" uname="nodo1"/>
      <node id="2" uname="nodo2"/>
      <node id="3" uname="nodo3"/>
    </nodes>
    <resources>
      <primitive class="ocf" id="ip" provider="heartbeat" type="IPaddr">
        <instance_attributes id="ip-instance_attributes">
          <nvpair id="ip-instance_attributes-ip" name="ip" value="192.168.0.120"/>
          <nvpair id="ip-instance_attributes-cidr_netmask" name="cidr_netmask" value="32"/>
```

Figura 1.12. Archivo CIB de un cluster de prueba

1.5.3.2.5 Stonithd

Es el nombre del demonio de *fencing*, que como se indicó anteriormente se encarga de aislar un nodo para quitarlo del *cluster*, a fin de asegurar la integridad de los datos del almacenamiento compartido.

1.5.3.3 Tipos de cluster con Pacemaker [23]

De acuerdo a los requerimientos que el *cluster* tenga que cumplir, Pacemaker permite implementar diferentes tipos de *cluster*.

1.5.3.3.1 Cluster de alta disponibilidad activo/pasivo

Es la configuración más sencilla de realizar, consiste de dos nodos, denominados activo y pasivo, el pasivo funciona como respaldo del nodo activo y solo se activa si este deja de funcionar.

En este tipo de *cluster* no es necesario que el almacenamiento compartido tenga un sistema de archivos del tipo *cluster*, ya que los nodos no necesitan acceder simultáneamente al almacenamiento.

1.5.3.3.2 Cluster de alta disponibilidad tipo failover compartido

Consiste en combinar varios *clusters* del tipo activo/pasivo, pero que comparten un nodo de respaldo común. Por ejemplo en un *cluster* formado por cuatro nodos, tres de ellos están activos y brindan los servicios de alta disponibilidad, si uno de los nodos activos falla el cuarto nodo se activa y toma su lugar.

Esta configuración se conoce también como *cluster* de alta disponibilidad N + 1

1.5.3.3.3 Cluster de alta disponibilidad tipo activo/activo

En esta configuración dos o más nodos ejecutan los recursos a los que el *cluster* da alta disponibilidad, para este tipo de configuración es necesario disponer de un almacenamiento compartido con un sistema de archivos tipo *cluster*.

Cuando se tienen más de dos nodos esta configuración se conoce como *cluster* de alta disponibilidad N a N, los servicios se reparten entre los N nodos que forman el *cluster* si un nodo falla cualquiera de los nodos restantes puede tomar su lugar.

1.5.4 AGENTE DE RECURSOS

Es una interfaz estándar que sirve como intermediario entre el recurso o servicio y el software del *cluster*.

Un agente de recursos se identifica por su clase, proveedor y tipo. Como se indicó en la Sección 0 Pacemaker soporta los agentes de recursos: OCF, LSB y Systemd.

La clase se refiere a las especificaciones que se siguieron para escribir el agente, el estándar por defecto que se utiliza es el correspondiente a OCF.

El proveedor se refiere a la empresa o proyecto que escribió el código del agente, así por ejemplo existen agentes de clase OCF desarrollados por los proyectos Pacemaker, Hearbeat, Linbit[57], etc.

Por último, tipo se refiere al servicio que el agente maneja. Por ejemplo un agente de recursos encargado de iniciar, detener, monitorizar, etc. un servidor web Apache se identificará como `ocf:Heartbeat:apache`, un agente encargado de monitorizar el estado del *cluster* se identificará como `ocf:Pacemaker:clustermon`, etc.

1.5.5 RECURSOS [25]

Un recurso es cualquier servicio que el *cluster* pueda administrar mediante un agente de recursos. Puede tratarse de una dirección IP, un sistema de archivos, un bloque iSCSI, un servidor web, etc.

1.5.5.1 Propiedades generales de un recurso

De forma general todo recurso tiene las propiedades que se indican en la Tabla 1.7 y que determinan que agente de recursos usará Pacemaker para administrarlo.

Propiedad	Descripción
id	Nombre que se le asigna al recurso, puede contener letras y números, debe ser único.
class	Es el estándar usado para escribir el recurso, puede ser OCF, Systemd, LSB, etc.
type	El nombre del agente de recurso que se utilizará
provider	Especifica el proyecto o empresa que escribió el agente de recursos

Tabla 1.7. Propiedades de un recurso

[57] Linbit: es la empresa que da soporte a la tecnología de replicación de datos DRBD, para más información se recomienda revisar la referencia [30]

En la Línea de Comandos 1.1 se presentan los comandos para ver la lista de estándares y proveedores disponibles en un *cluster*.

```
# pcs resource standard
ocf      service
lsb      systemd
stonith
# pcs resource providers
heartbeat
pacemaker
```

Línea de Comandos 1.1. Comandos para ver los estándares y proveedores disponibles

1.5.5.2 Propiedades específicas de un recurso

Cada recurso dependiendo de su tipo, dispone de parámetros adicionales que pueden configurarse de forma opcional u obligatoria, por ejemplo un recurso del tipo IP tiene como parámetro obligatorio la dirección IP, mientras que de forma opcional puede configurarse la máscara de red, la dirección MAC[58], dirección de *broadcast*, entre otros parámetros. En la Línea de Comandos 1.2 se presentan algunas de las propiedades específicas de un recurso del tipo `IPaddr`.

1.5.5.3 Opciones de recursos

Es posible configurar opciones adicionales para un recurso, las mismas que determinan como el *cluster* debe controlarlo. En la Tabla 1.8 se presentan las opciones principales, y el valor por defecto, a continuación se detalla cada una de ellas.

[58] MAC: es un identificador único asignado a una interfaz de red.

1.5.5.3.1 Priority

En el caso que no existan suficientes recursos de hardware para tener activos todos los recursos, el *cluster* detendrá aquellos que se les haya asignado una baja prioridad.

```
# pcs resource describe ocf:heartbeat:IPaddr

Resource options:

ip (required): The IPv4 (dotted quad notation) or IPv6 address
(colon hexadecimal notation) example IPv4 "192.168.1.1".

cidr_netmask: The netmask for the interface in CIDR format
(e.g., 24 and not 255.255.255.0) If unspecified, the script
will also try to determine this from the routing table.
```

Línea de Comandos 1.2. Descripción de las propiedades específicas del recurso IP

1.5.5.3.2 Target-role

Determina cual es el estado por defecto del recurso, los estados posibles son detenido (*Stopped*), iniciado (*Started*) o maestro (*Master*).

Opción	Valor por defecto
Priority	0
Target-role	*Started*
Is-managed	*true*
Requires	(Depende de otros parámetros del *cluster*)
Migration-threshold	*INFINITY* (Deshabilitado por defecto)
Failure-timeout	0 (Deshabilitado por defecto)

Tabla 1.8. Opciones de un recurso

1.5.5.3.3 Is-managed

Determina si el *cluster* puede administrar el recurso, es decir si el *cluster* está encargado de iniciar y detener el recurso.

1.5.5.3.4 Requires

Indica las condiciones que deben cumplirse para que el recurso pueda iniciar. Los valores posibles son:

- *Nothing*: el *cluster* siempre puede iniciar el recurso, no existen restricciones.
- *Quorum*: el *cluster* solo arranca el recurso si la mayoría de nodos del *cluster* están activos. Si la propiedad *stonith-enabled* tiene el valor falso, los recursos tendrán está opción por defecto.
- *Fencing*: el *cluster* puede iniciar este recurso si existe quórum y cualquier nodo con fallas ha sido apagado (*fenced*).

1.5.5.3.5 Migration-threshold

Determina las veces que un recurso puede fallar en un nodo antes de determinar que el nodo no es capaz de ejecutar el mismo.

1.5.5.3.6 Failure-timeout

Determina cuantos segundos esperar antes de permitir al recurso arrancar otra vez en un nodo en el que el recurso presentó un fallo.

1.5.5.4 Operaciones de monitorización de un recurso

Es posible monitorizar un recurso a fin de determinar si está funcionando de forma correcta, por defecto el *cluster* monitoriza el recurso en un intervalo de 60 segundos.

Las operaciones de monitorización tienen las propiedades que se indican en la Tabla 1.9.

Propiedad	Descripción
id	Es el nombre único que el sistema asigna a la operación de monitorización.
name	La acción que la operación tiene que realizar, las opciones son *monitor*, *start*, *stop*.
timeout	Cuanto tiempo debe el software del *cluster* esperar antes de determinar que el recurso ha fallado. Tiene un valor por defecto de 20 segundos.
On-fail	La acción que debe tomar el software del *cluster* si la acción de monitoreo falla. Las posibles opciones son: *Ignore*: asumir que el recurso no falló. *Block*: no realizar ninguna acción. *Stop*: detener el recurso y no iniciarlo en ningún otro nodo. *Restart*: detener el recurso y arrancarlo nuevamente, incluso en un nodo diferente. *Fence*: aísla el nodo en el que el recurso falló. *Standby*: mueve todos los recursos fuera del nodo en el que el recurso falló.
Enabled	Habilita o deshabilita la operación de monitoreo

Tabla 1.9. Propiedades de una operación de monitoreo

Por defecto si la operación de monitoreo falla el *cluster* aislará el nodo empleando el dispositivo de *fencing*.

Es posible crear operaciones de monitoreo al momento de crear el recurso o posteriormente, si se desea modificar los valores de la operación es necesario primero eliminar la operación y crear una nueva.

1.5.5.5 Restricciones de recursos

Permiten configurar el comportamiento de los recursos, existen restricciones de tres tipos.

1.5.5.5.1 Restricción de localización

Permite configurar un recurso para que prefiera ejecutarse o no en un nodo.

La configuración por defecto hace que el *cluster* trate a todos los nodos del *cluster* por igual y que los recursos puedan ejecutarse en cualquier nodo.

Puede configurarse un recurso que debe ejecutarse obligatoriamente en el nodo asignado. Si por ejemplo uno de los nodos del *cluster* tiene más capacidad de memoria y CPU que el resto, es posible obligar a los recursos más importantes a ejecutarse en ese nodo.

1.5.5.5.2 Restricción de orden

Permiten configurar el orden en el que se ejecutan los recursos del *cluster*, por ejemplo, un recurso del tipo base de datos deberá ejecutarse siempre luego del recurso dirección IP que se haya asignado a esa base.

Si el primer recurso se detiene, el software del *cluster* detendrá también al segundo recurso, hasta que el primero este activo nuevamente, si no es posible volver a activarlo el segundo recurso tampoco se volverá a activar.

1.5.5.5.3 Restricción de colocación

Permiten configurar en donde deben ejecutarse un recurso tomando como referencia donde se ejecuta otro recurso, por ejemplo es posible configurar que la base de datos pueda ejecutarse solamente en el nodo en el que se ejecuta la dirección IP asociada con esa base de datos.

1.5.5.6 Tipos de recursos

Pacemaker clasifica los recursos en las categorías que se presentan a continuación:

1.5.5.6.1 Recurso de tipo primitivo

Es un recurso que no está agrupado, que no depende de otro recurso para funcionar. En el Archivo de Configuración 1.1 se presenta un recurso IP tal como se encuentran en el archivo `CIB.xml` de un *cluster* de prueba.

```
<primitive id="Public-IP" class="ocf" type="IPaddr"
provider="heartbeat">

<instance_attributes id="params-public-ip">

<nvpair id="public-ip-addr" name="ip" value="10.14.14.14"/>

</instance_attributes>

</primitive>
```

Archivo de Configuración 1.1. Ejemplo de un recurso primitivo

La mayoría de los servicios que un *cluster* maneja son de este tipo , como por ejemplo un sistema de archivos, un volumen lógico, una máquina virtual.

1.5.5.6.2 Recurso de tipo agrupado

Es un conjunto de recursos primitivos que tienen que estar localizados en un mismo servidor, además de iniciar y detenerse de una forma ordenada. Un ejemplo de un recurso agrupado sería un servidor de correo y la dirección IP asociada a ese servidor, como se presenta en el Archivo de Configuración 1.2.

```
<group id="ServidorEmail">

<primitive id="Public-IP" class="ocf" type="IPaddr"
provider="heartbeat">

<instance_attributes id="params-public-ip">

<nvpair id="public-ip-addr" name="ip" value="10.14.14.1"/>

</instance_attributes>

</primitive>

<primitive id="Email" class="lsb" type="sendmail"/>

</group>
```

Archivo de Configuración 1.2. Ejemplo de un recurso agrupado

No existe un límite para el número de recursos primitivos que pueden agruparse, los recursos primitivos inician en el orden en el que aparecen dentro del grupo, es decir en el ejemplo presentado en el Archivo de Configuración 1.2, primero se ejecutará el recurso IP, y en segundo lugar el servidor de correo, si el primer recurso no puede arrancar, tampoco lo hará el segundo.

1.5.5.6.3 Recurso de tipo clon

Este tipo de recurso permite que varias copias del mismo estén activas en diferentes nodos, así por ejemplo es posible clonar un recurso del tipo dirección IP en los nodos de un *cluster* para realizar balanceo de carga.

Es posible clonar un recurso de tipo primitivo o agrupado, siempre que el recurso soporte esta configuración, así por ejemplo clonar un recurso del tipo sistema de archivos o base de datos podría llegar a corromper la información que esos recursos manejan.

1.5.5.6.4 Recursos de estado múltiple

Son una variante de los recursos de tipo clon, ya que asignan roles a los recursos clonados, uno de los recursos asume el rol de maestro (*master*) y el otro el rol de esclavo (*slave*), así por ejemplo en el caso de un dispositivo DRBD, el que actúa como primario se le asigna el rol de maestro, mientras que al servidor DRBD secundario se le asigna el rol de esclavo.

Si el servidor con el rol maestro falla el *cluster* promueve al recurso clon esclavo y le asigna el rol de maestro.

1.5.6 PACEMAKER REMOTE

Una de las características de las versiones 1.1.10 y posteriores de Pacemaker es la posibilidad de administrar recursos que se ejecuten dentro de una máquina virtual o física sin tener que instalar el software del *cluster* en esa máquina.

En versiones anteriores esto podía hacerse solamente agregando la máquina como un nodo más del *cluster*, lo que era un problema debido a problemas de escalabilidad.

Esta nueva característica permite integrar una máquina virtual o física al *cluster* como un nodo en el que es posible ejecutar y monitorizar recursos, pero que no tiene voto al momento de decidir el quórum del *cluster*.

Esto se consigue mediante la instalación del programa `pacemaker_remote` en un nodo virtual implementado mediante KVM[59] o LXC[60], o en un nodo físico que ejecuta un sistema operativo Linux.

1.5.7 PROPIEDADES DEL CLUSTER

Existen ciertos parámetros que determinan el comportamiento del *cluster* ante determinadas situaciones.

En la Tabla 1.10 se presentan los parámetros más importantes, su valor por defecto y a continuación se realiza una breve descripción de estos parámetros.

Parámetro	Valor por defecto
no-quorum-policy	*stop*
Symmetric-cluster	*true*
Stonith-enabled	*true*
Stonith-action	*reboot*

Tabla 1.10. Propiedades de un cluster Pacemaker

[59] KVM: es una plataforma de virtualización *open source* que permite virtualizar sistemas operativos Linux y Windows.

[60] Linux Containers: es una tecnología de virtualización que permite ejecutar varios sistemas operativos Linux en un único servidor físico que tenga el kernel Linux instalado.

1.5.7.1 No-quorum-policy

Determina las acciones a tomar con los recursos cuando el *cluster* no tiene quórum, las posibles opciones son:

- *Ignore*: el *cluster* continúa administrando los recursos como si nada hubiera pasado.
- *Freeze*: igual que la opción anterior, pero el *cluster* no trata de iniciar los recursos en los nodos que no fueron afectados por el error.
- *Stop*: con esta opción el *cluster* detendrá todos los recursos que se estén ejecutando en los nodos con errores.
- *Suicide*: esta opción provoca que el *cluster* aísle (*fence*) todos los nodos con error.

1.5.7.2 Symmetric-cluster

Este parámetro indica si los recursos del *cluster* pueden ejecutarse en cualquiera de los nodos que forman el *cluster*, o si por el contrario es necesario especificar el lugar en el que los recursos se deben ejecutar.

1.5.7.3 Stonith-enabled

Esta propiedad indica que si no es posible detener los recursos que se ejecutan en un nodo que presenta fallas, la acción a tomar es aislar el nodo mediante el dispositivo de *fencing* que se haya configurado.

Si el valor de esta propiedad es *true*, el *cluster* no permitirá ejecutar ningún recurso hasta que al menos un dispositivo de *fencing* haya sido agregado al *cluster*.

1.5.7.4 Stonith-action

Es la acción por defecto que realiza el dispositivo de *fencing* contra el nodo que presenta un error. Las opciones son reiniciar el nodo o apagarlo.

1.5.8 HERRAMIENTAS DE CONFIGURACIÓN DE PACEMAKER: PCS

En las primeras versiones de Pacemaker la configuración del *cluster* se tenía que hacer editando un archivo XML, posteriormente la configuración se podía realizar mediante la consola `crm`, sin embargo a partir de la versión Pacemaker 1.1.8, se descontinuó su uso y se emplea en su lugar la consola `pcs`, e inclusive es posible configurar el *cluster* mediante una interfaz web.

La consola `pcs` contiene una serie de comandos que permite configurar Corosync y Pacemaker de forma fácil, sin embargo el *cluster* debe tener Pacemaker en su versión 1.1.8 o mayor, y Corosync en la versión 2.0 para poder utilizar `pcs`.

1.6 TECNOLOGÍAS DE ALMACENAMIENTO

En esta sección se presentarán algunos conceptos para entender el almacenamiento compartido que el *cluster* de alta disponibilidad necesitará.

1.6.1 RAID (REDUNDANT ARRAY OF INDEPENDENT DISK) [26] [27]

Esta tecnología permite la replicación de datos entre múltiples discos, asegurando así la tolerancia a fallos mediante redundancia. Es posible implementar RAID a nivel de hardware o software. En el primer caso es necesaria una tarjeta destinada específicamente para ello, mientras que en el segundo caso un programa se encarga de replicar los datos entre los discos duros. En Linux se emplea el software `md`[61] para implementar RAID por software, el cual soporta niveles de RAID 0, 1, 4, 5, 6 y 1+0.

---

[61] Para mayor información sobre el software `md` se recomienda revisar la referencia [83]

1.6.1.1 RAID implementado mediante software

En este tipo de RAID el procesador realiza las operaciones necesarias a nivel de kernel, en lugar de utilizar hardware especializado. El kernel se encarga de organizar los datos en varios discos a la vez que presenta al sistema operativo un único dispositivo virtual.

1.6.1.2 RAID implementado mediante hardware

En hardware RAID se utilizan procesadores especializados que se encuentran en tarjetas o controladores de discos, y son esos procesadores los que realizan las tareas de administración del RAID.

Existen tarjetas controladoras de RAID, a las que se conectan los discos duros, existen también gabinetes externos que contienen controladores SCSI y que forman una SAN. En ambos casos el arreglo lo maneja exclusivamente el sistema operativo del gabinete o controladora.

Las soluciones de hardware de alto costo son más rápidas que una solución por software, además no agregan carga al CPU. Sin embargo la implementación de RAID de Linux puede superar a algunas soluciones de hardware de gama baja, debido a que los CPU actuales son mucho más rápidos y a que el código del programa `md` se encuentra mejorando constantemente.

1.6.1.3 Niveles de RAID

El nivel de RAID que se vaya a utilizar dependerá de las aplicaciones que correrán sobre el dispositivo RAID y del presupuesto del que se disponga.

Los niveles de RAID no están organizados de forma jerárquica y cada uno tiene diferentes características de desempeño y redundancia. Por ejemplo los niveles más rápidos no ofrecen más confiabilidad que la que ofrecería un único disco. La elección debe tomar en cuenta que un nivel RAID puede brindar redundancia pero no un buen rendimiento.

1.6.1.3.1 RAID 0

También se denomina *striping* y consiste en combinar múltiples discos para presentar la ilusión de un único disco de gran tamaño. Los discos se combinan para que la escritura de datos sea entrelazada (*interleaving*) de tal forma que el acceder a un archivo de gran tamaño, se accede a todos los dispositivos que forman el RAID, lo que mejora el desempeño de los discos, sin embargo si un disco falla se perderán todos los datos, eliminando la confiabilidad que este tipo de RAID brinda.

1.6.1.3.2 RAID 1

Se lo conoce también como *mirroring* y consiste en crear una copia exacta del contenido de un disco en uno más discos, si un disco falla los otros pueden tomar su lugar haciendo que este sistema tenga gran confiabilidad. El inconveniente de este tipo de arreglo es que la escritura en disco se demora más debido a que los datos deben escribirse en varios discos.

En este nivel el espacio del que se dispondrá será igual al tamaño del disco multiplicado por $1/n$ donde n es el número de discos del arreglo, mientras que la tolerancia a fallos será $(n - 1)$ discos.

1.6.1.3.3 RAID 4

En este arreglo los datos se distribuyen de manera similar que en RAID 0, pero un disco que se conoce como disco de paridad se encarga de almacenar un *checksum* de los datos, si alguno de los discos falla la información en el disco de paridad permite recuperar los datos perdidos.

En este nivel el espacio total del que se dispondrá será igual al tamaño del disco multiplicado por $(1-1/n)$ donde n es el número de discos del arreglo, mientras que el número máximo de discos que pueden dañarse es uno. Para implementar este RAID es necesario un mínimo de 3 discos. Así por ejemplo si se tiene tres discos de 1 TB, el tamaño del arreglo será 2 TB aplicando la Ecuación 1.2.

```
n * Tamaño disco * (1-1/n) = Tamaño disponible
```

Ecuación 1.2. Tamaño del arreglo en RAID 4

Dado que para cada escritura en el disco es necesario realizar el *checksum* de los datos la velocidad de escritura se ve disminuida.

1.6.1.3.4 RAID 5

Tiene el mismo funcionamiento que RAID 4 con la diferencia que la información de paridad se escribe en todos los discos del arreglo, el tamaño que puede alcanzar se calcula con la Ecuación 1.2. Al igual que en RAID 4 si un disco falla la información del arreglo aún estará disponible para el usuario. Para implementar este RAID es necesario un mínimo de 3 discos.

1.6.1.3.5 RAID 6

Si en los arreglos de tipo RAID 4 y RAID 5 dos discos fallan el resultado es la pérdida de datos, para evitar esto RAID 6 aumenta los datos de *checksum* para incrementar la resistencia a fallas, pero es necesario contar por lo menos con 4 discos.

En este nivel el espacio total del que se dispondrá será igual al tamaño del disco multiplicado por (`1-2/n`) donde `n` es el número de discos del arreglo, mientras que el número máximo de discos que pueden dañarse es dos.

Si por ejemplo, se tiene cuatro discos de 1 TB, el tamaño del arreglo será 2 TB, valor que se obtiene aplicando la Ecuación 1.3.

```
n * Tamaño disco * (1-2/n) = Tamaño disponible
```

Ecuación 1.3. Tamaño del arreglo en RAID 6

1.6.1.3.6 RAID 1+0

Se conoce también como RAID 10, es una combinación de RAID 1 y RAID 0, tiene redundancia y un buen rendimiento, sin embargo es la configuración más costosa, debido a la forma en que utiliza el espacio de los discos que forman el arreglo.

Consiste en dos arreglos RAID 1 que se unen para formar un RAID 0, por lo que es necesario un mínimo de 4 discos como puede verse en la Figura 1.13.

En este nivel el espacio total del que se dispondrá será igual al tamaño del disco multiplicado por (2/n) donde n es el número de discos del arreglo, mientras que el número máximo de discos que pueden dañarse es un disco de cada RAID 1.

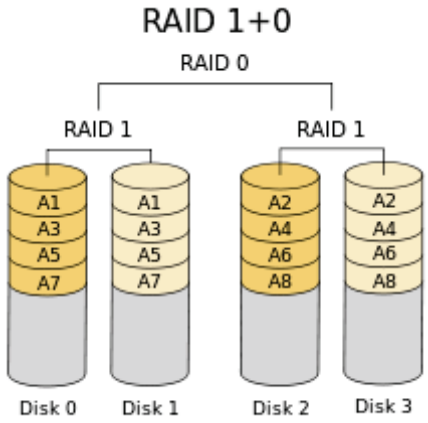

Figura 1.13. Diagrama de RAID 1+0 [28]

Si por ejemplo se tiene cuatro discos de 1 TB, el tamaño del arreglo será 2 TB, valor que se obtiene aplicando la Ecuación 1.4.

Dado que RAID 1+0 es la unión de dos arreglos RAID 1, las velocidades de escritura y lectura para estos dos niveles son similares [29].

```
n * Tamaño disco * (2/n) = Tamaño disponible
```

Ecuación 1.4. Tamaño del arreglo en RAID 1+0

1.6.2 DRBD [30]

El sistema DRBD (*Distributed Replicated Block Device*) es software libre que permite replicar la información de un dispositivo de bloque a través de la red.

Una forma de entender DRBD es imaginar que se trata de un RAID de nivel 1, solo que la replicación de datos se realiza entre dos dispositivos de almacenamiento pero de diferentes servidores.

El servidor activo se conoce como servidor primario, el servidor de pasivo se denomina servidor secundario, y al dispositivo replicado se le denomina recurso DRBD.

1.6.2.1 Características

- El recurso DRBD puede crearse a partir de cualquier dispositivo de bloque que se utilice en Linux, como por ejemplo, una partición de disco o un disco completo, un disco RAID implementado mediante hardware o software o un volumen LVM.

- DRBD utiliza el puerto 7788 TCP para las comunicaciones entre los nodos DRBD, por lo que este puerto debe habilitarse en el firewall de ambos nodos.

- DRBD replica en tiempo real los datos del servidor primario al secundario, de tal forma que ambas copias sean idénticas.

- Los datos que se replican no están cifrados, lo que podría suponer un riesgo de seguridad.

1.6.2.2 Funcionamiento

DRBD está formado por un módulo del kernel y dos aplicaciones que son `drbdadm` y `drbdsetup` y sirven para crear y administrar el dispositivo DRBD.

La replicación de datos en DRBD se realiza del servidor primario a un servidor secundario, es al servidor primario al que los nodos se conectan para operaciones de lectura y escritura.

Mientras tanto el servidor secundario recibe los datos replicados desde el servidor primario y los escribe en su disco local, listo para activarse en caso de falla del servidor primario.

Cuando el servidor primario falla el servidor secundario adquiere el rol primario sin que el sistema que utiliza el recurso DRBD se vea afectado por el cambio, este procedimiento se conoce como conmutación por error y para que se realice de forma automática debe estar controlado por un administrador de recursos de un *cluster*, como por ejemplo Pacemaker.

1.6.2.3 Tipos de sincronización de datos en DRBD

DRBD utiliza tres formas de replicación de datos entre el nodo primario y secundario, que se revisan a continuación.

1.6.2.3.1 Replicación asincrónica

La operación de escritura se completa cuando se ha escrito en el disco del nodo principal y la información se ha enviado al búfer TCP local.

1.6.2.3.2 Replicación semi-sincrónica

La operación de escritura se completa cuando se ha escrito en el disco principal y los paquetes TCP han alcanzado el otro nodo.

1.6.2.3.3 Replicación síncrona

Los datos deben escribirse en el disco local y el disco remoto para que la operación se complete.

La replicación síncrona es la que se emplea y recomienda para entornos de producción y se conoce como protocolo C.

1.6.3 ISCSI [31]

La tecnología iSCSI se deriva de SCSI (*Small Computer System Interface*), que es a su vez un conjunto de estándares para conectar y transferir datos entre un computador y dispositivos externos, como unidades de CD, escáner, etc.

iSCSI (Internet SCSI) es una implementación de SCSI para redes IP, permite que un dispositivo SCSI que se encuentra en red funcione como un dispositivo local.

iSCSI está definido en el RFC[62] 3720 desde el año 2004, y ha sido ampliamente adoptado y soportado como un estándar para conectividad SAN, reemplazando en gran medida a almacenamientos basados en canales de fibra. Las soluciones de almacenamiento de alta disponibilidad y de bajo costo usan predominantemente iSCSI.

iSCSI puede describirse como un protocolo cliente/servidor, pero para evitar confusiones se emplea el término *target* para el servidor e *initiator* para el cliente.

1.6.3.1 Target iSCSI

El *target* es el nodo que tiene un dispositivo de bloque al que el *initiator* puede acceder a través de una conexión de red y escribir en este dispositivo como si se tratara de un disco local.

Para implementar un *target* iSCSI existen cuatro opciones de código abierto que pueden ser empleadas:

- *IET* (iSCSI *Enterprise Target*): es una implementación disponible como un módulo externo del kernel, es decir que no se planea incluirla en el mismo.

[62] Request For Comments: son un conjunto de publicaciones que describen estándares usados en Internet

Proviene de una implementación propietaria bajo licencia GPL. El desarrollo del proyecto se encuentra detenido.

- STGT (SCSI *Target Framework*): fue creado por un ex programador de IET con el fin de remplazar al *target* IET; puede emplearse para implementar *targets* iSCSI o SCSI. Red Hat adoptó STGT como su *target* iSCSI para RHEL 5 y 6. Al igual que IET, el desarrollo del proyecto se encuentra detenido.

- SCST (SCSI *Target Subsystem*): también se originó a partir del proyecto IET con el objetivo de arreglar todos los problemas que el target IET tenía. Tiene un gran número de usuarios y se contempló incluirlo en el kernel de Linux, pero al final se optó por incluir LIO como el *target* por defecto.

- LIO (Linux iSCSI Org): es un *target* genérico, del cual el *target* iSCSI es solo una parte. En el 2011 LIO venció a SCST en convertirse en el remplazo de STGT[63], como el *target* por defecto en el kernel de Linux.

1.6.3.1.1 IQN (iSCSI Qualified Name)

Es un identificador permanente y único para el *target* y el *initiator* iSCSI, sigue un formato específico, por lo general se genera de forma automática por el software del target iSCSI, pero es posible definirlo de forma manual.

El estándar usado puede verse en la Figura 1.14, en la que se indica los campos que forman el identificador IQN, el campo *type* identifica el formato usado, el segundo la fecha en la que se asignó el dominio de Internet de la organización a la que el target pertenece, el tercer campo *Naming Auth* corresponde a la autoridad que otorgo el uso del dominio de Internet a la organización, el último campo lo define también la autoridad que asignó el uso del dominio.

[63] Para más información se recomienda revisar la referencia [85].

Figura 1.14. Formato para crear un nombre IQN [32]

1.6.3.1.2 Portal iSCSI

Permite identificar en la red al nodo que actúa como *target* iSCSI, y está formado por la dirección IP del *target* iSCSI y un puerto TCP, por defecto se utiliza el puerto 3260.

1.6.3.1.3 ACL (Access Control List)

Permite configurar los *initiator* iSCSI que tendrán permiso para conectarse al *target* iSCSI y montar el dispositivo iSCSI.

1.6.3.1.4 LUN (Logical Unit Number)

Se denomina así al dispositivo de bloque que se pone a disposición del iSCSI *initiator,* este dispositivo puede ser una partición de disco, un volumen lógico, un disco RAID, etc.

1.6.3.2 Initiator iSCSI

Es el cliente que accede al almacenamiento iSCSI, el software empleado por defecto como *initiator* en sistemas Linux es el desarrollado por el proyecto Open-iSCSI[64].

Cada *initiator* tiene definido un IQN único, y que se genera al momento de instalar el software de forma automática, el mismo se encuentra en el archivo

[64] Para más información se recomienda revisar la referencia [84].

`/etc/iscsi/initiatorname.iscsi`, como se presenta en la Línea de Comandos 1.3.

```
# cat /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.1994-05.com.redhat:b7f474e4dc2
```

Línea de Comandos 1.3. Identificador IQN de un nodo iSCSI initiator

1.6.4 SISTEMA DE ARCHIVOS PARA CLUSTER

Un sistema de archivos del tipo *cluster* es aquel que puede utilizarse simultáneamente en varios sistemas, es decir que puede montarse al mismo tiempo en varios nodos de un *cluster* y permite a los mismos leer y escribir de forma simultánea en un almacenamiento compartido, como por ejemplo una SAN.

1.6.4.1 OCFS2 [33]

Es un sistema de archivos para *cluster*, creado por Oracle para facilitar la administración de bases de datos RAC[65] en sistemas Linux y Windows, pero actualmente sirve como un sistema de archivos *cluster* para cualquier tipo de archivos.

OCFS2 es una tecnología libre y de código abierto, pero sus últimas versiones son incompatibles con los sistemas operativos Red Hat, CentOS y Fedora[66], etc.

1.6.4.1.1 Características de OCFS2

- El acceso a los archivos se coordina a través de DLM.
- Soporta *clusters* de hasta 255 nodos.
- Soporta un tamaño máximo de 16 TB.

---

[65] Oracle Real Application Clusters: es una versión *cluster* de una base de datos Oracle.

[66] Fedora: es un sistema operativo para computadores de escritorio, libre y de código abierto basado en Linux.

- Todos los nodos de un *cluster* pueden leer y escribir directamente en el sistema de almacenamiento.

1.6.4.2 GFS2 [34]

Es un sistema de archivos para *cluster* que tiene como característica principal permitir el acceso coordinado de los nodos de un *cluster* al mismo dispositivo de bloque. GFS2 ofrece compartición de datos entre los nodos GFS2 del *cluster* con una vista consistente del sistema de archivos a lo largo de todo el *cluster*, lo que permite que procesos ejecutándose en diferentes nodos compartan archivos GFS2 como si fueran archivos de un sistema local. GFS2 está implementado mediante el módulo del kernel `gfs2.ko`.

Para mantener la integridad del sistema de archivos GFS2 utiliza un administrador de bloqueos (*lock manager*) para coordinar la lectura y escritura, cuando un nodo cambia datos en un sistema GFS2 el cambio es visible de manera inmediata para el resto de nodos del *cluster*.

1.6.4.2.1 Características de GFS2

- Soporta dispositivos de hasta 100 TB en sistemas de 64 bits, y 16 TB en sistemas de 32 bits.
- Permite agregar nuevos nodos de forma dinámica.
- Puede utilizarse como sistema de archivos para un solo nodo (no soportado por Red Hat).
- Soporta un máximo de 16 nodos.
- Necesita de un programa que coordine el acceso de los nodos al sistema de archivos, este trabajo lo realiza el programa `dlm_controld`.

- Para garantizar la integridad del sistema de archivos se recomienda utilizar GFS2 sobre un volumen lógico del tipo *cluster*, empleando CLVM.

- No es compatible con SELinux[67] por lo que se recomienda deshabilitarlo, debido a que afecta el desempeño de GFS2.

1.6.4.2.2 Conceptos empleados en GFS2

- Nodos GFS2: son los nodos en los que se montará el sistema de archivos GFS2.

- Nombre de la Tabla de Bloqueos (*Lock Table Name*): es un identificador único formado por dos campos separados por dos puntos, estos campos son el nombre del *cluster* que utilizará el sistema de archivos y el nombre que se asignará al sistema de archivos, por ejemplo `clusterTest:gfs2Test`

- Número de registros (*Journals*): determina el número de registros para el sistema de archivos GFS2, se necesita un registro por cada nodo que vaya a montar el bloque GFS2, es posible agregar registros dinámicamente a medida que se agregan nuevos servidores.

1.6.5 CLVM

Es una versión clusterizada de LVM2, que permite a los nodos de un *cluster* acceder simultáneamente a un almacenamiento compartido, tal como una SAN. Esta opción está disponible a partir de la versión 6 del sistema operativo Red Hat, y está implementada mediante el demonio `clvmd` que está encargado de comunicar los cambios en el volumen lógico de tipo *cluster* a todos los nodos del *cluster*. Hay que aclarar que este tipo de volumen lógico necesita obligatoriamente la utilización de un sistema de archivos de tipo *cluster*, tal como OCFS2 o GFS2.

---

[67] SELinux: es un conjunto de módulos y políticas de seguridad usado en sistemas operativos CentOS, Fedora, etc.

1.7 VIRTUALIZACIÓN

En general la virtualización permite crear una capa lógica a partir de una capa física, por ejemplo la tecnología LVM crea un disco virtual a partir de uno o varios discos o particiones físicas sin repercusiones para el sistema operativo que utiliza el almacenamiento virtual [35].

De los diferentes tipos de virtualización que existen se explicarán ciertos detalles que son de interés para el presente Proyecto.

1.7.1 VIRTUALIZACIÓN DE RED

Este tipo de virtualización permite crear una red lógica o red virtual desacoplada del entorno de la red física.

Mediante software es posible recrear conmutadores, enrutadores y cortafuegos, es decir reproducir los servicios de la capa 2 a la capa 7 del modelo OSI, que un programa o un servidor necesita.

Mediante una red virtualizada es posible combinar diferentes redes físicas en una sola red virtual o al contrario crear múltiples redes virtuales a partir de una red física, como es el caso de una VLAN[68].

1.7.2 VIRTUALIZACIÓN DE ALMACENAMIENTO [36]

Es la abstracción lógica del almacenamiento físico, permitiendo que varios dispositivos físicos de almacenamiento aparezcan como un solo recurso a uno o varios clientes; así por ejemplo una SAN consiste de varios discos físicos que pueden agruparse en forma de LUN (*Logical Unit Number*) que son unidades de almacenamiento virtual que se presentan a un computador mediante SCSI. El computador utiliza la LUN como si se tratara de un disco de almacenamiento local, sin percatarse que se trata de un almacenamiento virtual.

---

[68] Virtual Local Area Network: consiste en dividir una única red en varios dominios de *broadcast* aislados entre sí.

En la Figura 1.15 se presenta un diagrama de la virtualización de almacenamiento, en la parte inferior se encuentra la infraestructura física, es decir los discos duros, que mediante la capa de virtualización se agrupan en discos virtuales que se ponen a disposición de un computador que los utiliza como si se tratará de un disco local.

Esta forma de virtualización es utilizada por las soluciones de almacenamiento de empresas como IBM, NetApp[69], EMC[70], etc.

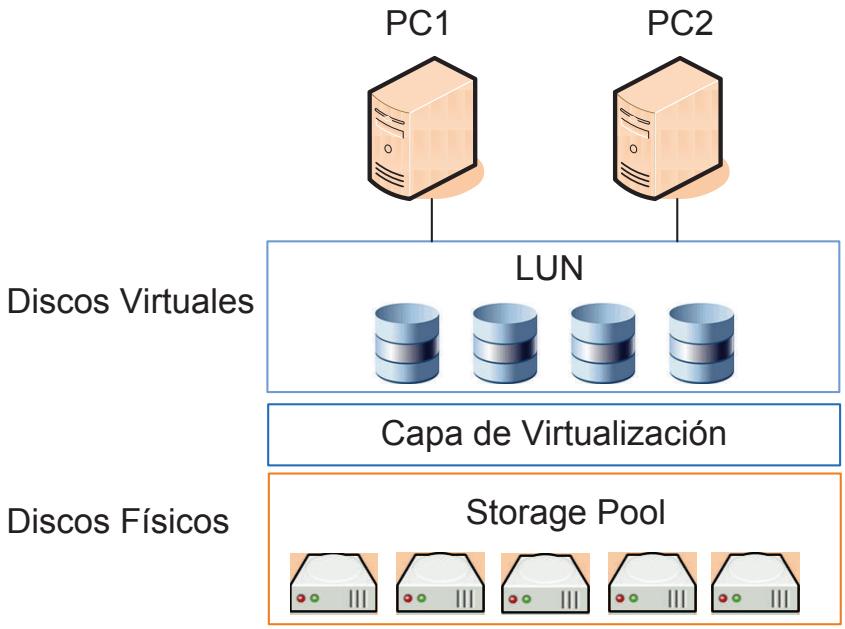

Figura 1.15. Esquema de virtualización de almacenamiento

1.7.3 VIRTUALIZACIÓN DE SERVIDORES [37]

Es la tecnología que permite crear entornos de hardware virtuales para que un sistema operativo se ejecute en esos entornos como si lo hicieran en uno real.

Existen diferentes opciones para virtualizar servidores, a continuación se indican las más importantes.

---

[69] NetApp: es una empresa estadounidense que fabrica soluciones de almacenamiento.

[70] EMC: es una empresa estadounidense dedicadas a fabricar soluciones para el almacenamiento y administración de datos.

1.7.3.1 Hipervisor

Hipervisor o *Virtual Machine Monitor* es el software encargado de manejar la asignación de recursos y memoria para las máquinas virtuales.

El hipervisor se encarga de iniciar, detener y administrar cada máquina virtual y coordinar el acceso de las diferentes máquinas virtuales al hardware del computador anfitrión.

1.7.3.2 Virtualización con hipervisor de tipo 2 o virtualización de sistema operativo

En este tipo de virtualización se tiene un computador que ejecuta un sistema operativo común y corriente, sobre el que se instala el hipervisor que se ejecuta de la misma manera que un programa cualquiera.

Programas como VMware Workstation y VirtualBox[71] son ejemplos de hipervisores de tipo 2. En la Figura 1.16 se indica el lugar que el hipervisor de tipo 2 ocupa en este tipo de virtualización, entre el sistema operativo del servidor y el sistema operativo de la máquina virtual.

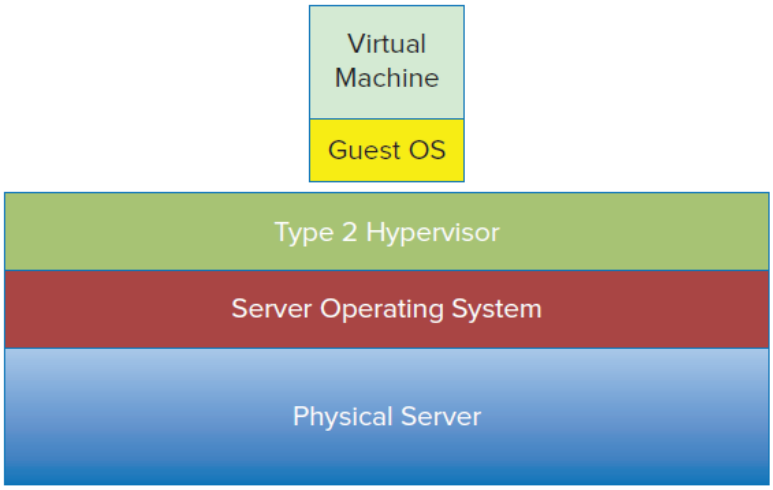

Figura 1.16. Hipervisor de tipo 2 [35]

[71] Para más información sobre este hipervisor se recomienda revisar la referencia [86] y [87].

1.7.3.3 Virtualización con hipervisor de tipo 1

En la virtualización con hipervisor de tipo 1, el hipervisor se ejecuta directamente en el hardware del servidor, por lo que a este hipervisor se lo conoce también como hipervisor tipo *bare metal*[72], es decir no se necesita que exista un sistema operativo instalado previamente en el servidor de virtualización, también se denominan hipervisores nativos. La Figura 1.17 representa este tipo de hipervisor.

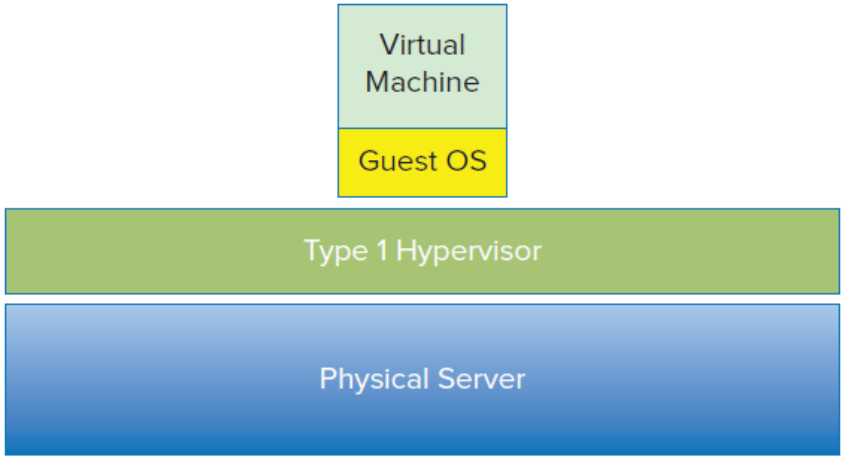

Figura 1.17. Hipervisor de tipo 1 [37]

1.7.3.3.1 Paravirtualización [38]

En este tipo de virtualización el sistema operativo se modifica para correr específicamente sobre el hipervisor, lo que implica reemplazar cualquier operación privilegiada que se necesite del procesador con llamadas al hipervisor, quien se encarga de realizar la operación solicitada.

Este tipo de virtualización no soporta sistemas operativos Windows, lo que limita su compatibilidad. Originalmente Xen[73] era un hipervisor que utilizaba solamente este tipo de virtualización.

---

[72] *Bare metal*: se refiere a un computador que no tiene instalado ningún sistema operativo o software.

[73] La Sección 1.7.4.2 de este documento trata sobre el hipervisor Xen.

1.7.3.3.2 Virtualizacion completa con Binary Translation [36]

Este tipo de virtualización soporta sistemas operativos que no han sido modificados. El hipervisor incorpora código que emula al CPU y que se encarga de ejecutar las operaciones privilegiadas que el sistema operativo virtualizado necesita, sin embargo este proceso de emulación o *binary translation*, necesita tiempo y recursos del sistema, lo que disminuye el desempeño con respecto al método anterior. Esta técnica de virtualización era empleada por VMware y Microsoft Parallels.

1.7.3.3.3 Virtualizacion completa asistido por hardware [38]

A partir del año 2006 los procesadores Intel y AMD incluyen las características VT-x y AMD-V, respectivamente, que facilitan la virtualización completa.

Este tipo de procesadores ofrece una forma de ejecutar instrucciones privilegiadas en el CPU a los sistemas operativos virtuales, con lo que ya no fue necesario utilizar la técnica *binary translation*, lo que mejora el desempeño que este tipo de virtualización ofrece en comparación a las dos tecnologías presentadas anteriormente.

El primer hipervisor que empleó virtualización asistida por hardware fue KVM[74], que es la primera solución de virtualización que se revisará.

1.7.4 SOLUCIONES DE VIRTUALIZACIÓN DE CÓDIGO ABIERTO

A continuación se presentan tres de las tecnologías de virtualización de código abierto que se emplean actualmente en ambientes Linux.

1.7.4.1 KVM [39] [40]

Es una solución del tipo virtualización completa asistida por hardware, los módulos que forman KVM permiten al kernel de Linux trabajar como un hipervisor del tipo *bare*

[74] *Kernel-based Virtual Machine* es un hipervisor de virtualización completa de código abierto.

metal. KVM necesita ejecutarse en servidores con procesadores que soporten tecnología de virtualización, en el caso de Intel procesadores que tengan habilitada la bandera VT-x y en el caso de AMD procesadores con la bandera AMD-V habilitada.

1.7.4.1.1 Componentes

Un esquema con los componentes de KVM se puede observar en la Figura 1.18.

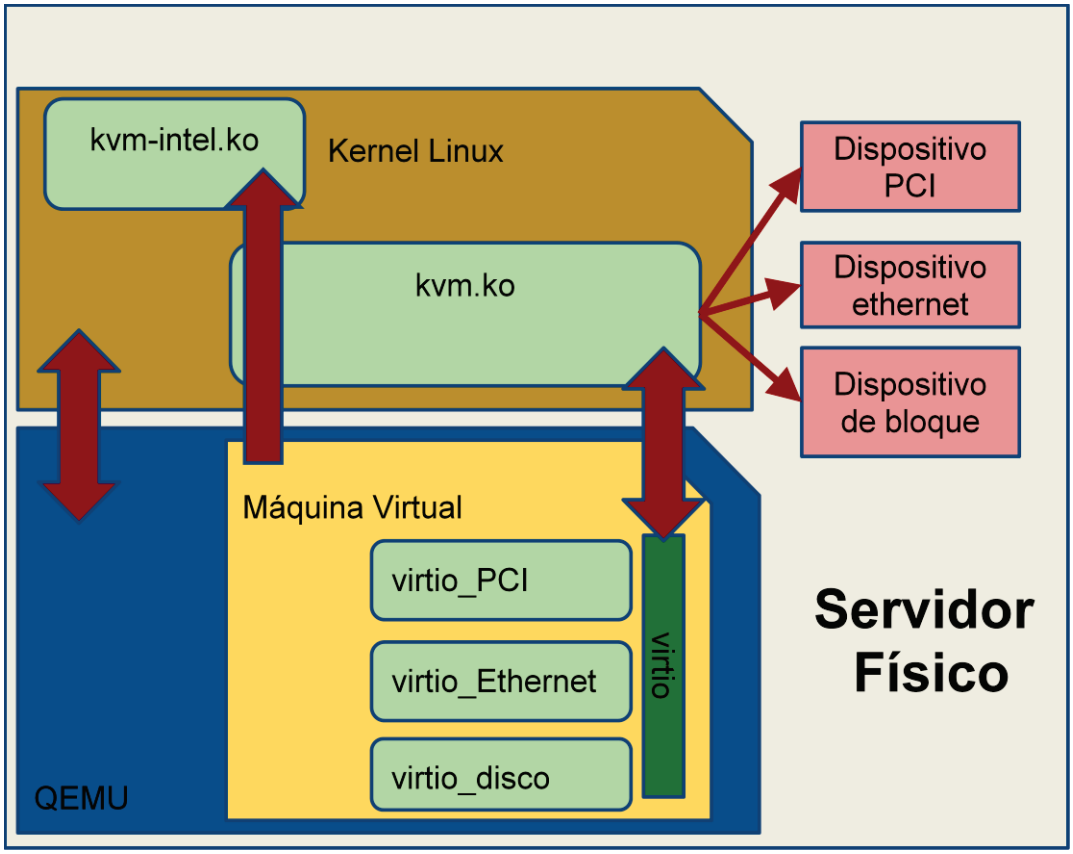

Figura 1.18. Arquitectura de KVM [39]

- `kvm.ko`: módulo para que el *kernel* de Linux funcione como un hipervisor de tipo 1.

- `kvm_intel.ko` o `kvm_amd.ko`: módulos para los procesadores Intel o AMD que permiten a KVM realizar la virtualización completa asistida por hardware.

- Virtio: es un proyecto que suministra drivers para que la máquina virtual interactúe con dispositivos de disco duro, tarjetas Ethernet y dispositivos PCI[75] virtuales [41].

1.7.4.1.2 Funcionamiento

KVM convierte el kernel de Linux en un hipervisor de tipo 1 o *bare metal* lo que hace que no sea necesario instalar un programa adicional que actúe como administrador de las máquinas virtuales.

El módulo `kvm.ko` genera el dispositivo `/dev/kvm` que es el que permite crear y ejecutar máquinas virtuales.

Mediante el módulo `kvm-intel.ko` o `kvm-amd.ko` es posible implementar el CPU virtual mediante un nuevo modo de ejecución llamado *guest mode*, el cual permite al procesador virtual ejecutar ciertas instrucciones especiales. Este modo de operación es posible solamente si la bandera VT-x o AMD-V existe y está habilitada en el procesador.

La emulación de dispositivos se realiza mediante una versión modificada de QEMU[76], el cual suministra: BIOS[77], puertos PCI, puertos USB[78], controladores IDE[79], tarjetas de red, de audio y video, etc.

1.7.4.2 XEN [42]

El tipo de virtualización que Xen utiliza es paravirtualización cuando se dispone de un sistema operativo invitado con kernel modificado, y de virtualización completa

---

[75] *Peripheral Component Interconnect*: es un conector de computación que permite conectar dispositivos tales como tarjetas Ethernet, tarjetas de audio, etc.

[76] QEMU: es un emulador y virtualizador de servidores físicos de código abierto.

[77] BIOS: es el software que utiliza un computador para iniciar el proceso de encendido del computador.

[78] USB: Universal Serial Bus es una interfaz estándar usada para conectar periféricos a un computador.

[79] IDE: es una interfaz estándar para conectar discos duros a un computador.

cuando no es posible realizar modificaciones al sistema operativo invitado, como es el caso de los sistemas operativos Windows.

1.7.4.2.1 Componentes

Como puede verse en la Figura 1.19 los componentes principales de Xen son:

- Hipervisor Xen: es un software o microkernel que se ejecuta directamente en el hardware, y es el único que tiene completo acceso al mismo, a través de `dom0`, que es el que interactúa con los recursos del hardware.

- `Dom0` (Dominio 0 o *Driver Domain*): es un sistema operativo huésped que utiliza el hipervisor para generar el entorno virtual para el resto de las máquinas virtuales o dominios. El dominio 0 es el único que tiene acceso a los recursos de disco y de red.

- `DomU` (*Unprivileged Domains*): se conocen así a los dominios sin privilegios o máquinas virtuales. Todos las interacciones que los sistemas operativos de las máquinas virtuales necesiten del hardware se realizan a través de `dom0`.

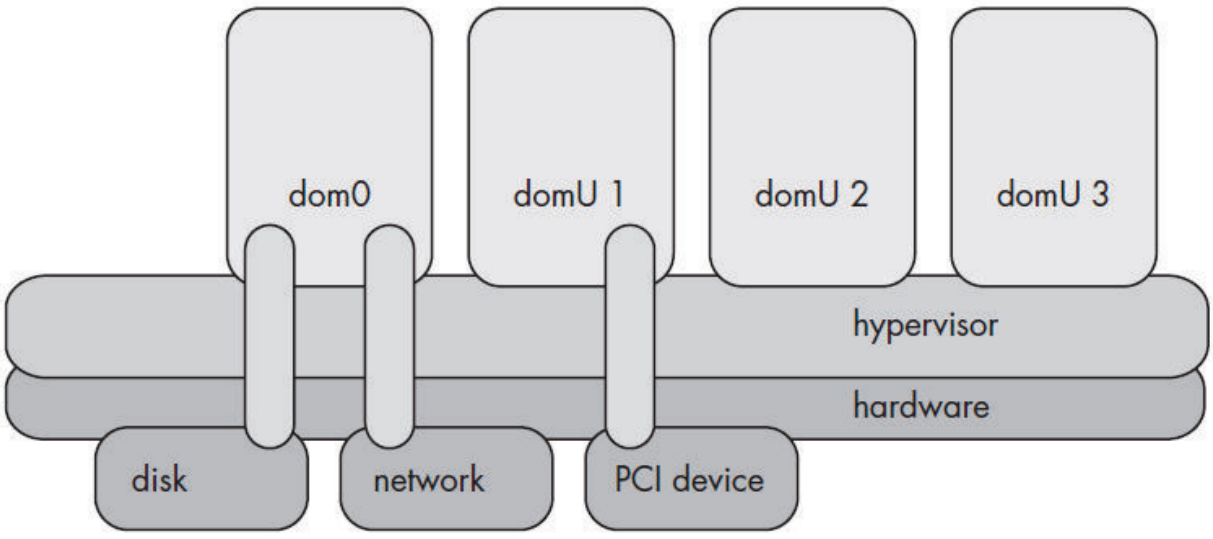

Figura 1.19. Componentes del hipervisor Xen [43]

1.7.4.2.2 Funcionamiento

El hipervisor arranca primero y ejecuta `dom0` como su primer dominio o sistema operativo invitado.

Mientras el hipervisor realiza tareas de virtualización de bajo nivel, `dom0` se encarga de crear los dominios sin privilegios, controlar el acceso de las máquinas virtuales a los recursos del sistema, controlar la cantidad de memoria que se asigna a las máquinas virtuales, etc.

1.7.4.3 QEMU [40]

QEMU es un emulador e hipervisor de tipo 2, es capaz de emular procesadores del tipo x86[80], SPARC[81], PowerPC[82], entre otros, es capaz de emular incluso dispositivos de red de las marcas Cisco y Juniper[83].

1.7.4.3.1 Funcionamiento

En QEMU el hardware que se desea emular se denomina *Target*, mientras que la máquina real se denomina *Host*, la emulación se consigue traduciendo el código *Target* a código *Host* mediante el módulo TCG (*Tiny Code Generator*) como se indica en la Figura 1.20.

En base al software del proyecto QEMU se creó un proyecto llamado `qemu-kvm` que permite a KVM utilizar la capacidad de emulación de dispositivos de QEMU, tal como puertos PCI, tarjetas de audio, teclado, etc. y utilizar esos dispositivos en los servidores virtuales de KVM.

---

[80] x86: es una familia de microprocesadores Intel.

[81] SPARC: es una arquitectura de microprocesadores desarrollado por Sun Microsystems.

[82] PowerPC: es una arquitectura de microprocesadores desarrollados por Motorola, IBM y Apple.

[83] Cisco, Juniper: son empresas dedicadas a la manufactura y venta de dispositivos de red.

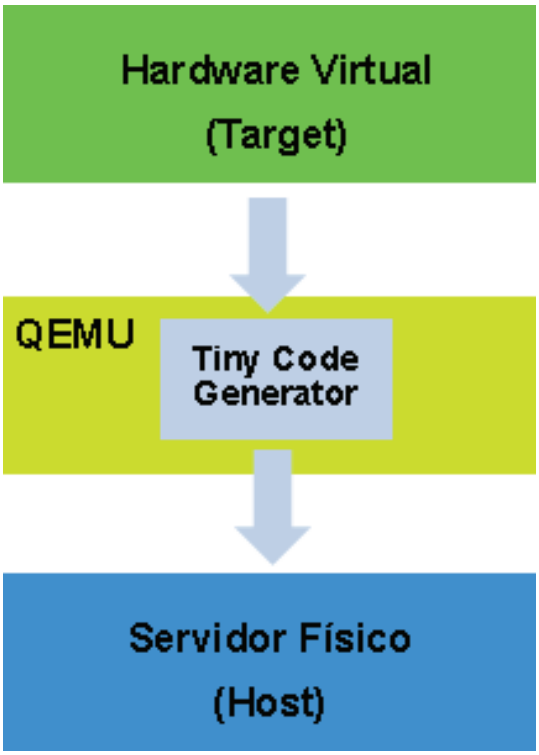

Figura 1.20. Funcionamiento Básico de QEMU

1.7.5 COMPARACIÓN DE LAS SOLUCIONES PRESENTADAS [42]

De las soluciones presentadas, KVM es la que ofrece mejores características, como se indica en la Tabla 1.11.

KVM tiene más sistemas operativos invitados compatibles que Xen, su desempeño es el más alto de soluciones libres o privadas de acuerdo al factor denominado *Virtualization Density*[84], que determina el número de máquinas virtuales que pueden ejecutarse en un servidor con determinado hipervisor.

KVM es el hipervisor más utilizado en las implementaciones de plataformas de *cloud computing*[85] basadas en OpenStack[86], con un porcentaje del 71% en comparación con Xen que es solo del 8% [44].

---

[84] Para mayor información se recomienda revisar la referencia [88].

[85] *Cloud Computing*: consiste en ofrecer servicios como almacenamiento, procesamiento o aplicaciones a través de Internet.

KVM forma parte también de la Open Virtualization Alliance [45], una organización formada por empresas tales como IBM, HP, Red Hat, Intel, NetApp, etc., cuyo objetivo es promover la utilización de KVM como plataforma de virtualización. Otra característica importante es su integración con el software de *clustering* Pacemaker, que permite que recursos que se ejecuten dentro del servidor virtual, puedan ser administrados por el software del *cluster*.

Por estos motivos se decide utilizar KVM como el software de virtualización para el *cluster* de alta disponibilidad del presente Proyecto de Titulación.

1.8 REVISIÓN DEL SOFTWARE DE VIRTUALIZACIÓN SELECCIONADA

Adicionalmente a los componentes presentados en la Sección 1.7.4.1 la plataforma de virtualización basada en KVM necesita de otras herramientas que funcionan como complementos para KVM, una de ellas, como se indicó al final de la Sección 1.7.4.3 es código del proyecto QEMU, otros proyectos relacionados con KVM son libvirt y virsh.

1.8.1 LIBVIRT [40]

Es una API que se utiliza junto con otros programas para administrar plataformas de virtualización como KVM, Xen, VMware ESX, etc.

El programa `libvirtd` es un demonio de Linux con el que se comunican los programas de administración para realizar tareas como crear, detener, migrar máquinas virtuales.

La herramienta de administración que se utilizará para administrar KVM es `virsh`, la misma que se revisa a continuación.

---

[86] OpenStack: es el software de código abierto más utilizado para implementar una plataforma de *cloud computing*.

Características	KVM	Xen	QEMU
Licenciamiento	GPLV2	GPL	GPL, LGPL87
Empresa que respalda el proyecto	Red Hat Inc. , Comunidad Linux.	Citrix System, Comunidad Linux.	Comunidad Linux.
OS anfitrión	Linux, FreeBSD	Linux, Solaris, NetBSD	Windows, Linux, Mac OS X, Solaris, FreeBSD
OS Invitado	Linux, Windows, FreeBSD, Solaris	FreeBSD, Linux, NetBSD, Windows XP, Windows 2003	Windows, Linux, Mac OS X, Solaris, FreeBSD, Android
Hardware	x86 ,x86-64	x86 ,x86-64, ARM	x86 ,x86-64, ARM, PowerPC.
Desempeño del sistema operativo invitado	Cercano al del OS anfitrión.	Cercano al del OS anfitrión.	Bajo comparado con el OS anfitrión
Modo de funcionamiento	Hipervisor de tipo 1 asistido por hardware	Hipervisor de tipo 1 con paravirtualización y virtualización asistida por hardware	Emulación de hardware.
Live Migration88	Soporta	Soporta	No soporta
Seguridad	Certificación EAL4+89	No certificado	No certificado
Integración con Pacemaker	Sí	No	No
Precio	No tiene costo	No tiene costo	No tiene costo

Tabla 1.11. Comparación de las soluciones de virtualización

[87] *Lesser General Public License*: es una licencia de software libre, que no obliga a hacer público el código fuente.

[88] *Live Migration*: migración en caliente es la capacidad de mover una máquina virtual entre dos nodos sin tener que apagar la máquina virtual.

[89] La evaluación EAL de un producto informático se asigna luego del cumplimiento de ciertos criterios de seguridad.

1.8.2 VIRSH [40]

Es una consola de comandos que utiliza la API de `libvirt` para conectarse con un hipervisor y facilitar la administración de máquinas virtuales.

Mediante `virsh` es posible administrar máquinas virtuales que se ejecutan en un servidor local o remoto.

`Virsh` permite también controlar la red virtual a la que los servidores virtuales se conectan, un dispositivos de red se crea automáticamente una vez que se instala el software del proyecto `libvirt` y crea la red que permite al servidor invitado conectarse a la red externa.

1.8.2.1 Virt-install

Es un comando que facilita la creación de máquinas virtuales. Para crear un servidor virtual mediante `virsh` es necesario crear un archivo XML que contiene toda la información del hardware del servidor virtual, mientras que mediante `virt-install` es posible crear un servidor con una sola línea de comandos, como se indica en la Línea de Comandos 1.4.

En la Sección 1.8.4 se revisarán las opciones principales de este programa.

```
# virt-install --name ubuntu2 --ram 512 --disk
path=/mnt/discosVirtuales/ubuntu.img,size=5 --network
network:default --graphics vnc,listen=0.0.0.0,port=5901 --
cdrom /mnt/discosVirtuales/iso/ubuntu-12.04.1-server-i386.iso
```

Línea de Comandos 1.4. Creación de una máquina virtual empleando virt-install

En la Figura 1.21 se presenta un esquema de los programas que forman un servidor de virtualización KVM básico.

En el nivel correspondiente a las herramientas de administración es posible emplear también programas con interfaz gráfica e incluso programas que permiten realizar las

tareas de administración vía web, sin embargo estos programas utilizan el comando `virsh` y la API de `libvirt` para funcionar.

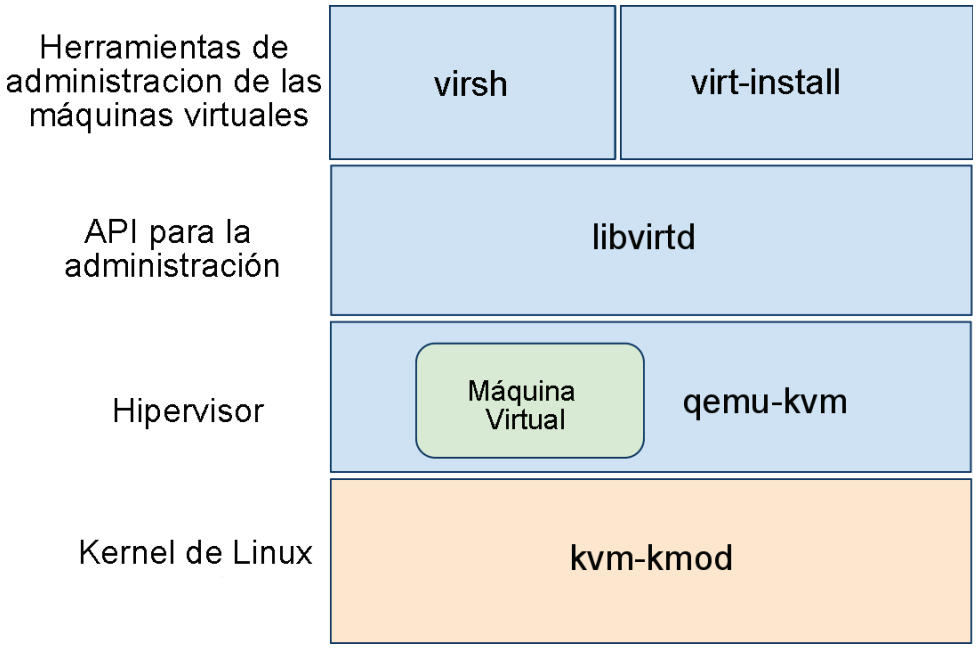

Figura 1.21. Capas de un servidor virtual KVM [46]

1.8.3 RED VIRTUAL [47]

Es necesario conectar el servidor virtual a un conmutador virtual de tal forma que pueda comunicarse con otros servidores virtuales, o con la red del servidor físico, para esto el servidor virtual cuenta con una interfaz de red virtual o VNIC la que se conecta a un conmutador virtual creado al instalar el software `libvirt`.

Este conmutador virtual se denomina `virbr0` y puede trabajar en varios modos, que se revisan a continuación.

1.8.3.1 Modo NAT

Es el modo de operación por defecto, permite a los servidores virtuales comunicarse con la red externa a través de la IP del servidor físico, empleando la traducción de direcciones de red (NAT).

1.8.3.2 Modo Bridge

En esta configuración el conmutador virtual se conecta a la red del servidor huésped y permite a los servidores virtuales acceder a LAN del servidor huésped con su propia dirección IP.

1.8.3.3 Modo Aislado

En esta configuración los servidores virtuales pueden comunicarse entre sí y con el servidor físico, pero no tienen acceso a la red externa ni recibir tráfico desde fuera.

1.8.4 OPERACIONES CON MÁQUINAS VIRTUALES

En esta sección se presentarán las tareas más comunes relacionadas con la administración de los servidores virtuales.

1.8.4.1 Creación de un servidor virtual

La creación del servidor virtual puede realizarse fácilmente empleando el comando `virt-install`, las opciones más importantes de este programa se indican en la Tabla 1.12.

El comando para crear una máquina virtual se presentó en la Línea de Comandos 1.4.

1.8.4.2 Administración del servidor virtual

Se revisarán a continuación las principales tareas administrativas relacionadas con los servidores virtuales.

1.8.4.2.1 Listar los servidores en ejecución

El comando presentado en la Línea de Comandos 1.5 genera una lista de los servidores hospedados en el nodo físico y el estado en el que se encuentran los mismos.

Opción	Descripción
`--name NOMBRE`	Especifica el nombre que tendrá el servidor virtual, no debe incluir espacios en blanco.
`--ram 1024`	Especifica la cantidad de memoria RAM asignada al servidor.
`--disk path="RUTA"`	Determina el lugar en el que se almacenará el disco duro del servidor virtual.
`--network network:default`	Indica la red virtual a la que se conectará el servidor virtual, por defecto se utilizará la red creada al momento de instalar libvirtd.
`--graphics PROTOCOLO`	Esta opción permite configurar el protocolo que los clients utilizarán para conectarse al servidor virtual, es posible emplear VNC[90] o SPICE[91].
`--cdrom "RUTA"`	Esta opción se utiliza para indicar que el servidor se instalar a partir de una imagen ISO.

Tabla 1.12. Opciones principales del comando virt-install

```
# virsh list --all
Id    Name                           State
----------------------------------------------------
4     ubusc3                         running
5     ubuserver                      running
```

Línea de Comandos 1.5. Presentar las máquinas virtuales en ejecución

1.8.4.2.2 Iniciar un servidor virtual

El comando presentado en la Línea de Comandos 1.6 permite arrancar una máquina virtual existente en el nodo de virtualización.

[90] Virtual Network Computing: es un protocolo que permite compartir el entorno gráfico de un computador a través de la red.

[91] SPICE: es un protocolo para acceder de forma remota al escritorio de un computador.

```
# virsh start ubuserver

Domain ubuserver started
```

Línea de Comandos 1.6. Encender un servidor virtual

Es posible también arrancar una máquina virtual desde un archivo de configuración XML, mediante el comando presentado en la Línea de Comandos 1.7.

```
# virsh create /mnt/xml/ubuserver.xml

Domain ubuserver created from /mnt/xml/ubuserver.xml
```

Línea de Comandos 1.7. Encender un servidor virtual a partir de un archivo XML

1.8.4.2.3 Detener un servidor virtual

En la Línea de Comandos 1.8 se indica el comando para detener un servidor virtual. En ocasiones este comando no es capaz de apagar el nodo virtual, por lo que es necesario emplear la opción `destroy`, con el comando presentado en la Línea de Comandos 1.9.

Esta opción realiza un apagado por la fuerza y garantiza que el servidor virtual ya no está encendido.

```
# virsh shutdown ubuserver

Domain ubuserver is being shutdown
```

Línea de Comandos 1.8. Apagar un servidor virtual

```
# virsh destroy ubuserver

Domain ubuserver destroyed
```

Línea de Comandos 1.9. Forzar el apagado de un servidor virtual

1.8.4.3 Migración de un servidor virtual [48]

Consiste en mover una máquina virtual de un servidor físico a otro, sin tener que apagar la máquina virtual, procedimiento que se conoce como *live migration*. Este procedimiento permite que en caso de falla del servidor de virtualización el funcionamiento de la máquina virtual no se interrumpa, y pueda seguir funcionando en otro servidor de virtualización.

Para que la migración de una máquina virtual sea posible es necesario que las máquinas virtuales se encuentren en un almacenamiento al que los nodos físicos que realizarán la migración puedan acceder, tal como un servidor NFS[92], un volumen GFS2, un volumen iSCSI, etc.

Es necesario también que los servidores físicos que participan en la migración tengan configuradas las mismas redes físicas y virtuales.

En la Línea de Comandos 1.10 se indica el comando que permite migrar la máquina de un servidor a otro, una vez que se ejecuta el comando el sistema solicita la clave de `root`, con lo que el servidor ya no se ejecuta en el `nodo3` sino que se ejecuta en el `nodo2` sin haber interrumpido su operación.

```
[nodo3]# virsh migrate --live ubuserver
qemu+ssh://nodo2/system
```

Línea de Comandos 1.10. Comando para migrar en caliente una máquina virtual

[92] Network File System: es un protocolo que permite acceder a un sistema de archivos de forma remota a través de una red TCP/IP

CAPÍTULO II

2. SITUACIÓN ACTUAL Y ANÁLISIS DE REQUERIMIENTOS

En este capítulo se revisarán los sistemas de adquisición y procesamiento del Instituto Geofísico, que serán administrados por el software del *cluster* a fin de brindar a estos sistemas alta disponibilidad.

Al inicio del capítulo se presentan algunos términos usados en sismología, posteriormente se describe los sistemas que se pretende ejecutar en el *cluster* de alta disponibilidad, los cuales son: SeisComP3[93], Earthworm[94] y ShakeMap[95].

Se indicará brevemente el funcionamiento de estos sistemas, características principales, de que programas están formados, etc., a fin de determinar la forma en que el *cluster* de alta disponibilidad controlará estos sistemas.

En la sección de Anexos se incluyen los procedimientos de instalación y configuración de los tres sistemas, documentación que es necesaria para poder instalarlos en los servidores virtuales que se crearán en el *cluster* de alta disponibilidad.

Una vez revisado el funcionamiento de los sistemas de adquisición y procesamiento, se determinarán los requerimientos de hardware que tienen los mismos, tal como espacio de disco, memoria RAM, uso de CPU, etc. para poder dimensionar los servidores virtuales en los que se instalarán estos sistemas.

---

[93] SeisComP3: es un software de adquisición y procesamiento de formas de onda sísmicas de código abierto para sistemas Linux.

[94] Earthworm: es un software de adquisición y procesamiento sísmico para plataformas Windows y Linux.

[95] ShakeMap: es un conjunto de *scripts* para la generación de mapas de movimiento sísmico.

2.1 INSTITUTO GEOFÍSICO DE LA ESCUELA POLITÉCNICA NACIONAL

El Instituto Geofísico de la Escuela Politécnica Nacional[96] es el principal centro de investigación en el área de la sismología y vulcanología del Ecuador, encargado también de la vigilancia y diagnostico de los eventos sísmicos y volcánicos, a fin de ayudar a reducir el impacto que esos eventos generan en la población.

Para realizar estas tareas el Instituto mantiene un programa de monitoreo en tiempo real con sensores sísmicos, que asegura la vigilancia permanente sobre volcanes activos y fallas tectónicas en el Ecuador continental e Insular.

Los sistemas de adquisición y procesamiento, que se ejecutan en los servidores del Centro Terras[97], recuperan la información de los sensores sísmicos para su procesamiento, detección de eventos y difusión a la autoridades competentes y al público en general.

Este Proyecto de Titulación tiene como objetivo virtualizar esos sistemas en un *cluster* de alta disponibilidad para tratar que los servicios que dichos sistemas brindan estén disponibles la mayor cantidad de tiempo posible.

2.2 TÉRMINOS UTILIZADOS EN SISMOLOGÍA [49]

Se definirán de forma breve algunos conceptos que se utilizarán en esta sección.

- Sensores sísmicos: son instrumentos que miden movimientos del suelo generados por un evento sísmico, volcánico o de otro origen, como una explosión fuerte.

---

[96] Para más información de la actividad del Instituto Geofísico se recomienda revisar la referencia [90]

[97] El Centro Terras se encuentra ubicado en Quito en el sexto piso de la Facultad de Ingeniería Civil de la Escuela Politécnica Nacional.

Los sensores modernos son completamente electrónicos, y disponen de un sistema de comunicación TCP/IP para recuperar las formas de onda en tiempo real. Los sistemas a virtualizar trabajan solamente con este tipo de sensor.

- Sistemas de adquisición. Los programas de adquisición se desarrollaron con el objetivo de unificar la recuperación de datos de los sensores sísmicos independientemente del fabricante. Para conseguir esto el programa de adquisición cuenta con módulos que le permiten comunicarse con sensores de diferentes marcas, formatos y protocolos.

- Sistemas de procesamiento. Son los programas encargados del filtrado de señales, visualización de formas de onda, detección y localización de sismos, etc. para lo que cuentan con algoritmos que a partir de variaciones en la amplitud de una forma de onda, causadas por un evento sísmico, determinan la magnitud de ese evento, su localización, la hora en la que sucedió, etc.

2.3 SISTEMAS DE ADQUISICIÓN Y/O PROCESAMIENTO DEL INSTITUTO GEOFÍSICO

El primer sistema que se describirá, SeisComP3, es al momento el más importante del Instituto Geofísico, realiza la adquisición de datos de aproximadamente 400 sensores sísmicos, se encarga del procesamiento de esas señales, determina la existencia de eventos sísmicos y disemina esa información a través de varios medios de comunicación.

El segundo sistema, Earthworm, se utiliza para adquirir datos de sensores analógicos ubicados en los volcanes Tungurahua y Cotopaxi, tareas de procesamiento y visualización de formas de onda.

El último sistema, ShakeMap, es un nuevo servicio que se pondrá a disposición del público, y que genera mapas que indican el movimiento experimentado en un área geográfica luego de un sismo de gran magnitud.

2.3.1 SEISCOMP3[50]

2.3.1.1 Introducción

Es un software para adquisición, procesamiento y distribución de datos desarrollado por GEOFON y Gempa[98].

En sus 7 años de existencia ha pasado de ser un conjunto de módulos de adquisición a ser una plataforma de software en tiempo real para el monitoreo de sismos, mediante el desarrollo e inclusión de programas de detección y localización de eventos, determinación de magnitud, generación de alarmas y difusión de boletines sísmicos.

SeisComP3 es el software más utilizado para adquisición, procesamiento e intercambio de datos sismológicos en tiempo real por institutos de sismología en todo el mundo [51].

2.3.1.2 Características

- Funciona solamente en sistemas operativos Linux de 32 o 64 bits.
- Módulos independientes para cada función.
- Uso de paquetes TCP/IP para la comunicación entre los módulos que forman el sistema.
- Comunicación con los sensores sísmicos mediante un protocolo basado en TCP/IP.
- Almacenamiento de las formas de onda adquiridas de forma local o remota.

[98] Geofon y Gempa son institutos dedicados a la investigación sismológica. Para más información se recomienda revisar la referencia [91] y [54] .

- Utiliza una base de datos MySQL para almacenar la información de sismos detectados.

- Los módulos pueden activarse, detenerse o monitorizarse mediante la línea de comandos o de forma gráfica.

Estas características hacen que SeisComP3 sea un sistema que el administrador de recursos del *cluster* sea capaz de manejar. Una vez revisado los componentes y su función en el sistema SeisComP3 se los clasificará según su importancia, para poder, en el siguiente capítulo, definir que agentes y con qué políticas el *cluster* los manejará.

2.3.1.3 Módulos de comunicación

SeisComP3 es un conjunto de módulos independientes que se comunican entre sí mediante mensajes TCP/IP, cada módulo puede suscribirse a uno o varios grupos de mensajes, dependiendo de la información que el módulo requiera para funcionar. Los grupos de mensajes son por ejemplo: PICK, LOCATION, EVENT, etc. Cada módulo envía y recibe información de los grupos a los que está suscrito.

2.3.1.3.1 Spread

Es el programa encargado del envío y recepción de mensajes entre los distintos módulos; utiliza un conjunto de herramientas de código abierto basadas en TCP/IP, denominado Spread Toolkit[99], lo que permite que los módulos de SeisComP3 se comuniquen estando inclusive en diferentes servidores, el programa utiliza por defecto el puerto 4803.

2.3.1.3.2 Scmaster

El módulo Scmaster se encarga de coordinar los componentes del sistema.

---

[99] Spread Toolkit: es un conjunto de herramientas software utilizadas para implementar sistemas de mensajes con alta disponibilidad. Para mayor información sobre Spread Toolkit se recomienda revisar la referencia [92]

Cuando un módulo desea conectarse al sistema realiza una solicitud de conexión a Scmaster, el cual responde con un mensaje de acuse de recibo que informa al módulo sobre su admisión o rechazo.

Si se aceptó la conexión del módulo, Scmaster le informa al módulo solicitante los grupos de mensajes a los que está suscrito.

En la Figura 2.1 puede verse la solicitud la conexión del módulo Scolv[100] al sistema SeisComP3. Es necesario llenar los campos `User` con el nombre del módulo que se conecta, el campo `Server` es la dirección IP del servidor que ejecuta Scmaster y Spread, `Timeout in sec` es el tiempo máximo que espera el cliente sin obtener una respuesta, `Primary group` es el grupo al que el módulo pertenece, y por último `Subscriptions` es la lista de grupos a los que el módulo se conectará.

Scmaster está encargado también de las operaciones de escritura y lectura de la base de datos, es el único módulo que interactúa con la base de datos, el resto de módulos accede a la base de datos solamente mediante él.

2.3.1.4 Módulos de adquisición y almacenamiento

2.3.1.4.1 SeedLink

Este módulo tiene dos funciones, la primera es la adquisición de datos en tiempo real desde los sensores sísmicos que se encuentran en el campo o desde otros servidores de datos sísmicos.

La segunda función es entregar, en tiempo real, los datos al resto de módulos del sistema, para ambas tareas utiliza el protocolo SeedLink[101], que es un protocolo de comunicaciones basado en TCP/IP que utiliza el puerto 18000 por defecto.

---

[100] Scolv: es el módulo que se utiliza para corregir de forma manual información de eventos sísmicos tales como magnitud, localización geográfica, profundidad.

[101] Para mayor información sobre el protocolo SeedLink se recomienda revisar la referencia [93]

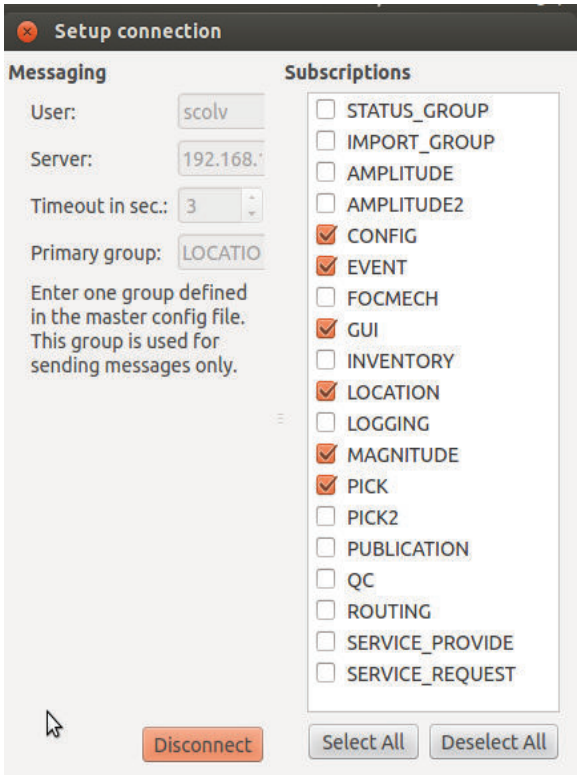

Figura 2.1. Solicitud de conexión a los grupos de SeisComP3

SeedLink dispone de varios *plugins*[102] para comunicarse con los diferentes tipos de sensores sísmicos, recuperar la información que el sensor genera y convertir esa información al formato miniSEED[103], formato con el que SeisComP3 trabaja por defecto.

2.3.1.4.2 ArcLink

Es un protocolo de comunicación que permite a otros módulos de SeisComP3 o programas externos acceder a la información de los sensores sísmicos que se encuentran almacenada, de manera local o remota, en formato miniSEED. A diferencia de SeedLink, no funciona en tiempo real, sino que los datos que suministra

---

[102] *Plugin*: es una aplicación pequeña que se instala en un programa principal para agregarle alguna funcionalidad.

[103] miniSEED (*Standard for the Exchange of Earthquake Data*): es un formato de almacenamiento de datos sísmicos. Para más información se recomienda revisar la referencia [94]

pueden tener varios días, meses o inclusive años de antigüedad. Igual que SeedLink está basado en TCP/IP y funciona por defecto en el puerto 18001.

2.3.1.4.3 Slarchive

Su función es conectarse a un servidor SeedLink solicitando formas de onda para su almacenamiento.

El almacenamiento puede referirse a un disco duro local, o un almacenamiento de red mediante NFS o Samba[104].

En la Figura 2.2 se presenta un diagrama con un esquema del funcionamiento de los tres módulos presentados, en la parte superior se representan los sensores sísmicos a los que se conectan los *plugin* de SeedLink mediante TCP/IP para obtener los datos crudos y convertirlos al formato miniSEED, el módulo SeedLink toma esos datos y los pone a disposición en el puerto 18000 en tiempo real.

A su vez el módulo Slarchive almacena los datos para que el módulo Arclink ponga a disposición el acceso a los datos antiguos.

2.3.1.5 Módulos de procesamiento

2.3.1.5.1 Scautopick

Se encarga de buscar cambios de amplitud en las formas de onda. Recibe las formas de onda del módulo SeedLink, las filtra y luego aplica un algoritmo STA/LTA [105].

Si existe una anomalía en la amplitud de la señal, el valor generado por el algoritmo superará un valor umbral, lo que hará que el módulo registre esa variación en la amplitud como un pico en la forma de onda.

---

[104] Samba: es un protocolo que permite compartir sistemas de archivos e impresoras entre una red de computadores.

[105] *Short Term Averaging/Long Term Averaging*: es un algoritmo para detectar picos en las formas de onda.

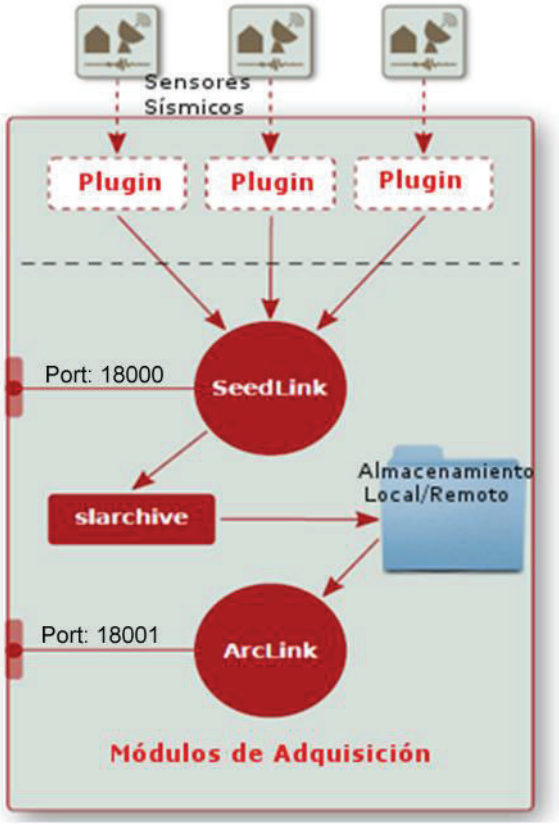

Figura 2.2. Módulos de adquisición y almacenamiento [52]

Scautopick crea un objeto `pick` que contiene información sobre la hora, minuto y segundo en que sucedió la variación, que sensores detectaron el pico, latitud y longitud del sensor, etc.

La información del objeto se envía mediante mensajes a los módulos correspondientes, en este caso a los que estén suscritos al grupo de mensajes PICK, la información también se almacena en la base de datos a través de Scmaster.

Este módulo se encarga también de calcular la amplitud a la que el pico equivale, crea un objeto `amplitude`, envía esa información al grupo AMPLITUDE y la almacena en la base de datos.

2.3.1.5.2 Scautoloc

Es el encargado de procesar los picos y las amplitudes que genera Scautopick para determinar el lugar geográfico en el que el evento sísmico se originó.

El módulo lee los picos generados y las amplitudes asociadas a esos picos y mediante algoritmos y modelos sísmicos, trata de identificar la combinación de picos que corresponde a un evento sísmico común. Como resultado crea un objeto del tipo `origin` y lo envía a la base de datos y al grupo LOCATION. El objeto `origin` contiene datos de latitud, longitud, profundidad, hora, etc. de un posible evento sísmico.

2.3.1.5.3 Scevent

Recibe los objetos de tipo `origin` y crea un objeto del tipo `event`, con información sobre la hora en la que sucedió el evento, magnitud, latitud y longitud del epicentro, que sensores lo detectaron, etc. esta información se envía a la base de datos y al grupo EVENT. A partir de este momento ya puede hablarse de la existencia de un evento sísmico, por lo que una vez calculado un valor preliminar de magnitud, inicia el proceso de difusión de esta información a las autoridades y al público en general, mediante los módulos de difusión de SeisComP3.

2.3.1.5.4 Scamp

Se encarga de calcular amplitudes adicionales a partir de objetos origen y sus objetos picos correspondientes.

Los objetos `amplitude` que este módulo crea se envían al grupo AMPLITUDE, y se almacenan en la base de datos. El objetivo de este módulo es mejorar la información del evento sísmico que SeisComP3 detectó.

2.3.1.5.5 Scmag

Su tarea es calcular la magnitud de un evento sísmico mediante los datos de los objetos `amplitude` y `origin`. Crea un objeto `magnitude` con el valor de la magnitud detectada por el sensor, y otro objeto `magnitude` con el valor de la magnitud asociada con el sismo. La información de ambos objetos se envía al grupo MAGNITUDE, y a la base de datos.

En la Figura 2.3 se presenta un esquema del funcionamiento de los módulos de procesamiento, las flechas en azul son objetos usados por el módulo como entrada, mientras que las flechas de color rojo son los objetos que el módulo genera. Las letras corresponden a los tipos de objetos, `A: amplitud, P: pick, O: origin, M: magnitude, E: event`.

Como puede verse el módulo SeedLink actúa como servidor de formas de onda para Scautopick y Scamp. Puede verse también que el único módulo que escribe en la base de datos es Scmaster.

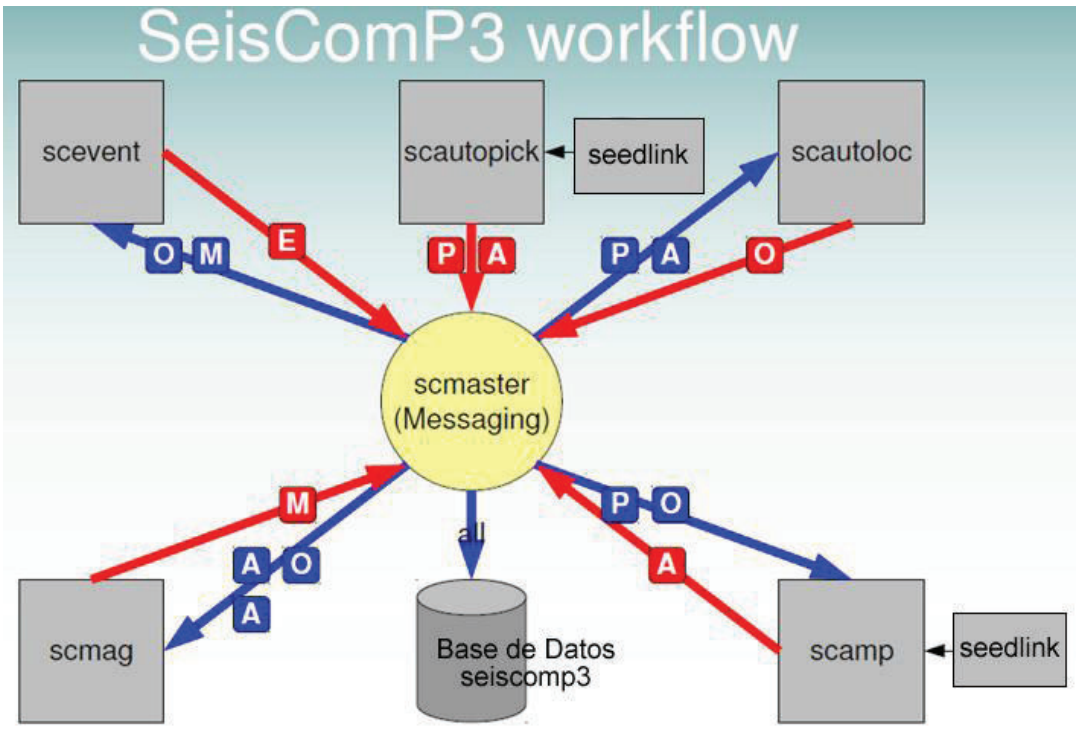

Figura 2.3. Funcionamiento de los módulos de procesamiento de SeisComP3 [53]

2.3.1.6 Módulos de diseminación

Una vez que Scevent recibe la magnitud calculada actualiza el objeto `event` en la base de datos y si cumple con algunas condiciones establecidas, como magnitud mínima, número de estaciones que detectaron el evento, área de interés, etc. el sistema SeisComP3 generará una alerta auditiva y diseminará la información del

evento vía web, SMS[106], correo electrónico, Twitter[107] y en la página de Facebook[108] del Instituto Geofísico.

2.3.1.6.1 QuakeLink

Es un módulo que implementa el protocolo QuakeLink[109], y se utiliza para intercambiar información de un evento sísmico en tiempo real. Una vez que SeisComP3 tiene la información del evento, QuakeLink se encarga de extraer esa información de la base de datos y la pone a disposición a través del puerto 18010.

2.3.1.6.2 GDS (Gempa Dissemination Server)

Es un sistema desarrollado por Gempa, está formado por varios módulos escritos en Python[110]. El sistema recibe información del módulo QuakeLink, la procesa y adapta según el medio de comunicación que usará para informar sobre el evento. En la Figura 2.4 se presenta un esquema del funcionamiento de este servidor. Los rectángulos en azul representan los módulos encargados de difundir la información vía email, SMS, correo electrónico, etc.

El sistema dispone de una interfaz web, representada en el rectángulo rojo, que permite configurar parámetros como que usuarios recibirán la información, a partir de que magnitud difundir alertas, etc.

2.3.1.7 Módulos para análisis

El sistema cuenta también con módulos para visualizar las señales de entrada, corregir los parámetros de un sismo, monitorizar el estado de las estaciones, etc. , a continuación se describe brevemente los principales.

---

[106] *Short Message System*: permite el envío de mensajes de texto a través de la red celular.

[107] Twitter: es una red social que permite el envío y recepción de mensajes cortos denominados tweets

[108] Facebook: es una red social que permite la publicación de imágenes, vídeos, etc.

[109] Quakelink es un protocolo propietario, para más información se recomienda revisar la referencia [54]

[110] Python: es un lenguaje de programación que no requiere compilación, para más información se recomienda revisar la referencia [95]

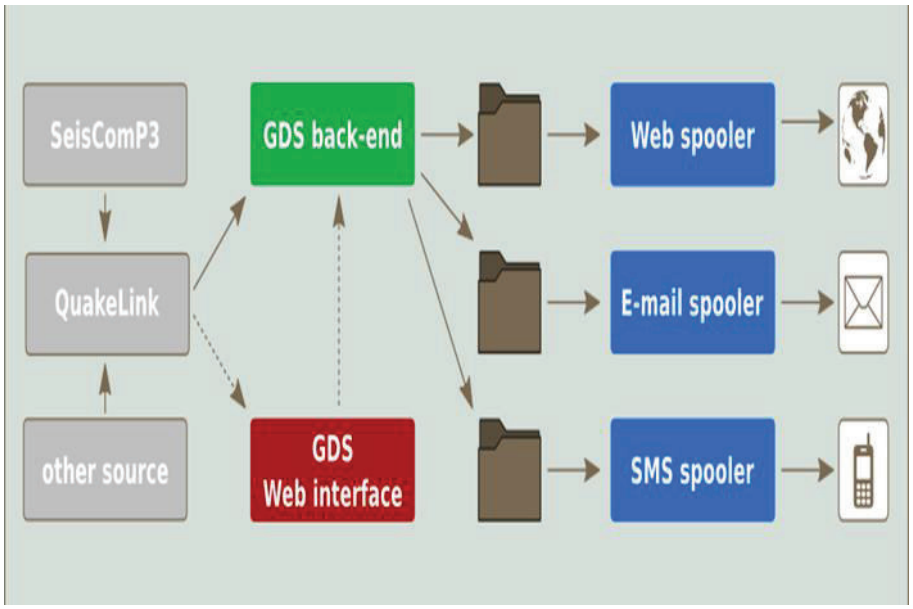

Figura 2.4. Diagrama del funcionamiento del módulo GDS [54]

2.3.1.7.1 Scrttv

Permite visualizar en tiempo real las formas de onda que ingresan al SeisComP3 a través del módulo SeedLink. En la Figura 2.5 se presenta la información generada por 10 estaciones dedicadas a monitorizar la actividad volcánica.

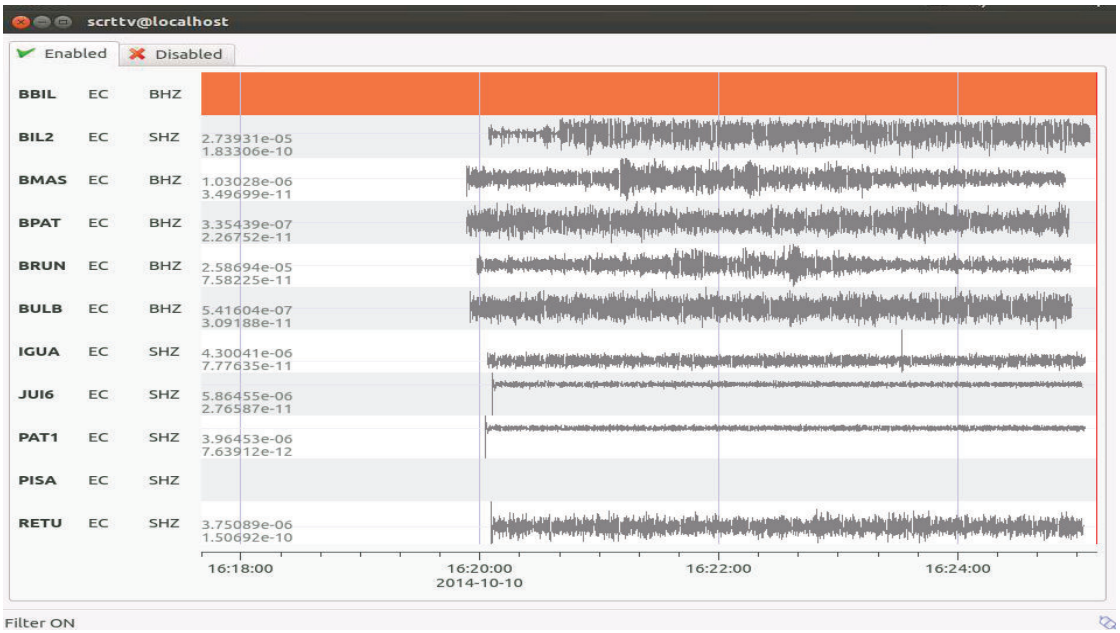

Figura 2.5. Módulo de visualización de formas de onda Scrttv

2.3.1.7.2 Scqcv

Este módulo despliega valores relacionados al estado de un sensor sísmico y los datos que suministra, tales como tiempo de retardo, latencia, vacíos en la formas de onda, etc.

2.3.1.7.3 Scolv

Es el módulo de análisis más importante, ya que permite revisar y corregir ciertos parámetros de un evento sísmico.

En la Figura 2.6 se presenta la ventana principal de este módulo, en el lado izquierdo existe un mapa con el que indica el lugar geográfico en el que el sistema estima se originó el evento y las estaciones que lo detectaron, en el centro de la ventana contiene información relacionada con el sismo, profundidad, latitud, longitud, hora, número de estaciones que lo detectaron, etc.

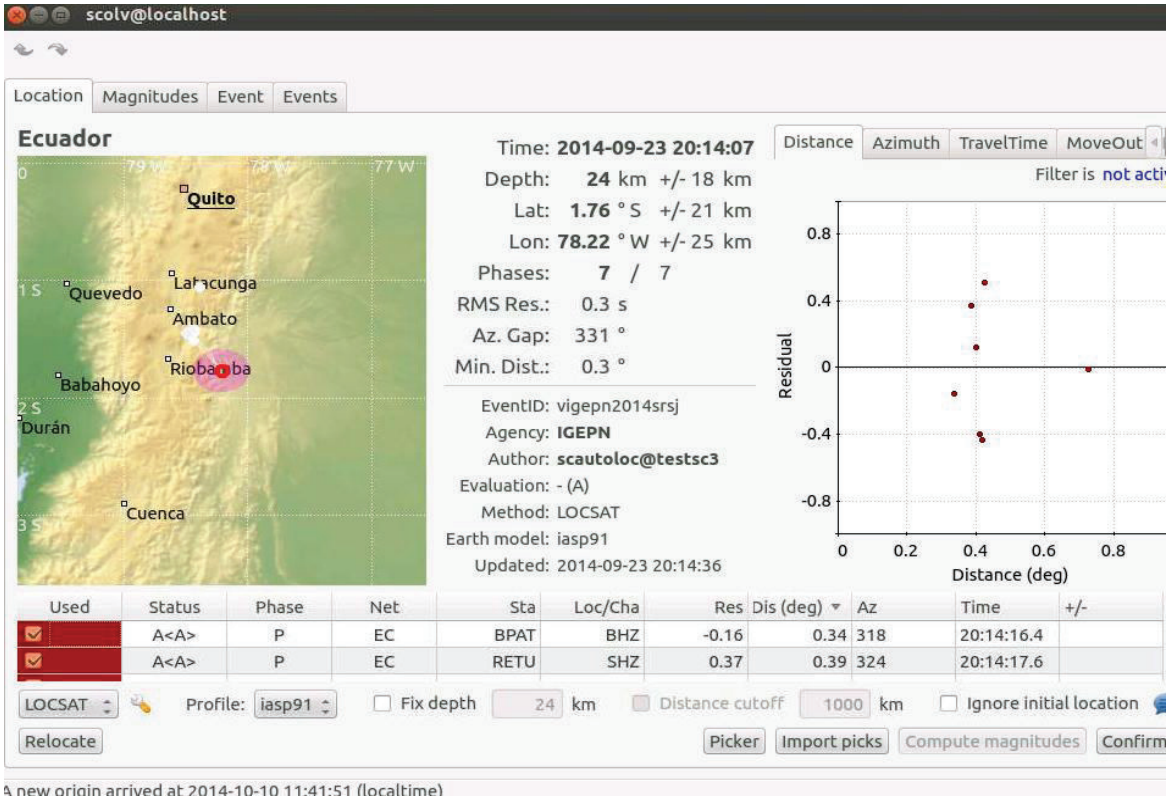

Figura 2.6. Ventana principal del módulo scolv

2.3.1.8 Base de datos

Como se ha descrito, cada módulo almacena la información que genera en una base de datos, y desde allí otros módulos recuperan esa información para revisar o corregir información de sismos pasados. La base de datos es uno de los componentes principales del sistema, ya que contiene la información procesada de todos los sismos desde el año 2011 hasta el momento, que han sido detectados por el sistema y corregidos por el usuario. Por defecto el sistema utiliza una base del tipo MySQL.

2.3.2 EARTHWORM [55][56]

2.3.2.1 Introducción

El proyecto Earthworm se inició en 1993 con el objetivo de tener un sistema que permita procesar los datos de decenas de estaciones de forma rápida, confiable y en tiempo real, así como generar de forma rápida notificaciones y alertas de eventos sísmicos.

2.3.2.2 Características

- Funciona sobre sistemas operativos Windows o Linux de 32 o 64 bits.
- Modularidad: cada función del programa se implementa como un módulo independiente, de tal forma que si un módulo secundario falla, los programas más importantes no se verán afectados.
- Conectividad: el sistema puede trabajar en tiempo real con otras aplicaciones de procesamiento, análisis o de notificación de eventos.
- Configuración basada en archivos de texto.
- Los módulos solo pueden activarse, detenerse o monitorizarse mediante línea de comandos.

El sistema Earthworm es un conjunto de módulos unidos por un sistema de mensajes tipo *broadcast* que se denomina “anillo de mensajes”. Cada módulo desempeña una tarea, tal como adquisición de datos, almacenamiento, detección de eventos, etc., y se configura mediante un archivo de texto que contiene los parámetros con los que debe ejecutarse.

El anillo de mensajes permite la comunicación entre módulos del mismo servidor o de diferentes servidores. En la Figura 2.7 puede verse un esquema del funcionamiento de los módulos de Earthworm.

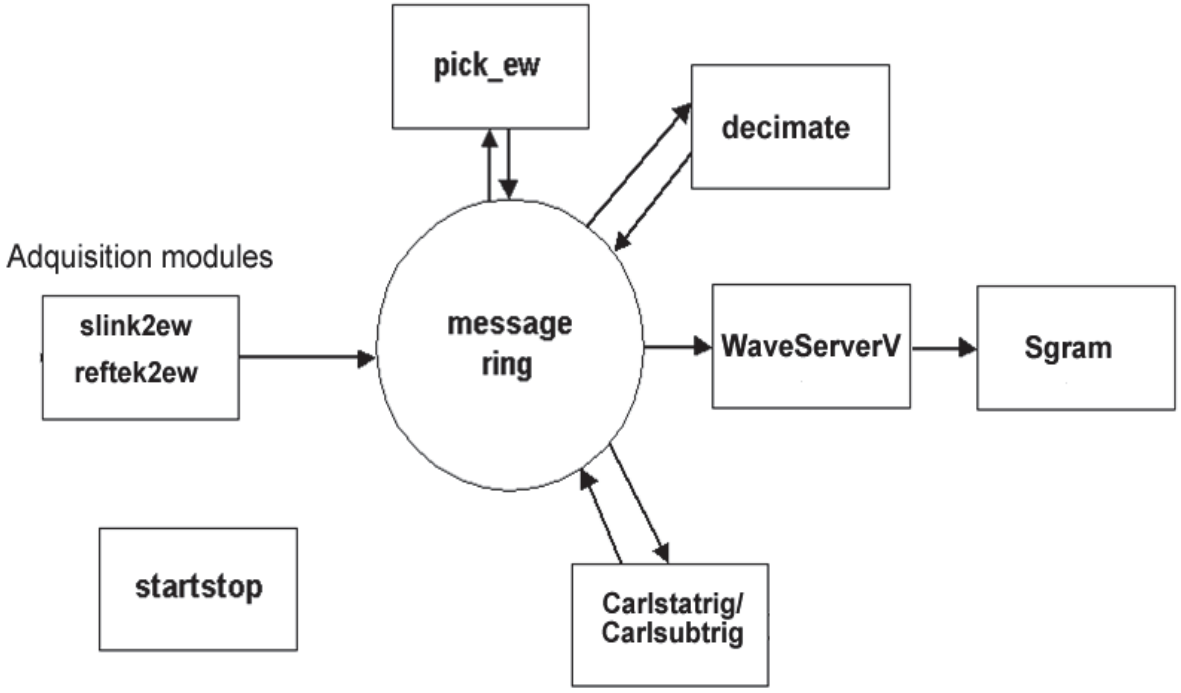

Figura 2.7. Esquema del funcionamiento de Earthworm [55]

2.3.2.3 Módulos de administración

2.3.2.3.1 startstop

Es el módulo principal del sistema Earthworm y es el encargado de crear el anillo de mensajes y arrancar los módulos con los parámetros que indique cada archivo de configuración.

Este módulo también permite monitorizar el estado y detener la ejecución del resto de módulos.

2.3.2.3.2 Statmgr

Monitoriza el estado de los módulos de Earthworm mediante mensajes de tipo *heartbeat*[111], si detecta errores puede reiniciar el módulo y enviar alertas mediante correo.

2.3.2.4 Módulos de comunicación

2.3.2.4.1 Import_generic y Export_scnl_ack

Estos módulos permiten el intercambio de mensajes entre dos o más servidores Earthworm empleando el protocolo TCP/IP.

2.3.2.4.2 Ringdup_scn

Es un programa que lee mensajes de un anillo de mensajes y los exporta a un segundo anillo.

2.3.2.4.3 Scnl2scn

Módulo que se encarga de transformar datos de entrada del formato antiguo de Earthworm al nuevo formato que utiliza desde la versión 7.

2.3.2.5 Módulos para adquisición de datos

2.3.2.5.1 Slink2ew

Es un cliente SeedLink para Earthworm, es decir el módulo se conecta a un servidor SeedLink del que obtiene formas de onda para entregarlas a los módulos de procesamiento del sistema Earthworm.

---

[111] Mensajes de tipo *heartbeat:* son mensajes que indican que la aplicación está activa y funcionando normalmente.

2.3.2.5.2 Scream2ew

Es un módulo que convierte datos de un sensor Scream[112] al formato que utiliza Earthworm.

2.3.2.5.3 Reftek2ew

Es un módulo que obtiene datos de un sensor Reftek[113] y los convierte al formato que utiliza Earthworm.

2.3.2.6 Módulos de procesamiento

2.3.2.6.1 Decimate

Es un módulo que implementa operaciones de filtrado en las formas de onda entrantes.

2.3.2.6.2 Carlstatrig y Carlsubtrig

Son módulos que implementan un algoritmo STA/LTA para detectar variaciones en la amplitud de las formas de onda que han sido filtradas por el módulo `Decimate`.

2.3.2.7 Módulos de visualización y almacenamiento

2.3.2.7.1 Wave_serverV

Este módulo permite acceder en tiempo real a las formas que ingresan al sistema, soporta la conexión de hasta diez clientes o programas para visualizar las formas de onda.

---

[112] Scream: es un protocolo para adquisición de datos en tiempo real desarrollado por la empresa fabricante de sensores sísmicos Guralp.

[113] Reftek: es una empresa dedicada a la manufactura y venta de sensores sísmicos.

2.3.2.7.2 Heli_ewII

Este módulo genera de forma automática imágenes en formato GIF[114] de las formas de onda que ingresan al sistema Earthworm en tiempo real.

2.3.2.7.3 Sgram

Este módulo genera en tiempo real imágenes en formato GIF del espectro de frecuencias de las formas de onda que ingresan al sistema Earthworm.

2.3.3 SHAKEMAP

2.3.3.1 Introducción [57]

ShakeMap es un sistema que genera mapas que indican el movimiento del suelo generado por un sismo en un área determinada. El USGS (*United States Geological Survey*)[115] desarrolló el programa ShakeMap a fin de ofrecer mapas de movimientos del suelo e intensidad de movimiento casi en tiempo real, con el objetivo de que se los use para planes de reacción ante un evento sísmico de magnitud considerable.

Cuando un evento sísmico ocurre genera diferentes niveles de movimiento lo que depende de la distancia de la zona al epicentro, tipo del material de la zona y variaciones debido a la estructura de la corteza de la tierra.

El mapa representa, mediante una escala de colores, con que fuerza se sacudió el suelo en una zona, como puede verse en la Figura 2.8, los colores claros indican que el sismo no generó sacudidas, por ejemplo el color amarillo indica una sacudida mediana, mientras que el color rojo indica que el sismo generó una gran sacudida. Esta información permitiría a los organismos de gestión de riesgos, defensa civil, cruz roja, bomberos, etc., concentrar sus esfuerzos en las áreas más afectadas por la sacudida.

---

[114] *Graphics Interchange Format*: es un formato de imagen de mapa de bits.

[115] USGS: es un instituto orientado a la investigación geológica y sismológica.

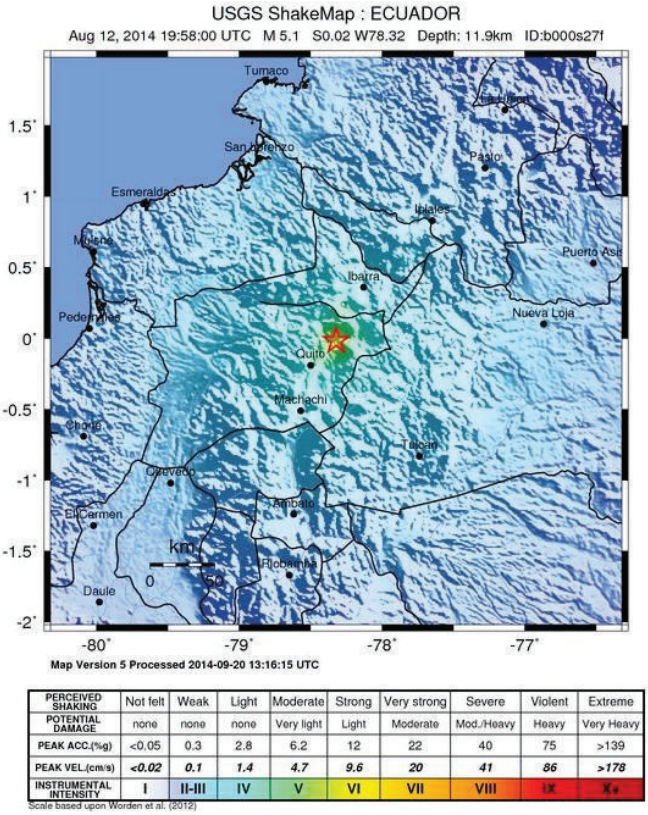

Figura 2.8. ShakeMap del sismo ocurrido en Quito el 2014-08-12

2.3.3.2 Características

- Puede instalarse en sistemas operativos Linux o MacOS X.
- Software de código abierto bajo la licencia GPL.
- El sistema está formado por módulos escritos en Perl[116].
- Depende de otros programas de código abierto como GMT[117], NetCDF[118], etc. lo que disminuye su portabilidad.
- Se tiene un registro de los mapas generados en una base de datos MySQL.

[116] Perl: es un lenguaje interpretado de alto nivel, con características del lenguaje de programación C.

[117] GMT: es un conjunto de módulos para procesamiento de datos geográficos.

[118] NetCDF: es un conjunto de librerías usadas para tratamiento de datos científicos

- Permite la integración con programas como Google Earth[119], ArcGIS[120], etc.
- Configuración mediante archivos de texto.
- Todo el procesamiento se activa mediante un único comando
- Genera páginas web que presentan la información generada.

2.3.3.3 Componentes y funcionamiento de ShakeMap

ShakeMap está formado por módulos o *scripts* escritos en Perl, que se ejecutan de forma secuencial para producir una página web que contiene mapas e información relacionada a los movimientos del suelo, producidos por un evento sísmico de gran magnitud. El programa puede generar también alertas vía correo.

2.3.3.3.1 Datos de entrada

El programa necesita de dos archivos XML para funcionar: `event.xml` y `event_data.xml`. Estos archivos contienen información sobre el sismo, tal como magnitud, latitud, longitud, estaciones que detectaron el evento, y otros parámetros que los programas que conforman ShakeMap necesitan.

Estos archivos son generados por el módulo `scwfparam`[121] del sistema SeisComP3.

2.3.3.3.2 Shake

Es el programa principal, funciona como *wrapper*[122], es decir que su tarea es llamar al resto de programas.

---

[119] Google Earth: es un programa de información geográfica, usado para visualizar mapas en 3D, para más información se recomienda revisar la referencia [96].

[120] ArcGIS: es un programa usado para generar mapas y bases de datos geográficas.

[121] Scwfparam: es un módulo de procesamiento que calcula valores de desplazamiento, velocidad y aceleración producidos por un sismo.

[122] *Wrapper*: es un programa, subrutina o librería cuya función es llamar a una segunda rutina.

2.3.3.3.3 Retrieve

Es el primer programa que el módulo `shake` llama y se encarga de recuperar los archivos `event.xml` y `event_dat.xml` y colocarlos en la carpeta `input`.

2.3.3.3.4 Grind

Lee los parámetros que se encuentran en `event_data.xml` y estima los movimientos que el evento sísmico generó en el área cercana. Luego de calcular esos valores los asocia a una latitud y longitud y los guarda en un archivo de texto con el nombre `grid.xyz`, que el resto de programas utiliza para generar el mapa de movimiento.

2.3.3.3.5 Tag

Este módulo escribe en la base de datos información sobre el evento sísmico del que se han creado los mapas de movimiento.

2.3.3.3.6 Mapping

A partir del archivo `grid.xyz` crea los mapas de movimiento en formato PostScript[123].

2.3.3.3.7 Genex

Utiliza la información creada por `mapping` y `grind` para crear los mapas en formato JPEG[124], los archivos HTML para la página web, etc.

2.3.3.3.8 Transfer

Este módulo se encarga de copiar los archivos creados por `genex` a un servidor web local o remoto

---

[123] PostScript: es un lenguaje de computación para crear gráficos vectoriales.

[124] JPEG: es un formato de imagen sin compresión.

2.3.3.3.9 Base de datos

Se utiliza básicamente para llevar un registro de los sismos de los que se han generado los mapas de movimiento, el espacio que la base de datos utiliza es mínimo.

2.4 ANÁLISIS DE REQUERIMIENTOS

En esta sección se examina los requerimientos de los sistemas presentados, para lo que se ha monitorizado el uso de recursos físicos de los servidores físicosen los que están instalados los tres sistemas, para lo que se ha utilizado el programa Zabbix[125].

Así mismo se determina que módulos de cada sistema serán administrados por el software del *cluster*.

2.4.1 SEISCOMP3

Al momento el sistema SeisComP3 está instalado en tres servidores, diariamente ingresan al sistema información de 170 sensores sísmicos localizados en territorio ecuatoriano, y 220 sensores de varios países como Colombia, Chile, EEUU, Alemania, etc.

En uno de los servidores denominado `scauto2`, ingresan solamente los datos de la mitad de los sensores, y este servidor actúa como servidor secundario y de pruebas, los módulos de procesamiento están activos pero no están configurados, por lo que este servidor no será virtualizado.

En el segundo servidor, denominado `scauto1` ingresa toda la información de los sensores sísmicos y es en el que se realiza el procesamiento de la información, para su posterior diseminación, por lo que este es el servidor que se virtualizará.

---

[125] Zabbix: es un software de monitorización de código abierto. Para más información se recomienda revisar la referencia [97].

El tercer servidor funciona solamente para tareas de visualización de formas de onda, estado de las estaciones y revisión de eventos sísmicos, en este servidor solamente están activos los módulos gráficos, en este servidor no ingresan datos ni se ejecuta ningún módulo de procesamiento, por lo que tampoco se virtualizará este servidor.

A continuación se revisará la información de utilización de los recursos de hardware del servidor `scauto1`.

2.4.1.1 Estimación de uso del hardware del servidor SeisComP3

El servidor `scauto1` en el que se ejecuta el sistema SeisComP3 es en un servidor IBM X3400 M3[126], en el que está instalado el sistema operativo Ubuntu 12.04 de 64 bits. En la Tabla 2.1 se presenta la información de hardware de `scauto1`. De acuerdo a la información registrada por Zabbix se estimará el porcentaje de utilización del hardware para determinar los requisitos que tendrá la máquina virtual a crear, para las estimaciones se utilizarán los valores máximos registrados, en los casos que sea posible.

Característica	Valor
RAM	8 GB
Disco Duro	2 x 500 GB RAID 1
CPU	16 núcleos
Red	2 puertos Ethernet 1 GB

Tabla 2.1. Información del hardware del servidor SeisComP3

En la Figura 2.9, se presenta el historial de utilización de CPU del servidor `scauto1`, en un período de un mes. Como puede verse en la columna `max` el porcentaje de utilización máximo registrado en este período de tiempo es igual 9.09%. No es

[126] IBM X3400 M3: es un servidor IBM de tipo torre de alto desempeño, escalabilidad y de bajo consumo de energía.

posible ver este valor gráficamente en la Figura 2.9 debido a que el período de uso máximo fue solo de unos minutos mientras que la Figura 2.9 presenta el historial de todo un mes.

Dado que el servidor posee 16 núcleos, un porcentaje de utilización del 9.09% indica que el sistema SeisComP3 necesita solamente 1.45 núcleos.

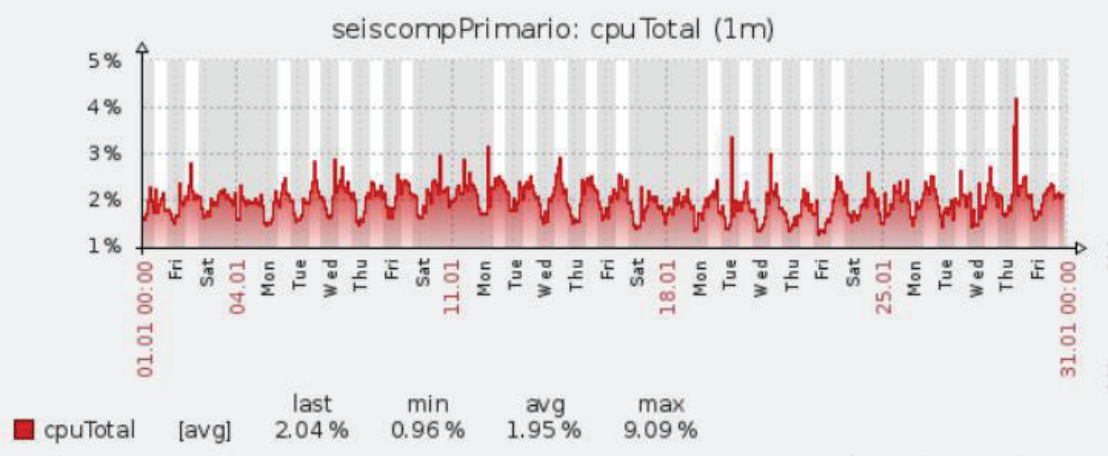

Figura 2.9. Uso del CPU del servidor SeisComP3

En la Figura 2.10 se presentan los registros de utilización de memoria RAM en un período de un mes.

En los sistemas operativos basados en Linux para obtener la cantidad de memoria RAM utilizada por los procesos que se ejecutan en el servidor es necesario utilizar la Ecuación 2.1, debido a que el sistema operativo considera como memoria usada a la memoria caché[127] y a la memoria búfer[128].

```
Uso real de memoria = Memoria utilizada – (Memoria Cache + Memoria Buffer)
```

Ecuación 2.1. Determinar la utilización real de memoria RAM

[127] Memoria Caché: es un tipo de memoria al que el procesador accede más rápidamente y permite acelerar la lectura de datos.

[128] Memoria búfer: es un tipo de memoria RAM reservado para acelerar la escritura de datos al disco.

Para aplicar la Ecuación 2.1 se utilizan los valores promedios de la columna `avg`, de la Figura 2.10, lo que da como resultado 2,77 GB.

En la Figura 2.11 se presenta el uso de la interfaz de red del servidor SeisComP3 en un mes, los valores máximos de utilización de la red son 8.23 Mbps para el tráfico de salida y 1.56 Mbps para el tráfico de entrada, sumados ambos valores la utilización de red sería igual a 9.79 Mbps

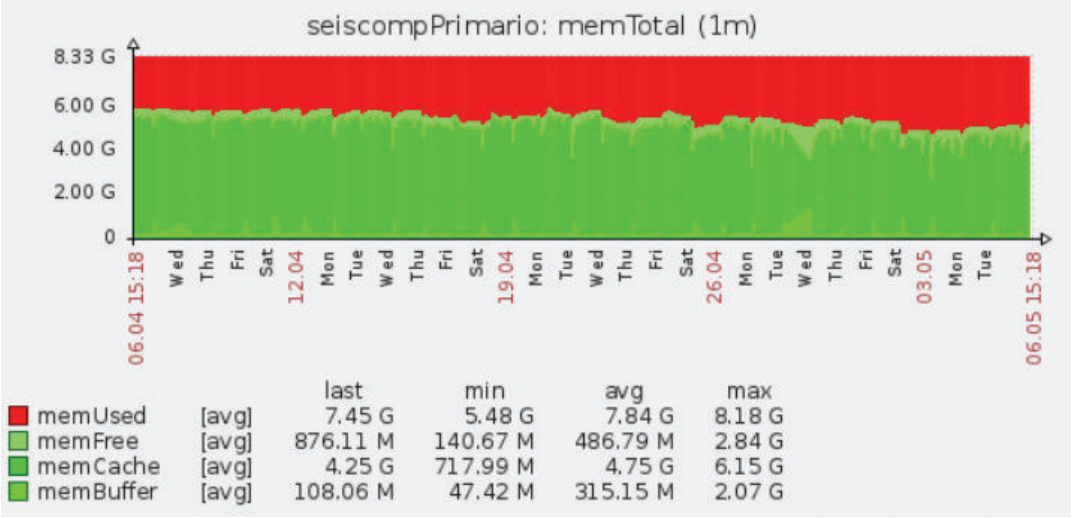

Figura 2.10. Uso de memoria RAM en el servidor SeisComP3 primario

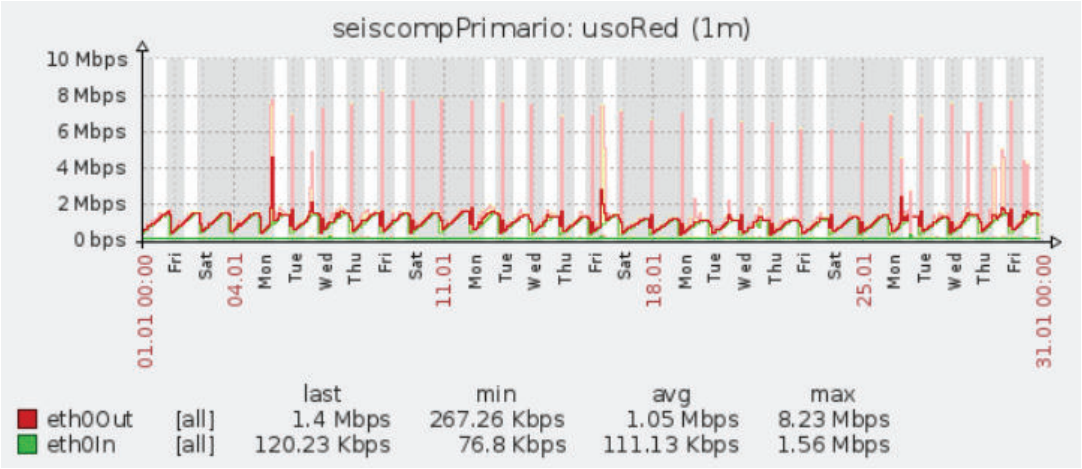

Figura 2.11. Uso de la interfaz de red eth0 del servidor SeisComP3

Por último se determinará la cantidad de espacio en disco duro que necesita el sistema SeisComP3 diariamente. La Figura 2.12 representa el espacio libre del disco

duro en el que SeisComP3 almacena las formas de onda, en el período de una semana, como puede apreciarse el espacio libre disminuye de forma constante. En la parte inferior de la Figura 2.12 puede verse que el valor máximo, que representa el espacio libre al inicio de la semana, es de 50.67 GB y el valor mínimo, que representa el espacio libre al final de la semana, es 23.47 GB, la diferencia entre estos dos valores será igual al espacio requerido para almacenar los datos por una semana, valor que es igual a 27.2 GB.

No es necesario que la información de los sensores permanezca en el disco duro por mucho tiempo, ya que semanalmente se realiza un respaldo en cintas y en el sistema de almacenamiento central por lo que basta con tener los datos de las últimas dos semanas, es decir 54 GB.

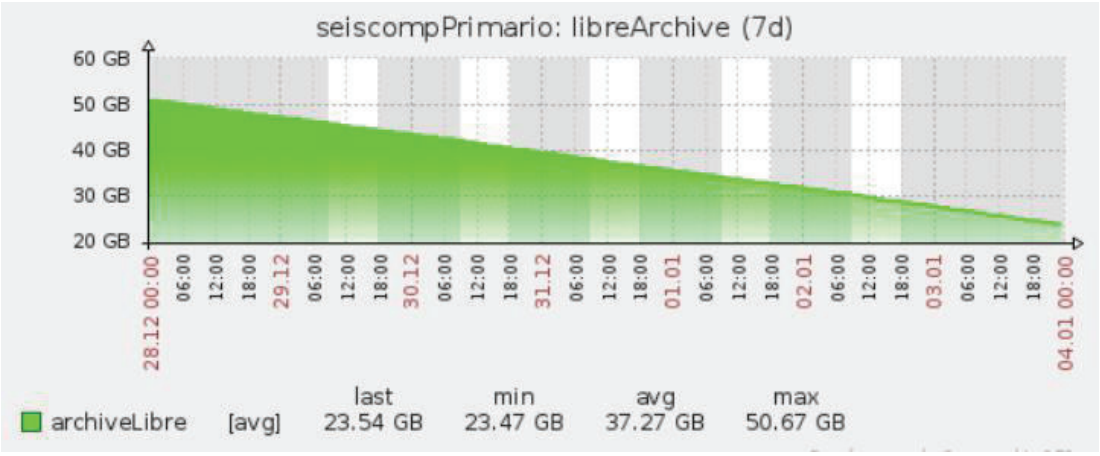

Figura 2.12. Utilización del disco en el sistema SeisComP3

El espacio necesario para los datos procesados, que incluyen las páginas web generadas, imágenes de las formas de onda, informes sísmicos, base de datos de los eventos, etc. se decide asignar 30 GB. Al igual que en el caso de las formas de onda, estos datos se respaldan y luego se eliminan para dejar espacio libre a nueva información.

Una vez revisada la información de utilización del hardware del servidor `scauto1` se realiza una estimación del hardware virtual que necesitará el sistema virtualizado para funcionar, estos valores se indican en la Tabla 2.2.

Característica	Requisitos
RAM	2.77 GB
Espacio en disco para adquisición de datos	54 GB
Espacio en disco para datos generados	30 GB
CPU	1.45 núcleos
Red	9.29 Mbps

Tabla 2.2. Requerimientos del hardware virtualizado para el sistema SeisComP3

2.4.1.2 Módulos a administrar

En la Tabla 2.3 se indican los módulos que se ejecutan en el servidor SeisComP3 primario y que deben ser administrados por el software del *cluster*.

Importancia	Componente
Crítica	`Scmaster`
Crítica	`Spread`
Crítica	`Seedlink`
Crítica	Base de datos
Crítica	Módulos de procesamiento (`scautopick`, `scautolock`, etc.)
Alta	`Arclink`
Media	`Slarchive`

Tabla 2.3. Componentes del sistema SeisComP3 a administrar

2.4.2 EARTHWORM

Al momento el sistema está instalado en dos computadores denominados `Earthworm1` y `Earthworm2` el primero dispone de tarjetas de hardware especiales dedicadas al ingreso de sensores volcánicos utilizando comunicación serial, y su tarea es principalmente la adquisición de formas de onda.

Debido a los requerimientos especiales de hardware que tiene el servidor `Earthworm1` no es posible virtualizarlo.

`Earthworm2` toma la información de `Earthworm1` y permite realizar tareas de procesamiento y visualización de esos datos, este servidor no tiene requerimientos especiales de hardware, por lo que es el servidor que se virtualizará.

El computador `Earthworm2` es una estación de trabajo, que ejecuta el sistema operativo Windows Server 2008, en la Tabla 2.4 se presenta la información del servidor en el que el sistema Earthworm se está ejecutando.

Característica	Valor
RAM	12 GB
Disco Duro	1 x 1 TB
CPU	4 núcleos
Red	1 puerto Ethernet 1 GB

Tabla 2.4. Información del hardware del servidor Earthworm

2.4.2.1 Estimación del uso del hardware del servidor físico Earthworm

Como puede verse en la parte baja de la Figura 2.13, en la columna `max` la utilización del procesador tiene un valor igual a 59.69%, es decir que de los seis núcleos que el servidor posee utiliza como máximo 2.38 núcleos.

A continuación se determina la utilización de memoria RAM que hace el servidor `Earthworm2`, como puede verse en la Figura 2.14 en la columna `max` el valor correspondiente a la memoria usada es igual a 6.01 GB.

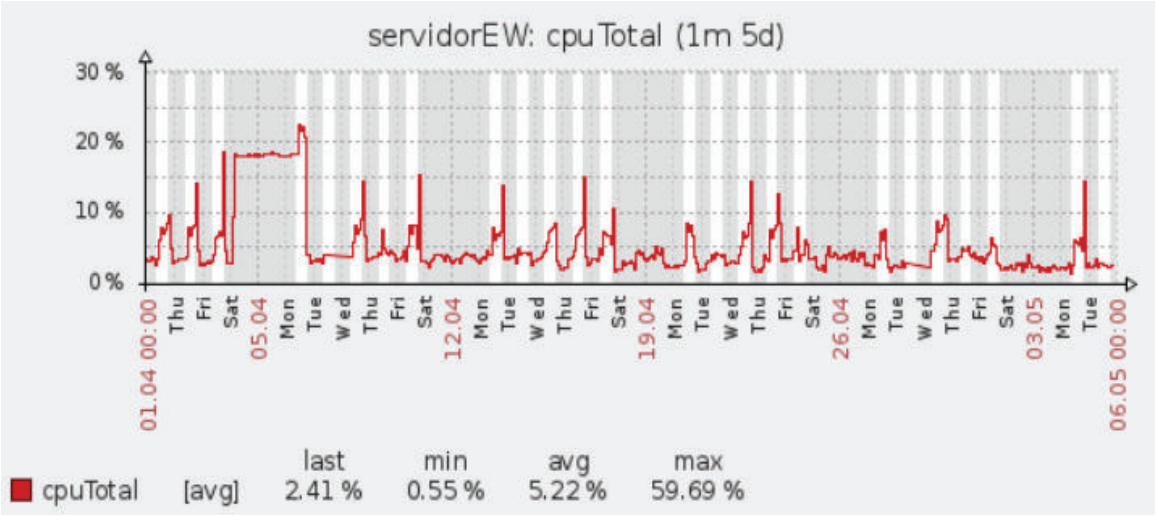

Figura 2.13. Utilización del CPU en el sistema Earthworm

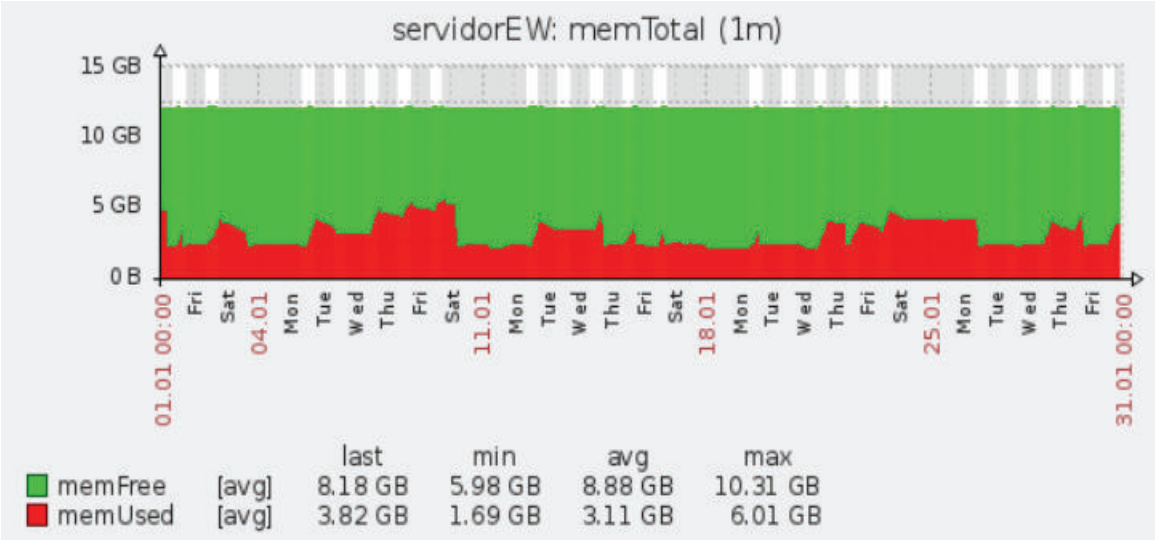

Figura 2.14. Utilización de la memoria RAM en el servidor Earthworm

En la Figura 2.15 se indica la utilización de la interfaz de red en el servidor Earthworm, el valor máximo para los datos de entrada es 600.17 Kbps, mientras que para los datos de salida es de 2.04 Mbps.

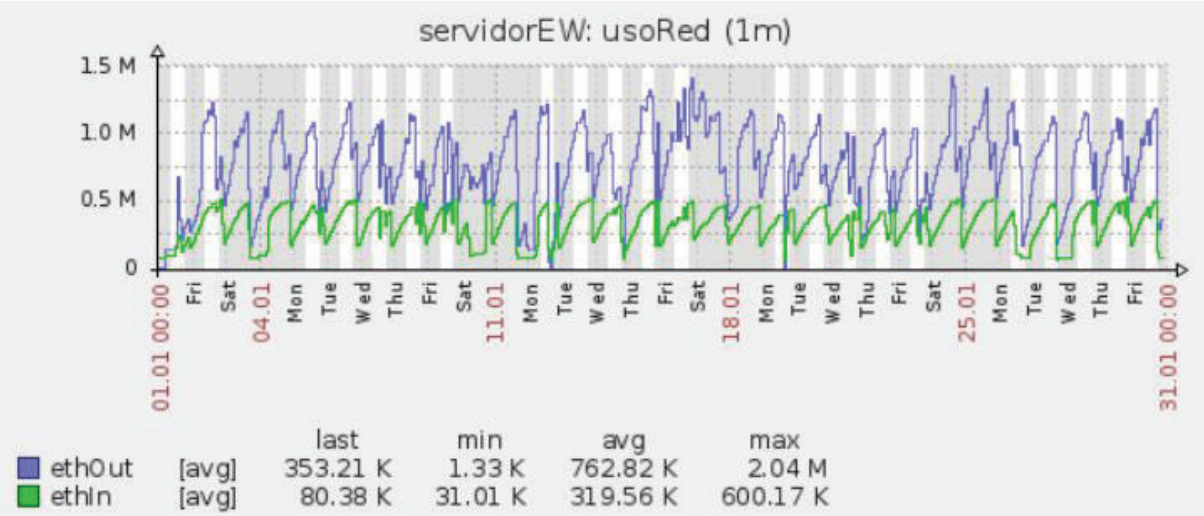

Figura 2.15. Utilización de la interfaz de Ethernet en el servidor Earthworm

En cuanto al uso de disco, la utilización del mismo en un período de una semana se presenta en la Figura 2.16, la cual indica el espacio libre del disco duro en el que el sistema Earthworm almacena las formas de onda durante una semana.

En la parte inferior de la Figura 2.16 puede verse que el valor mínimo, que corresponde al inicio de una semana, es igual a 120.9 GB, mientras que el valor máximo, que corresponde al final de la semana, es igual a 123.9 GB, la diferencia entre estos valores indica la cantidad de espacio que necesita el sistema para almacenar los datos durante siete días, valor que es igual a 3 GB.

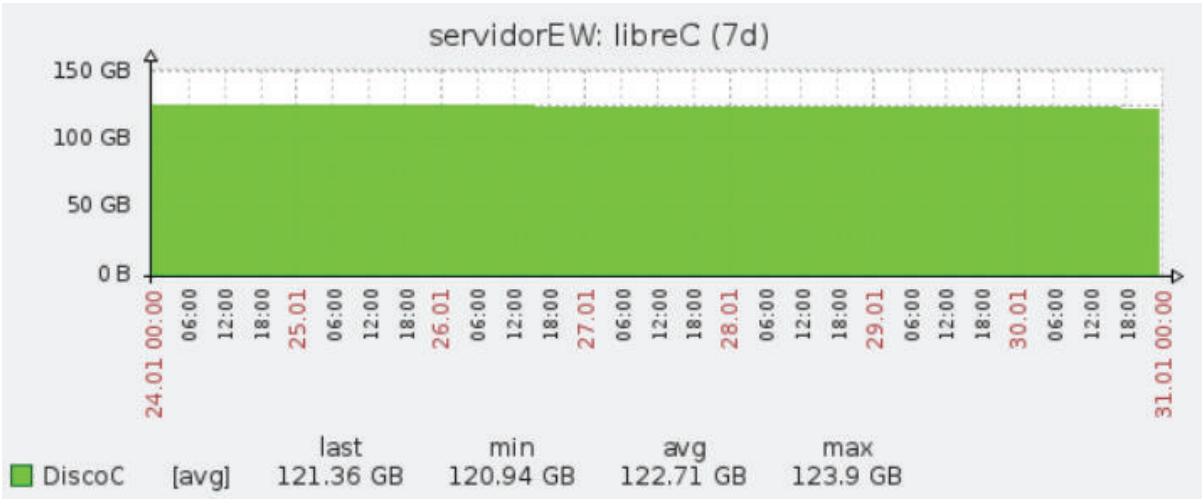

Figura 2.16. Utilización del disco duro en el servidor Earthworm

Al igual que en sistemas SeisComP3, los datos que el sistema adquiere se almacenan en cintas y en un repositorio general, por lo que no es necesario almacenar estos datos por un período mayor a dos semanas.

Con esta información ya es posible decidir los requisitos que el servidor virtual Earthworm debería tener, los mismos que se indican en la Tabla 2.5.

Característica	Requisitos
RAM	6 GB
Espacio en disco para los datos durante dos semanas	6 GB
CPU	2.38 núcleos
Red	10 Mbps

Tabla 2.5. Requerimientos del hardware virtualizado para el sistema Earthworm

2.4.2.2 Módulos a administrar

El sistema Earthworm al momento funciona como sistema de adquisición y visualización de formas de onda, es decir adquiere datos de estaciones y genera imágenes de esas formas de onda, tales como espectrogramas, que son imágenes que presentan la información que el sensor genera en el dominio de la frecuencia y *helicorders*, que presentan la información del sensor con respecto al tiempo, y que pueden visualizarse en la página web del Instituto Geofísico [58], [59]. Por este motivo los módulos más importantes son los de adquisición y visualización, como se indica en la Tabla 2.6.

Importancia	Componente
Crítica	`Startstop`
Crítica	`Slink2ew`
Crítica	`Wave_serverV`

Tabla 2.6. Componentes de Earthworm a monitorizar

2.4.3 SHAKEMAP

ShakeMap se encuentra instalado en un computador de escritorio que tiene el sistema operativo Ubuntu 14.04, y dado que no procesa datos continuamente, sino bajo demanda, se utilizarán los valores indicados en la Tabla 2.7, para crear el servidor virtual para este sistema.

Característica	Valor
RAM	2 GB
Disco Duro 1	30 GB
Disco Duro 2	30 GB
CPU	1 núcleo
Red	1 puerto Fast Ethernet

Tabla 2.7. Información del hardware del servidor ShakeMap

En el primer disco duro se encuentra el sistema operativo Linux, mientras que en el segundo está instalado el ShakeMap y los productos que genera para cada sismo, que son mapas, páginas web, etc. los mismos que ocupan alrededor de 30 MB por cada evento, estimando un evento sísmico diario durante un año el espacio que necesitaría sería de solo 10 GB, pero se elige emplear también un disco de 30 GB, en caso de que sea necesario realizar procesamientos adicional de los datos y mapas generados.

2.4.3.1 Módulos a administrar

A diferencia de los sistemas SeisComP3 y Earthworm los módulos que componen ShakeMap no se ejecutan en forma de servicio, sino que funcionan bajo demanda y una vez terminado su trabajo se cierran, por lo que no es posible monitorizar de forma continua su funcionamiento.

Debido a esto los módulos que Pacemaker administrará serán la base de datos MySQL y el servidor Apache.

En la Tabla 2.8 se presentan los componentes que el software del *cluster* administrará.

Importancia	Componente
Alta	Servidor web Apache
Media	Base de datos MySQL

Tabla 2.8. Componentes de ShakeMap a monitorizar

2.4.4 REQUERIMIENTOS PARA EL CLUSTER DE ALTA DISPONIBILIDAD

Una vez revisados los requerimientos de hardware de los tres sistemas, con esta información se determinará en primer lugar el tamaño del almacenamiento compartido del *cluster*, mientras que con la información de requerimientos de RAM, CPU y red se determinarán las características de hardware que debe tener un nodo del *cluster* de alta disponibilidad.

2.4.4.1 Tamaño del almacenamiento compartido

El tamaño del almacenamiento compartido deberá ser el adecuado para almacenar las tres máquinas virtuales en las que se ejecutarán los sistemas de adquisición y procesamiento revisados.

En primer lugar se determina el espacio que el sistema SeisComP3 necesita, como se indicó en la Sección 2.4.1.1 es necesario un espacio de 54 GB para almacenar datos de dos semanas, mientras que para los datos procesados se estima un espacio necesario de 40 GB, para el sistema operativo Linux básico se estima necesario un espacio de 20 GB.

SeisComP3	Tamaño necesario (GB)	Tamaño de disco duro (GB)
Sistema Operativo Linux	20	30
Adquisición de Datos (2 semanas)	54	64
Datos Procesados	40	40
Total		134

Tabla 2.9. Espacio requerido por el sistema SeisComP3

El servidor virtual que ejecutará el sistema SeisComP3 contará con tres discos duros virtuales, uno destinado al sistema operativo, otro destinado a la adquisición de datos y un tercero en el que se almacenarán los datos procesados. En la tercera columna de la Tabla 2.9 se indica el tamaño que tendrán estos discos duros, a los que se ha agregado una holgura dependiendo de para que se utilizarán, para el disco en el que se instalará el sistema operativo se agregó 10 GB de holgura, espacio que se considera suficiente en caso de que sea necesario instalar programas o librerías. Para el disco de adquisición de datos también se agrega una holgura de 10 GB, este valor bastaría para agregar las 10 nuevas estaciones que el Instituto planea instalar en el año presente; una estación ocupa en un período de dos semanas aproximadamente 600 MB, por lo que 10 nuevas estaciones necesitarían 6 GB.

En el disco correspondiente a los datos procesados no se agrego ninguna holgura, ya que la información procesada que se guarda en la base de datos, imágenes y páginas web generadas se almacenarán en otro servidor. El servidor virtual en el que se instalará el sistema SeisComP3 necesitará por lo tanto un espacio de 134 GB.

En la Tabla 2.10 se estima el espacio que Earthworm necesita. El servidor virtual en el que se instalará el sistema Earthworm contará con dos discos duros, en uno se instalará el sistema operativo y en el otro se almacenarán los datos de los sensores sísmicos, en ambos casos se ha dejado una holgura suficiente. El servidor virtual en el que se instalará el sistema Earthworm necesita un almacenamiento de 60 GB.

Earthworm	Tamaño necesario (GB)	Tamaño de disco duro (GB)
Sistema Operativo Linux	20	30
Adquisición de Datos (2 semanas)	6	30
Total		60

Tabla 2.10. Espacio requerido para el sistema Earthworm

Como se indicó en la Tabla 2.7 el espacio para el sistema ShakeMap es de 60 GB, con lo que el espacio requerido por los tres sistemas sería de 254 GB.

Es importante mencionar que de ser necesario es posible aumentar o disminuir el tamaño de los discos virtuales mediante el comando `qemu-img`[129].

2.4.4.2 Requerimientos físicos de los nodos del cluster

En un *cluster* de alta disponibilidad todos los nodos que pertenecen al *cluster* son capaces de ejecutar recursos, sin embargo en caso de que solamente uno de los nodos del *cluster* esté activo, el mismo debe ser capaz de poder ejecutar todos los recursos del *cluster*.

Teniendo esta situación en cuenta sería deseable que un nodo del *cluster* cuente con las características de RAM, CPU y red que se presentan a continuación.

2.4.4.2.1 Tamaño de memoria RAM

A partir de los requisitos de memoria RAM para cada sistema, se determina que la cantidad de memoria con la que un nodo del *cluster* debería contar es 11 GB, como se indica en la Tabla 2.11.

[129] Qemu-img: es un comando que permite realizar tareas como revisar un disco, redimensionarlo, crear nuevos discos, etc.

Sistema	RAM (GB)
SeisComP3	3
Earthworm	6
ShakeMap	2
Total	11

Tabla 2.11. Memoria RAM requerida para un nodo del cluster

2.4.4.2.2 Número de Procesadores

En base a los requisitos estimados sobre el uso de CPU para cada uno de los sistemas se tiene que en total se requieren 6 CPU, como se indica en la Tabla 2.12, que es el mínimo número de núcleos que los servidores físicos que se usarán en el *cluster* necesitará en caso de que tenga que ejecutar los tres servidores virtuales simultáneamente.

Sistema	Número de CPU
SeisComP3	2
Earthworm	3
ShakeMap	1
Total	6

Tabla 2.12. Número de procesadores necesarios

2.4.4.2.3 Requisitos de red

Como se revisó en las secciones anteriores, el tráfico de red de los sistemas presentados no es intenso, por lo que no sería necesario que los servidores virtuales se conecten a una red separada de la red del *cluster*.

Sin embargo es necesario separar la red de comunicaciones del *cluster* y la red que utilizarán los servidores virtuales, por motivos de seguridad y para impedir que el tráfico de los sistemas virtuales interfiera con la red del *cluster*.

Por este motivo es necesario que los nodos del *cluster* de alta disponibilidad tengan por lo menos dos interfaces de red, una dedicada a la red de las máquinas virtuales y otra que se utilice en la red de comunicaciones del *cluster*.

2.4.5 REQUISITOS ADICIONALES DE LOS NODOS DEL CLUSTER

2.4.5.1 Tamaño del disco duro local

En cada nodo físico es necesario instalar un sistema operativo Linux, el software del *cluster* y algún otro software adicional, por lo que cada nodo deberá contar con un disco local de por lo menos 30 GB.

2.4.5.2 Características de procesador

Los procesadores de los servidores físicos que se van a emplear para el *cluster* de alta disponibilidad requieren soportar la virtualización asistida por hardware, es decir que deben tener la característica VT-x o AMD-V, dependiendo de si el procesador es Intel o AMD, y la misma debe estar habilitada.

2.4.5.3 Características de red

Como se determinó en la Sección 2.4.4.2.3, las máquinas virtuales deben estar conectadas a una red separada de la red del *cluster*, por lo que es necesario que cada nodo del *cluster* cuente con dos interfaces de red.

Sin embargo a fin de evitar que el fallo de una interfaz de red genere un error en la red de comunicaciones del *cluster* o en la red de las máquinas virtuales, es necesario contar con redundancia a nivel de red, por lo que los nodos del *cluster* deberán tener 4 interfaces de red, dos de las cuales se usarán en la red de comunicaciones del

cluster y las otras dos para la red de los servidores virtuales a la que se conectarán los clientes de los sistemas de adquisición y procesamiento.

Con esto ya es posible definir las características de hardware que deberá tener un nodo del *cluster* de alta disponibilidad, las mismas que se indican en la Tabla 2.13.

Característica	Requisito
CPU	6 CPU VT-x / AMD-V
RAM	11 GB
Disco Local	30 GB
Red	2 interfaces Gigabit Ethernet 2 interfaces Fast Ethernet

Tabla 2.13. Requisitos físicos de un nodo del cluster

CAPÍTULO III

3. DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN, PRUEBAS, RESULTADOS Y COSTOS

3.1 INTRODUCCIÓN

En este Capítulo se presenta el diseño del *cluster* de alta disponibilidad utilizando el software que se eligió en la Sección 1.4.4.

Además se presenta el diseño del almacenamiento compartido, el mismo brinda las características que el *cluster* de alta disponibilidad necesita.

Por último se presentan los pasos seguidos para la implementación de los diseños realizados, las pruebas correspondientes y una estimación de costos.

3.2 DISEÑO DEL CLUSTER DE ALTA DISPONIBILIDAD

3.2.1 ELECCIÓN DEL TIPO DE CLUSTER

De los tipos de *cluster* que se pueden implementar utilizando Pacemaker, presentados en la Sección 1.5.3.3, se descarta implementar un *cluster* de tipo activo/pasivo y de *failover* compartido, ya que en ambos casos uno de los nodos del *cluster* está encendido sin ejecutar ningún recurso, lo que puede considerarse como un desperdicio de recursos de hardware. En un *cluster* de alta disponibilidad de tipo activo/activo en cambio todos los nodos del *cluster* están activos y son capaces de ejecutar recursos y a la vez funcionar como respaldo en caso de que algún nodo del *cluster* falle.

Por esta razón se elige implementar un *cluster* de alta disponibilidad tipo activo/activo para el presente Proyecto de Titulación, al que se denominará solamente como *cluster* de alta disponibilidad.

3.2.2 ELECCIÓN DEL SISTEMA OPERATIVO

El software del *cluster* puede instalarse sobre diferentes sistemas operativos basados en Linux, se descarta utilizar sistemas operativos de escritorio como Ubuntu, Fedora, OpenSuse, etc. debido a que por lo general poseen un tiempo de soporte solamente de uno a tres años, y además porque por defecto incluyen programas que no son necesarios para un *cluster* como por ejemplo entorno gráfico, software para procesamiento de texto, software de audio, video, etc.

En cuanto a los sistemas operativos de tipo servidor, es posible utilizar Ubuntu Server LTS[130], Debian, CentOS, SUSE Linux Enterprise Server, Red Hat, entre otros.

Como se indica en la Tabla 3.1 SLES y Red Hat tienen los tiempos de soporte más altos, pero en ambos casos es necesario comprar una licencia especial para recibir lo que denominan soporte extendido, CentOS por otra parte cuenta con 10 años de soporte, es un sistema operativo sin costo de licenciamiento, es bastaste estable, tiene una amplia documentación y soporte por parte de la comunidad Linux, por lo que se elige este sistema operativo para implementar el *cluster* de alta disponibilidad.

Sistema Operativo	Tiempo de soporte
Ubuntu Server LTS	5 años [60]
Debian	3 años [61]
CentOS	10 años [62]
SUSE Linux Enterprise Server	5 a 13 años [63]
Red Hat	5 a 13 años [64]

Tabla 3.1. Tiempo de soporte de sistemas operativos Linux

[130] Ubuntu *Server Long Term Support*: es una versión para servidores del sistema operativo Ubuntu que brinda un tiempo de soporte mayor que la versión para escritorio.

3.2.3 DISEÑO DE LA RED PARA EL CLUSTER

Como se mencionó en la Sección 2.4.5.3, es necesario que la red de comunicaciones para el *cluster* y la red a la que se conectarán los servidores virtuales tengan redundancia, por lo que cada servidor debe contar con al menos cuatro interfaces de red, las que estarán enlazadas en modo activo/pasivo, dos interfaces de red estarán configuradas para que formen una interfaz denominada `bond0`, mientras que las otras dos interfaces de red formarán la interfaz `bond1`, la interfaz `bond0` se utilizará para las comunicaciones del *cluster*, mientras que la interfaz `bond1` servirá para que los clientes accedan a los servicios que brindan los servidores virtuales. Como se indica en la Figura 3.1, las interfaces de red que forman una interfaz enlazada están conectadas a diferentes conmutadores, de tal forma que si uno de los conmutadores falla, queda otro en funcionamiento lo que aumenta el nivel de redundancia de la red para el *cluster* y la red para los servidores virtuales.

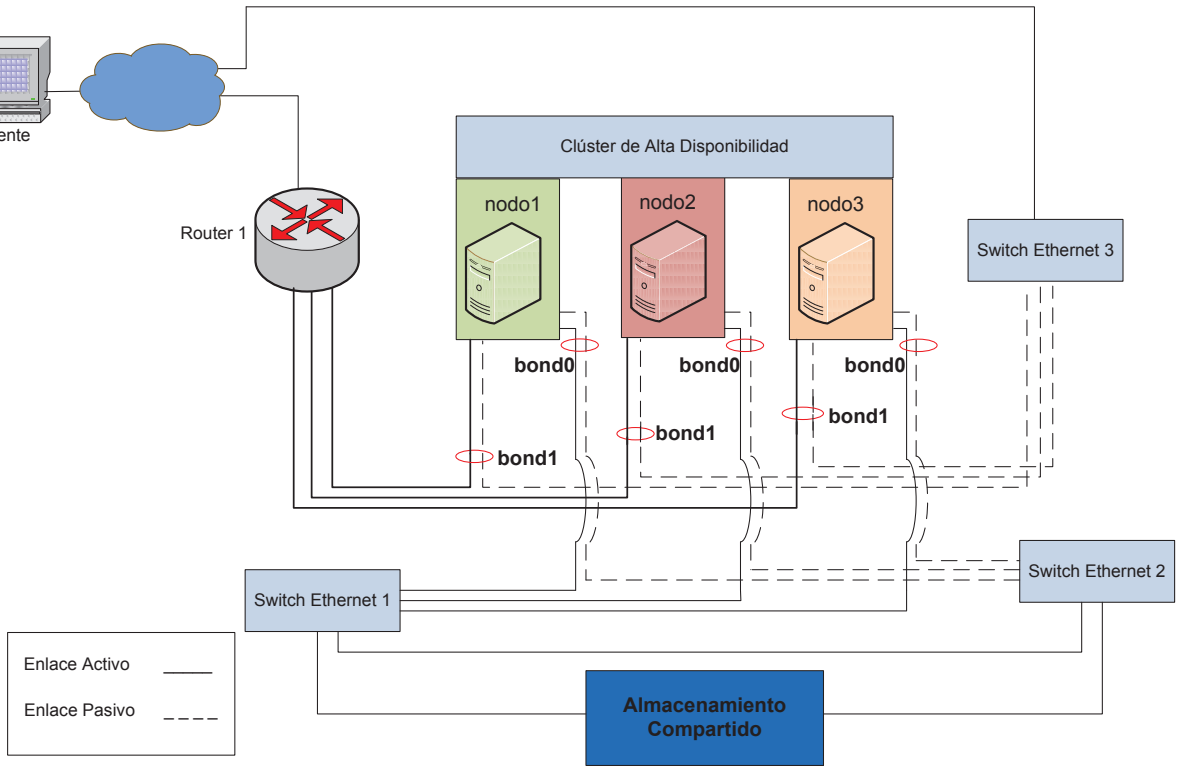

Figura 3.1. Diagrama de red para cluster de alta disponibilidad

Las direcciones IP para la red del *cluster* estarán en el rango `10.14.14.0` con la máscara de red `255.255.255.0`.

La red pública usará el rango de direcciones IP que sean asignadas por el Instituto Geofísico.

Para implementar la red del *cluster* se eligieron dos conmutadores de ocho puertos Gigabit Ethernet, mientras que para la red pública se usará un enrutador inalámbrico, que actúa como conmutador y cuenta con 4 puertos Gigabit Ethernet y un conmutador con ocho puertos Gigabit Ethernet.

El nombre de cada nodo seguirá el patrón: `nodoX` donde `X` corresponde al número del nodo. La red del *cluster* tiene el nombre `redwa.local`, y los nombres de los nodos serán: `nodo1.redwa.local`, `nodo2.redwa.local` y `nodo3.redwa.local`

3.2.4 ELECCIÓN DEL DISPOSITIVO DE FENCING

Para decidir qué dispositivo de *fencing* se utilizará para implementar el *cluster* se consultó la lista de dispositivos que recomienda Red Hat [65]. En esta lista existen tres marcas: WTI, APC y Baytech[131], que se ajustan al tipo de dispositivo *fencing* que se necesita, es decir un controlador de alimentación externo.

Se revisaron las hojas de datos de los modelos ofrecidos por estas marcas, los productos de APC y Baytech ofrecen conexión para veinte o más dispositivos, lo que es excesivo para el *cluster*. En cambio los dispositivos de la marca WTI de la serie NPS (*Network Power Switch*) tienen las características que el *cluster* requiere:

- Administración de hasta ocho dispositivos
- Conexión mediante puerto serial y Ethernet

[131] WTI, APC y Baytech: son empresas dedicadas al diseño y manufactura de dispositivos electrónicos relacionados con fuentes de alimentación eléctrica.

- Conexión para dos fuentes de alimentación
- Soporta cargas de hasta 15 Amperios
- Control de direcciones IP que pueden acceder al dispositivo
- Control de seguridad por contraseña a nivel de dispositivo y de puerto

Luego de revisar varios modelos de la serie NPS se eligió el modelo NPS-115 [66]. No es posible adquirir este dispositivo a nivel nacional, por lo que es necesario comprarlo en el exterior.

3.2.5 ELECCIÓN DEL SISTEMA DE ARCHIVOS DEL ALMACENAMIENTO COMPARTIDO

Como se mencionó en la Sección 1.8.4.3, para que el software de administración de recursos sea capaz de migrar las máquinas virtuales en caso que uno de los nodos del *cluster* falle, es necesario que todos los nodos del *cluster* sean capaces de leer y escribir en el almacenamiento compartido de forma simultánea. Para esto el almacenamiento debe contar con un sistema de archivos del tipo *cluster* como por ejemplo OCFS2 o GFS2.

OCFS2 no es compatible con las últimas versiones del sistema operativo CentOS, siendo necesario instalar un kernel especial para poder utilizar este sistema de archivos.

Por otra parte GFS2 es compatible con el sistema operativo CentOS, tiene bastante documentación y tiene soporte por parte de Red Hat , por lo que se elige utilizar GFS2 como sistema de archivos para el almacenamiento compartido.

3.2.6 CARACTERÍSTICAS DE LOS NODOS DEL CLUSTER

Como se determinó en la Sección 2.4.5, los nodos necesitan tener por lo menos 6 procesadores a fin de poder ejecutar las máquinas virtuales, así mismo necesitan tener por lo menos 11 GB de memoria RAM y cuatro interfaces de red a fin de tener

redundancia para la red pública y para la red de comunicaciones del *cluster*, mientras que para el almacenamiento local basta con 30 GB de espacio.

Para implementar el *cluster* se dispone de los computadores que se indican en la Tabla 3.2.

Computador	Dell PowerEdge T410	HP Proliant ML350e	IBM X3400 M3
Número de procesadores	12	4	16
Memoria RAM (GB)	12	8	8
Espacio en disco (GB)	1000	320	500
Interfaces de red	4	4	4
Nombre del servidor	nodo1	nodo2	nodo3

Tabla 3.2. Características de los servidores físicos

De los servidores presentados solamente el `nodo1` cumple con los requerimientos de memoria RAM y número de procesadores. Estos requisitos son necesarios en el caso que solamente uno de los nodos del *cluster* funcione y deba ejecutar todos los servidores virtuales al mismo tiempo. Sin embargo en condiciones normales los servidores `nodo2` y `nodo3` pueden tienen características de hardware que les permitirían ejecutar sin problema los servidores virtuales, razón por la que se los incluye en la implementación del *cluster*.

3.3 DISEÑO DEL ALMACENAMIENTO COMPARTIDO

El almacenamiento compartido brinda al *cluster* de alta disponibilidad la capa de almacenamiento que necesita para funcionar.

De los tipos de almacenamiento compartido que el *cluster* puede usar, presentados en la Sección 1.3.2.1, si se utiliza un almacenamiento de instancia única o SIS, como por ejemplo una SAN, el almacenamiento se convertiría en un punto único de falla,

es decir todo el *cluster* y los servicios de alta disponibilidad dependerían del correcto funcionamiento de la SAN.

El almacenamiento de tipo replicado elimina este punto único de falla, al sincronizar los datos del almacenamiento compartido con uno o más nodos. En este tipo de almacenamiento existe un nodo que actúa como servidor de datos principal, en caso que éste falle, uno de los nodos de respaldo toma su lugar, sin que exista pérdida de datos o que los servidores conectados al almacenamiento se percaten del cambio.

Tomando esto en cuenta se optó por emplear el almacenamiento compartido de tipo replicado en el Proyecto de Titulación. En la Tabla 3.3 están indicadas las capas con las que contará el almacenamiento compartido replicado, así como las tecnologías que pueden usarse para su implementación, las mismas que se revisaron en la Sección 1.6.

Capa	Tecnología disponible
Sistema de Archivos	GFS2, OCFS2
Tecnología de Acceso al Almacenamiento Compartido	iSCSI, NFS
Tecnología de Replicación	Replicación vía hardware o software
Redundancia de Capa Física	RAID vía hardware o software
Capa Física	Discos duros SATA

Tabla 3.3. Capas de un almacenamiento compartido replicado con alta disponibilidad

El diseño empezará con la capa de tecnología de replicación ya que de la misma dependerán el resto de niveles.

3.3.1 ELECCIÓN DE LA TECNOLOGÍA DE REPLICACIÓN

Como se mencionó en la Sección 1.3.2.1.2, el almacenamiento replicado puede implementarse mediante hardware o software, sin embargo la primera opción a más de ser bastante costosa, implicaría utilizar la tecnología de un solo fabricante. Por

ejemplo, para utilizar el almacenamiento replicado de IBM, denominado PPRC[132], sería necesario adquirir dos sistemas de almacenamiento SAN, el software para administrar la SAN, enlaces de fibra, etc., además los servidores físicos del *cluster* de alta disponibilidad necesitarían tarjetas de fibra compatibles con IBM, entre otros requisitos.

La replicación mediante software DRBD, por otra parte, es de bajo costo, puede implementarse con partes fáciles de conseguir en el mercado nacional, es de código abierto, no requiere enlaces de fibra, sino que puede utilizar enlaces Gigabit Ethernet, entre otras ventajas [67].

Para que el *cluster* de alta disponibilidad pueda acceder al dispositivo de bloque DRBD es necesario que el mismo esté disponible a través de la red, para lo que se recomienda utilizar la tecnología iSCSI, se descarta utilizar NFS debido a que NFS brinda acceso remoto a un sistema de archivos local, tal como ext4[133], y el *cluster* requiere un sistema de archivos *cluster*, tal como GFS2.

Como se indicó en la Sección 1.6.2.2, para que la conmutación por error entre los servidores DRBD se realice de forma automática, el recurso DRBD debe ser controlado por un administrador de recursos de un *cluster*, es decir que para poder emplear la tecnología DRBD en el almacenamiento compartido es necesario que el almacenamiento compartido sea a su vez un *cluster* de alta disponibilidad donde el servicio que tiene alta disponibilidad será el dispositivo iSCSI que utiliza el bloque DRBD.

Para evitar confusiones se denominará al *cluster* de almacenamiento con alta disponibilidad simplemente como *cluster* de almacenamiento.

---

[132] *Peer to Peer Remote Copy*: es una tecnología que permite replicar datos entre dos sistemas de almacenamiento IBM.

[133] Ext4: es un sistema de archivos usado en sistemas operativos Linux. Es el sistema de archivos por defecto para muchas distribuciones de Linux.

3.3.2 DETALLES DE LA CAPA FÍSICA

Los computadores disponibles para realizar el almacenamiento compartido tienen puertos SATA versión 1.0, es decir velocidades de 150 MB/s, siendo esta tecnología la que se utilizará para los discos duros.

3.3.3 ELECCIÓN DEL TIPO DE REDUNDANCIA DE CAPA FÍSICA

Con el fin de aumentar la redundancia del almacenamiento compartido se ha decidido implementar un RAID con los discos físicos que se dispone.

Como se mencionó en la Sección 1.6.2.3 una solución RAID implementada mediante software puede competir, en ciertos casos, con un RAID del tipo hardware, gracias a los nuevos procesadores y al mejoramiento del software RAID empleado en Linux, por lo que se opta por implementar el RAID empleando el driver `md`.

Los niveles de RAID que pueden implementarse mediante software en Linux son RAID 0, 1, 4, 5, 6, 1+0. Se descartan los niveles 0, por falta de redundancia y 4 por bajo desempeño en escritura y lectura.

En un arreglo RAID 1 se requiere un mínimo de 2 discos de igual tamaño, en un arreglo nivel 5 se requiere un mínimo de 3 discos, mientras que los niveles 6 y 1+0 necesitan un mínimo de 4 discos.

De las tres opciones la que tiene mejores características de desempeño y confiabilidad es RAID 1+0.

En la Figura 3.2 puede verse las diferentes velocidades de escritura para los distintos niveles de RAID medido en IOPS[134].

---

[134] *Input/Output Operations Per Second*: es una forma usual de medir el desempeño de dispositivos de almacenamiento tal como discos duros, discos de estado sólido, etc.

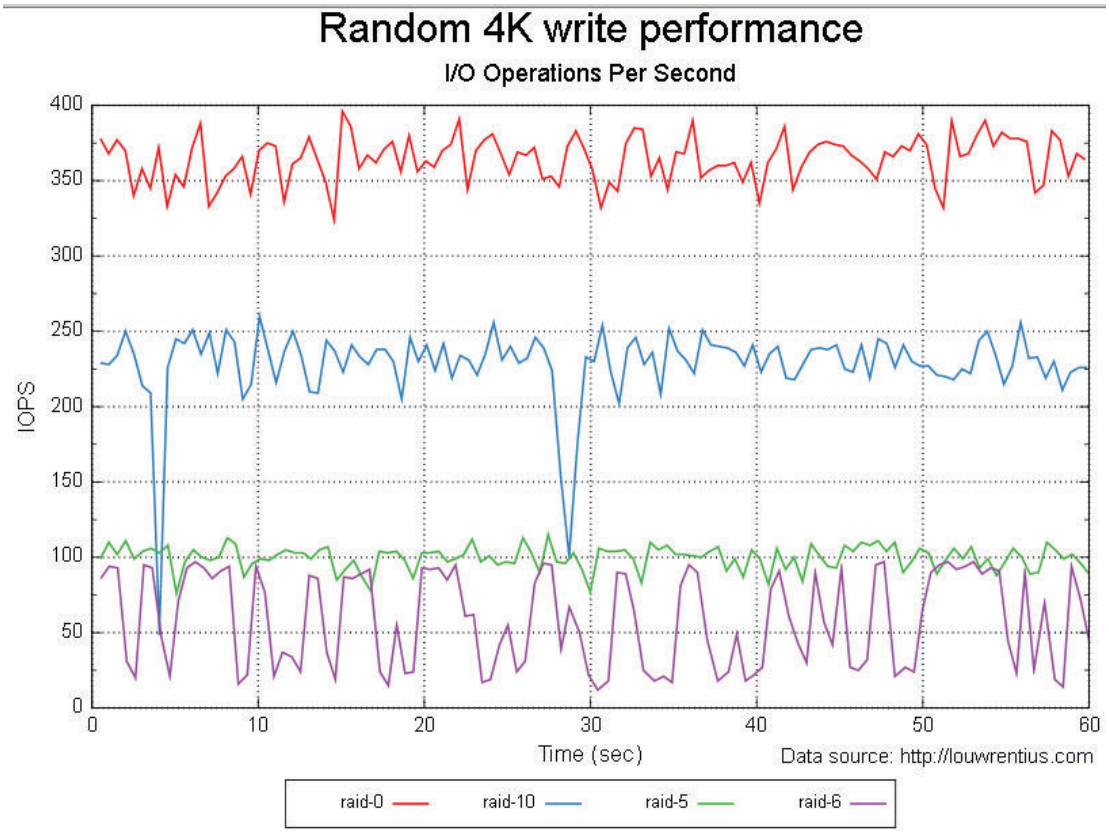

Figura 3.2. Comparación del desempeño de escritura de niveles de RAID [68]

El nivel 0 es el que tiene el mejor desempeño, pero no ofrece ninguna redundancia, mientras que el nivel 1+0 , tiene el segundo mejor desempeño.

En cuanto a los niveles 5 y 6, ambos presentan desempeños bajos de escritura, debido a que ambos deben realizar cálculos de paridad antes de escribir en el disco.

En la Figura 3.3 se presentan las velocidades de lectura, que no varían demasiado para los diferentes niveles.

El inconveniente de utilizar RAID 1+0 es que el número mínimo de discos necesarios es 4, el doble de RAID 1, teniendo esto en cuenta se decide utilizar RAID 1 para la primera capa de redundancia, ya que como se indicó en la Sección 1.6.1.3.6. tiene un desempeño similar al de RAID 1+0.

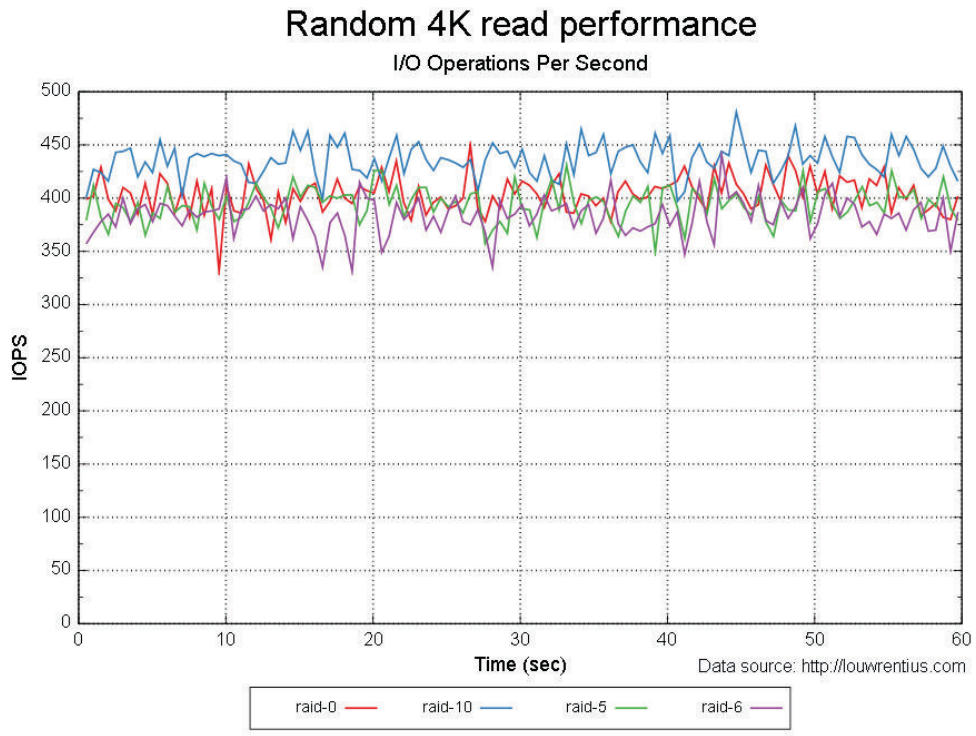

Figura 3.3. Comparación del desempeño de lectura de niveles de RAID [68]

3.3.4 DETALLES DE LA TECNOLOGÍA DE ACCESO

La siguiente capa tiene que ver con la tecnología que se usará para que los nodos del *cluster* accedan al almacenamiento compartido. Como se indicó en la Sección 3.3.1, es necesario utilizar la tecnología iSCSI para conseguir que el dispositivo DRBD sea accesible a través de la red, y dado que a partir del año 2011 el software del proyecto LIO se convirtió en el software por defecto para implementar un *target* iSCSI en Linux, este es el software que se empleará para esta capa del *cluster* del almacenamiento.

3.3.5 DETALLES DEL SISTEMA DE ARCHIVOS SELECCIONADO

Como se indicó en la Sección 3.2.5, el *cluster* de alta disponibilidad necesita un sistema de archivos de tipo *cluster* compatible con CentOS 7, por lo que se utilizará GFS2. Este sistema de archivos requiere, para garantizar la integridad del sistema de archivos, que se utilice junto con CLVM, como se indicó en la Sección 1.6.4.2.1.

Al momento de crear un sistema de archivos de este tipo, es necesario especificar algunos parámetros, los mismos que se presentan en la Tabla 3.4 y que fueron indicados en la Sección 1.6.4.2.

Característica	Valor
Número de registros	3
Nombre del *cluster*	clusterwa
Nombre del sistema de archivos	iscsiwa

Tabla 3.4. Parámetros del sistema de archivos GFS2

3.3.6 DISEÑO DEL CLUSTER DE ALMACENAMIENTO

Como se indicó en la Sección 3.3.1, el almacenamiento compartido se configurará como un *cluster* de alta disponibilidad. Este *cluster* estará formado por dos computadores que serán los servidores DRBD primario y secundario, y empleará el mismo software que se eligió para el *cluster* de alta disponibilidad, es decir Pacemaker y Corosync.

3.3.6.1 Elección del tipo de cluster

El tipo de *cluster* que se ajusta a los requerimientos de la tecnología de replicación de datos DRBD es un *cluster* de alta disponibilidad activo/pasivo, que se presenta en la Sección 1.5.3.3.1.

DRBD necesita de un *cluster* tipo activo/pasivo debido a que, como se indicó en la Sección 1.6.2.2, solo el servidor DRBD primario está activo y es en el que los nodos del *cluster* leerán y escribirán datos, mientras que en el servidor DRBD secundario o pasivo se realiza la replicación de datos. No es posible implementar el *cluster* de almacenamiento como un *cluster* activo/activo, porque eso implicaría tener dos servidores DRBD primarios y ese tipo de configuración podría generar corrupción en los datos del almacenamiento compartido [69].

Para el *cluster* de almacenamiento se seguirá el mismo diseño que para el *cluster* de alta disponibilidad, en cuanto a sistema operativo, direcciones de red, dispositivo de *fencing*, etc.

Los nombres de los nodos serán `iscs1` y `iscs2` y se encontrarán en el dominio `almwa.local`, las direcciones IP de cada nodo serán `10.14.14.11` y `10.14.14.12` respectivamente. El *cluster* creado se denominará `clusteralmawa.`

3.3.6.2 Requisitos de los nodos del cluster de almacenamiento

Debido a que los nodos que formen el *cluster* de almacenamiento no tendrán una carga de trabajo excesiva se emplearán dos computadores de escritorio a los que se agregará el almacenamiento necesario y una tarjeta Ethernet adicional, a fin de que exista redundancia a nivel de red.

Como se determinó en la Sección 2.4.4.1 el almacenamiento debe tener un tamaño aproximadamente igual a 240 GB, a fin de que contenga los discos de las tres máquinas virtuales que se crearán.

Una vez terminados los diseños se procede a implementar en primer lugar el *cluster* de almacenamiento, ya que la capa de almacenamiento es la base del *cluster* de alta disponibilidad.

3.4 IMPLEMENTACIÓN DEL CLUSTER DE ALMACENAMIENTO

En esta sección se presenta la implementación de un target iSCSI sobre un volumen DRBD, el que a su vez empleará un disco RAID 1. Estos recursos estarán a su vez administrados en un *cluster* activo/pasivo empleando Pacemaker y Corosync.

El objetivo es conseguir un sistema de almacenamiento compartido y replicado como el que se presenta en la Figura 3.4.

Se dispone de dos computadores con tres discos duros cada uno, dos de esos discos son de 500 GB cada uno y se emplearán para crear el disco RAID de nivel 1.

En el tercer disco se instalará el sistema operativo CentOS 7. Los discos se han conectado en los puertos SATA de los computadores de tal forma que el sistema operativo le asigne el nombre `sda` al disco en el que se instalará el sistema operativo y los nombres `sdb` y `sdc` a los discos que formarán el RAID.

Cada computador dispone de dos interfaces de red, las mismas que se enlazarán para formar la interfaz `bond0`. Estas interfaces se conectan a dos conmutadores que permitirán a los nodos del *cluster* de alta disponibilidad acceder al almacenamiento compartido.

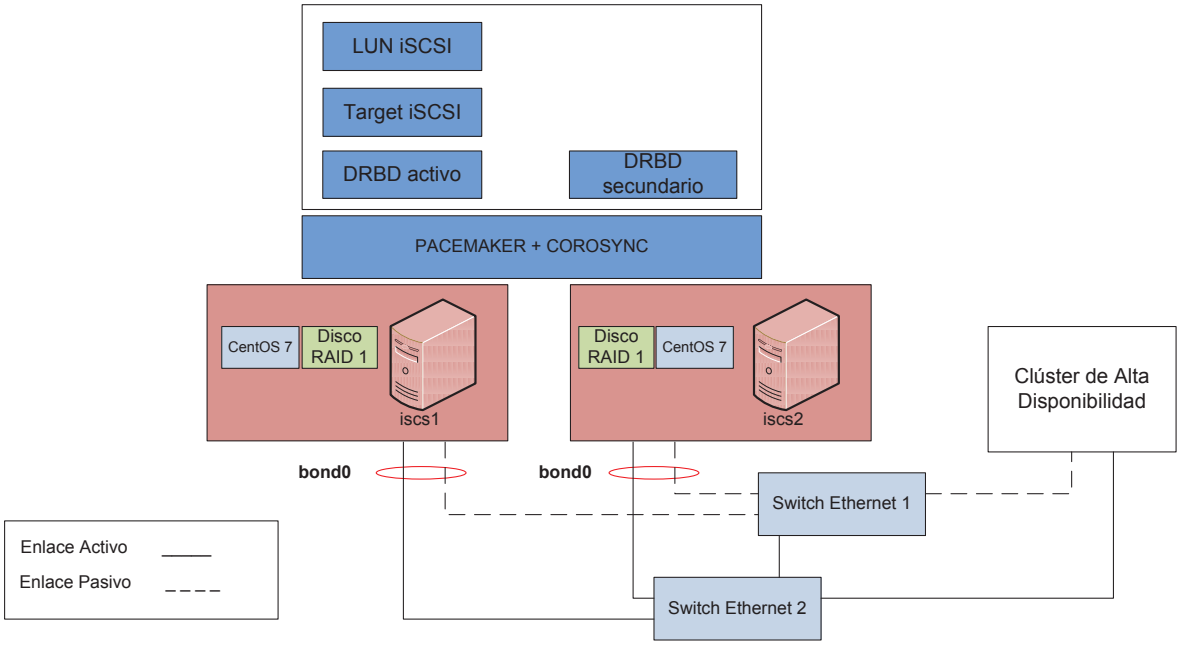

Figura 3.4. Diagrama del cluster de almacenamiento a implementar

3.4.1 INSTALACIÓN DEL SISTEMA OPERATIVO

Una imagen ISO del DVD de CentOS 7 se puede descargar de [70]. El proceso de instalación es bastante sencillo, la información que se configura es el nombre del *host*, la dirección IP, la zona horaria.

Se elige instalar un servidor básico sin interfaz gráfica y se utiliza la configuración por defecto para las particiones del disco duro.

Una vez instalado el sistema operativo es necesario realizar algunas configuraciones antes de continuar.

3.4.2 INSTALACIÓN DE NTP[135]

Es muy importante que los relojes de los computadores que forman el *cluster* estén sincronizados, para lo que se instala, activa e inicia el cliente NTP mediante los comandos que se presentan en la Línea de Comandos 3.1

```
# yum install ntp
# systemctl enable ntpd
# systemctl start ntpd
```

Línea de Comandos 3.1. Instalar y activar el servicio NTP

3.4.3 CONFIGURACIÓN DE SEGURIDAD

En la documentación de Red Hat [71] y [72], se recomienda fijar las políticas de SELinux en el nivel deshabilitado, debido a problemas de compatibilidad con Pacemaker y a que las políticas de SELinux para Pacemaker aún no se han completado, por lo que en el archivo `/etc/selinux/config` la variable `SELINUX` tiene el valor `disabled`, como se indica en el Archivo de configuración 3.1.

```
SELINUX=disabled
SELINUXTYPE=targeted
```

Archivo de configuración 3.1. Contenido del archivo /etc/selinux/config

Debido a que la red del *cluster* estará separada de la red pública y no tendrá una salida hacia ninguna red externa, se detiene y deshabilita el cortafuegos interno de los servidores, con los comandos presentados en la Línea de Comandos 3.2.

---

[135] *Network Time Protocol*: es un protocolo para sincronizar el reloj de un computador mediante Internet.

```
# systemctl stop firewalld
# systemctl disable firewalld
```

Línea de Comandos 3.2. Deshabilitar el cortafuegos incluido en CentOS 7

La resolución de nombres se realizará configurando los archivos `/etc/hosts` de cada nodo que forma el *cluster*, este archivo contendrá el nombre de cada nodo y su correspondiente dirección IP como se indica en el Archivo de configuración 3.2.

```
127.0.0.1   localhost localhost.localdomain
10.14.14.11 iscs1 iscs1.almwa.local
10.14.14.12 iscs2 iscs2.almwa.local
```

Archivo de configuración 3.2. Contenido del archivo /etc/hosts

3.4.4 CONFIGURACIÓN DE LAS INTERFACES DE RED [73]

Como se indicó en la Sección 3.3.6.2, los nodos del *cluster* de almacenamiento requieren dos interfaces de red que se enlazarán para formar una interfaz de red virtual `bond0`, que brinde redundancia, a continuación se presenta el procedimiento para enlazar las interfaces de red de uno de los computadores, el mismo procedimiento se repite en el segundo nodo, con las respectivas modificaciones.

En la Línea de Comandos 3.3 se presentan las interfaces de red disponibles, las mismas son `enp0s25` y `enp6s1`.

```
# ifstat
enp6s1
enp0s25
```

Línea de Comandos 3.3. Comando para ver las interfaces disponibles en el servidor iscs1

Para configurar estas interfaces se crean sus respectivos archivos de configuración con los comandos presentados en la Línea de Comandos 3.4.

```
# touch /etc/sysconfig/network-scripts/ifcfg-enp0s25
# touch /etc/sysconfig/network-scripts/ifcfg-enp6s1
```

Línea de Comandos 3.4. Creación de los archivos de configuración de las interfaces de red

El contenido de estos archivos puede verse en el Archivo de Configuración 3.3, es necesario remplazar el contenido del campo `DEVICE` con el nombre de cada interfaz.

```
DEVICE=XXX
MASTER=bond0
SLAVE=yes
USERCTL=no
ONBOOT=yes
BOOTPROTO=none
```

Archivo de Configuración 3.3. Contenido del archivo ifcfg-XXX

A continuación se crea el archivo de configuración para la interfaz enlazada, denominada `bond0`, como se indica en la Línea de Comandos 3.5. El contenido de este archivo se presenta en el Archivo de Configuración 3.4.

```
# touch /etc/sysconfig/network-scripts/ifcfg-bond0
```

Línea de Comandos 3.5. Creación del archivo de configuración de la interfaz de red bond0

Una vez realizados estos pasos para activar las interfaces basta con reiniciar el servicio de red con el comando de la Línea de Comandos 3.6.

```
DEVICE=bond0

NAME=bond0

BOOTPROTO=none

ONBOOT=yes

IPADDR=10.14.14.11

PREFIX=24

GATEWAY=10.14.14.253

DNS1=192.168.2.22

DNS2=192.168.3.33

BONDING_MASTER=yes

BONDING_OPTS="mode=active-backup miimon=100"
```

Archivo de Configuración 3.4. Contenido del archivo /etc/sysconfig/network-scripts/ifcfg-bond0

Puede probarse la interfaz `bond0` mediante el comando que se presenta en la Línea de Comandos 3.7.

Como puede verse la interfaz `bond0` está activa y utiliza la interfaz esclava `enp0s25`, mientras que `enp6s1` funciona como respaldo.

Una vez realizadas estas configuraciones ya es posible instalar el software del *cluster*, procedimiento que se indica en la siguiente sección.

```
# systemctl restart network.service
```

Línea de Comandos 3.6. Comando para crear y cargar la interfaz bond0

```
# watch –n 2 cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)
Bonding Mode: fault-tolerance (active-backup)
Currently Active Slave: enp0s25
Slave Interface: enp0s25
Speed: 1000 Mbps
Slave Interface: enp6s1
Speed: 1000 Mbps
```

Línea de Comandos 3.7. Comando para revisar el funcionamiento de la interfaz bond0

3.4.5 INSTALACIÓN DEL SOFTWARE DEL CLUSTER

El comando presentado en la Línea de Comandos 3.8 se ejecuta en los dos nodos del *cluster* ya que instala el software de comunicación del *cluster* Corosync, el software del *cluster* de alta disponibilidad Pacemaker, y la consola de comandos pcs que se utiliza para configurar el *cluster*.

```
# yum install pacemaker Corosync pcs
```

Línea de Comandos 3.8. Instalación del software requerido para el cluster de alta disponibilidad

Una vez terminada la instalación es necesario habilitar e iniciar `pcs` como un servicio del sistema y activarlo, como se indica en la Línea de Comandos 3.9.

Estos comandos deben repetirse en el segundo nodo.

```
# systemctl enable pcsd.service
# systemctl  start pcsd
```

Línea de Comandos 3.9. Habilitar el programa pcs como un servicio

Al instalar el software del *cluster* se creó automáticamente el usuario `hacluster`, que es el usuario que el *cluster* utiliza para tareas de configuración.

Es necesario agregar una clave a esa cuenta con el procedimiento que se indica en la Línea de Comandos 3.10.

Este procedimiento debe repetirse en el segundo nodo.

```
# echo ********* | passwd --stdin hacluster
##Resultado
Changing password for user hacluster
passwd: all authentication tokens updated successfully.
```

Línea de Comandos 3.10. Cambio de clave del usuario hacluster

```
# pcs cluster auth iscs1 iscs2
Username: hacluster
Password: **********
##Resultado si la autenticación es exitosa:
iscs1: Authorized
iscs2: Authorized
```

Línea de Comandos 3.11. Autenticación del usuario hacluster en los nodos del cluster de almacenamiento

Para que las operaciones de configuración del *cluster*, se repliquen en todos los nodos del *cluster*, de forma automática, se ejecuta el comando que se indica en la Línea de Comandos 3.11.

El comando solicitará el usuario que se usará para las operaciones de configuración, es decir `hacluster` y la clave que se le asignó en la Línea de Comandos 3.10.

Este comando solo se ejecuta en uno de los nodos, no importa en cual.

Una vez realizadas estas configuraciones ya es posible administrar todas las operaciones del *cluster* desde cualquiera de los nodos que forman el *cluster*.

El comando presentado en la Línea de Comandos 3.12 se encarga de crear el *cluster* `clusteralmawa`, y luego inicia Corosync y Pacemaker.

Como resultado del comando de la Línea de Comandos 3.12 se crea el archivo de configuración del software de comunicaciones Corosync, que se indica en el Archivo de Configuración 3.5, el cual contiene: el nombre del *cluster*, los nodos que son parte del mismo y la forma de comunicar los mensajes.

Para arrancar el *cluster* se ejecuta el comando presentado en la Línea de Comandos 3.13, como puede verse este comando inicia el *cluster* en ambos nodos.

```
# pcs cluster setup --name clusteralmawa iscs1 iscs2
##Resultado si el comando es exitoso:
iscs1: Succeeded
iscs2: Succeeded
```

Línea de Comandos 3.12. Creación del cluster “clusteralmawa” formado por iscs1 y iscs2

Para iniciar el software del *cluster* junto con el sistema operativo se utiliza el comando presentado en la Línea de Comandos 3.14.

En la Línea de Comandos 3.15 se puede ver el estado actual del *cluster*. El *cluster* está compuesto por dos nodos que están activos, el *cluster* tiene quórum para funcionar y no existen recursos configurados. Pacemaker ha designado al servidor `iscs2` como DC o coordinador del *cluster* e identifica que se utiliza Corosync como software de comunicaciones del *cluster*. El mensaje de alerta `WARNING` se debe a que aún no se ha configurado un dispositivo de *fencing* que es lo que se realizará en la siguiente sección.

```
totem {
version: 2
secauth: off
cluster_name: clusteralmawa
transport: udpu
}
nodelist {
node {
ring0_addr: iscs1
nodeid: 1
}
node
{
ring0_addr: iscs2
nodeid: 2
}}
```

Archivo de Configuración 3.5. Contenido del archivo /etc/corosync/corosync.conf

```
# pcs cluster start --all
##Resultado del comando:
iscs1: Starting Cluster...
iscs2: Starting Cluster...
```

Línea de Comandos 3.13. Comando para iniciar el cluster

```
# pcs cluster enable --all
##Resultado del comando:
iscs1: Cluster Enabled
iscs1: Cluster Enabled
```

Línea de Comandos 3.14. Comando para arrancar el cluster junto con el sistema operativo

3.4.6 CONFIGURACIÓN DEL CLUSTER DE ALMACENAMIENTO

Para realizar la configuración del *cluster* es posible utilizar la consola de comandos `pcs` o hacerlo vía web, por facilidad se elige utilizar la interfaz de comandos.

3.4.6.1 Agregar un dispositivo de fencing al cluster

3.4.6.1.1 Configuración del dispositivo NPS-115

La configuración del dispositivo se realizó conectando un computador al puerto serial[136] del dispositivo NPS-115 y utilizando el programa HyperTerminal[137].

Una vez se conecta al dispositivo aparece la información que se presenta en la Figura 3.5, como los puertos disponibles, el nonmbre asignado a cada uno de ellos, el estado del puerto activo/apagado, la clave asignada a cada puerto, etc.

---

[136] Puerto Serial: es una interfaz de comunicaciones en la que la información se transmitía un bit a la vez.

[137] HyperTerminal: es un programa que permite la comunicación remota entre un computador y un dispositivo de red.

Se configurará una clave para acceder al dispositivo, las direcciones IP que pueden acceder al mismo y la información de red. En la Línea de Comandos 3.16 se presenta la configuración de la clave de acceso, todos los comandos tienen el formato `/COMANDO`, por ejemplo, para configurar los parámetros generales se utiliza el comando `/G` y luego se ingresa la opción `1` para ingresar la clave general.

```
# pcs status
Cluster name: clusteralmawa
WARNING: no stonith devices and stonith-enabled is not false
Stack: corosync
Current DC: iscs2 (2) - partition with quorum
2 Nodes configured
0 Resources configured
Online: [ iscs1 iscs2 ]
Full list of resources:
PCSD Status:
iscs1: Online
iscs2: Online
Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Línea de Comandos 3.15. Comando para verificar el estado del cluster

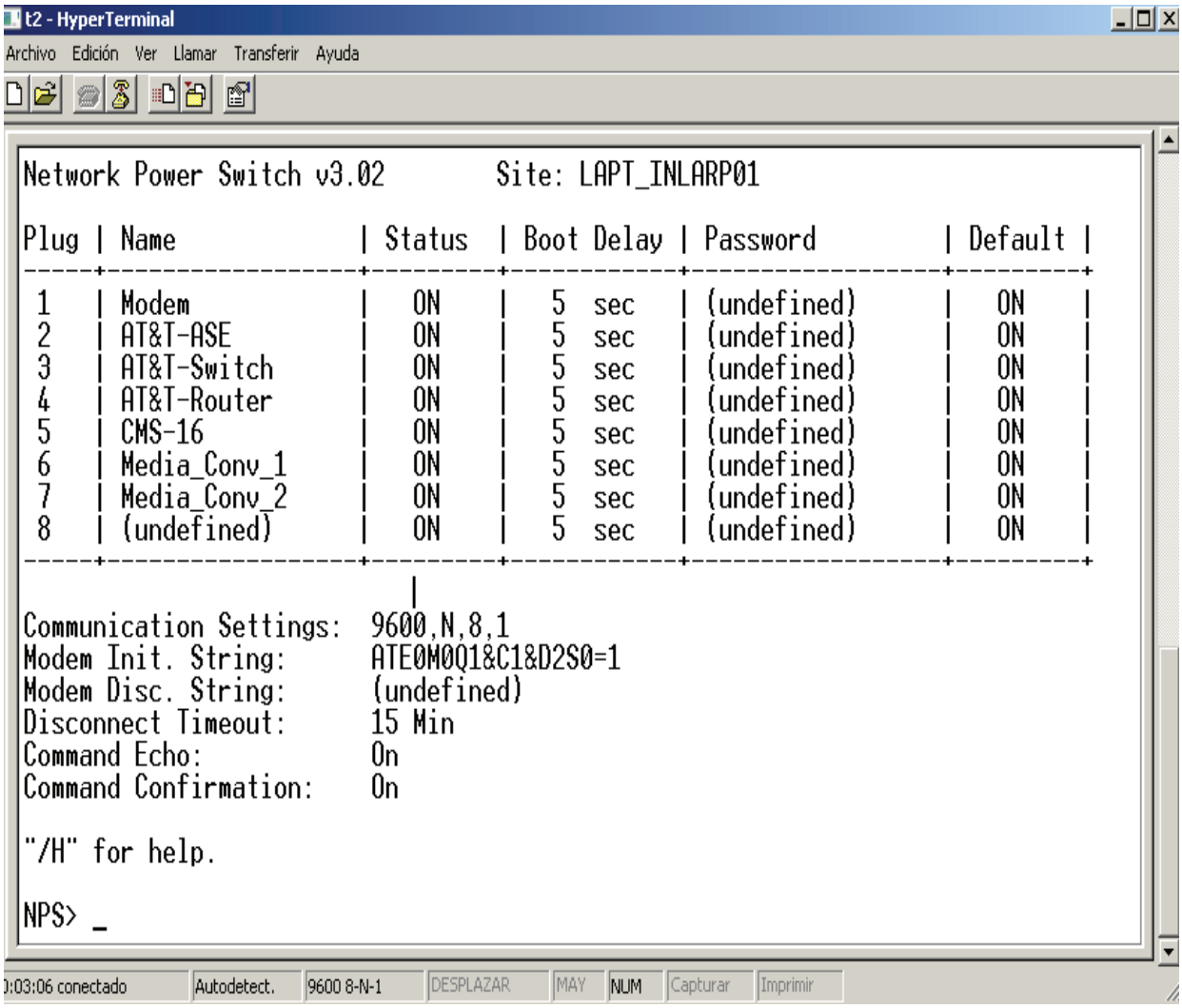

Figura 3.5. Conexión inicial del dispositivo NPS-115

3.4.6.1.2 Configuración de la información de red del dispositivo

Se usará la información que se indica en la Tabla 3.5 para configurar la interfaz de red del dispositivo NPS-115.

Para acceder a la configuración de red se utiliza el comando `/N`, luego se elige el parámetro a configurar, por ejemplo al elegir la opción 1 se configura la dirección IP, como se indica en la Línea de Comandos 3.17, la opción 2 configura la máscara de red, la opción 3 configura la dirección IP de la compuerta de enlace y la opción 4 configura la lista de direcciones IP que tienen acceso al dispositivo.

```
NPS> /G

GENERAL PARAMETERS:

1. System Password:       (defined)

2. Site ID:               CLUSTERWA

Enter Selection or <ESC> to Exit ... 1

SYSTEM PASSWORD:

Up to 16 characters are allowed for the password.

******

Enter password again to verify: ******
```

Línea de Comandos 3.16. Configuración de la clave de acceso al dispositivo NPS-115

Característica	Valor
IP	10.14.14.31
Máscara	255.255.255.0
Puerta de enlace	10.14.14.253

Tabla 3.5. Datos de configuración para el dispositivo de NPS-115

3.4.6.1.3 Agregar el dispositivo NPS-115 al cluster

Pacemaker dispone de un agente para administrar el dispositivo de *fencing*, para instalarlo se ejecuta el comando que se indica en la Línea de Comandos 3.18, en cada uno de los nodos que pertenecen al *cluster*.

Una vez instalado el agente de recursos se realiza el procedimiento de la Línea de Comandos 3.19 para agregar el dispositivo al *cluster* de almacenamiento, los

parámetros que el comando configura son: el nombre del recurso, la dirección IP, la clave de acceso y `pcmk_host_map`, en donde se ingresa el nombre de los nodos y el puerto del dispositivo de *fencing* en el que están conectados en el formato `Nombre_Nodo:Número_Puerto`, por ejemplo `iscs1:5;iscs2:6`.

```
NPS> /N

NETWORK PARAMETERS:

1. IP Address:          1.1.1.1

2. Subnet Mask:         255.255.255.0

3. Gateway Address:     10.14.14.253

4. IP Security

Enter Selection or <ESC> to Exit ... 1

IP ADDRESS:

Enter the IP Address.

10.14.14.31
```

Línea de Comandos 3.17. Configuración de la interfaz de red del dispositivo NPS-115

```
# yum install fence-agents-wti.x86_64
```

Línea de Comandos 3.18. Instalación del agente para el dispositivo NPS-115

3.4.6.2 Activar el encendido automático en los nodos

Para que el dispositivo NPS-115 sea capaz de apagar, encender o reiniciar los nodos que forman el *cluster*, es necesario activar la opción `AC/POWER LOSS` en el BIOS de los nodos, esta opción permite que el nodo se encienda en cuanto la fuente de alimentación se conecte, sin necesidad de presionar el botón de encendido.

```
# pcs cluster cib stonith_cfg

# pcs -f stonith_cfg stonith create wtialma fence_wti
pcmk_host_map="iscs1:5;iscs2:6" ipaddr=10.14.14.31
passwd=****** op monitor interval=60s

# pcs -f stonith_cfg stonith

# pcs -f stonith_cfg property set stonith-enabled=true

# pcs -f stonith_cfg property

# pcs cluster cib-push stonith_cfg
```

Línea de Comandos 3.19. Agregar el dispositivo NPS-115 al cluster

Una vez activada esta opción ya es posible realizar la prueba del dispositivo *fencing*, mediante el comando que se indica en la Línea de Comandos 3.20, que en caso de funcionar correctamente debe apagar y encender nuevamente el nodo `iscs1`.

```
# stonith_admin --reboot iscs1
```

Línea de Comandos 3.20. Comando para probar el funcionamiento del fencing

El siguiente paso es configurar los recursos del *cluster*, pero para ello primero es necesario crear el disco RAID 1 mediante el software `md`, lo que se revisa en la siguiente sección.

3.4.7 CREACIÓN DEL RAID 1

Antes de iniciar la instalación es necesario que el kernel de Linux soporte el módulo `md` y que el comando `mdadm` esté instalado.

Por lo general la mayoría de distribuciones de Linux cumplen con estos requisitos, con el comando presentado en la Línea de Comandos 3.21 se verifica que el kernel tiene soporte para `md`.

```
# modinfo raid1

filename:       /lib/modules/3.10.0-
123.el7.x86_64/kernel/drivers/md/raid1.ko

alias:          md-level-1

alias:          md-raid1

description:    RAID1 (mirroring) personality for MD
```

Línea de Comandos 3.21. Comando para verificar el soporte para el módulo de RAID 1

Mediante el comando presentado en la Línea de Comandos 3.22 puede conocerse los discos duros de los que se dispone para realizar la instalación.

```
##Comando ejecutado en iscs1##

# fdisk -l

Disk /dev/sda: 250.1 GB

Disk /dev/sdb: 500.1 GB

Disk /dev/sdc: 500.1 GB

##Comando ejecutado en iscs2##

# fdisk -l

Disk /dev/sda: 320.1 GB

Disk /dev/sdb: 500.1 GB

Disk /dev/sdc: 500.1 GB
```

Línea de Comandos 3.22. Comando para presentar la información de los discos disponibles en cada nodo

Cada computador dispone de tres discos duros, uno de ellos, `/dev/sda` ha sido utilizado para instalar el sistema operativo, mientras que `/dev/sdb y /dev/sdc` se utilizarán para crear el volumen RAID 1. El comando que se indica en la Línea de

Comandos 3.23 carga el módulo `md` en el kernel de Linux. Este comando debe ejecutarse en los dos nodos.

```
# modprobe --verbose raid1
```

Línea de Comandos 3.23. Comando para cargar el módulo md en el kernel

3.4.7.1 Creación de particiones

El comando para crear la partición se presenta en la Línea de Comandos 3.24

```
# fdisk /dev/sdb

Command (m for help): n

Command action

e   extended

p   primary partition (1-4): p

Partition number (1-4): 1

First cylinder : 2048

Last cylinder or +sizeM: 490000M

Command (m for help): t

Selected partition 1

Hex code (type L to list codes): fd

Command (m for help): w

The partition table has been altered!
```

Línea de Comandos 3.24. Procedimiento para crear la partición para el disco RAID 1

Se presenta el proceso para el nodo `iscs1` y el mismo procedimiento se debe repetir luego en el nodo `iscs2`. Como se presentó en la Línea de Comandos 3.22 el nodo `iscs1` dispone de 2 discos de 500 GB para crear el RAID 1.

Si bien es posible utilizar todo el disco para el arreglo, es preferible realizar antes una partición de 490 GB, debido a que los fabricantes no siempre utilizan los mismos tamaños de disco y si fuera necesario remplazar uno de los discos su tamaño podría no coincidir.

Para realizar la partición las opciones a ingresar son `n` para crear una nueva partición, `p` para crear una partición primaria, el número de partición es `1`, el primer cilindro a utilizar es el `1`, para el último cilindro se puede indicar el valor en MB, por lo que se especifica el valor de 490000M, el tipo de partición se elige mediante la opción `t,` la partición es `1`, el código hexadecimal es `fd`, y para confirmar todo el procedimiento se ingresa la opción `w`. Este mismo procedimiento se repite para el segundo disco `/dev/sdc`.

3.4.7.2 Creación del RAID

Una vez realizada la partición en los dos discos el siguiente paso es crear el dispositivo RAID 1 con el nombre `md1` como se indica en la Línea de Comandos 3.25.

```
# mdadm --create --verbose /dev/md1 --level=1 --raid-
devices=2 /dev/sdb1 /dev/sdc1

##resultado del commando anterior

mdadm: Defaulting to version 1.2 metadata

mdadm: array /dev/md1 started.
```

Línea de Comandos 3.25. Creación del disco RAID 1

Con el comando presentado en la Línea de Comandos 3.26 se verifica que el dispositivo RAID 1 haya sido creado.

```
# mdadm --detail /dev/md1

/dev/md1:

        Version : 1.2

  Creation Time : Mon Dec 29 10:52:33 2014

     Raid Level : raid1

     Array Size : 478488704 (456.32 GiB 489.97 GB)

   Raid Devices : 2

  Total Devices : 2

    Persistence : Superblock is persistent



Number   Major   Minor   RaidDevice State

   0       8       17        0      active sync   /dev/sdb1

   1       8       33        1      active sync   /dev/sdc1
```

Línea de Comandos 3.26. Verificación del disco RAID

Los procedimientos indicados en esta Sección se repiten en el nodo `iscs2`.

Una vez completado estos procedimientos se dispone de dos servidores con un disco RAID 1 de 489.97 GB cada uno.

Estos discos RAID se usarán para crear los dispositivos DRBD.

3.4.8 CREACIÓN DEL DISPOSITIVO DRBD

En primer lugar se crearán los dispositivos DRBD y una vez que se hayan configurado correctamente y verificado su funcionamiento se agregará el dispositivo de bloque DRBD como un recurso del *cluster* de almacenamiento.

3.4.8.1 Diseño del dispositivo DRBD

Es necesario determinar algunos valores relacionados con el desempeño de escritura a nivel de discos y a nivel de red, con el fin de poder configurar correctamente los parámetros del servidor DRBD.

3.4.8.1.1 Determinación de la velocidad de sincronización

Las pruebas de escritura y lectura en disco se realizaron mediante el comando `dd`, como se indica en la Línea de Comandos 3.27, las pruebas consistieron en que `dd` escriba en el disco RAID 1 una determinada cantidad de *bytes* y al terminar presenta la velocidad con la que lo hizo

```
##Test 1
# dd if=/dev/zero of=/dev/md1 bs=1G count=1 oflag=direct
1073741824 bytes (1.1 GB) copied, 12.8812 s, 83.4 MB/s
##Test 2
# dd if=/dev/zero of=/dev/md1 bs=1GB oflag=direct count=2
2000000000 bytes (2.0 GB) copied, 24.1583 s, 82.8 MB/s
##Test 3
# dd if=/dev/zero of=/dev/md1 bs=2GB oflag=direct count=2
4000000000 bytes (4.0 GB) copied, 48.4291 s, 82.6 MB/s
```

Línea de Comandos 3.27. Test para determinar la velocidad de escritura en el disco

. Se realizó cada prueba 10 veces con valores de 1, 2 y 4 GB siendo el valor promedio igual a 82.93 MB/s.Para determinar el desempeño de la red se siguió el procedimiento que se indica en la Línea de Comandos 3.28, en el nodo `iscs1` se ejecuta el comando `ncat` para que escuche en el puerto 2222 y escriba el resultado en el dispositivo `/dev/null`.

```
##Test 1: 1GB
##Nodo iscsi1
# ncat -v -v -l p 2222 > /dev/null
##Nodo iscsi2
# dd if=/dev/zero bs=1M count=1000 | ncat 10.14.14.11 2222
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB) copied, 8.98664 s, 117 MB/s
##Test 2: 5GB
# dd if=/dev/zero bs=1M count=5000 | ncat 10.14.14.11 2222
5000+0 records in
5000+0 records out
5242880000 bytes (5.2 GB) copied, 44.7942 s, 114 MB/s
##Test 3: 10GB
# dd if=/dev/zero bs=1M count=10000 | ncat 10.14.14.11 2222
10000+0 records in
10000+0 records out
10485760000 bytes (10 GB) copied, 89.5857 s, 116 MB/s
```

Línea de Comandos 3.28. Test para determinar la velocidad de la red

Mientras tanto en el nodo `iscs2`, se ejecuta el comando `dd` para que envíe 1 GB al nodo `iscs1` a través del puerto 2222. Se repitió este proceso enviando 5 GB y 10 GB, obteniendo resultados similares para la velocidad de escritura de 1, 5 y 10 GB, velocidad que en promedio es igual a 115.6 MB/s.

De los resultados obtenidos en las pruebas de la Línea de Comandos 3.27 y de la Línea de Comandos 3.28 se tiene que el menor rendimiento es el de escritura en el disco.

Por regla general en DRBD se espera que la velocidad con la que los datos se sincronizan entre los nodos, sea igual a la tercera parte de la menor velocidad, es decir la tercera parte de 82.9 MB/s, por lo que se esperaría tener una velocidad de sincronización de aproximadamente 27.6 MB/s.

3.4.8.2 Instalación

Los paquetes de DRBD necesarios se encuentran en los repositorios de software[138] de ELRepo[139], una vez que se instala ese repositorio con el comando presentado en la Línea de Comandos 3.29, se procede a instalar DRBD con el comando presentado en la Línea de Comandos 3.30.

```
# rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
# yum install http://www.elrepo.org/elrepo-release-7.0-
2.el7.elrepo.noarch.rpm
```

Línea de Comandos 3.29. Agregar el repositorio para instalar DRBD

[138] Repositorio de software: Un repositorio es un servidor del que es posible descargar software para un sistema operativo Linux.

[139] ELRepo: *Community Enterprise* Linux *Repository*: es un repositorio para CentOS mantenido por la comunidad Linux.

```
# yum install drbd84-utils  kmod-drbd84
```

Línea de Comandos 3.30. Instalar el software DRBD

3.4.8.3 Configuración

DRBD se configura mediante el archivo `/etc/drbd.conf`, y también mediante archivos que se localizan en la carpeta `/etc/drbd.d/`. Los archivos de configuración deben ser los mismos en los nodos que vayan a formar el almacenamiento replicado, en este caso `iscs1` y `iscs2`.

```
include "drbd.d/global_common.conf";
include "drbd.d/*.res";
```

Archivo de Configuración 3.6. Contenido del archivo /etc/drbd.conf

En el archivo `/etc/drbd.conf` se especifica la carpeta que contiene el archivo de configuración común para todos los recursos DRBD o archivo de configuración global, y los archivos de configuración individuales para cada recurso, como se indica en el Archivo de Configuración 3.6.

El archivo `/etc/drbd.d/global_common.conf` es el archivo de configuración global, es decir las opciones que este archivo contenga afectarán a todos los discos DRBD que se encuentren en el servidor. El contenido de este archivo se presenta en el Archivo de configuración 3.7, e indica que se utilizará el protocolo de sincronización tipo C.

```
global { }
net { protocol C;}
```

Archivo de configuración 3.7. Contenido del archivo /etc/drbd.d/global_common.conf

En el archivo `/etc/drbd.d/drbd0.res` se configura de manera individual cada dispositivo DRBD o recurso DRBD. El contenido de este archivo contiene todas las opciones importantes del recurso DRBD, por lo que su contenido se presenta en el Archivo de configuración 3.8.

```
resource drbd0 { disk    /dev/md1;
device /dev/drbd0;
meta-disk internal;
on iscs1.almwa.local {address 10.14.14.11:7788;}
on iscs2.almwa.local {address 10.14.14.12:7788;}
syncer{ c-plan-ahead 20;
c-min-rate 1M;
c-max-rate 300M;
c-fill-target 2M;
verify-alg md5;}
net {   sndbuf-size 10M;
rcvbuf-size 10M;
ping-int 10;
ping-timeout 5;
connect-int 2;
timeout 10;
ko-count 5;
max-buffers 128k;
max-epoch-size 8192;}}
```

Archivo de configuración 3.8. Contenido del archivo /etc/drbd.d/drbd0.res

La primera línea corresponde al nombre del recurso, la siguiente asigna el nombre al dispositivo DRBD, la tercera línea determina en donde se almacenarán los metadatos, por defecto se guardan internamente, es decir en el dispositivo DRBD.

La sección `on` incluye información sobre los nodos que forman parte de DRBD, el nombre del *host*, la direccion IP y el puerto a utilizar para la sincronización, el disco que se utilizará en cada *host*, etc. La sección `syncer` contiene algunos parámetros relacionados con la sincronización de los datos, en la sección `net` se configuran parámetros de conectividad.

Las secciones `net` y `syncer` son las de mayor importancia para conseguir un desempeño alto en la sincronización del disco DRBD a través de la red.

Para verificar que el archivo de configuración esté correcto se ejecuta el comando presentado en la Línea de Comandos 3.31, si la configuración no tiene errores el comando presenta en pantalla el contenido del Archivo de configuración 3.8.

```
# drbdadm dump all
```

Línea de Comandos 3.31. Comando para verificar la configuración del dispositivo drbd0

3.4.8.4 Creación del dispositivo DRBD

Los comandos de la Línea de Comandos 3.32 se utilizan para crear y activar el dispositivo DRBD, según los parámetros especificados en el Archivo de configuración 3.8, estos comandos deben ejecutarse en los dos nodos.

```
# drbdadm --verbose create-md drbd0
# drbdadm --verbose up drbd0
```

Línea de Comandos 3.32. Comandos para crear el dispositivo drbd0

Para revisar el estado del dispositivo DRBD que se acaba de crear se utiliza el comando presentado en la Línea de Comandos 3.33, como puede verse el estado de ambos nodos DRBD es `Secondary` e `Inconsistent`, debido a que aún no se han sincronizado los dispositivos DRBD.

```
# watch –n 2 cat /proc/drbd
0: cs:Connected r0:Secondary/Secondary
ds:Inconsistent/Inconsistent
```

Línea de Comandos 3.33. Comando para monitorizar el estado del dispositivo drbd0

Para iniciar la sincronización se emplea el comando presentado en la Línea de Comandos 3.34, este comando debe ejecutarse solamente en el nodo DRBD que se eligió como primario, en este caso `iscs1`.

```
# drbdadm --verbose primary --force drbd0
```

Línea de Comandos 3.34. Comando para sincronizar los datos entre los nodos

Después de iniciar la sincronización entre los dispositivos DRBD, se vuelve a monitorizar su estado, como se indica en la Línea de Comandos 3.35.

```
# watch –n 2 cat /proc/drbd
0: cs:SyncTarget r0:Primary/Secondary ds:
UpToDate/Inconsistent
[=========>.........] sync'ed: 54.3% (26956/58880)M
finish: 0:14:27 speed: 31,792 (31,220) want: 26,240 K/s
```

Línea de Comandos 3.35. Sincronización de los nodos DRBD

El estado debe de los nodos cambia y uno de los nodos obtiene el rol `Primary` y sus datos estan en estado `UpToDate`, también se puede observar que la velocidad de sincronización es de 31.79 MB/s, ligeramente superior a la velocidad de 27.6 MB/s que se determinó en la Sección 3.4.8.1.1.

3.4.8.5 Agregar el dispositivo DRBD al cluster

Una vez creado el dispositivo de bloque DRBD y con los roles de los servidores DRBD definidos, el siguiente paso es encargar al software del *cluster* el manejo del dispositivo DRBD, es decir agregarlo como un recurso del *cluster*.

El procedimiento se indica en la Línea de Comandos 3.36. En primer lugar se crea un CIB temporal con el nombre `drbdtemp`, todas las configuraciones se harán en este archivo. El segundo comando crea el recurso, mientras que el tercero lo convierte en un recurso del tipo maestro-esclavo. El último comando remplaza el CIB del *cluster* con el CIB temporal y actualiza la configuración del *cluster*. El recurso DRBD es un recurso del tipo clon, es decir que se ejecuta en más de un nodo del *cluster*, pero también se trata de un recurso del tipo maestro/esclavo es decir que el recurso se ejecuta en más de un nodo pero está activo o con el rol maestro solamente en uno de los nodos.

```
# pcs cluster cib drbdtemp

# pcs -f drbdtemp resource create drbdwa ocf:linbit:drbd
drbd_resource=drbdwa

# pcs -f drbdtemp resource master masterdrbd drbdwa meta
master-max=1 master-node-max=1 clone-max=2 clone-node-max=1
notify=true

# pcs cluster cib-push drbdtemp
```

Línea de Comandos 3.36. Agregar al recurso drbdwa al cluster

El nuevo recurso se denomina `drbdwa` y puede verse su estado en la Línea de Comandos 3.37. El *cluster* ha asignado al nodo `iscs2` como el servidor DRBD maestro, y al nodo `iscs1` como servidor esclavo o secundario.

```
# pcs resource show
Master/Slave Set: masterdrbd [drbdwa]
Masters: [ iscs2 ]
Slaves: [ iscs1 ]
```

Línea de Comandos 3.37. Comando para ver el estado del recurso drbd_wa

Con el recurso DRBD configurado y funcionando el siguiente paso es crear el recurso iSCSI, configuración que se presenta a continuación.

3.4.9 CREACIÓN DEL TARGET ISCSI

3.4.9.1 Instalación del software targetcli

Para instalar el programa se ejecuta el comando presentado en la Línea de Comandos 3.38, en los dos nodos del *cluster*. El recurso iSCSI se creará con la información que se indica en la Tabla 3.6.

```
# yum install targetcli
```

Línea de Comandos 3.38. Comando para instalar el software del target iSCSI

Un recurso LUN iSCSI es un recurso del tipo agrupado y con restricción de orden, es decir que para ejecutarse la LUN iSCSI necesita un recurso tipo dirección IP y un recurso *target* iSCSI, ambos recursos deben ejecutarse en un orden determinado. Además los tres recursos, LUN iSCSI, *target* iSCSI, y la dirección IP tienen que ejecutarse en el nodo en el que el recurso DRBD tenga el rol *master*, es decir en el nodo que funciona como DRBD primario. Esto se consigue con una restricción de colocación.

3.4.9.2 Creación del target iSCSI

Parámetro	Valor
IQN	iqn.2015-01.local.almwa:iscswa
Dirección IP	10.14.14.100
Puerto	3260
Dispositivo de bloque	/dev/drbd0

Tabla 3.6. Propiedades del target iSCSI

En primer lugar se crea un recurso del tipo dirección IP, que es la dirección IP que usará el *target* iSCSI para crear el portal, como se indica en la Línea de Comandos 3.39.

```
# pcs resource create iscsiIP ocf:heartbeat:IPaddr2
ip=10.14.14.100 cidr_netmask=24 op monitor interval=10
```

Línea de Comandos 3.39. Comando para agregar la dirección IP como recurso del cluster

A continuación se crea el grupo `grupiSCSI` en el que se agruparán la IP, y posteriormente el *target* y la LUN, como se indica en la Línea de Comandos 3.40.

```
# pcs resource group add grupiSCSI iscsiIP
```

Línea de Comandos 3.40. Comando para agregar el recurso iscsiIP al grupo grupiSCSI

El siguiente paso, que se indica en la Línea de Comandos 3.41, es agregar la restricción de colocación que obliga al grupo `grupISCSI` a ejecutarse solamente en el nodo donde el recurso DRBD tenga el rol *master*, es decir en el nodo que sea el servidor DRBD primario. Si el nodo con el rol *master* falla, Pacemaker asigna el rol *master* al otro nodo DRBD y arranca los recursos del grupo `grupISCSI` en este nodo.

```
# pcs constraint colocation add grupiSCSI masterdrbd with-
rsc-role=Master
```

Línea de Comandos 3.41. Agregar una restricción de colocación al grupo de recursos grupiSCSI

Con esto ya es posible crear el recurso *target* iSCSI, como se indica en la Línea de Comandos 3.42, el parámetro `allowed_initiators` permite configurar la lista de iSCSI *initiator* o clientes iSCSI que pueden conectarse al *target* iSCSI.

```
# pcs resource create iscsTgt ocf:heartbeat:iSCSITarget
iqn=iqn.2015-01.local.almwa:iscsiwa
portals=10.14.14.100:3260 allowed_initiators="iqn.1994-
05.com.redhat:17d73505287 iqn.1994-05.com.redhat:b7f474e4dc2
iqn.1994-05.com.redhat:371737cf4719" --group grupiSCSI
```

Línea de Comandos 3.42. Comando para crear el recurso target iSCSI en el grupo grupiSCSI

A continuación se agrega el recurso LUN iSCSI como se indica en la Línea de Comandos 3.43.

El recurso se llama `iscsLUN` y usa al dispositivo `/dev/drbd0` como dispositivo de bloque.

```
# pcs resource create iscsLUN ocf:heartbeat:iSCSILogicalUnit
target_iqn=iqn.2015-01.local.almwa:iscsiwa lun=1
path=/dev/drbd0 --group grupiSCSI
```

Línea de Comandos 3.43. Comando para crear el recurso iscsLUN y agregarlo al grupo grupiSCSI

Por último se agrega la restricción de orden, para garantizar que los recursos del grupo `grupiSCS` arranquen luego que el recurso DRBD haya iniciado.

Este comando se presenta en la Línea de Comandos 3.44.

```
# pcs constraint order promote masterdrbd then start
grupiSCSI
```

Línea de Comandos 3.44. Comando para crear la restricción de orden para el grupo grupiSCSI y el recurso masterdrbd

Una vez realizadas estas configuraciones ya se cuenta con un *cluster* de almacenamiento del tipo activo/pasivo con alta disponibilidad, el servicio altamente disponible que este *cluster* brinda es un dispositivo de bloque DRBD de 450 GB al que es posible acceder mediante la tecnología iSCSI.

Este dispositivo de bloque se usará como almacenamiento compartido en el *cluster* de alta disponibilidad cuya implementación se indica a continuación.

3.5 IMPLEMENTACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD

Se sigue el mismo procedimiento que el presentado para la instalación del *cluster* de almacenamiento en lo relacionado a la instalación del sistema operativo, la configuración de seguridad, las interfaces de red, la instalación del software del *cluster* y el dispositivo de *fencing*.

La información que se modifica corresponde a los nombres de los nodos, las direcciones IP y el nombre del *cluster*, como se indica en la Tabla 3.7.

Nombre del cluster	clusterwa
Nombre de los nodos	nodo1, nodo2, nodo3
Direcciones IP	10.14.14.21, 10.14.14.22, 10.14.14.23

Tabla 3.7. Información del cluster clusterwa

Como se mencionó en la Sección 3.2.3 el *cluster* de alta disponibilidad tendrá una red adicional destinada al tráfico de las máquinas virtuales cuya configuración se indica a continuación.

3.5.1 CONFIGURACIÓN DE LA RED PARA LAS MÁQUINAS VIRTUALES

En primer lugar se crea el archivo de configuración para la interfaz enlazada `bond1` con el comando que se indica en la Línea de Comandos 3.45, el contenido de este archivo se presenta en el Archivo de Configuración 3.9.

```
# touch /etc/sysconfig/network-scripts/ifcfg-bond1
```

Línea de Comandos 3.45. Creación del archivo de configuración de la interfaz enlazada bond1

```
DEVICE=bond1

NAME=bond1

BOOTPROTO=none

ONBOOT=yes

BONDING_MASTER=yes

BONDING_OPTS="mode=active-backup miimon=100"

BRIDGE=br0
```

Archivo de Configuración 3.9. Contenido del archivo de configuración de la interfaz bond1

Para las interfaces físicas se repite el procedimiento que se indica en la Sección 3.4.4, con los cambios necesarios.

Para conectar las máquinas virtuales es necesario crear una interfaz del tipo puente o *bridge* que permita a los servidores virtuales conectarse y acceder a la red del instituto. La interfaz puente utilizará la interfaz `bond1` que se acaba de crear, con lo que la red de las máquinas virtuales tendrá alta disponibilidad.

Para la interfaz puente `br0` se crea el archivo de configuración como se indica en la Línea de Comandos 3.46, el contenido de este archivo se presenta en el Archivo de Configuración 3.10.

```
# touch /etc/sysconfig/network-scripts/ifcfg-br0
```

Línea de Comandos 3.46. Creación del archivo de configuración de la interfaz enlazada br0

```
DEVICE=br0

TYPE=Bridge

BOOTPROTO=none

ONBOOT=yes

DELAY=0

IPADDR=192.168.1.229

PREFIX=24

GATEWAY=192.168.1.250

DNS1=192.168.1.10

DNS2=192.168.1.11

IPV6INIT=no

USERCTL=no

NM_CONTROLLED=no
```

Archivo de Configuración 3.10. Contenido del archivo de configuración de la interfaz puente br0

Una vez terminada la configuración de las interfaces de red ya es posible empezar a configurar el *cluster* de alta disponibilidad.

3.5.2 CONFIGURACIÓN DEL CLUSTER DE ALTA DISPONIBILIDAD

Como se indicó en la Sección 3.4.6, para realizar la configuración del *cluster* es posible utilizar la consola de comandos `pcs` o la interfaz web, a continuación se mostrará brevemente la forma de acceder a la interfaz web.

3.5.2.1 Interfaz web de pcs

La URL del portal de configuración del *cluster* es `https://10.14.14.21:2224`, una vez aceptado el certificado correspondiente aparece la pantalla de ingreso que se indica en la Figura 3.6. El nombre de usuario y la clave son los que se especificaron en el procedimiento mostrado en la Línea de Comandos 3.10.

Una vez que se inicia la sesión, el portal direccionará a la sección de administración del *cluster* que se indica en la Figura 3.7. Mediante la interfaz web se tiene la posibilidad de administrar y configurar uno o más *clusters*. Para ello se puede hacer clic en `Add Existing` y se solicitará ingresar el nombre de uno de los nodos que forman parte del *cluster* que se desea administrar.

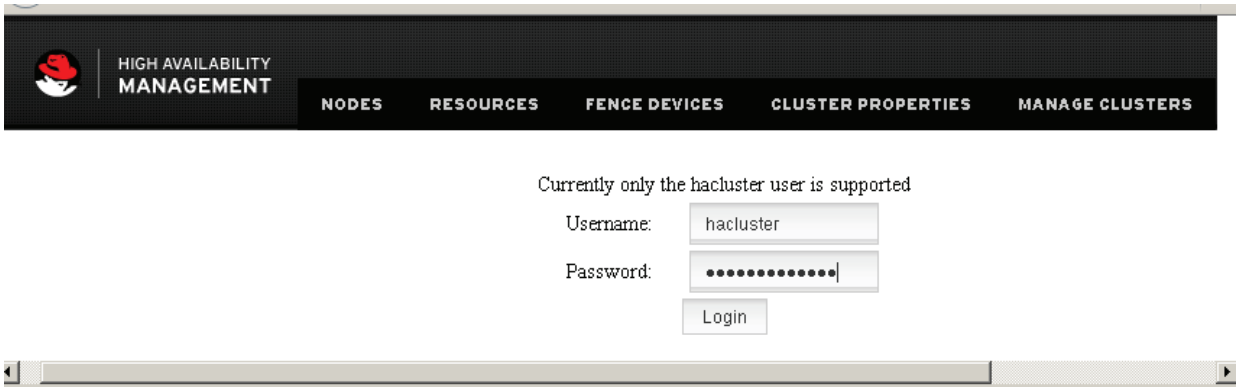

Figura 3.6. Ingresar al portal de configuración del cluster

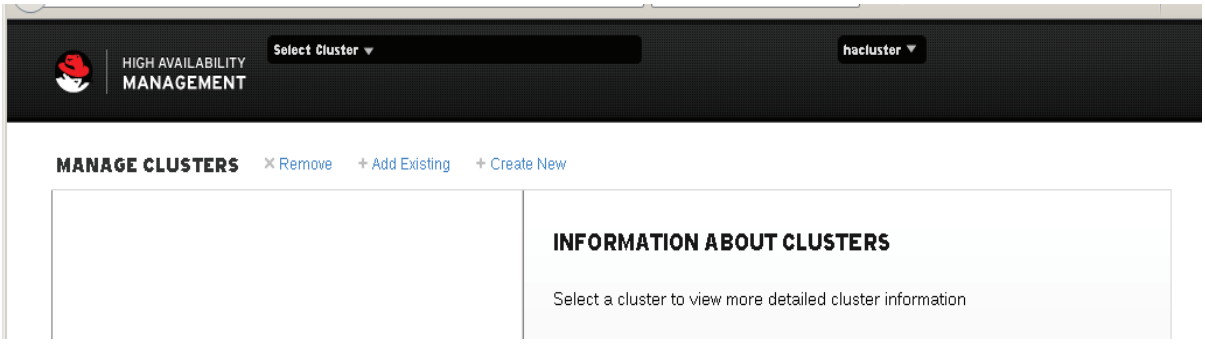

Figura 3.7. Sección de administración de cluster

Como se indica en la Figura 3.8, se ingresó el nombre de uno de los servidores `nodo1`, se hace clic en el botón `Add Existing` y el *cluster* `clusterwa` aparece

en la lista de *cluster* administrados en la parte derecha del portal web, como se indica en la Figura 3.9.

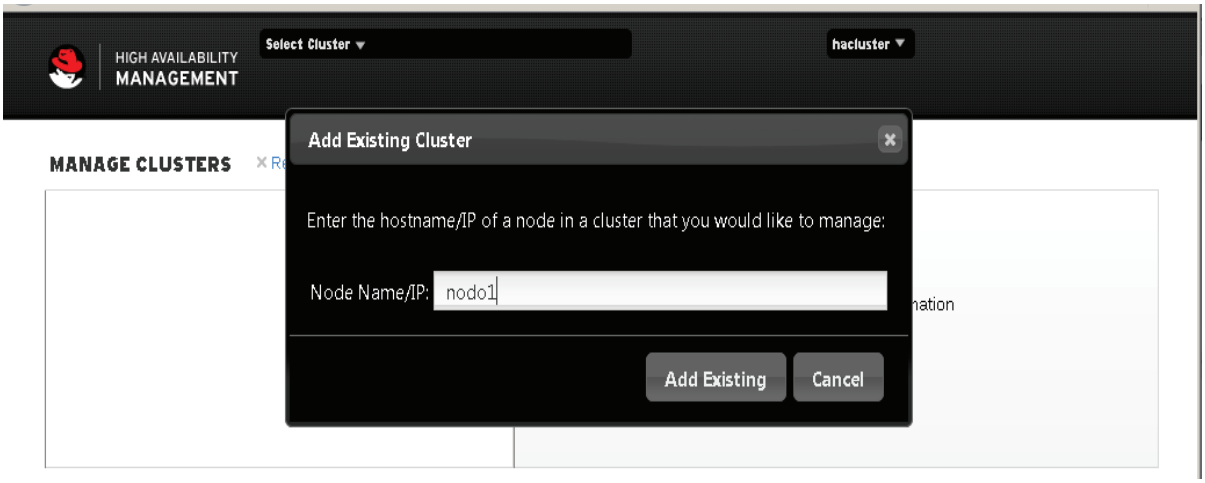

Figura 3.8. Agregar un cluster para administrar

Al hacer clic sobre el nombre del *cluster* `clusterwa` se presentan las opciones de configuración disponibles, tal como agregar un dispositivo de *fencing*, agregar nodos al *cluster*, agregar recursos, arrancar/detener nodos, entre otros, como puede verse en la Figura 3.10.

El resto de configuraciones del *cluster* se realizará mediante la línea de comandos, empleando el programa `pcs`.

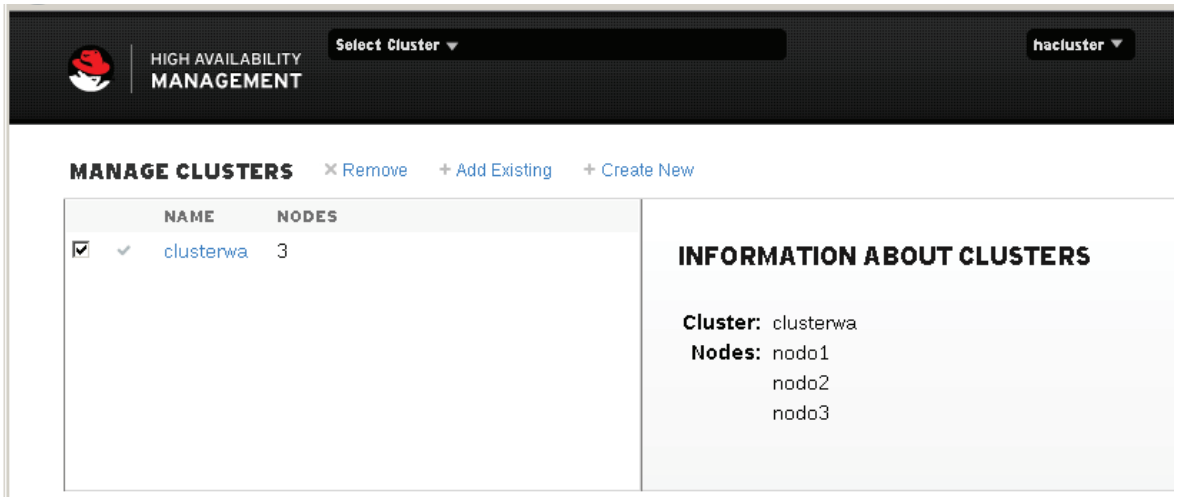

Figura 3.9. Cluster clusterwa agregado al portal de administración

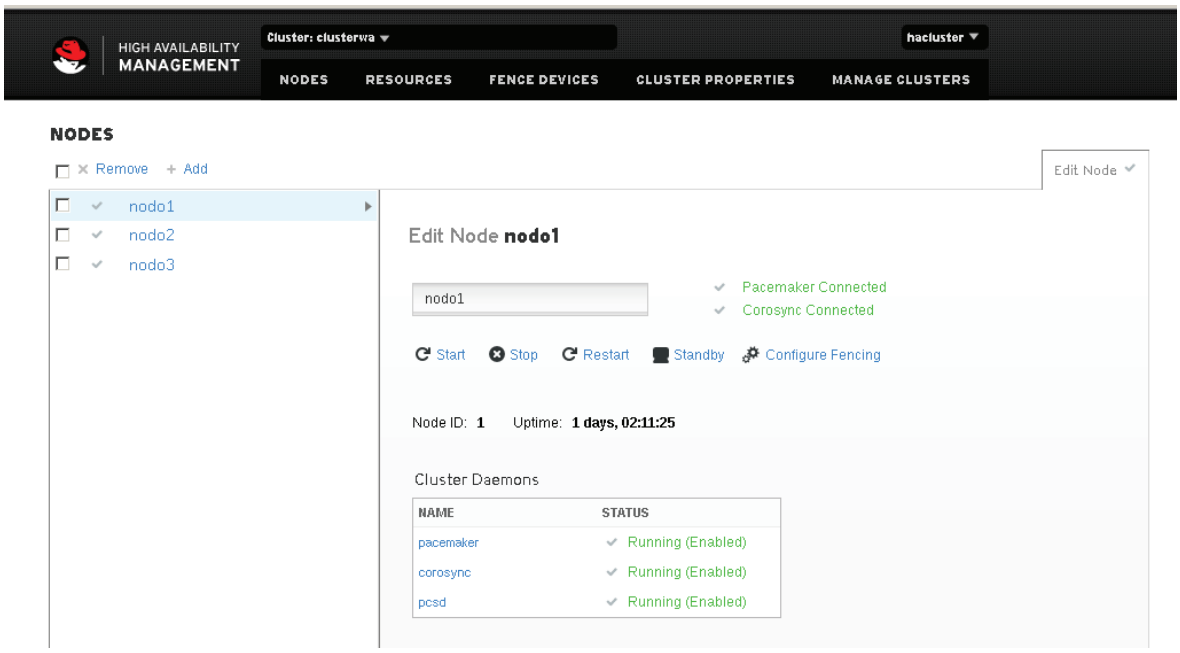

Figura 3.10. Sección de configuración del cluster

3.5.3 AGREGAR EL ALMACENAMIENTO COMPARTIDO COMO UN RECURSO [34]

El primer recurso que se agregará es la LUN iSCSI que se creó en la configuración del *cluster* de almacenamiento, primero es necesario instalar el software del iSCSI *initiator* con el comando que se presenta en la Línea de Comandos 3.47. Este comando se repite en los tres nodos que conforman el *cluster*.

```
# yum install iscsi-initiator-utils

# systemctl disable iscsid

# systemctl disable iscsi
```

Línea de Comandos 3.47. Comando para instalar el software del initiator iSCSI

El siguiente paso es crear el recurso LUN iSCSI, como se indica en la Línea de Comandos 3.48, para crear el recurso se necesita la dirección IP, puerto y el `iqn` del target iSCSI. Este recurso debe ejecutarse en los tres nodos del *cluster*, es decir que

se trata de un recurso del tipo clon y se lo configura con las opciones correspondientes a esta clase de recurso.

```
# pcs resource create iscsiwa ocf:heartbeat:iscsi
portal=10.14.14.100:3260 target=iqn.2015-
01.local.almwa:iscsiwa udev=no op monitor interval=30
timeout=10 on-fail=fence clone interleave=true ordered=true
```

Línea de Comandos 3.48. Comando para agregar la LUN iSCSI como un recurso

Si el comando es exitoso, en cada uno de los nodos aparecerá un nuevo disco de 450 GB. Antes de continuar es necesario instalar el software que el sistema de archivos GFS2 necesita.

3.5.3.1 Instalación de software para el sistema de archivos tipo cluster

Como se mencionó en la Sección 1.6.4.2, el sistema de archivos GFS2 necesita de programas de otros proyectos para funcionar correctamente, el programa GFS2 y sus requisitos se instalan con los comandos de la Línea de Comandos 3.49. Este comando debe ejecutarse en los tres nodos que forman el *cluster*.

```
# yum install gfs2-utils

# yum install dlm

# yum install lvm2-cluster
```

Línea de Comandos 3.49. Instalación del software necesario para GFS2

Una vez instalados el administrador de bloqueo DLM y la versión *cluster* del administrador de volúmenes LVM, es necesario crear dos recursos del tipo clon, un recurso para el demonio del administrador de bloqueo, al que se llamará `dlmwa` y otro para el demonio encargado de controlar un LVM del tipo *cluster*, al que se llamará `clvmwa`. En primer lugar se crea el recurso DLM, como se indica en la Línea de Comandos 3.50.

```
# pcs resource create dlmwa ocf:pacemaker:controld op
monitor interval=30s on-fail=fence clone interleave=true
ordered=true
```

Línea de Comandos 3.50. Comando para crear el recurso dlmwa

El siguiente paso, que se indica en la Línea de Comandos 3.51, es agregar las restricciones de orden y colocación, necesarias para que el recurso DLM inicie luego de que el recurso que controla la LUN iSCSI haya arrancado correctamente y para que ambos recursos arranquen en el mismo servidor.

```
# pcs constraint order  start iscsiwa-clone then dlmwa-clone

# pcs constraint colocation add dlmwa-clone with iscsiwa-
clone
```

Línea de Comandos 3.51. Comando para agregar las restricciones para el recurso dlmwa

El siguiente paso es configurar LVM para que funcione en modo *cluster*, con el comando que se indica en la Línea de Comandos 3.52. Este comando se ejecuta en los tres nodos que forman el *cluster*.

```
# lvmconf --enable-cluster
```

Línea de Comandos 3.52. Comando para habilitar el modo cluster de LVM

Una vez realizado ese cambio en la configuración de LVM, se crea un recurso encargado de controlar el demonio `clvmd`, como se indica en la Línea de Comandos 3.53

```
# pcs resource create clvmwa ocf:heartbeat:clvm op monitor
interval=30s on-fail=fence clone interleave=true
ordered=true
```

Línea de Comandos 3.53. Comando para crear el recurso clvmwa

Lo siguiente es agregar las restricciones de orden y colocación para que el recurso `clvmwa` inicie luego de que el recurso `dlmwa` haya arrancado, como se indica en la Línea de Comandos 3.54.

```
# pcs constraint order  start dlmwa-clone then start clvmwa-
clone

# pcs constraint colocation add clvmwa-clone with dlmwa-
clone
```

Línea de Comandos 3.54. Comando para agregar las restricciones para el recurso clvmwa

A continuación se crea una partición usando la LUN iSCSI, ejecutando el comando que se indica en la Línea de Comandos 3.55, solamente en uno de los nodos del *cluster*.

Para crear la partición se utilizan los valores que por defecto indica el comando, para que la nueva partición sea visible en el resto de nodos es necesario detener y arrancar el recurso `iscsiwa`, mediante lo indicado en la Línea de Comandos 3.56.

```
# fdisk /dev/sdb

Partition 1 of type Linux and of size 456.3 GiB is set
```

Línea de Comandos 3.55. Comando para crear una partición en la LUN iSCSI (/dev/sdb)

```
# pcs cluster disable iscsiwa

# pcs  cluster enable iscsiwa
```

Línea de Comandos 3.56. Comando para reiniciar el recurso iscsiwa

El siguiente paso es crear un volumen físico, con la partición que se acaba de crear, como se indica en la Línea de Comandos 3.57.

```
# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created
```

Línea de Comandos 3.57. Comando para crear el volumen físico empleando la LUN iSCSI

Debe ser posible ver el volumen físico que se acaba de crear en todos los nodos del *cluster*, como se presenta en la Línea de Comandos 3.58.

```
# pvs
  PV         VG            Fmt  Attr PSize   PFree
/dev/sdb1                lvm2 a--  456.30g 456.30g
```

Línea de Comandos 3.58. Comando para mostrar los volúmenes físicos disponibles

A continuación se crea un grupo de volúmenes utilizando el volumen físico que se acaba de crear, como se indica en la Línea de Comandos 3.59.

```
# vgcreate --clustered y vgiscsiwa /dev/sdb1
Clustered volume group "vgiscsiwa" successfully created
```

Línea de Comandos 3.59. Comando para crear el grupo de volúmenes vgiscsiwa

El siguiente paso es crear el volumen lógico, según lo indicado en la Línea de Comandos 3.60.

```
# lvcreate -l 116812 -n lviscsiwa vgiscsiwa
Logical volume "lviscsiwa" created
```

Línea de Comandos 3.60. Comando para crear el volumen lógico lviscsiwa

3.5.4 CREAR EL SISTEMA DE ARCHIVOS GFS2

Una vez creado el volumen lógico ya es posible dar formato a ese volumen con el sistema de archivos GFS2, con el comando que se indica en la Línea de Comandos 3.61.

```
# mkfs.gfs2 -j 3 -t clusterwa:iscsiwa
/dev/vgiscsiwa/lviscsiwa

##Resultado del comando:

Device:                    /dev/vgiscsiwa/lviscsiwa

Block size:                4096

Device size:               456.30 GB (119615488 blocks)

Filesystem size:           456.30 GB (119615486 blocks)

Journals:                  3

Resource groups:           1826

Locking protocol:          "lock_dlm"

Lock table:                "clusterwa:iscsiwa"

UUID:                      ba53927f-0345-4fbb-7046-
9455c71a031f
```

Línea de Comandos 3.61. Comando para dar formato al volumen lógico lviscsiwa con el sistema de archivos GFS2

Las opciones empleadas en el comando corresponden a las presentadas en la Sección 1.6.4.2.2, la opción `j` indica el número de registros de datos o *journals* que se crearan y que es igual al número de nodos del *cluster* de alta disponibilidad que es igual a tres, la opción `t` es para indicar el nombre de la tabla de bloqueo, el cual es `clusterwa:iscsiwa`, la última parte corresponde a la partición en la que se creará el sistema de archivos.

El encargado de administrar el montaje y desmontaje del volumen lógico con el sistema de archivos GFS2 será el software del *cluster*, para lo que se crea el recurso `gfs2wa`, como se indica en la Línea de Comandos 3.62.

```
# pcs resource create gfs2wa ocf:heartbeat:Filesystem
device="/dev/vgiscsiwa/lviscsiwa"
directory="/mnt/discosVirtuales/" fstype="gfs2"
"options=noatime" op monitor interval=10 on-fail=fence clone
interleave=true
```

Línea de Comandos 3.62. Comando para crear el recurso gfs2wa

Luego se agregan las restricciones necesarias para que el recurso `gfs2wa-clone` se inicie en el mismo nodo que el recurso `clvmwa-clone` y solo si el mismo ha arrancado correctamente.

```
# pcs constraint order start clvmwa-clone then gfs2wa-clone

# pcs constraint colocation add gfs2wa-clone with clvmwa-
clone
```

Línea de Comandos 3.63. Comandos para agregar las restricciones de orden y colocación del recurso gfs2wa

Es necesario configurar la propiedad `no-quorum-policy` del *cluster* con el valor `freeze`, ya que es un requisito para el correcto funcionamiento de GFS2. El comando usado se indica en la Línea de Comandos 3.64.

```
# pcs property set no-quorum-policy=freeze
```

Línea de Comandos 3.64. Comando para configurar la propiedad del cluster no-quorum-policy

Una vez creados estos recursos ya se dispone de un almacenamiento compartido al que todos los nodos del *cluster* de alta disponibilidad pueden acceder, y donde se almacenarán las máquinas virtuales de los sistemas de adquisición y procesamiento.

En la siguiente sección se presentan los pasos para instalar y configurar el software de virtualización KVM.

3.5.5 INSTALACIÓN DE LA PLATAFORMA DE VIRTUALIZACIÓN KVM

Todos los nodos del *cluster* deben soportar la virtualización asistida por hardware que KVM necesita, para verificarlo se ejecuta el comando presentado en la Línea de Comandos 3.65, como puede verse el CPU soporta la bandera `vmx`.

```
# cat /proc/cpuinfo | grep vmx
flags           : ... monitor ds_cpl vmx smx est cx16
```

Línea de Comandos 3.65. Verificar que los nodos soportan la virtualización asistida por hardware

A continuación es necesario verificar que el módulo de KVM esté instalado en el kernel de Linux, con el comando presentado en la Línea de Comandos 3.66. Debe mencionarse que las últimas versiones de Linux incluyen este módulo por defecto.

```
# lsmod | grep kvm
kvm_intel             138567  0
kvm                   441119  1 kvm_intel
```

Línea de Comandos 3.66. Verificar que el kernel soporta la virtualización con KVM

Como ambos requisitos se cumplen ya es posible instalar el software de virtualización básico, con el comando presentado en la Línea de Comandos 3.67.

```
# yum install  qemu-kvm libvirt virt-install
```

Línea de Comandos 3.67. Instalar el software de virtualización necesario

Al instalar el programa `virt-install` se instala automáticamente el programa `virsh`. Una vez terminada la instalación es necesario arrancar el demonio `libvirtd` que actúa como intermediario entre el administrador de las máquinas virtuales y el hipervisor.

3.5.6 CONFIGURACIÓN DE LA MIGRACIÓN EN CALIENTE

Para que los nodos del *cluster* sean capaces de realizar la migración de las máquinas virtuales es necesario compartir las SSH *keys*[140] con el procedimiento que se indica en la Línea de Comandos 3.68, este procedimiento se realiza en cada uno de los nodos del *cluster*.

```
##Generar el archivo llave
# ssh-keygen
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
## Copiar el archivo llave al resto de nodos del cluster
# ssh-copy-id -i /root/.ssh/id_rsa.pub root@nodo3
# ssh-copy-id -i /root/.ssh/id_rsa.pub root@nodo1
```

Línea de Comandos 3.68. Comandos para compartir las llaves SSH

3.5.7 CREACIÓN DE LAS MÁQUINAS VIRTUALES

En esta sección se presenta la instalación y configuración de las máquinas virtuales, para que funcionen como nodos en los que el *cluster* pueda ejecutar recursos. En estos servidores se ejecutarán los sistemas de adquisición y procesamiento, se mostrará el proceso para el servidor virtual en el que se ejecutará el sistema

[140] SSH *Keys*: son archivos que el protocolo SSH utiliza para crear una conexión sin requerir de una clave de acceso.

SeisComP3, ya que para el resto de los servidores el procedimiento es similar y solamente es necesario realizar modificaciones en ciertos parámetros.

Dado que estos servidores virtuales serán a la vez recursos del tipo máquina virtual y nodos remotos del *cluster* de alta disponibilidad Pacemaker necesita referirse a ellos con nombres distintos según su función, debido a esto los servidores virtuales como recursos tipo máquina virtual se identificarán como: `vmseisc`, `vmearth` y `vmshake`, mientras que los nodos remotos del *cluster* se identificarán como: `seisc`, `earth` y `shake`.

En el resto del documento para evitar confusiones se empleará `seisc`, `earth` y `shake` para hacer referencia tanto al recurso como al nodo remoto.

3.5.7.1 Servidor virtual seisc

La primera máquina virtual que se creará será el servidor `seisc` en el que se instalará el sistema de adquisición y procesamiento SeisComP3, de acuerdo a las características que se determinaron en la Sección 2.4.1 y que se especifican mediante los parámetros del comando `virt-install`, que se presenta en la Línea de Comandos 3.69. Los parámetros de `virt-install` fueron explicados en la Sección 1.8.4.1.

```
##Nodo1

# virt-install --name seisc --ram 3072 --vcpus=2 --disk
path=/mnt/discosVirtuales/seisc1.img,size=30,bus=virtio --
network bridge=br0 --graphics vnc,listen=0.0.0.0,port=5911 -
-cdrom /mnt/discosVirtuales/ISOS/CentOS-7-x86_64-DVD-1503-
01.iso

Starting install...

Creating storage file seisc1.img
|  30 30 GB  00:00:00
```

Línea de Comandos 3.69. Comando para crear el servidor virtual seisc

Este comando puede ejecutarse en cualquiera de los nodos, se ha elegido el `nodo1`, una vez ejecutado , el proceso de instalación de la máquina virtual iniciará, pero para continuar con la instalación se necesita conectarse al servidor VNC[141] del `nodo1`, para ello en la opción `graphics` del comando `virt-install` se ha indicado que se usará el protocolo VNC, que se aceptarán conexiones desde cualquier origen y que el puerto a utilizar será el 5911.

La conexión al servidor VNC se realiza mediante un cliente VNC[142], como se presenta en la Figura 3.11.

La información con la que se configurará el sistema operativo del servidor virtual `seisc` se indica en la Tabla 3.8. Es necesario agregar el nombre del servidor virtual y la dirección IP en los archivos `/etc/hosts` de los nodos del *cluster*, como se indica en la Línea de Comandos 3.70.

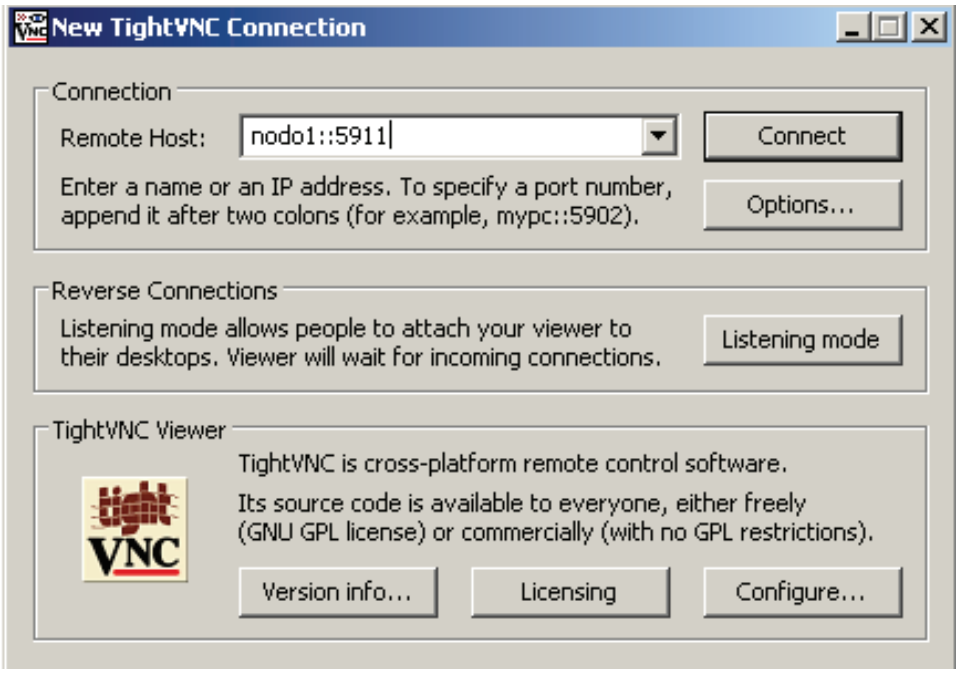

Figura 3.11 Conexión al servidor VNC del nodo1

[141] Servidor VNC: es el programa que permite que otros computadores controlen al servidor de forma remota.

[142] Cliente VNC: Se lo denomina también visor y es el programa que permite tomar control del servidor VNC

Parámetro	Valor
Nombre del computador	seisc
Dirección IP	192.168.1.35
Usuario	seiscomp

Tabla 3.8. Propiedades del servidor virtual seisc

```
# echo "192.168.1.35 seisc" >> /etc/hosts
```

Línea de Comandos 3.70. Comando para agregar la dirección IP del servidor virtual seisc a la lista de /etc/hosts

Una vez terminada la instalación del servidor virtual `seisc`, es necesario crear el archivo XML que define al servidor virtual y que el software de recursos usará para administrar la máquina virtual, este archivo se crea mediante el comando de la Línea de Comandos 3.71.

```
##nodo1
# virsh dumpxml seisc > /mnt/discosVirtuales/xml/seisc.xml
```

Línea de Comandos 3.71. Comando para crear el servidor virtual seisc

A continuación se crean los dos discos que el sistema utilizará para almacenar los datos adquiridos y la información procesada, con los comandos que se presentan en la Línea de Comandos 3.72.

```
# qemu-img create -f raw /mnt/discosVirtuales/seisc2.img 64G
# qemu-img create -f raw /mnt/discosVirtuales/seisc3.img 50G
```

Línea de Comandos 3.72. Comando para crear los discos virtuales seisc2 y seisc3

Estos discos se conectan al servidor virtual `seisc` agregando el contenido que se indica en el Archivo de Configuración 3.11 al archivo `/mnt/discosVirtuales/xml/seisc.xml`.

3.5.7.1.1 Instalación y configuración del sistema de adquisición y procesamiento SeisComP3

Los pasos de esta sección se realizan de acuerdo a lo indicado en el Anexo 1. Una vez instalado y configurado el sistema SeisComP3 en el servidor virtual `seisc` el siguiente paso es agregar el mismo como un recurso que el *cluster* administrará, como se indica en la siguiente sección.

```
<disk type='file' device='disk'>
      <driver name='qemu' type='raw'/>
      <source file='/mnt/discosVirtuales/seisc2.img'/>
      <target dev='vdb' bus='virtio'/>
</disk>
<disk type='file' device='disk'>
      <driver name='qemu' type='raw'/>
      <source file='/mnt/discosVirtuales/seisc3.img'/>
      <target dev='vdb' bus='virtio'/>
</disk>
```

Archivo de Configuración 3.12. Contenido para agregar dos discos al servidor virtual seisc

3.5.7.1.2 Configuración del servidor SeisComP3 como un servidor Pacemaker remoto

El servidor virtual `seisc` se agregará como un recurso de tipo máquina virtual, pero que actuará como un nodo remoto, para ello es necesario ejecutar en uno de los servidores físicos los comandos que se presentan en la Línea de Comandos 3.73, este comando crea un archivo de autenticación que permitirá al programa

`pacemaker_remote` que se ejecuta en la máquina virtual autenticarse con el *cluster* de alta disponibilidad.

```
# mkdir /etc/pacemaker

# dd if=/dev/urandom of=/etc/pacemaker/authkey bs=4096
count=1

# scp -pr /etc/pacemaker nodo2:/etc/

# scp –pr /etc/pacemaker nodo3:/etc/

# scp –pr /etc/pacemaker root@seisc:/etc/
```

Línea de Comandos 3.73. Generación y copia de la clave de autenticación

En el servidor virtual se instala el software `pacemaker-remote`, se agrega el mismo como un servicio del sistema y se abre el puerto 3121 del firewall, con los comandos presentados en la Línea de Comandos 3.74. A fin de evitar problemas con los módulos de SeisComP3 se deshabilitará el sistema SELinux como se presentó en el Archivo de configuración 3.1.

```
# yum install pacemaker-remote resource-agents

# systemctl start pacemaker_remote.service

# systemctl enable pacemaker_remote.service

# firewall-cmd --add-port 3121/tcp –permanent
```

Línea de Comandos 3.74. Instalación del software de administración de Pacemaker remoto

Una vez instalado el software se agrega la máquina virtual como un recurso del *cluster* con el comando que se indica en la Línea de Comandos 3.75.

En el parámetro `remote-node` se utiliza el nombre del servidor virtual, es importante no utilizar el nombre del recurso tipo máquina virtual ya que eso confundiría a Pacemaker y generaría errores al arrancar el recurso.

El parámetro `remote-connect-timeout` es muy importante ya que determina cuanto tiempo debe esperar Pacemaker hasta que la máquina virtual haya arrancado y pueda empezar a formar parte del *cluster*, por defecto utiliza un valor de 120 segundos, pero para evitar errores en caso que el servidor virtual se demore en arrancar se ha duplicado este valor.

Se agrega también una restricción de orden para que la máquina virtual arranque solamente si el recurso que controla el sistema de archivos GFS2 ya se está ejecutando.

```
# pcs resource create vmseisc VirtualDomain
hypervisor="qemu:///system"
config="/mnt/discosVirtuales/xml/seisc.xml" force_stop=true
migration_transport=ssh meta remote-node=seisc remote-
addr=192.168.1.35 remote-connect-timeout=240

# pcs constraint order start gfs2wa-clone then start vmseisc
```

Línea de Comandos 3.75. Instalación del software de administración de Pacemaker remoto

Una vez que el *cluster* reconozca a la máquina virtual como un nodo remoto, ya es posible empezar a agregar los componentes del sistema SeisComP3 como recursos administrados por Pacemaker, este procedimiento se revisa en la Sección 3.5.8

3.5.7.2 Servidor virtual earth

En la Línea de Comandos 3.76 se presenta el comando utilizado para crear el servidor virtual `earth` que ejecutará el sistema de adquisición y procesamiento Earthworm de acuerdo a las características que se indicaron en la Sección 2.4.2.

El procedimiento que se sigue es similar al realizado para el servidor virtual `seisc`.

```
# virt-install --name earth --ram 6144 --vcpus=3 --disk
path=/mnt/discosVirtuales/earth.img,size=30,bus=virtio --
disk path=/mnt/discosVirtuales/earth1.img,size=30,bus=virtio
--network bridge=br0 --graphics vnc,listen=0.0.0.0,port=5912
--cdrom /mnt/discosVirtuales/ISOS/CentOS-7.0-1406-x86_64-
Minimal.iso

Starting install...

Creating storage file earth.img               |  30 GB
00:00:00

Creating storage file earth1.img              |  30 GB
00:00:00
```

Línea de Comandos 3.76. Creación del servidor virtual Earthworm

Una vez terminada la instalación del servidor se procede a la configuración del sistema de adquisición tal como se indica en el Anexo 2.

Para configurar el servidor virtual como un nodo remoto del *cluster* se sigue el mismo procedimiento que el indicado en la Línea de Comandos 3.73 y la Línea de Comandos 3.74, una vez realizado esos procedimientos ya es posible crear el recurso `vmearth` que permitirá a Pacemaker administrar la máquina virtual, el recurso se crea con el comando que se indica en la Línea de Comandos 3.76.

```
pcs resource create vmearth VirtualDomain
hypervisor="qemu:///system"
config="/mnt/discosVirtuales/xml/earth.xml" force_stop=true
migration_transport=ssh meta allow-migrate=true  remote-
node=earth remote-addr=192.168.1.36 remote-port=3121
remote-connect-timeout=240
```

Línea de Comandos 3.77. Creación del servidor recurso vmearth

3.5.7.3 Servidor virtual shake

Con el comando presentado en la Línea de Comandos 3.78 se crea la máquina virtual para el sistema de generación de mapas ShakeMap.

```
# virt-install --name shake --ram 2048 --vcpus=2 --disk
path=/mnt/discosVirtuales/shake.img,size=30,bus=virtio --
network bridge=br0 --graphics vnc,listen=0.0.0.0,port=5913 -
-cdrom /home/ISO/Centos7.DVD.iso

Starting install...
Creating storage file shake.img            |  36 GB  00:00:00
```

Línea de Comandos 3.78. Creación del servidor virtual shake

Para agregar el servidor virtual como un nodo remoto se sigue el procedimiento indicado en la Línea de Comandos 3.73 y la Línea de Comandos 3.74.

Una vez terminada la instalación y configuración del servidor virtual `shake`, se instala el sistema ShakeMap, de acuerdo a lo indicado en el Anexo 3.

Como ser verá en la siguiente sección, el servidor virtual `shake` se utilizará como servidor de respaldo para los sistemas que se ejecutan en los servidores `seisc` y `earth`, por lo que es necesario instalar en este servidor los sistemas de adquisición SeisComP3 y Earthworm, de acuerdo a los Anexos 1 y 2, respectivamente.

3.5.8 INSTALACIÓN DE LOS SISTEMAS DE ADQUISICIÓN COMO RECURSOS CON ALTA DISPONIBILIDAD

A fin de garantizar alta disponibilidad para los sistemas de adquisición y procesamiento que se ejecutan en los servidores virtuales, Pacemaker debe ser capaz de ejecutar estos recursos en un servidor virtual de respaldo, para lo que se empleará el servidor virtual `shake`, es decir que si por ejemplo la máquina virtual `seisc` falla, Pacemaker detectará esta falla y arrancará los módulos de SeisComP3 en el servidor virtual `shake`, con una interrupción en los servicios mínima.

3.5.8.1.1 Configuración de los Módulos de SeisComP3 como recursos del cluster con alta disponibilidad

El primer recurso que se creará será un recurso tipo dirección IP al que se denominará `ipsc3`, a este recurso se añadirán los módulos de SeisComP3 formando un grupo que se llamará `seiscomp`.

La creación del recurso IP, las restricciones necesarias y el grupo se presentan en la Línea de Comandos 3.79, como puede verse se configura el recurso `ipsc3` y los recursos agrupados con el mismo, de tal forma que se ejecute en los servidores virtuales `seisc` y `shake`, se establece una mayor preferencia para el servidor virtual `seisc`, y también se agregan restricciones para que el recurso evite ejecutarse en los nodos físicos del *cluster*.

```
# pcs resource create ipsc3 IPaddr2 ip=192.168.1.93
cidr_netmask=32

# pcs constraint location ipsc3 prefers seisc

# pcs constraint location ipsc3 prefers shake=200

# pcs constraint location ipsc3 avoids nodo1

# pcs constraint location ipsc3 avoids nodo2

# pcs constraint location ipsc3 avoids nodo3

# pcs resource group add seiscomp ipsc3
```

Línea de Comandos 3.79. Creación y configuración del recurso ipsc3

El siguiente paso es configurar los módulos de SeisComP3 que se indicaron en la Sección 2.4.1.2.

En la Línea de Comandos 3.80 se presenta la configuración del módulo `spread` y el comando usado para agregar `spread` al grupo `seiscomp`.

```
# pcs resource create spread ocf:heartbeat:sc3
binfile="/home/seiscomp/seiscomp3/bin/run_with_lock"
user=seiscomp
cmdline_options="/home/seiscomp/seiscomp3/var/run/spread.pid
/home/seiscomp/seiscomp3/sbin/spread -n localhost -c
/home/seiscomp/seiscomp3/var/lib/spread/spread.conf &"
pidfile="/home/seiscomp/seiscomp3/var/run/spread.pid”

# pcs resource group add seiscomp spread --after ipsc3
```

Línea de Comandos 3.80. Creación del recurso spread

El siguiente módulo que se configura es `scmaster`, lo cual se indica en la Línea de Comandos 3.81.

```
# pcs resource create scmaster ocf:heartbeat:anything
binfile=/home/seiscomp/seiscomp3/sbin/scmaster
cmdline_options="-D -l
/home/seiscomp/seiscomp3/var/run/scmaster.pid -H
localhost:4803"
pidfile=/home/seiscomp/seiscomp3/var/run/scmaster.pid
user=seiscomp

# pcs resource group add seiscomp scmaster --after spread
```

Línea de Comandos 3.81. Creación del recurso scmaster

Para el resto de módulos se sigue el mismo procedimiento siendo necesario modificar solamente el nombre del módulo y las opciones de ejecución.

3.5.8.1.2 Configuración de los Módulos de Earthworm como recursos del cluster con alta disponibilidad

El procedimiento seguido para configurar los módulos del sistema Earthworm como recursos del *cluster* con alta disponibilidad son similares a los seguidos en la sección anterior y se presentan en la Línea de Comandos 3.82.

En primer lugar se crea un recurso IP al que se denomina `ipew`, a este recurso IP se le agregan restricciones de localización para que prefiera ejecutarse en el servidor virtual `earth`, pero en caso de ser necesario que pueda ejecutarse también en el servidor `shake`. El grupo que formarán los módulos de Earthworm se denomina `earthworm`.

```
# pcs resource create ipew IPaddr2 ip=192.168.1.94
cidr_netmask=32

# pcs resource enable ipew
# pcs constraint location ipew prefers earth1
# pcs constraint location ipew prefers shake=200
# pcs constraint location ipew avoids nodo1
# pcs constraint location ipew avoids nodo2
# pcs constraint location ipew avoids nodo3
# pcs resource group add earthworm ipew
```

Línea de Comandos 3.82. Creación y configuración del recurso ipew

3.5.8.1.3 Configuración de los Módulos de ShakeMap como recursos del cluster

Como se indicó en la Sección 2.4.3.1, los módulos de ShakeMap no se ejecutan como servicios, por lo que no es posible que se configuren como recursos que Pacemaker pueda administrar, por lo que se configurarán la base de datos y el servidor Apache que el sistema utiliza, lo cual se indica en la Línea de Comandos 3.83.

Una vez terminada la configuración del *cluster*, el estado de los nodos es el que se presenta en la Figura 3.12, en la cual se aprecia que existen 6 nodos, 3 nodos físicos y 3 nodos remotos.

```
# pcs resource create mysql mysql
binary="/usr/libexec/mysqld" meta target-role=stopped

# pcs constraint location mysql prefers shake

# pcs constraint order start vmshake then start mysql

# pcs resource create web apache
configfile=/etc/httpd/conf/httpd.conf meta target-
role=stopped

# pcs constraint location web prefers shake

# pcs constraint order start vmshake then start web
```

Línea de Comandos 3.83. Creación y configuración de los recursos del sistema ShakeMap

```
Every 2.0s: pcs status                Fri Sep 25 12:13:22 2015
Cluster name: clusterwa
Last updated: Fri Sep 25 12:13:23 2015          Last change:
Fri Sep 25 12:12:55 2015
by hacluster via crmd on nodo1
Stack: corosync
Current DC: nodo2 (version 1.1.12-a14efad) - partition with
quorum
6 nodes and 45 resources configured

Online: [ nodo1 nodo2 nodo3 ]
GuestOnline: [ earth@nodo3 seisc@nodo1 shake@nodo3 ]
```

Figura 3.12 Estado de los nodos físicos y nodos remotos del cluster

En la Figura 3.13 se indica el estado de los recursos que el *cluster* está ejecutando.

```
vmshake        (ocf::heartbeat:VirtualDomain): Started nodo3
 vmseisc       (ocf::heartbeat:VirtualDomain): Started nodo1
 Resource Group: seiscomp
     ipsc3     (ocf::heartbeat:IPaddr2):       Started seisc
     spread    (ocf::heartbeat:sc3):   Started seisc
     scmaster  (ocf::heartbeat:anything):      Started seisc
     seedlink  (ocf::heartbeat:sc3wa): Started seisc
     slarchive (ocf::heartbeat:sc3wa): Started seisc
     arclink   (ocf::heartbeat:sc3wa): Started seisc
     scautopick (ocf::heartbeat:sc3wa): Started seisc
     scautoloc (ocf::heartbeat:sc3wa): Started seisc
 mysql  (ocf::heartbeat:mysql): Started shake
 web    (ocf::heartbeat:apache):        Started shake
 vmearth       (ocf::heartbeat:VirtualDomain): Started nodo3
 Resource Group: earthworm
     ipew      (ocf::heartbeat:IPaddr2):       Started earth1
     startstop (ocf::heartbeat:sc3):   Started earth1
     slink2ew  (ocf::heartbeat:sc3):   Started earth1
     wave_serverV      (ocf::heartbeat:sc3):   Started earth1
```

Figura 3.13 Estado de los recursos que el cluster ejecuta

3.6 PRUEBAS

A continuación se presentan las pruebas realizadas para revisar el correcto funcionamiento de los componentes del *cluster* de alta disponibilidad.

3.6.1 PRUEBA DEL ALMACENAMIENTO COMPARTIDO

3.6.1.1 Prueba del disco RAID 1 implementado por software

Se recomiendan dos procedimientos para comprobar el funcionamiento de un disco RAID implementado por software, uno mediante hardware y otro empleando el comando `mdadm`, en ningún caso se recomienda desconectar el disco en caliente para simular un fallo de disco [74], por lo que se decide utilizar el comando `mdadm` para realizar las pruebas.

Para mostrar que la redundancia del disco RAID 1 funciona se monitoriza el estado del dispositivo DRBD `/dev/drbd1`, que utiliza el disco `/dev/md1` como dispositivo

de bloque. En primer lugar se ejecuta el comando presentado en la Línea de Comandos 3.84 para simular un fallo en uno de las particiones que forman el RAID 1, se simulará el fallo en la partición `/dev/sdc1`.

```
# mdadm --manage --set-faulty /dev/md1 /dev/sdc1

mdadm: set /dev/sdc1 faulty in /dev/md1
```

Línea de Comandos 3.84. Comando para simular el fallo en un disco del arreglo RAID1

Como puede verse en la Línea de Comandos 3.85 una vez ejecutado el comando de la Línea de Comandos 3.84, el servidor detecta un fallo en uno de las particiones físicas que forman el disco RAID 1, pero el disco `/dev/md1` continua funcionando sobre una sola partición.

Así también el funcionamiento del disco DRBD no se ha visto afectado, lo cual se indica en la Línea de Comandos 3.86, en la que se puede ver que el disco se encuentra en estado `UpToDate`.

```
# tail -1000 /var/log/messages | grep raid

Mar  9 08:38:04 iscs2 kernel: md/raid1:md1: Disk failure on
sdc1, disabling device.

md/raid1:md1: Operation continuing on 1 devices.
```

Línea de Comandos 3.85. Estado del disco RAID1 detectado por el sistema operativo

```
# cat /proc/drbd

version: 8.4.5 (api:1/proto:86-101)

1: cs:Connected ro: Primary/Secondary ds:UpToDate/UpToDate C
```

Línea de Comandos 3.86. Estado del dispositivo DRBD

Mediante el comando presentado en la Línea de Comandos 3.87 se conoce el estado del disco RAID 1, luego de simular el fallo, el sistema detecta un disco fallido y que solo un disco está funcionando.

```
# mdadm --detail /dev/md1

/dev/md1:

Raid Level : raid1

Raid Devices : 2

Total Devices : 2

State : clean, degraded

Working Devices : 1

Failed Devices : 1

Name : iscs2.almwa.local:1

Number   Major   Minor   RaidDevice State

0        8       17        0      active sync   /dev/sdb1

1        0        0        1      removed

1        8       33        -      faulty   /dev/sdc1
```

Línea de Comandos 3.87. Estado del disco RAID 1

Para volver a agregar el disco fallido es necesario removerlo del arreglo y volver a agregarlo, con lo que el proceso de sincronización empieza. Estos comandos se indican en la Línea de Comandos 3.88.

```
# mdadm /dev/md1 -r /dev/sdc1

mdadm: hot removed /dev/sdc1 from /dev/md1

# mdadm /dev/md1 -a /dev/sdc1
```

Línea de Comandos 3.88. Remover y agregar el disco /dev/sdc1 al RAID 1

3.6.1.2 Pruebas del dispositivo DRBD y del target iSCSI

Existen dos formas de realizar esta prueba, de forma ordenada o simular una situación crítica. En ambos casos los servidores virtuales no experimentan tiempos de caída, pero el proceso de recuperación es diferente. En la forma ordenada se apaga el nodo que tenga el rol DRBD primario mediante el comando `shutdown`, con lo que el software del *cluster* tiene tiempo de asignar al segundo nodo como DRBD primario y activar el resto de recursos en el segundo nodo. Este escenario puede servir si lo que se desea es darle mantenimiento a uno de los nodos y una vez terminado, se enciende nuevamente el nodo y sin intervención del administrador el nodo vuelve a formar parte del *cluster* de almacenamiento. La segunda forma es desconectar la fuente de alimentación del nodo con que tiene el rol DRBD maestro, este escenario simula un fallo crítica del nodo y es la prueba que se presenta a continuación. Antes de realizar esta prueba es necesario que el sistema de *fencing* esté funcionando correctamente, ya que de lo contrario el *cluster* no será capaz de realizar la migración de los recursos.

Para verificar la disponibilidad de los servidores virtuales una computadora portátil ejecutará el comando ping de forma constante al servidor `earth` con dirección IP 192.168.1.36, mientras que en uno de los nodos del *cluster* de alta disponibilidad y en otro nodo del *cluster* de almacenamiento se monitoriza los logs del sistema. En cuanto se desconecta la fuente del servidor primario `iscs1`, Pacemaker detecta el fallo e inicia el proceso para promover al nodo `iscs2` de servidor DRBD secundario a primario. El módulo *Policy Engine* (`pengine`) decide la acción a ejecutar y delega al administrador de recursos local (`crmd`) para que asigne el rol DRBD primario al nodo `iscs2`, una vez completado este procedimiento Pacemaker levanta el recurso iSCSI, tal como puede verse en la Figura 3.14. Una vez terminadas estas operaciones el nuevo estado del *cluster* se indica en la Figura 3.15, en la cual se observa que el nodo `iscs1` está fuera de línea y todos los recursos que se ejecutaban en el nodo `iscs1` están ahora en el nodo `iscs2`, así mismo el recurso

de tipo maestro/esclavo `drbdwa`, está detenido en el nodo `iscs1`, y se está ejecutando en el nodo `iscs2` con el rol maestro.

```
iscs2 pengine[2021]:notice LogActions: Promote drbdwa:0 (Slave -> Master iscs2)

iscs2 crmd[2022]: notice: te_rsc_command: Initiating action 6: promote
drbdwa_promote_0 on iscs2 (local)

iscs2 kernel: block drbd1: role(Secondary -> Primary)

iscs2 iSCSILogicalUnit(iscsLUN)[26840]: INFO: Create block storage object
iscsLUN using /dev/drbd1
```

Figura 3.14. Detección del error y cambio de rol DRBD secundario a rol DRBD primario

Como se esperaba el cambio del recurso DRBD y LUN iSCSI del nodo fallido al nodo `iscs2` no afecta a las máquinas virtuales que el *cluster* de alta disponibilidad ejecuta.

```
Cluster name: clusteralmawa
Last updated: Tue Mar 10 12:36:49 2015
Last change: Tue Mar 10 12:36:49 2015 via crmd on iscs1
Stack: corosync
Current DC: iscs2 (2) - partition with quorum
Version: 1.1.10-32.el7_0.1-368c726
2 Nodes configured
6 Resources configured

Online: [ iscs2]
OFFLINE: [ iscs1]

Full list of resources:

 Master/Slave Set: masterdrbd [drbdwa]
     Masters: [ iscs2 ]
     Stopped: [ iscs1 ]
 Resource Group: grupiSCSI
     iscsiIP    (ocf::heartbeat:IPaddr2):       Started iscs2
     iscsTgt    (ocf::heartbeat:iSCSITarget):   Started iscs2
     iscsLUN    (ocf::heartbeat:iSCSILogicalUnit):      Started iscs2
 wtialma        (stonith:fence_wti):    Started iscs2

PCSD Status:
  iscs1: Offline
  iscs2: Online
```

Figura 3.15. Estado del cluster de almacenamiento luego del fallo del nodo iscs1

El error fue detectado por los nodos del *cluster* de alta disponibilidad lo cual se indica en la Figura 3.16, pero los nodos pudieron recuperarse ante esta falla, mientras que el servidor virtual no experimentó ninguna interrupción, como se indica en la Figura 3.17.

```
nodo2 iscsid: Kernel reported iSCSI connection 2:0 error
(1011 - ISCSI_ERR_CONN_FAILED: iSCSI connection failed)
state (3)

nodo2 iscsid: connection2:0 is operational after recovery (1
attempts)
```

Figura 3.16. Detección del error y recuperación de la LUN iSCSI en los nodos del cluster de alta disponibilidad

Sin embargo el escenario que se acaba de simular genera una situación de s*plit-brain*, en el recurso DRBD, ya que el nodo `iscs1` no tuvo tiempo de dejar el rol DRBD primario. Cuando el nodo `iscs1` se recupera del fallo, en los dos nodos que pertenecen el *cluster* de almacenamiento aparecerá el mensaje que se indica en la Figura 3.18, el cual alerta acerca de la detección de un problema de *split-brain* que no ha podido resolverse de forma automática, por lo que es necesario hacerlo de forma manual.

```
64 bytes from 192.168.1.36: icmp_req=34 ttl=63 time=4.86 ms
64 bytes from 192.168.1.36: icmp_req=35 ttl=63 time=4.52 ms
64 bytes from 192.168.1.36: icmp_req=36 ttl=63 time=6.88 ms
64 bytes from 192.168.1.36: icmp_req=37 ttl=63 time=4.85 ms
64 bytes from 192.168.1.36: icmp_req=38 ttl=63 time=4.09 ms
64 bytes from 192.168.1.36: icmp_req=39 ttl=63 time=9.42 ms
64 bytes from 192.168.1.36: icmp_req=40 ttl=63 time=4.06 ms
64 bytes from 192.168.1.36: icmp_req=41 ttl=63 time=104 ms
64 bytes from 192.168.1.36: icmp_req=42 ttl=63 time=40.5 ms
64 bytes from 192.168.1.36: icmp_req=43 ttl=63 time=55.5 ms
64 bytes from 192.168.1.36: icmp_req=44 ttl=63 time=5.18 ms
^C
--- 192.168.1.36 ping statistics ---
44 packets transmitted, 44 received, 0% packet loss, time 43066ms
rtt min/avg/max/mdev = 2.341/10.724/104.382/17.404 ms
wacero@wacerodell:~$
```

Figura 3.17. Prueba de conectividad con el comando ping al servidor virtual earth

```
iscs2 kernel: block drbd1: helper command: /sbin/drbdadm initial-split-brain
minor-1

iscs2 kernel: block drbd1: Split-Brain detected but unresolved, dropping
connection!

iscs2 kernel: block drbd1: helper command: /sbin/drbdadm initial-split-brain
minor-1 exit code 0
```

Figura 3.18. Detección de una situación split-brain

Para poder resolver esta situación es necesario desactivar en primer lugar los recursos del *cluster* de alta disponibilidad, luego desactivar los recursos del *cluster* de almacenamiento y por último realizar el procedimiento que se indica en la Línea de Comandos 3.89

```
##Desactivar el recurso drbdwa

# pcs resource disable drbdwa

##iscs1: Nodo del que se conservarán los datos (sobreviviente)

##iscs2: Nodo del que se descartará los datos (víctima)

##Activar DRBD

iscs1# drbdadm up drbdwa

iscs2# drbdadm up drbdwa

## Desconectar DRBD y descartar datos del nodo víctima

iscs2# drbdadm disconnect drbdwa

iscs2# drbdadm secondary drbdwa

iscs2# drbdadm connect --discard-my-data drbdwa

## Conectar el dispositivo DRBD en el nodo sobreviviente

iscs1# drbdadm connect drbdwa
```

Línea de Comandos 3.89. Procedimiento para la recuperación de una situación DRBD split-brain

Luego de esto iniciará un proceso de sincronización entre los dos nodos, y una vez terminado se mostrará el mensaje de recuperación que aparece en la Figura 3.19.

```
iscs2 kernel: block drbd1: Split-Brain detected but unresolved, dropping
connection!

iscs2 kernel: block drbd1: helper command: /sbin/drbdadm
initial-split-brain minor-1 exit code 0

iscs2 kernel: block drbd1: Split-Brain detected, manually solved. Sync
from peer node
```

Figura 3.19. Resultado de la recuperación del split-brain

Luego de esto ya es posible activar los recursos en el *cluster* de almacenamiento y del *cluster* de alta disponibilidad.

3.6.1.3 Pruebas de escritura

En la Tabla 3.9 se presentan las pruebas de escritura realizadas sobre las diferentes capas del almacenamiento compartido, para realizar las pruebas se utilizó el comando `dd` para la escritura en dispositivos de bloque y sistemas de archivos, se realizaron 10 pruebas en cada nivel y se promediaron los resultados.

Nivel	Velocidad MB/s
Disco duro físico	85
Disco RAID 1	83
Disco DRBD	80
Disco iSCSI	64
Sistema de archivos GFS2	58
Discos duros de servidores virtuales	32

Tabla 3.9. Velocidades de escritura de los niveles del almacenamiento compartido

En la Línea de Comandos 3.90 se indica el comando usado para la prueba de escritura del disco duro del servidor virtual `seisc`, para el resto de niveles se utilizó el mismo comando con diferentes tamaños de bloque. Es importante destacar que esta prueba determina la velocidad real de escritura.

```
dd if=/dev/zero of=/dev/vdb bs=100M count=10 conv=fdatasync
10+0 records in
10+0 records out
1048576000 bytes (1.0 GB) copied, 32.0489 s, 32.7 MB/s
```

Línea de Comandos 3.90. Prueba de escritura

Es importante señalar que la velocidad de escritura en el disco duro de los servidores virtuales se obtuvo utilizando el driver Virtio y no el driver por defecto IDE, el cual generaba velocidades de escritura bastante bajas.

3.6.2 PRUEBAS DEL CLUSTER DE ALTA DISPONIBILIDAD

3.6.2.1 Prueba de migración en caliente

En esta prueba se mostrará la capacidad del *cluster* de migrar una máquina virtual de un nodo a otro sin necesidad de apagar primero el servidor virtual, es decir con un tiempo de caída igual a cero.

Esta situación puede presentarse si es necesario dar mantenimiento a uno de los servidores de virtualización o si el *cluster* detecta algún fallo en uno de los nodos y requiere migrar sus recursos hacia otro nodo del *cluster*.

La prueba se realiza de la siguiente manera: el servidor virtual `earth` con dirección IP 192.168.1.36 se está ejecutando en el `nodo1`, y se lo migrará al `nodo2`. Mientras tanto se revisarán los *log* del nodo que actúa como *Designated Coordinator* a fin de mostrar el procedimiento que el *cluster* sigue al realizar la migración. Se dispone

también de una computadora portátil que ejecutará de forma constante el comando ping a la dirección IP 192.168.1.36.

Esta prueba puede realizarse de tres formas distintas:

- Usando el comando presentado en la Línea de Comandos 3.91, en cuyo caso el *cluster* se encarga de mover el recurso de tipo máquina virtual al servidor indicado. Este comando permite al usuario decidir donde ejecutar el servidor virtual.

- Utilizando el comando presentado en la Línea de Comandos 3.92, en cuyo caso el *cluster* se encarga de mover todos los recursos que el nodo ejecuta hacia los otros nodos del *cluster* y coloca al nodo en estado de reposo o *standby* y no permite que ningún recurso se ejecute en el mismo. En esta opción el *cluster* decide en que nodo ejecutará los recursos del nodo en reposo.

- Apagando el nodo en el que se ejecuta el servidor virtual. Este método presenta un inconveniente ya que la versión de Pacemaker 1.1.12, que se utiliza en el Proyecto de Titulación, tiene un error de programación[143], que hace que Pacemaker entre en conflicto con el sistema operativo al momento de detener recursos del tipo máquina virtual. Por este motivo no es posible realizar la prueba de migración en caliente utilizando este método.

```
# pcs resource move vmearth nodo2
```

Línea de Comandos 3.91. Comando para mover el servidor virtual del nodo actual al nodo2

Se elige realizar la prueba de migración en caliente empleando el comando presentado en la Línea de Comandos 3.92. En cuanto se ejecuta este comando

[143] Este error se corregirá en la siguiente versión 1.1.13, para más información se recomienda revisar la referencia [102].

Pacemaker realiza las operaciones necesarias para detener todos los servicios del `nodo1` de forma ordenada.

En la Figura 3.20 se presentan las acciones que Pacemaker realiza en el `nodo1` para migrar el servidor virtual al `nodo2`, en primer lugar el módulo `pengine` indica la acción a realizar, esta acción la ejecuta el módulo `crmd` y por último informa acerca del éxito de la operación.

```
# pcs cluster standby nodo1
```

Línea de Comandos 3.92. Comando para colocar al nodo1 en modo de reposo y migrar todos sus recursos

Como puede verse el *cluster* tiene éxito migrando el servidor virtual, el procedimiento no toma más que unos segundos, sin que el servidor virtual experimente pérdida de conectividad, como se indica en la Figura 3.21.

3.6.2.2 Prueba de recuperación ante el fallo de un nodo físico

En esta prueba se simulará el fallo total de uno de los nodos del *cluster* junto con el servidor virtual y los recursos que ejecuta, a fin de demostrar la capacidad del software del *cluster* para detectar la caída de los servicios y levantarlos nuevamente.

```
Mar 26 16:11:12 [29207] nodo2.redwa.local  pengine:   info:
MigrateRsc: Migrating vmearth from nodo1 to nodo2

Mar 26 16:11:12 [29207] nodo2.redwa.local     crmd:   info:
match_graph_event:    Action vmearth_migrate_to_0 (79)confirmed on
nodo2 (rc=0)

Mar 26 16:11:12 [29207] nodo2.redwa.local      crmd:  notice:
     te_rsc_command:Initiating action 80: migrate_from
vmearth_migrate_from_0 on nodo2 (local)

VirtualDomain (vmearth)[582]: 2015/03/26_16:11:15 INFO: earth:live
migration from nodo1 succeded.
```

Figura 3.20. Proceso de migración de la máquina virtual earth

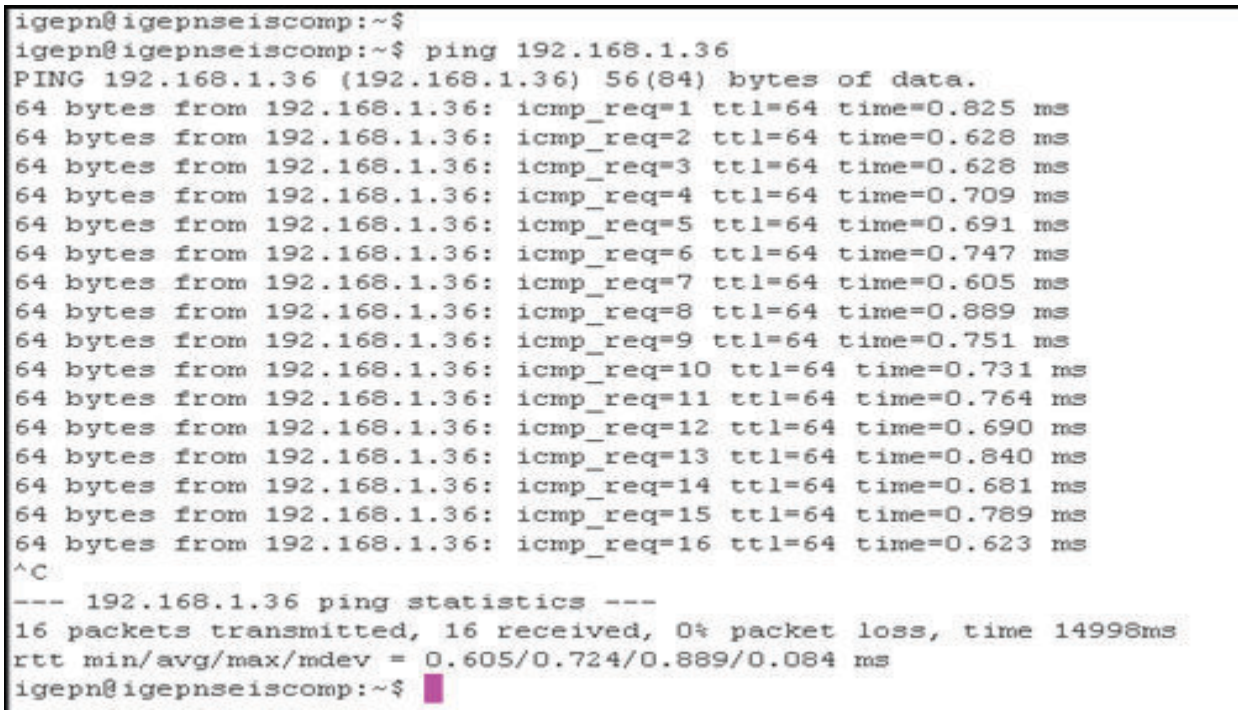

Figura 3.21. Test de conectividad con el servidor seisc

Para realizar esta prueba se desconectará la fuente de alimentación del nodo en el que se está ejecutando el servidor virtual `shake`, una vez que el *cluster* detecte que el servidor virtual no se está ejecutando arrancará el servidor virtual en uno de los nodos que esté disponible, con una interrupción a los servicios mínima.

Al desconectar el `nodo3`, el *cluster* se percata de la ausencia del mismo y toma las acciones correspondientes, como notificar al resto de nodos de lo sucedido y asegurarse de que el nodo realmente está apagado, para lo que mediante el dispositivo de *fencing*, conecta y desconecta la alimentación del nodo, como puede verse en la parte inferior de la Figura 3.22.

En cuanto al servidor virtual `shake` que se ejecutaba en el `nodo3`, Pacemaker se da cuenta que el recurso no puede ejecutarse en ese servidor, por lo que luego de que la operación de *fencing* se ha completado ordena que la máquina virtual `shake` arranque en uno de los nodos disponibles, en este caso el `nodo1`, como puede verse en el registro que presenta la Figura 3.23.

```
Sep 28 18:21:03 [2626] nodo2.redwa.local    crmd:    Node nodo3[2]
- state is now lost (was member)
Sep 28 18:21:03 [2629] nodo2.redwa.local    info:      Peer nodo3
left us
Sep 28 18:21:03 [2626] nodo2.redwa.local  notice:  Removing all
nodo3 attributes
Sep 28 18:21:05 [2628] nodo2.redwa.local    pengine:  warning:
Node nodo3 will be fenced because the node is no longer part of
the cluster
```

Figura 3.22 Proceso de fencing del nodo3

Como puede verse en la Figura 3.24, la pérdida de conectividad es mínima, ya que aún cuando existe un porcentaje de 20% de paquetes perdidos, esta pérdida corresponde al tiempo que le toma al servidor virtual arrancar, aproximadamente 1 minuto, ya que el *cluster* se apresura en volver a iniciar el servidor virtual y los recursos que se ejecutaban en el mismo, como se indica en la Figura 3.25.

```
Sep 28 18:21:05 [2628] nodo2.redwa.local    pengine:
vmshake_stop_0 on nodo3 is unrunnable (offline)
Sep 28 18:21:05 [2628] nodo2.redwa.local    pengine:
vmshake_stop_0 is implicit after nodo3 is fenced
Sep 28 18:21:05 [2628] nodo2.redwa.local    pengine:
Move    vmshake (Started nodo3 -> nodo1)
Sep 28 18:21:08 [2628] nodo2.redwa.local    pengine:
Start recurring monitor (10s) for vmshake on nodo1
Sep 28 18:21:11 [2629] nodo2.redwa.local       crmd:
vmshake_start_0 (59) confirmed on nodo1 (rc=0)
```

Figura 3.23 Recuperación del servidor virtual shake luego del fallo del servidor físico nodo3

```
64 bytes from 192.168.1.37: icmp_req=14 ttl=64 time=0.490 ms
64 bytes from 192.168.1.37: icmp_req=15 ttl=64 time=0.381 ms
64 bytes from 192.168.1.37: icmp_req=16 ttl=64 time=0.539 ms
64 bytes from 192.168.1.37: icmp_req=46 ttl=64 time=2909 ms
64 bytes from 192.168.1.37: icmp_req=47 ttl=64 time=1909 ms
64 bytes from 192.168.1.37: icmp_req=48 ttl=64 time=909 ms
64 bytes from 192.168.1.37: icmp_req=129 ttl=64 time=0.615 ms
64 bytes from 192.168.1.37: icmp_req=130 ttl=64 time=0.489 ms
64 bytes from 192.168.1.37: icmp_req=131 ttl=64 time=0.378 ms
--- 192.168.1.37 ping statistics ---
140 packets transmitted, 111 received, 20% packet loss, time
138999ms
rtt min/avg/max/mdev = 0.311/52.378/2909.614/337.392 ms, pipe 3
```

Figura 3.24 Test de conectividad durante la recuperación de shake

```
Sep 28 18:21:51 [2628] nodo2.redwa.local    pengine:     info:
native_print:       mysql    (ocf::heartbeat:mysql): Started shake
Sep 28 18:21:51 [2628] nodo2.redwa.local    pengine:     info:
native_print:       web      (ocf::heartbeat:apache):
Started shake
```

Figura 3.25 Proceso de recuperación de los servicios mysql y web en el servidor virtual shake

3.6.2.3 Prueba de alta disponibilidad de los sistemas de adquisición y procesamiento

La prueba se realizará con los sistemas SeisComP3 y Earthworm, ya que como se mencionó en la Sección 2.4.3.1, no es posible dar alta disponibilidad a los módulos que forman el sistema ShakeMap, sino solamente a la base de datos y al servidor web que ShakeMap utiliza.

La prueba del sistema SeisComP3 se realizará deteniendo el servidor virtual `seisc` en donde el sistema se ejecuta y se espera que Pacemaker migre los servicios de forma inmediata al servidor `shake`.

Para comprobar que los clientes aún puedan acceder a los servicios que SeisComP3 brinda se solicitarán formas de onda de algunas estaciones al módulo SeedLink mediante el programa `scrttv`, el resultado esperado es que el cliente `scrttv`

continúe graficando las formas de onda sin que se presente ninguna interrupción considerable.

```
# pcs cluster disable vmseisc
```

Línea de Comandos 3.93. Comando para detener el servidor virtual seisc

Una vez detenido el servidor virtual Pacemaker inicia la migración de los recursos, dado que los módulos de SeisComP3 están agrupados el sistema mueve el primer recurso `ipsc3` y debido a las restricciones que existen en los recursos agrupados, el resto de recursos migran y se activan en el servidor `shake`, como se indica en la Figura 3.26.

```
Sep 29 08:50:03 [2628] nodo2.redwa.local    pengine:   notice:
LogActions: Move    ipsc3   (Started seisc -> shake)

Sep 29 08:50:04 [2629] nodo2.redwa.local       crmd:     info:
match_graph_event:  Action ipsc3_stop_0 (79) confirmed on seisc (rc=0)

Sep 29 08:50:04 [2629] nodo2.redwa.local       crmd:     info:
match_graph_event:  Action ipsc3_start_0 (80) confirmed on shake (rc=0)
```

Figura 3.26 Proceso de migración del recurso ipsc3

El cliente `scrttv` se percata de la pérdida de conectividad pero se vuelve a conectar de forma inmediata luego de dos segundos, como se indica en la Figura 3.27.

Así mismo de forma gráfica no se presenta ninguna pérdida de datos en el período de tiempo que se realiza la prueba, lo cual se puede verificar en la Figura 3.28.

La prueba del sistema Earthworm se realizará de forma similar, se detendrá el servidor virtual `earth` y Pacemaker debe encargarse de ejecutar los módulos al servidor virtual `shake`.

La comprobación que los servicios que brinda Earthworm no pierdan conectividad se realiza mediante el programa `swarm`[144], que solicita formas de onda al módulo `wave_serverv` de Earthworm y presenta esos datos de forma gráfica.

```
13:49:43 [error] Timeout while waiting for acknowledgment message
13:49:43 [error] Reconnect failed, wait 2 sec and try again...
13:49:45 [info] Connecting to server: 192.168.1.93
13:49:45 [info] Connected to message server: 192.168.1.93
13:49:45 [info] Joining MASTER_GROUP group
13:49:45 [info] Sending connect message to server: 192.168.1.93
13:49:46 [info] Joining group: CONFIG
13:49:46 [info] Joining group: EVENT
13:49:46 [info] Joining group: GUI
13:49:46 [info] Joining group: LOCATION
13:49:46 [info] Joining group: PICK
13:49:46 [info] Client is reconnected to master client
```

Figura 3.27 Reconexión del módulo scrttv

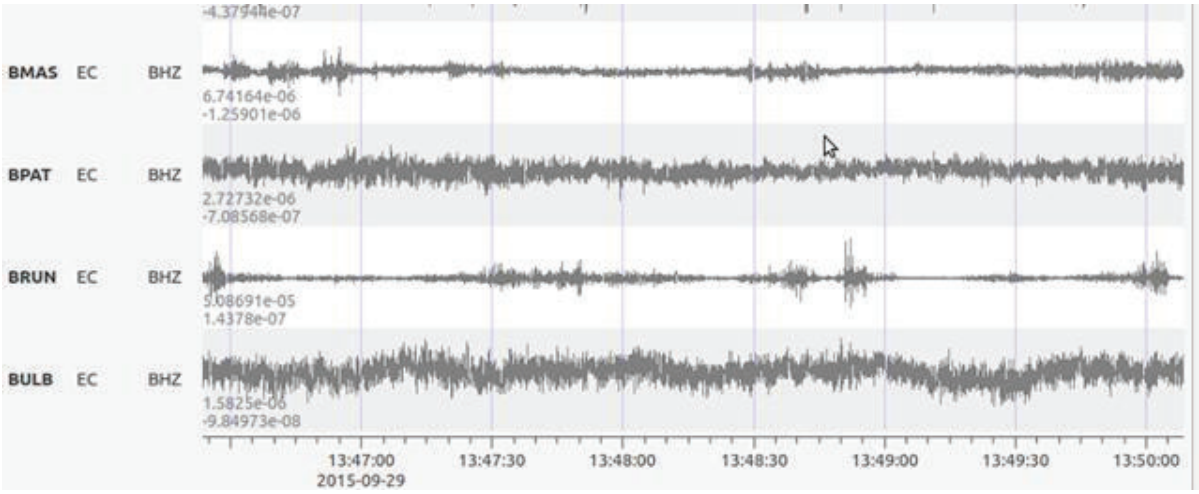

Figura 3.28 Funcionamiento ininterrumpido de scrttv durante la migración de Seedlink

[144] Swarm: *Seismic Wave Analysis and Realtime Monitor* es un programa que permite graficar formas de onda en tiempo real y graficarlas en función del tiempo o la frecuencia.

Luego de ejecutar el comando que se indica en la Línea de Comandos 3.94, los servicios que `earth` ejecutaba inician su migración al servidor `shake`, como puede verse en la Figura 3.29, una vez que migra el recurso `ipew`, el cual es un recurso del tipo IP y el primero del grupo de recursos `Earthworm`,el resto de módulos del grupo inician una migración y activación ordenada en el servidor `shake`.

```
# pcs cluster disable vmearth
```

Línea de Comandos 3.94. Comando para detener el servidor virtual earth

```
Sep 30 10:44:03 [1631] nodo1.redwa.local    pengine:   notice:
LogActions: Move    ipew    (Started earth -> shake)

Sep 30 10:44:03 [1631] nodo1.redwa.local       crmd:     info:
match_graph_event:  Action ipew_stop_0 (79) confirmed on earth (rc=0)

Sep 30 10:44:03 [1631] nodo1.redwa.local       crmd:     info:
match_graph_event:  Action ipew_start_0 (80) confirmed on shake (rc=0)
```

Figura 3.29 Proceso de migración del recurso ipew

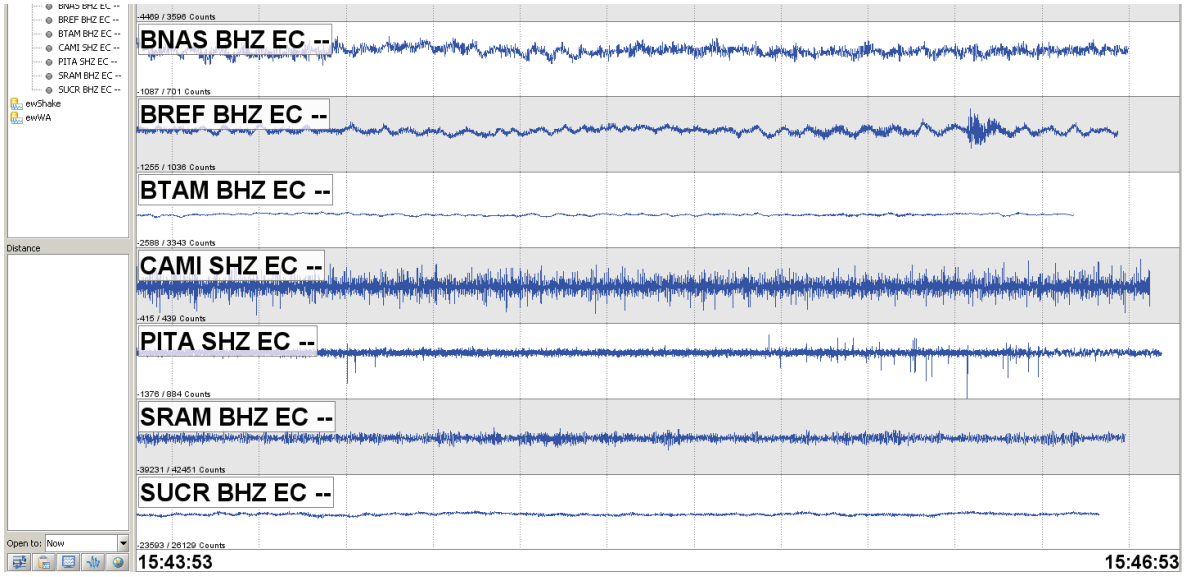

Figura 3.30 Funcionamiento ininterrumpido de Swarm durante la prueba de alta disponibilidad del sistema Earthworm

3.6.3 PRUEBAS DE LA RED DEL CLUSTER

La prueba de redundancia de la red del *cluster* consiste en desconectar uno de los conmutadores de la red de comunicaciones para que los enlaces pasivos del conmutador de respaldo se activen sin que por esta falla el funcionamiento del *cluster* se vea afectado. En la Figura 3.31, la Figura 3.32 y la Figura 3.33 se presentan los resultados de esta prueba, como puede verse los tres nodos que forman el *cluster* de alta disponibilidad se percatan de que se ha perdido conectividad en una de las interfaces, por lo que de forma inmediata activan la interfaz de respaldo, sin que exista perdida de conectividad como se presenta en la Figura 3.34.

```
Sep 28 19:33:06 nodo1 kernel: em1: NIC Copper Link is Down
Sep 28 19:33:06 nodo1 kernel: bonding: bond0: link status
definitely down for interface em1, disabling it
Sep 28 19:33:06 nodo1 kernel: bonding: bond0: making interface em2
the new active one.
```

Figura 3.31 Resultado del test de redundancia del conmutador en el nodo1

```
Sep 28 19:33:06 nodo2 kernel: eno1: igb: eno1 NIC Link is Down
Sep 28 19:33:06 nodo2 kernel: bonding: bond0: link status
definitely down for interface eno1, disabling it
Sep 28 19:33:06 nodo2 kernel: bonding: bond0: making interface
eno2 the new active one.
```

Figura 3.32 Resultado del test de redundancia del conmutador en el nodo2

```
Sep 28 19:33:06 nodo3 kernel: ens2: link down
Sep 28 19:33:06 nodo3 kernel: bonding: bond0: link status
definitely down for interface ens2, disabling it
Sep 28 19:33:06 nodo3 kernel: bonding: bond0: making interface
ens1 the new active one.
```

Figura 3.33 Resultado del test de redundancia del conmutador en el nodo3

```
igepn@igepnseiscomp:~$ ping nodo1
PING 192.168.1.90 (192.168.1.90) 56(84) bytes of data.
64 bytes from 192.168.1.90: icmp_req=1 ttl=64 time=0.506 ms
64 bytes from 192.168.1.90: icmp_req=2 ttl=64 time=0.475 ms
64 bytes from 192.168.1.90: icmp_req=3 ttl=64 time=0.489 ms
64 bytes from 192.168.1.90: icmp_req=4 ttl=64 time=0.423 ms
64 bytes from 192.168.1.90: icmp_req=5 ttl=64 time=0.509 ms
64 bytes from 192.168.1.90: icmp_req=6 ttl=64 time=0.483 ms
64 bytes from 192.168.1.90: icmp_req=7 ttl=64 time=0.409 ms
64 bytes from 192.168.1.90: icmp_req=8 ttl=64 time=0.490 ms
64 bytes from 192.168.1.90: icmp_req=9 ttl=64 time=0.467 ms
64 bytes from 192.168.1.90: icmp_req=10 ttl=64 time=0.485 ms
64 bytes from 192.168.1.90: icmp_req=11 ttl=64 time=0.475 ms
64 bytes from 192.168.1.90: icmp_req=12 ttl=64 time=0.376 ms
64 bytes from 192.168.1.90: icmp_req=13 ttl=64 time=0.519 ms
64 bytes from 192.168.1.90: icmp_req=14 ttl=64 time=0.387 ms
64 bytes from 192.168.1.90: icmp_req=15 ttl=64 time=0.437 ms

--- 192.168.1.90 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 13997ms
rtt min/avg/max/mdev = 0.376/0.462/0.519/0.043 ms
```

Figura 3.34 Prueba de conectividad con el comando ping al servidor nodo1

La segunda prueba consiste en desconectar el cable de red del enlace activo, para simular un fallo de la interfaz de red.

La segunda interfaz de red que forma el enlace activo – pasivo se activa de forma automática y evita que el nodo pierda comunicaciones con el resto del *cluster*, como puede verse en la Figura 3.35 y la Figura 3.36, el sistema operativo se percata de que el enlace de la interfaz de red `ens1` ha caído y de forma inmediata hace que la interfaz de respaldo `ens2` sea la nueva interfaz activa.

```
Sep 28 19:15:17 nodo3 kernel: r8169 0000:20:00.0 ens1: link down
Sep 28 19:15:17 nodo3 kernel: bonding: bond0: link status
definitely down for interface ens1, disabling it
Sep 28 19:15:17 nodo3 kernel: bonding: bond0: making interface
ens2 the new active one.
```

Figura 3.35 Resultado del test realizado a la interfaz enlazada bond0

```
[root@nodo3 ~]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: ens2
MII Status: up

Slave Interface: ens1
MII Status: down
Speed: 10 Mbps
Duplex: half
Link Failure Count: 2

Slave Interface: ens2
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 1
```

Figura 3.36 Cambio de estado de la interfaz de enlazada bond0

3.7 COSTOS REFERENCIALES

Para crear este presupuesto referencial se detallan los costos de los servidores presentados en la Tabla 3.2, y además de los equipos de red utilizados. Los precios unitarios de estos equipos y el costo total se indican en la Tabla 3.10.

Equipo	Número	Costo unitario	Subtotal
Servidor HP Proliant ML350e	1	1500	1500
Servidor Dell PowerEdge T410	1	1500	1500
Computador de escritorio HP d5800M	3	700	2100
Conmutador TP-Link SG108	3	45	135
Conmutador TP-Link SG1005D	1	25	25
Controladora de alimentación WTI NPS-115	1	110	140
Tarjeta Gigabit PCI Express	3	18	54
TOTAL			5454 $

Tabla 3.10. Costos referenciales del proyecto

Los precios de los servidores y equipos de red se obtuvieron de los facturas de adquisición del Instituto Geofísico.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- En el Proyecto se utilizó RAID implementado mediante software, esta tecnología no requiere la adquisición de tarjetas RAID adicionales, su instalación y configuración son bastante sencillas, tiene un buen desempeño y ha permitido implementar redundancia a nivel de disco sin tener que incurrir en gastos adicionales.

- La utilización de la tecnología RAID 1 hace posible que en caso de que un disco físico del arreglo falle, el disco RAID 1 continúe funcionando sin que el funcionamiento del resto del *cluster* se vea afectado. RAID 1 ha demostrado tener velocidades de lectura y escritura que cumplen con las requeridas por el *cluster* de alta disponibilidad y las máquinas virtuales que el *cluster* ejecuta.

- La tecnología DRBD utilizada en el presente Proyecto, ha sido fácil de instalar, medianamente compleja de configurar, ha demostrado tener un buen funcionamiento y una rápida recuperación ante fallas, además ha permitido agregar una capa de redundancia a nivel de almacenamiento, sin tener que emplear soluciones propietarias similares en funcionalidad pero con costos bastante elevados.

- Las tecnologías DRBD y iSCSI junto con el administrador de recursos de *cluster* Pacemaker constituyen el *cluster* de almacenamiento. El servicio con alta disponibilidad que este *cluster* brinda es un dispositivo de almacenamiento y garantiza que aunque uno de los dos servidores DRBD falle, la LUN iSCSI esté siempre disponible, lo que permite que las operaciones del *cluster* de alta disponibilidad no se vean afectadas y los servidores virtuales sigan brindando sus servicios sin presentar ninguna interrupción, y en el caso de situaciones más graves, minimizar los tiempos de caída.

- El *cluster* de alta disponibilidad en el que se ejecutan los servidores virtuales es del tipo activo – activo, es decir que no existen nodos destinados a activarse únicamente cuando un nodo activo falla, con lo que no existe un desperdicio de recursos de hardware, además que se realiza balanceo de carga, ya que no es un único nodo el que está encargado de ejecutar todos los servidores virtuales.

- El administrador de bloqueo DLM, la versión *cluster* de LVM y el sistema de archivos GFS2, administrados como recursos de tipo clon por Pacemaker, permiten que los nodos del *cluster* de alta disponibilidad accedan de forma simultánea y ordenada al dispositivo de bloque, sin que exista corrupción en el sistema de archivos. Tener un almacenamiento centralizado y replicado hace posible que si un nodo del *cluster* de alta disponibilidad falla, los nodos restantes puedan acceder a los datos del almacenamiento compartido y continuar con los servicios sin que exista interrupción o minimizándola.

- Implementar un *cluster* del tipo activo – activo solo es posible si se utiliza un almacenamiento compartido en el que los nodos del *cluster* puedan acceder de forma concurrente, tal como un servidor NFS, e inclusive un servidor Samba, este tipo de tecnologías bastaría para implementar un *cluster* de alta disponibilidad para un servicio web, pero para el caso del presente Proyecto en el que los servicios a los que se pretende dar alta disponibilidad son sistemas ejecutados en máquinas virtuales es necesario utilizar tecnologías como GFS2 que tienen un desempeño mucho mayor.

- La tecnología de virtualización KVM empleada en el Proyecto ha resultado ser confiable y tener un buen desempeño como plataforma de virtualización, a más de ser bastante fácil de instalar y configurar y disponer de varias herramientas para la creación, administración y respaldo de máquinas virtuales.

- La integración del administrador de recursos del *cluster* Pacemaker con máquinas virtuales KVM permite que los recursos que un servidor virtual ejecuta puedan ser controlados por el *cluster* sin tener que instalar todo el `stack` del *cluster* en un servidor virtual y sin afectar el quórum del *cluster*.

- El software de administración del *cluster* Pacemaker permite, mediante la aplicación de reglas y políticas configuradas previamente, automatizar la administración de los servicios que un sistema brinda, es decir que en caso que un recurso falle, el software del *cluster* se percata de este evento y hace lo posible por levantar nuevamente el recurso. En los casos en que no puede recuperarse ante un fallo, el software se encarga de apagar el recurso con errores y aquellos que dependan del mismo, pudiendo incluso apagar el nodo en el que el recurso se ejecutaba a fin de evitar errores más graves como por ejemplo la corrupción del sistema de archivos.

- Es importante entender la instalación, configuración y funcionamiento de un servicio antes de agregarlo como un recurso del *cluster* controlado por el respectivo agente de recursos y Pacemaker. Así por ejemplo antes de crear el recurso *target* iSCSI denominado `iscsTgt` se instalaron y configuraron varios *target* iSCSI de prueba y una vez comprendido el funcionamiento de los comandos y las opciones que debían configurarse ya fue posible agregar el target iSCSI como un recurso administrado por el *cluster* y el agente de recursos `ocf:Heartbeat:scsi`. Se siguió este procedimiento con todos los recursos creados en el *cluster*, lo que facilitaba la creación del recurso y la resolución de errores. Seguir este procedimiento ayuda incluso a detectar posibles errores en el agente de recursos y facilita su corrección.

- Existe gran variedad de agentes de recursos que se incluyen al momento de instalar Pacemaker, los mismos funcionan bastante bien y permiten a Pacemaker administrar y monitorizar los recursos, sin embargo en algunos casos fue necesario realizar pequeñas modificaciones, a fin de corregir algunos errores.

- Aún cuando la gran mayoría de aplicaciones puede ejecutarse en forma de servicios y ser controlados por un agente de recursos, como los módulos que pertenecen a los sistemas Earthworm y SeisComP3, existen también programas que no cumplen con los requisitos que un agente de recursos del *cluster* tiene, como por ejemplo los módulos que forman el sistema ShakeMap, los que al no funcionar como servicios que se ejecutan de forma continua, impiden a Pacemaker monitorizarlos para determinar su estado.

- El dispositivo de *fencing* utilizado ha funcionado bastante bien y cumplido con los requerimientos requeridos por el *cluster*, el dispositivo es compartido por el *cluster* de alta disponibilidad y el *cluster* de almacenamiento y permite controlar el encendido o apagado de los nodos de ambos *clusters*, este dispositivo es tan importante que sin el mismo no sería posible implementar de forma segura el *cluster* de alta disponibilidad ni el *cluster* de almacenamiento.

- El sistema operativo CentOS utilizado para implementar el *cluster* de almacenamiento y el *cluster* de alta disponibilidad es un sistema operativo fácil de instalar, no tiene licenciamiento, cuenta con gran variedad de software y ha demostrado ser bastante confiable y estable, además de ser compatible con todas las tecnologías utilizadas en el presente Proyecto.

- Los nodos físicos empleados para el *cluster* de alta disponibilidad tienen diferentes características de hardware, RAM, CPU, etc. lo que comprueba que todo el software empleado en el proyecto es independiente del hardware que se utilice, a diferencia de algunas de las soluciones de software privado que requieren la utilización de hardware especializado y de alto costo.

- Las pruebas que se realizaron demostraron la efectividad del uso de interfaces de red enlazadas en el modo activo-pasivo, ya que permiten que en el caso que una de las interfaces de red falle la otra tome su lugar, lo que permite también que si uno de los conmutadores experimenta un fallo, el *cluster* no pierda comunicaciones y continúe en funcionamiento.

4.2 RECOMENDACIONES

- A fin de mejorar la utilización del espacio de los discos físicos se podría emplear RAID 5, para lo que se requiere por lo menos tres particiones de tamaños similares, sin embargo es necesario tomar en cuenta que en comparación con RAID 1 el nivel de redundancia no mejora y que la velocidad de escritura puede disminuir.

- Para agregar escalabilidad al *cluster* de almacenamiento, entre la capa de RAID y la capa de DRBD, se podría agregar una capa LVM. Es decir el disco RAID 1 se utilizaría para crear un volumen lógico que de ser necesario puede aumentar de tamaño agregando más discos RAID 1 implementados mediante software, DRBD utilizaría entonces el volumen lógico en lugar de los discos RAID 1, sin embargo al agregar una nueva capa es posible que la velocidad de escritura disminuya.

- Se recomienda utilizar discos de 1 TB de capacidad lo que ampliaría el número de estaciones sísmicas que ingresan, así como el periodo de tiempo que esta información estaría disponible. Un almacenamiento compartido de mayor tamaño permitiría también poder realizar tareas administrativas sobre las máquinas virtuales, tales como: realizar copias de respaldo de los discos virtuales, creación de plantillas para servidores virtuales, etc.

- Sería posible agregar recuperación ante desastres al *cluster* de alta disponibilidad mediante la adición de un tercer servidor al *cluster* de almacenamiento DRBD que se encuentre en un área física alejada del Instituto Geofísico. La información se replicaría al tercer nodo DRBD y en caso de un desastre en el edificio en el que se encuentra el instituto, los datos de las máquinas virtuales aún estaría disponibles para su posterior recuperación.

- A fin de mejorar la velocidad de replicación de datos entre los servidores DRBD sería posible emplear una red exclusiva para este fin o inclusive

emplear cable UTP categoría 5 o superior, sin embargo cada nodo del *cluster* de almacenamiento requeriría dos interfaces de red adicionales, a fin de garantizar redundancia.

- La implementación de un almacenamiento iSCSI empleando una red Giga Ethernet tiene un desempeño que cumple con los requerimientos del presente Proyecto, sin embargo para aplicaciones que requieran mayores velocidades de escritura sería necesario emplear iSCSI sobre enlaces Infiniband o enlaces de fibra.

- Es necesario agregar un sistema que muestre el estado de los nodos del *cluster*, el estado de los recursos que el *cluster* ejecuta, la utilización de recursos de hardware, tal como CPU, RAM, interfaces de red, etc. Para esto podría utilizarse algún software de monitorización tal como Zabbix.

- Para facilitar las tareas de administración relacionadas con los servidores virtuales, podría utilizarse un sistema de administración con interfaz gráfica o un sistema de administración vía web, sin embargo ambos tipos de sistemas emplean la librería `libvirt` para funcionar, por lo que no agregan ninguna opción que la línea de comandos `virsh` no brinde.

- Se utilizó el agente de recursos `ocf:Heartbeat:anything` para controlar los módulos de los sistemas de adquisición y procesamiento, luego de una pequeña modificación para adaptar el agente a los requerimientos de los módulos, sin embargo puede ser necesario crear un agente de recursos para monitorizar módulos con requerimientos especiales, como por ejemplo aquellas aplicaciones que crean múltiples subprocesos.

- De ser necesario monitorizar y controlar recursos en un servidor físico existente con el sistema operativo Linux, esto podría hacerse empleando `Pacemaker_remote`, con lo que el servidor físico formaría parte del *cluster*

pero sin que forme parte del quórum, lo que no afectaría la escalabilidad del *cluster*.

- Es necesario separar la red de comunicaciones que utiliza el *cluster* y la red con la que el *cluster* de alta disponibilidad accede al almacenamiento compartido, para esto es necesario agregar dos tarjetas de red a cada uno de los nodos del *cluster* y dos conmutadores, lo que mejoraría el desempeño de lectura y escritura del *cluster*.

- El tipo de enlace de interfaces de red utilizado para garantizar redundancia en todas las interfaces es el conocido como modo activo/respaldo, sin embargo podría emplearse también el modo denominado como `Round Robin`, que incluye balanceo de carga y tolerancia a fallos, para la red de las máquinas virtuales, e incluso para la red de comunicaciones del *cluster*, pero no se recomendaría utilizarla para la red de replicación de datos de los servidores DRBD.

- Se recomienda no utilizar software en fase de desarrollo o *release candidate*, en un sistema que se utiliza en producción, al menos no sin antes haberlo ejecutado en un ambiente de pruebas, durante un tiempo prudente y luego de haber realizado varios ensayos. El mismo procedimiento se recomienda para cualquier actualización de software que se realice.

- Es importante tener en cuenta la familia a la que pertenecen los procesadores de los servidores físicos que se emplearán en la virtualización, si los procesadores no son de la misma generación es necesario definir en el archivo del servidor virtual una etiqueta que permita la compatibilidad con modelos de CPU anteriores.

- Al momento el *cluster* de alta disponibilidad y el *cluster* de almacenamiento cuentan con un único dispositivo de *fencing*, lo que representa un punto único de falla para ambos *clusters*, de ser posible es necesario adquirir otros

dispositivo de *fencing* o implementar otro método que permita al *cluster* desconectar a un nodo que presente fallas.

- Antes de ejecutar cualquiera de los recursos es importante verificar que todas las restricciones estén definidas correctamente, ya que el *cluster* tratará de ejecutar un recurso en cualquier nodo que esté disponible, así por ejemplo antes de ejecutar los servidores virtuales es necesario definir una regla que impida que los recursos como sistemas de archivos o volúmenes lógicos se ejecuten en los servidores virtuales, de lo contrario el *cluster* fallará al activarlos y detectará que existen fallas en el servidor virtual.

- Aunque es posible utilizar la interfaz web para realizar tareas de configuración de los nodos y recursos del *cluster*, este método aún presenta algunos inconvenientes en especial al tratar de configurar algunas opciones de los agentes de recursos, por ello se recomienda utilizar solamente la consola de configuración `pcs`.

- Es muy importante que los relojes de los nodos que forman el *cluster* estén sincronizados, de lo contrario el software de comunicaciones del *cluster* asumiría que existen retardos en el envío y recepción de mensajes, lo que generaría a su vez mensajes de alerta o inclusive que el software de administración del *cluster* decida apagar al nodo por considerar que presenta fallas.

- En relación con la tecnología DRBD es también importante que los relojes se encuentren sincronizados ya que para asegurar el estado de los servidores DRBD primario y secundario se envía un mensaje de monitoreo cada cierto período de tiempo, y al igual que con los relojes de los nodos del *cluster*, un falso retardo en las comunicaciones podría generar que ambos servidores asuman que el otro fallo y entrar en una situación de *split-brain*.

REFERENCIAS BIBLIOGRÁFICAS

[1] **Weygant, Peter.** *Clusters for High Availability: A Primer of HP Solutions.* EEUU : Prentice Hall, 2014.

[2] **Schmidt, Klaus.** *High Availability and Disaster Recovery.* EEUU : Springer, 2006.

[3] **Microsoft Technet.** Cluster Architecture Essentials. [Online] 11 2013. [Cited: 11 26, 2014.] http://technet.microsoft.com/en-us/library/cc737532%28v=ws.10%29.aspx.

[4] *Ahead of the Pack: The Pacemaker High-Availability Stack.* **Haas, Florian.** EEUU : Linux Journal, 2014, Vol. 216.

[5] **Cisco Systems.** Data Center High Availability Clusters Design Guide. [Online] 2006. [Cited: 11 26, 2014.] http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/HA_Clusters/HA_Clusters.html.

[6] **Roth, Tanja.** *SUSE Linux Enterprise High Availability Extension.* EEUU : Novell, Inc., 2013.

[7] *Replicate Everything! Highly Available iSCSI Storage with DRBD and Pacemaker.* **Haas, Florian.** EEUU : Linux Journal, 2012, Vol. 217.

[8] **Jones, Tim.** High availability with the Distributed Replicated Block Device. [Online] 2010. [Cited: 10 26, 2014.] http://www.ibm.com/developerworks/library/l-drbd/l-drbd-pdf.pdf .

[9] **Red Hat, Inc.** High Availability Add-On Overview. [Online] 2010. [Cited: 11 16, 2014.] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/High_Availability_Add-On_Overview/index.html.

[10] **Red Hat, Inc.** Using GNBD with Global File System. EEUU 2007. [Online] 2007. [Cited: 11 29, 2014.] https://www.centos.org/docs/5/html/Global_Network_Block_Device/index.html.

[11] —. Cluster Suite Overview. [Online] 2014. [Cited: 11 30, 2014.] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/Cluster_Suite_Overview/index.html .

[12] **GitHub.** Corosync Home Page. [Online] 2012. [Cited: 09 27, 2014.] https://github.com/corosync/corosync/wiki/.

[13] **Red Hat, Inc.** Openais. [Online] 2014. [Cited: 09 27, 2014.] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/html/5.6_Technical_Notes/openais.html.

[14] **Oracle.** Oracle Clusterware Administration and Deployment Guide. [Online] 2013. [Cited: 9 18, 2014.] http://docs.oracle.com/cd/E16655_01/rac.121/e17886/toc.htm.

[15] **Hussay, Syed.** *Expert Oracle RAC 12c.* EEUU : Apress, 2013.

[16] **Oracle.** Introduction to Oracle Automatic Storage Management. [Online] 2014. [Cited: 11 28, 2014.] https://docs.oracle.com/cd/E11882_01/server.112/e18951/asmcon.htm#OSTMG03601.

[17] **Shaw, Steve.** *Pro Oracle Database 11g RAC on Linux.* EEUU : Apress, 2010.

[18] **Red Hat, Inc.** Add-Ons for Red Hat Enterprise Linux Server. [Online] 2014. [Cited: 8 10, 2014.] https://www.redhat.com/apps/store/add-ons/.

[19] **SUSE.** How to Buy SUSE Linux Enterprise High Availability Extension Subscriptions. [Online] [Cited: 07 22, 2014.] https://www.suse.com/products/highavailability/how-to-buy/ .

[20] **Oracle.** Oracle Linux FAQ. [Online] 2014. [Cited: 07 22, 2014.] http://www.oracle.com/us/technologies/027617.pdf.

[21] **Red Hat, Inc.** RGManager Vs Pacemaker. [Online] 2014. [Cited: 12 08, 2014.] https://fedorahosted.org/cluster/wiki/RGManagerVsPacemaker#p7.

[22] **Oracle .** Oracle Clusterware 12.1. [Online] 2013. [Cited: 12 08, 2014.] http://www.oracle.com/technetwork/database/database-technologies/clusterware/overview/oracle-clusterware-12c-overview-1969750.pdf.

[23] **Beekhof, Andrew.** Cluster from scratch. [Online] 2009. [Cited: 09 18, 2014.] http://clusterlabs.org/doc/en-US/Pacemaker/1.1-plugin/html-single/Clusters_from_Scratch/ .

[24] **Pacemaker.** Pacemaker. [Online] 2010. [Cited: 08 16, 2014.] http://clusterlabs.org/wiki/Pacemaker .

[25] **Red Hat, Inc.** *Red Hat Enterprise Linux 7 High Availability Add-On Reference.* EEUU : Red Hat, 2014.

[26] **Smith, Roderick.** *Linux Professional Institute Certification 2.* EEUU : Sybex, 2011.

[27] **Vadala, Derek.** *Managing RAID on Linux.* EEUU : O'Reilly, 2003.

[28] **Wikipedia.** Standard RAID Levels. [Online] 2014. [Cited: 12 11, 2014.] http://en.wikipedia.org/wiki/Standard_RAID_levels.

[29] **Van Dyke, Kendal.** Disk Performance Hands On, Part 6: RAID 10 vs. RAID 1. [Online] 02 2009. [Cited: 05 13, 2014.] http://www.kendalvandyke.com/2009/02/disk-performance-hands-on-part-6-raid.html.

[30] **LINBIT.** About DRBD. [Online] 2014. [Cited: 12 01, 2014.] http://www.linbit.com/en/products-and-services/drbd.

[31] **Long, James.** *Storage Networking Protocol Fundamentals.* EEUU : Cisco Press, 2006.

[32] **Ben, Rockwood.** A Quick Guide to iSCSI on Linux. [Online] 2004. [Cited: 01 07, 2015.] http://www.cuddletech.com/articles/iscsi/ar01s02.html.

[33] **Novell, Inc.** *SUSE Linux Enterprise Server.* EEUU : s.n., 2011.

[34] **Red Hat, Inc.** *Red Hat Enterprise Linux 7 Global File System 2.* EEUU : s.n., 2014.

[35] **Ruest, Danielle.** *Virtualization: A Beginner's Guide.* EEUU : McGraw-Hill, 2009.

[36] **Barret, Diane.** *Virtualization and Forensics.* EEUU : Elsevier Inc., 2010.

[37] **Portnoy, Matthew.** *Virtualization Essentials.* EEUU : Sybex, 2012.

[38] **VMware, Inc. .** Understanding Full Virtualization, Paravirtualization, and Hardware Assist. [Online] 2007. [Cited: 12 12, 2014.] http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf.

[39] **Uphill, Thomas.** KVM: Kernel-based Virtual Machine. [Online] 2013. [Cited: 12 12, 2014.] https://docs.google.com/presentation/d/1ybDmE_SalaDTs70H8yBppMS2ovvBAGhIRQdsMWd1d80/edit#slide=id.g2f870fb6_0_135.

[40] **Svec, Michael.** Virtualization with KVM - SUSECon. [Online] 2012. [Cited: 12 12, 2014.] https://www.suse.com/events/susecon/sessions/presentations/SUSECon-2012-TT1523.pdf .

[41] **Red Hat, Inc.** *Virtualization Host Configuration and Guest Installation Guide.* EEUU : Red Hat, 2013.

[42] **Chisnall, David.** *The Definitive Guide to the Xen Hypervisor.* EEUU : Prentice Hall, 2008.

[43] **Takemura, Chris.** *The Book of XEN.* EEUU : No Starch Press, Inc., 2010.

[44] **IBM.** Cloud computing with KVM. [Online] 2013. [Cited: 01 01, 2015.] https://www.youtube.com/watch?v=Z2T4zORmtZg.

[45] **Open Virtualization Alliance.** Open Virtualization Alliance. [Online] 2014. [Cited: 01 02, 2015.] https://openvirtualizationalliance.org/.

[46] **IBM.** KVM Architecture: The Key Components of Open Virtualization with KVM . [Online] 2011. [Cited: 01 21, 2015.] https://www.ibm.com/developerworks/community/blogs/ibmvirtualization/entry/kvm_architecture_the_key_components_of_open_virtualization_with_kvm2?lang=en.

[47] **Libvirt.** Virtual Networking. [Online] 2013. [Cited: 01 03, 2015.] http://wiki.libvirt.org/page/VirtualNetworking.

[48] **Red Hat, Inc.** Virtualization Deployment and Administration Guide. [Online] 2014. [Cited: 02 02, 2015.] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Deployment_and_Administration_Guide/index.html .

[49] **Bormann, P.** New Manual of Seismological Observatory Practice (NMSOP-2). [Online] 2012. [Cited: 11 15, 2014.] http://bib.telegrafenberg.de/publizieren/vertrieb/nmsop/.

[50] **GFZ Potsdam, gempa GmbH.** SeisComP3 Seattle documentation. [Online] 2014. [Cited: 10 02, 2014.] http://www.seiscomp3.org/doc/seattle/2014.084/.

[51] **OSOP.** World-wide SeisComP installations. [Online] 2014. [Cited: 12 04, 2014.] http://www.osop.com.pa/world-wide-seiscomp-installations/ .

[52] **Weber, Bern.** SeiscomP3 Introduction. [Online] 2014. [Cited: 11 15, 2014.] http://ds.iris.edu/media/workshop/2014/07/managing-data-from-seismic-networks/files/presentations/29Jul2014-0830-Weber-SeisComP3%20Intropdf.pdf.

[53] —. Seiscomp3 Architecture and Workflow. [Online] 2009. [Cited: 11 15, 2014.] https://drive.google.com/file/d/0B7eg0JaxxaJVWWFVNVkyUkprSEU/view?usp=sharing.

[54] **Impress .** gempa GmbH. [Online] 2013. http://www.gempa.de/.

[55] **Lisowski, Stefan.** Earthworm Documentation. [Online] 09 12, 2014. [Cited: 10 28, 2014.] http://folkworm.ceri.memphis.edu/ew-doc/.

[56] **USGS.** Earthworm Modules. [Online] 11 14, 2012. [Cited: 10 29, 2014.] http://www.isti2.com/ew/modules.html.

[57] **Wald, David.** *ShakeMap Manual.* EEUU : USGS, 2006.

[58] **Instituto Geofísico EPN.** Instituto Geofísico EPN Helicorder. [Online] 2014. [Cited: 12 04, 2014.] http://www.igepn.edu.ec/index.php/tungurahua/helicorder-tungurahua.

[59] **Instituto Geofísico - EPN .** Instituto Geofísico - EPN - Espectrogramas. [Online] 2014. [Cited: 12 04, 2014.] http://www.igepn.edu.ec/index.php/tungurahua/espectrogramas-tungurahua.

[60] **Ubuntu.** LTS Ubuntu Wiki. [Online] 2013. [Cited: 11 25, 2014.] https://wiki.ubuntu.com/LTS.

[61] **Debian.** Debian Releases. [Online] 2014. [Cited: 11 26, 2014.] https://wiki.debian.org/DebianReleases.

[62] **The CentOS Project.** Frequently Asked Questions about CentOS in general. [Online] 2014. [Cited: 11 26, 2014.] http://wiki.centos.org/FAQ/General.

[63] **SUSE.** Product Support Lifecycle. [Online] 2014. [Cited: 11 26, 2014.] https://www.suse.com/support/policy.html.

[64] **Red Hat, Inc.** Red Hat Enterprise Linux Life Cycle. [Online] 2014. [Cited: 11 26, 2014.] https://access.redhat.com/support/policy/updates/errata.

[65] **Red Hat, Inc.** Fence Device and Agent Information for Red Hat Enterprise Linux. [Online] 2014. [Cited: 12 13, 2014.] https://access.redhat.com/articles/28603.

[66] **Western Telemati, Inc.** NPS Series Manual. [Online] 2000. [Cited: 01 27, 2015.] http://www.wti.com/guides/npsguide.pdf.

[67] **LINBIT.** Mirrored SAN vs. DRBD. [Online] 2012. [Cited: 12 12, 2014.] http://blogs.linbit.com/p/347/san-vs-drbd/.

[68] **ANONIMO.** Benchmark Results of Random I/O Performance of Different RAID Levels. [Online] 2013. [Cited: 12 12, 2014.] http://louwrentius.com/benchmark-results-of-random-io-performance-of-different-raid-levels.html.

[69] **LINBIT.** DRBD user mail list. [Online] 11 25, 2011. [Cited: 01 23, 2015.] http://lists.linbit.com/pipermail/drbd-user/2011-November/017284.html.

[70] **CentOS.** Centos Mirror. [Online] 03 31, 2015. [Cited: 01 31, 2015.] http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7.0-1406-x86_64-Everything.iso.

[71] **Cameron, Thomas.** Next-generation High Availability Linux Clustering. [Online] 01 20, 2014. [Cited: 11 21, 2014.] http://rhsummit.files.wordpress.com/2014/04/cameron_t_120_next-gen_ha_linux_clustering.pdf.

[72] **Beekhof, Andrew.** Pacemaker 1.1 Clusters from Scratch. [Online] 2012. [Cited: 11 21, 2014.] http://clusterlabs.org/doc/Cluster_from_Scratch.pdf.

[73] **Red Hat, Inc.** *Red Hat Enterprise Linux 7 Networking Guide.* EEUU : s.n., 2014.

[74] **Linux Raid Wiki.** Linux Raid Wiki Detecting, querying and testing. [Online] 10 2013. [Cited: 03 09, 2015.] https://raid.wiki.kernel.org/index.php/Detecting,_querying_and_testing.

[75] **Everett, Craig.** Software versus hardware replication for disaster recovery. [Online] 2002. [Cited: 09 25, 2014.] http://www.infostor.com/index/articles/display/146708/articles/infostor/volume-6/issue-6/features/software-versus-hardware-replication-for-disaster-recovery.html.

[76] **SUSE.** SUSE Linux Enterprise Server. [Online] 2014. [Cited: 12 01, 2014.] https://www.suse.com/products/server/.

[77] **Red Hat, Inc.** The Clustered Logical Volume Manager (CLVM). [Online] 2014. [Cited: 12 01, 2014.] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Logical_Volume_Manager_Administration/LVM_Cluster_Overview.html.

[78] —. Red Hat Enterprise Linux 5 Global Network Block Device. [Online] 2011. [Cited: 12 01, 2014.] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/5/pdf/Global_Network_Block_Device/Red_Hat_Enterprise_Linux-5-Global_Network_Block_Device-en-US.pdf.

[79] *The Totem Protocol.* **Ciarfella, P.** s.l. : IEEE , 1994, Vol. 20.

[80] **Service Availability™ Forum.** Service Availability Forum. [Online] 2014. [Cited: 12 01, 2014.] http://www.saforum.org/ .

[81] **The Linux-HA Project.** Linux High Availability. [Online] 2009. [Cited: 12 09, 2014.] http://www.linux-ha.org/wiki/Main_Page.

[82] **GitHub.** Cluster Management Shell. [Online] 2014. [Cited: 12 09, 2014.] http://crmsh.github.io/.

[83] **ANONIMO.** md man page. [Online] 2014. [Cited: 12 09, 2014.] http://linux.die.net/man/4/md.

[84] **Open-iSCSI Project.** Open-iSCSI . [Online] 2014. [Cited: 12 12, 2014.] http://www.open-iscsi.org/.

[85] **Eklektix, Inc.** A tale of two SCSI targets. [Online] 2011. [Cited: 12 12, 2014.] http://lwn.net/Articles/424004/.

[86] **VMware, Inc. .** VMware Workstation. [Online] 2014. [Cited: 12 12, 2014.] http://www.vmware.com/products/workstation.

[87] **Oracle.** Oracle VM Virtua Box. [Online] [Cited: 12 12, 2014.] https://www.virtualbox.org/.

[88] **Standard Performance Evaluation Corporation.** SPEC Virt 2013. [Online] 2013. [Cited: 01 01, 2015.] http://www.spec.org/virt_sc2013/.

[89] **Novell, Inc.** German Air Traffic Control. [Online] 2014. [Cited: 12 01, 2014.] http://www.novell.com/success/dfs.html.

[90] **Instituto Geofísico de la Escuela Politécnica Nacional.** Instituto Geofísico EPN. [Online] 2014. [Cited: 11 10, 2014.] http://www.igepn.edu.ec/.

[91] **Helmholtz-Zentrum Potsdam.** GEOFON Program. [Online] 2014. http://geofon.gfz-potsdam.de/.

[92] **Spread Concepts LLC.** The Spread Toolkit. [Online] 2014. http://www.spread.org/index.html.

[93] **IRIS.** Incorporated Research Institutions for Seismology. [Online] 2014. http://ds.iris.edu/ds/nodes/dmc/services/seedlink/.

[94] —. Data Formats. [Online] 2014. http://ds.iris.edu/ds/nodes/dmc/data/formats/.

[95] **Python Software Foundation.** Python.org. [Online] 2014. https://www.python.org/.

[96] **Google.** Google Earth. [Online] 2014. [Cited: 12 04, 2014.] https://www.google.com/earth/explore/products/plugin.html .

[97] **Zabbix SIA.** Zabbix Homepage. [Online] 2014. [Cited: 12 04, 2014.] www.zabbix.com.

[98] **Red Hat, Inc.** *Red Hat Enterprise Linux 7 Virtualization Deployment and Administration Guide.* EEUU : Red Hat, 2014.

[99] **Poulton, Nigel.** Virtualization experts debate ISCSI vs. NFS for shared storage in a virtual environment. [Online] 2014. [Cited: 12 12, 2014.] http://searchservervirtualization.techtarget.com/tip/ISCSI-vs-NFS-for-virtualization-shared-storage.

[100] **LINBIT.** Enabling your resource for the first time. [Online] 2010. [Cited: 12 12, 2014.] http://www.drbd.org/users-guide-8.3/s-first-time-up.html.

[101] **IEEE.** The Totem Redundant Ring Protocol. [Online] 2002. [Cited: 04 04, 2015.] http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1022310&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1022310.

[102] **Beekhof, Andrew.** ClusterLabs Mailing List. [Online] 03 30, 2015. http://clusterlabs.org/pipermail/users/2015-March/000169.html.

ANEXOS

ANEXO 1

1. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA SEISCOMP3

En este Anexo se presenta el procedimiento a seguir para instalar y configurar el sistema de adquisición y procesamiento SeisComP3.

1.1 INSTALACIÓN

1.1.1 REQUISITOS

El sistema puede instalarse sobre cualquier servidor físico o virtual con procesadores de 32 o 64 bits, para ejecutar ciertos módulos es necesario disponer de una licencia otorgada por la empresa desarrolladora del software Gempa. Los requisitos más importantes se muestran a continuación:

1.1.1.1 Sistema Operativo

SeisComP3 puede ejecutarse únicamente sobre un sistema operativo basado en Linux, en la página de descargas del programa [1], se encuentran versiones para los sistemas operativos Linux más utilizados tales como: Ubuntu, CentOS, Debian, Fedora, OpenSUSE, etc., por lo general existen instaladores para las versiones de 32 y 64 bits de los sistemas operativos listados, así como versiones para los *releases* más recientes.

1.1.1.2 Base de datos

SeisComP3 es compatible con dos servidores de base de datos: MySQL y PostgreSQL, el programa cuenta con scripts que instalan uno de estos servidores y que crean la base de datos que SeisComP3 necesita. Si ya existe un servidor de base de datos instalado es necesario conocer la clave de administrador de la base de datos.

1.1.1.3 Requerimientos de red

Los módulos que forman SeisComP3 necesitan de varios puertos de red para poder comunicarse entre sí y aceptar peticiones de clientes. En la Tabla 1.7 se presentan los puertos necesarios cuando se utiliza la configuración por defecto, además los diferentes tipos de sensores sísmicos que envían datos al sistema utilizan puertos UDP o TCP que dependen del fabricante.

Módulo	Puerto
`spread`	TCP/4803
`Seedlink`	TCP/18000
`Arclink`	TCP/18001
`GDS`	TCP/18008

Tabla 1.1. Puertos TCP utilizados por los módulos de SeisComP3

1.1.1.4 Paquete de Instalación

El paquete de instalación de SeisComP3 está disponible para descargarlo desde la URL [1]. La primera vez que se descarga el programa, la página web solicitará llenar un pequeño formulario antes de proceder a la descarga.

La versión de SeisComP3 a descargar depende del sistema operativo que se elija y del estado del proyecto, por ejemplo un archivo con el nombre `Seiscomp3-seattle-2013.274.01-centos5.3-i686.tar.gz`, corresponde a la distribución SeisComP3 Seattle liberada el día 274 del año 2013, para el sistema operativo CentOS 5.3 de 32 bits.

1.1.2 INSTALACIÓN DEL SOFTWARE

Para el proceso de instalación se utilizará el sistema operativo Ubuntu versión 12.04 de 64 bits.

Como se indica en la Línea de Comandos 1.1, el instalador de SeisComP3 se descarga en la carpeta $HOME, se descomprime el archivo descargado lo que creará la carpeta $HOME/seiscomp3 que contiene los ejecutables, librerías, archivos de configuración, etc.

```
$ wget
http://www.seiscomp3.org/downloader/download/file/1192 -O
$HOME

$ tar xzvf $HOME/Seiscomp3-seattle-2013.245.02-gempa-
2013.267.01-ubuntu12.04-x86_64.tar.gz
```

Línea de Comandos 1.1 Descomprimir el instalador de Seiscomp3

El siguiente paso es ejecutar los scripts para instalar las dependencias necesarias, y habilitar el servicio de base de datos, tal como se indica en la Línea de Comandos 1.2.

```
$ cd $HOME/seiscomp3/share/deps/Ubuntu/12.04/

$ sudo sh install-base.sh

$ sudo sh install-gui.sh

$ sudo sh install-mysql-server.sh

# systemctl enable mariadb

# systemctl start mariadb
```

Línea de Comandos 1.2 Instalar las dependencias necesarias

En la Línea de Comandos 1.3 se presenta el siguiente paso que consiste en modificar las variables de entorno de la consola de Linux para que el sistema sepa donde están los programas y librerías que necesita para ejecutarse.

```
$ $HOME/seiscomp3/bin/seiscomp print env >> $HOME/.bashrc
```

Línea de Comandos 1.3 Modificar las variables de entorno

1.2 CONFIGURACIÓN GENERAL

Es posible configurar el sistema empleando la línea de comandos o mediante una interfaz gráfica que es el procedimiento que se presenta en esta sección.

Para iniciar la configuración se ejecuta el módulo de configuración `scconfig`, con el comando que se presenta en la Línea de Comandos 1.4.

```
$ scconfig
```

Línea de Comandos 1.4 Ejecutar el módulo de configuración de SeisComP3

La primera vez que el comando `scconfig` se ejecuta, el módulo presenta el mensaje que puede verse en la Figura 1.1. Este mensaje indica que es la primera vez que se ejecuta el módulo de configuración en el servidor actual, al hacer clic en *Yes* indicamos que queremos iniciar el proceso de configuración inicial.

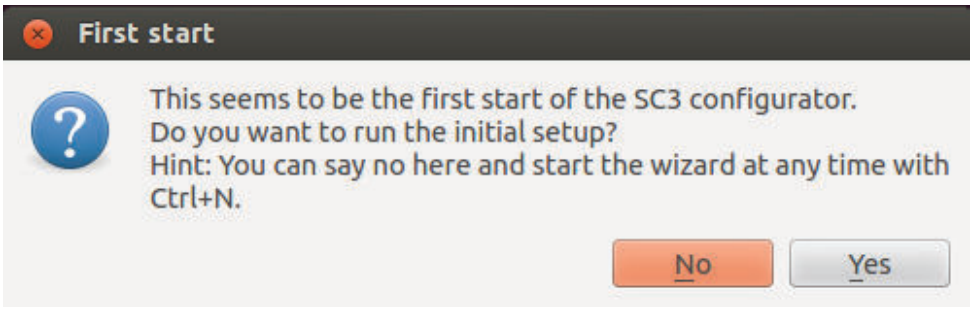

Figura 1.1 Mensaje de configuración inicial

El proceso de configuración empieza con un mensaje de introducción, que se presenta en la Figura 1.2, este mensaje nos indica que podemos utilizar los botones *Back* y *Next* para navegar avanzar o volver hacia atrás y corregir algún parámetro ingresado incorrectamente, al hacer clic en *Next* se puede empezar con el ingreso de parámetros.

Figura 1.2 Ventana inicial de scconfig

Figura 1.3: En el campo *Agency ID* escribir las siglas que identifican a la agencia de datos sismológicos, en este caso IGEPN.

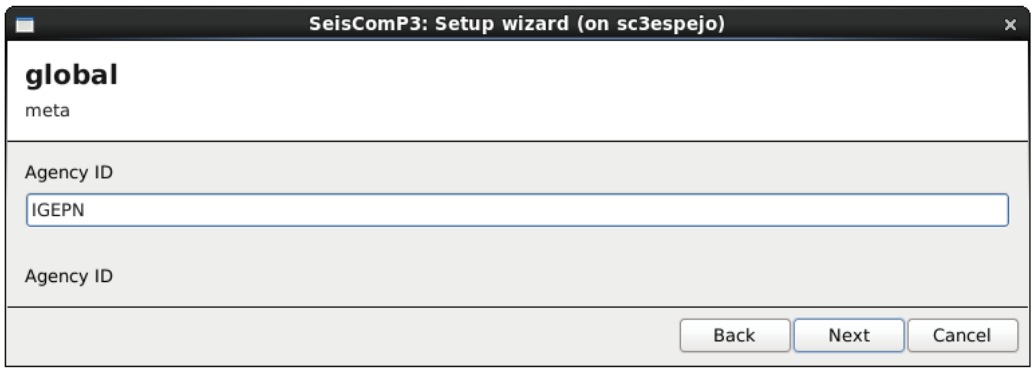

Figura 1.3 Cuadro de diálogo para ingresar el campo Agency ID

Figura 1.4: En el campo *Datacenter ID* escribir las siglas que identifican al centro sismológico, en este caso también IGEPN. Clic en *Next.*

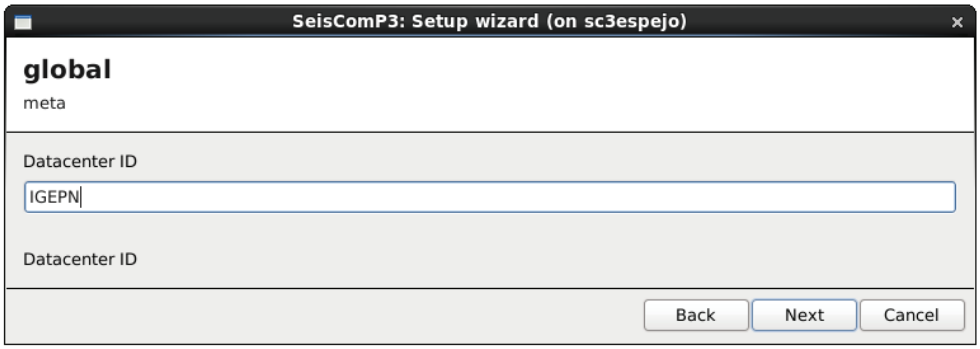

Figura 1.4 Cuadro de diálogo para ingresar el campo Datacenter ID

Figura 1.5: En el campo *Organization String* escribir las siglas que identifican a la organización, en este caso igepn. Clic en *Next.*

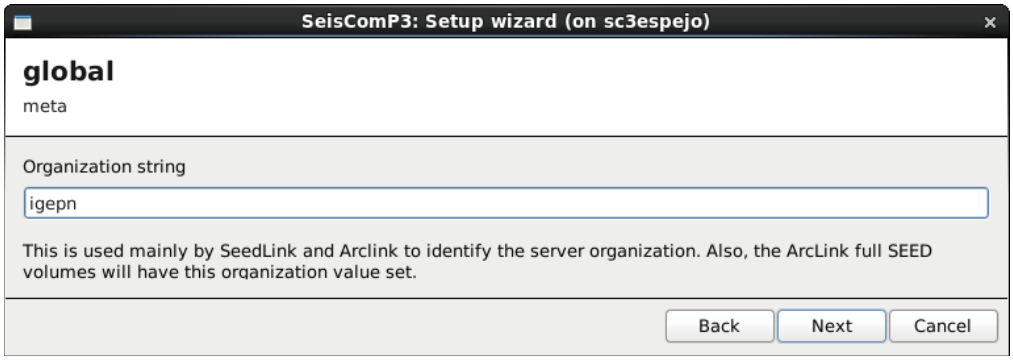

Figura 1.5 Cuadro de diálogo para ingresar el campo Organization String

Figura 1.6: Seleccionar la casilla *Enable database storage* si se desea que SeisComP3 almacene su información en una base de datos. Para un sistema que realice procesamiento de los datos adquiridos es indispensable que se habilite esta opción; si solo se utilizará las herramientas para monitorizar y revisar sismos esta opción no es necesaria. Clic en *Next.*

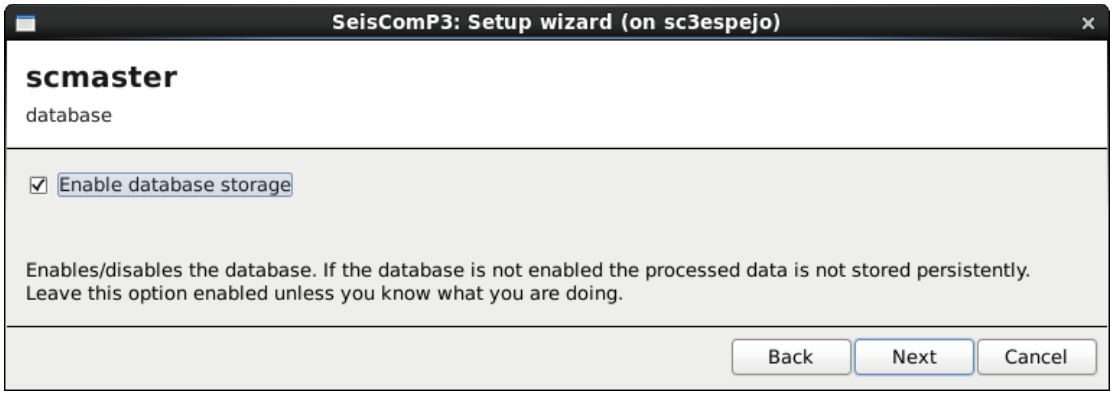

Figura 1.6 Cuadro de diálogo para habilitar el uso de la base de datos

Figura 1.7: Si se habilitó la opción *Enable database storage* en la Figura 1.7, la siguiente ventana tiene dos casillas para elegir MySQL o PostgreSQL como servidor de base de datos. Por defecto SeisComP3 trabaja con MySQL. Clic en *Next.*

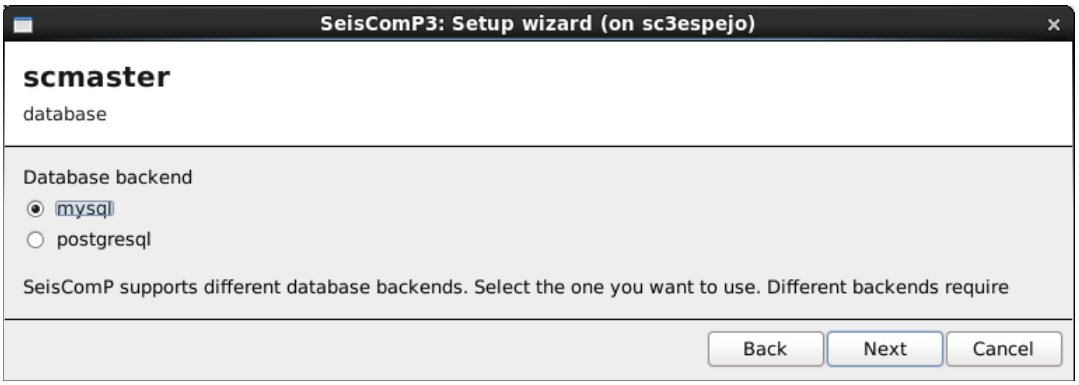

Figura 1.7 Cuadro de diálogo para elegir el tipo de base de datos a utilizar

Figura 1.8: Seleccionar la casilla *Create Database* para crear la base de datos SeisComP3, de lo contrario el sistema asume que existe una base de una instalación anterior. Clic en *Next.*

Figura 1.8 Cuadro de diálogo para confirmar la creación de la base de datos

Figura 1.9: En el cuadro de texto ingresar la clave de administrador del servidor MySQL. Clic en *Next.*

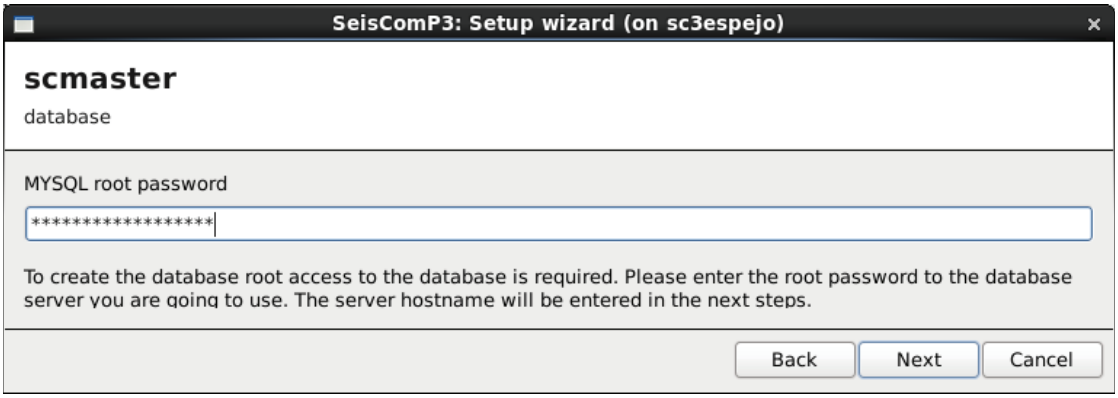

Figura 1.9 Cuadro de diálogo para ingresar la clave de root de MYSQL

Figura 1.10: Al hacer clic en la casilla *Drop existing database* se descarta una base SeisComP3 y los datos que la misma contenga, creándose una nueva completamente vacía, se recomienda nunca seleccionar esta opción. Clic en *Next*.

Figura 1.10 Cuadro de diálogo para descartar una base de datos existente

Figura 1.11: En el cuadro de texto ingresar el nombre de la base de datos que SeisComP3 utilizará, el valor por defecto es `seiscomp3`. Clic en *Next.*

Figura 1.11 Cuadro de diálogo para asignar el nombre a la base de datos

Figura 1.12: En el campo de texto ingresar el nombre del servidor en el que se encuentra la base de datos a la que SeisComP3 se conectará, el valor por defecto es `localhost`. Clic en *Next.*

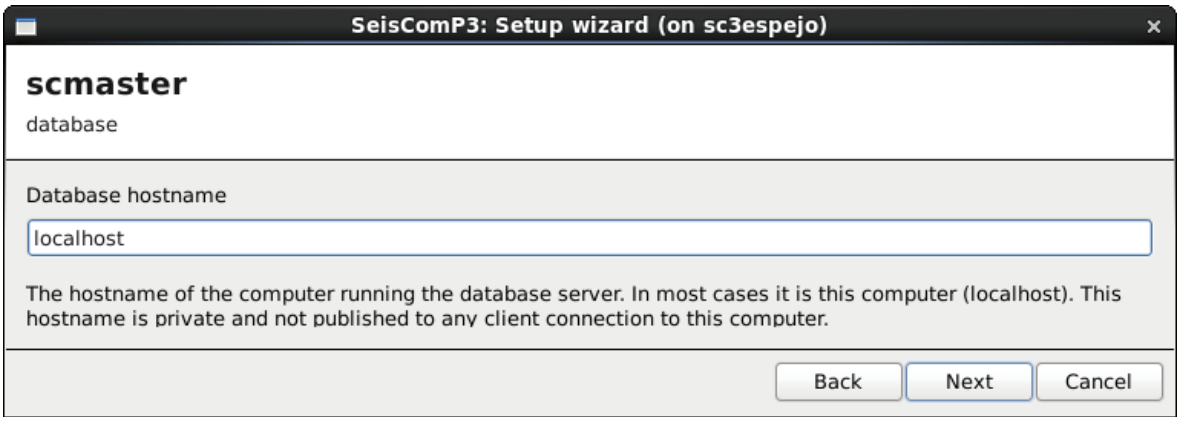

Figura 1.12 Cuadro de diálogo para asignar el servidor de la base de datos

Figura 1.13 y Figura 1.14: En el campo de texto ingresar el nombre del usuario con el que SeisComP3 se conectará a la base de datos, el sistema solicita posteriormente la clave de lectura y escritura que el usuario tendrá. Clic en *Next.*

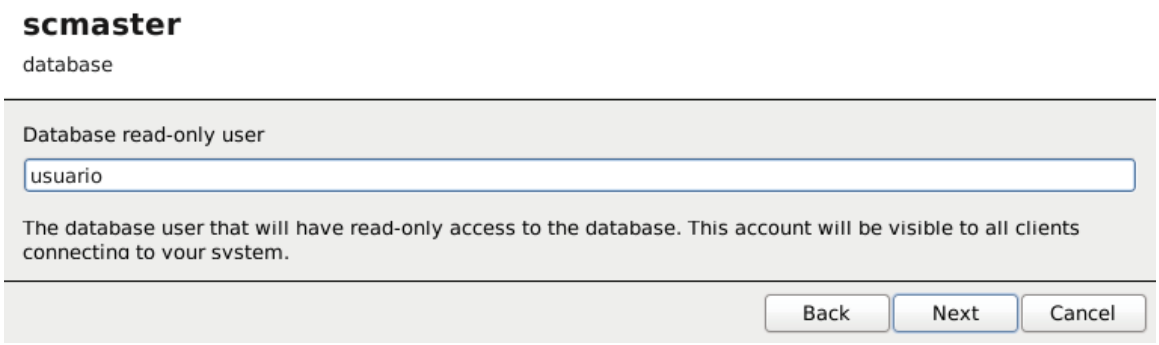

Figura 1.13 Cuadro de diálogo para asignar el usuario de la base de datos

Figura 1.14 Cuadro de diálogo para asignar la clave al usuario de la base de datos

Figura 1.15: En este cuadro de diálogo es posible finalizar el proceso de instalación o volver atrás para corregir algún valor. Clic en *Finish*.

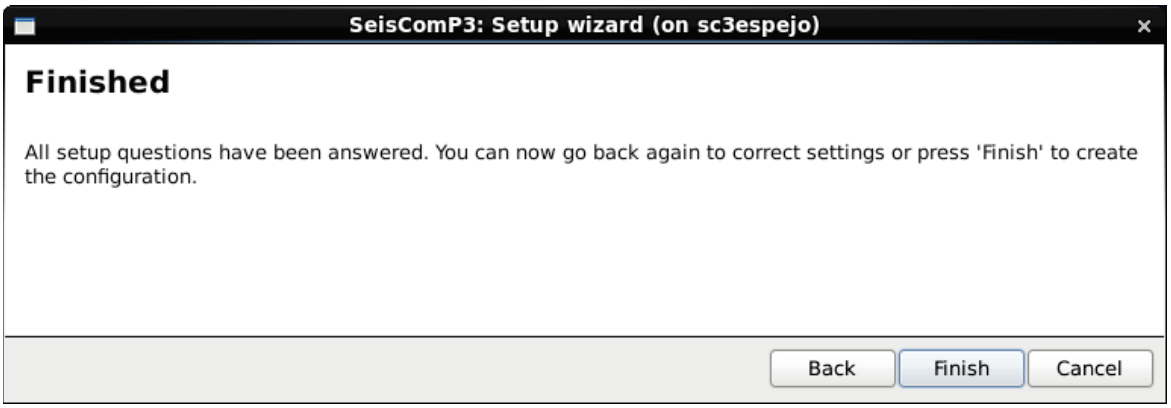

Figura 1.15 Cuadro de diálogo para finalizar la configuración general

Figura 1.16: Terminada la configuración se muestran los resultados del procedimiento, si todo se configuró correctamente el mensaje que aparece es *Setup ran successfully*. Si existe un error es posible regresar y corregir algún parámetro ingresado erróneamente.

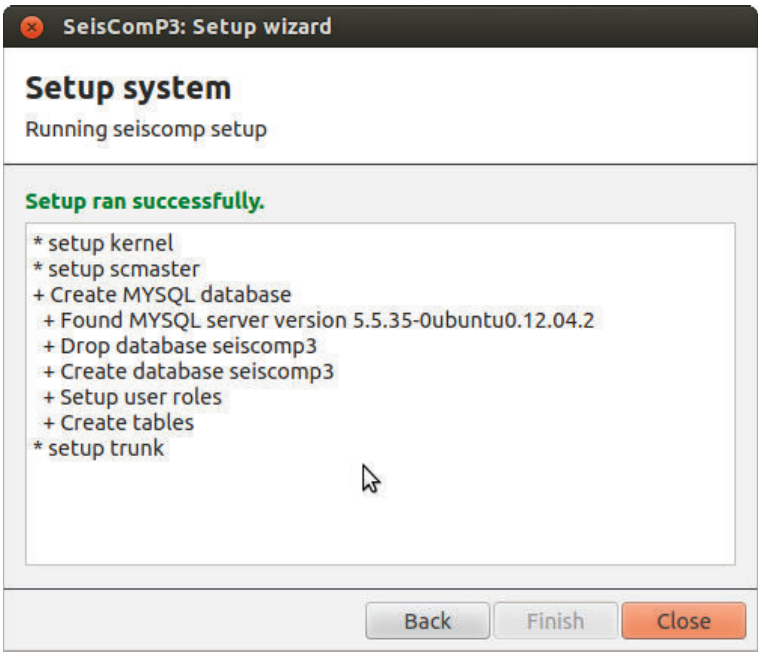

Figura 1.16 Configuración exitosa

Si se hace clic en *Close* la siguiente ventana en mostrarse es la de configuración de los módulos de SeisComP3 que puede verse en la Figura 1.17.

Una vez terminada la configuración general es necesario empezar a configurar cada uno de los módulos dependiendo del uso que se le quiera dar al sistema, así

por ejemplo si se utilizará para realizar procesamiento manual de los sismos ocurridos o visualización de formas de onda basta con iniciar los módulos `scolv`, `scesv`[145], etc.

Si se empleará SeisComP3 como un sistema de adquisición es necesario configurar primero los módulos `SeedLink` y `ArcLink` que se encargan de la adquisición y almacenamiento respectivamente. Si se desea que el sistema también procese los datos que ingresan se necesita configurar los módulos: `scautoloc`, `scautopick`, `scamp`, etc.

A continuación se muestran los pasos para configurar un sistema que realizará adquisición y procesamiento de formas de onda para detectar eventos sísmicos, para la configuración se utilizará la línea de comandos y la interfaz gráfica `Scconfig`, por lo que es necesario revisar de manera breve las características más importantes de esta interfaz de configuración.

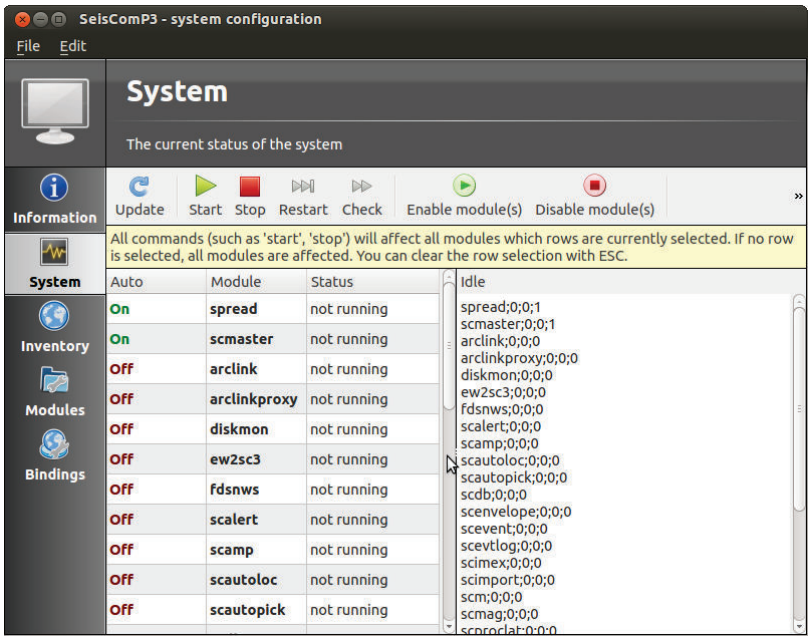

Figura 1.17 Módulo de configuración scconfig

[145] Módulo que muestra información sobre el último evento sísmico detectado.

1.2.1 SCCONFIG

Es una interfaz gráfica que permite configurar los módulos de SeisComP3 y definir las estaciones de las que el programa obtendrá las formas de onda.

`Scconfig` tiene como principales tareas:

- Iniciar/detener/monitorizar los módulos que forman SeisComP3.
- Importar metadatos de las estaciones y guardarlos en la base de datos.
- Configuración de los módulos de SeisComP3.
- Configuración de estaciones sísmicas.

Un modulo puede tener dos tipos distintos de configuración:

- Configuración del módulo o programa: consiste en la configuración de los parámetros con los que un programa de SeisComP3 se ejecuta.
- Vinculación con una estación (*Binding station*): establece la forma en que un módulo se relacionará con una o varias estaciones.

1.2.1.1 Descripción del modulo scconfig

Para iniciar `scconfig` se ejecuta el comando presentado en la Línea de Comandos 1.4. A continuación aparecerá la ventana principal de `Scconfig` que puede verse en la Figura 1.18, la ventana del programa se divide en cuatro secciones:

- Modo usuario o sistema, en rojo.
- Lista de Paneles, en amarillo.
- Título y descripción del panel elegido, en verde.
- Contenido del panel seleccionado, en azul.

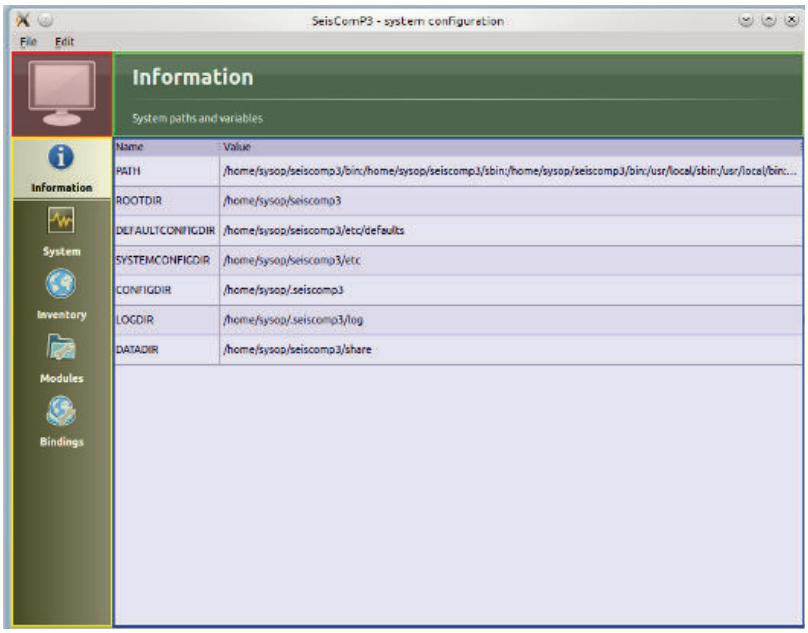

Figura 1.18 Secciones del módulo scconfig

En la parte superior existe una barra de herramientas con las opciones *File* y *Edit*. El menú *File* tiene las siguientes opciones:

- *Wizard*: ejecuta el asistente de configuración inicial.
- *Reload*: recarga la configuración de los módulos y estaciones desde los archivos de configuración.
- *Save*: guarda la configuración de todos los módulos.
- *Quit*: salir del módulo de configuración.

1.2.2 PANELES DEL MÓDULO SCCONFIG

En la parte izquierda se encuentran los paneles *Information*, *System*, *Inventory*, *Modules* y *Bindings*, cada una representa la información y configuraciones que pueden realizarse, en la parte central puede verse la información del panel seleccionado.

`Scconfig` no interactúa con la base de datos ni con su contenido, a excepción del panel *Inventory*, el resto de paneles permiten únicamente leer y escribir las

configuraciones que se encuentran en las carpetas `$SEISCOMP_ROOT/etc` o `$HOME/.seiscomp3`, que es donde se almacenan los archivos de configuración de los módulos y de las estaciones.

1.2.2.1 Information

El panel *Information*, que puede verse en la Figura 1.19, presenta los valores de algunas variables de entorno que SeisComP3 utiliza, por ejemplo en la variable `LOGDIR` se almacenan por defecto los log que generan todos los programas de SeisComP3, `CONFIGDIR`, `SYSTEMCONFIGDIR` y `DEFAULTCONFIGDIR` contienen los archivos de configuración de los módulos, `ROOTDIR` es la carpeta en la que está instalado todo el sistema, `PATH` es la carpeta que contiene todos los módulos y librearías necesarias, etc.

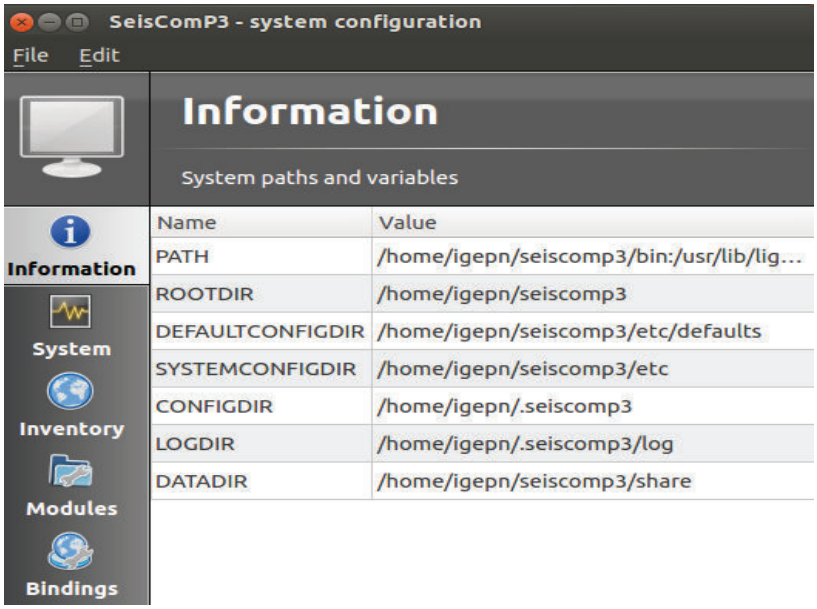

Figura 1.19 Panel Information

1.2.2.2 System

El panel *System* presenta el estado de los módulos de SeisComP3, en la Figura 1.19 puede verse en detalle esta sección.

El contenido de este panel se divide en tres partes, la barra de herramientas en rojo, las listas de módulos, en verde y la ventana de log, en azul.

La barra de herramientas permite iniciar o detener un módulo, así como habilitar o deshabilitarlo, es decir si arranca o no de forma automática junto con el sistema.

Tiene también un botón para que los cambios en la configuración se escriban en los correspondientes archivos de configuración.

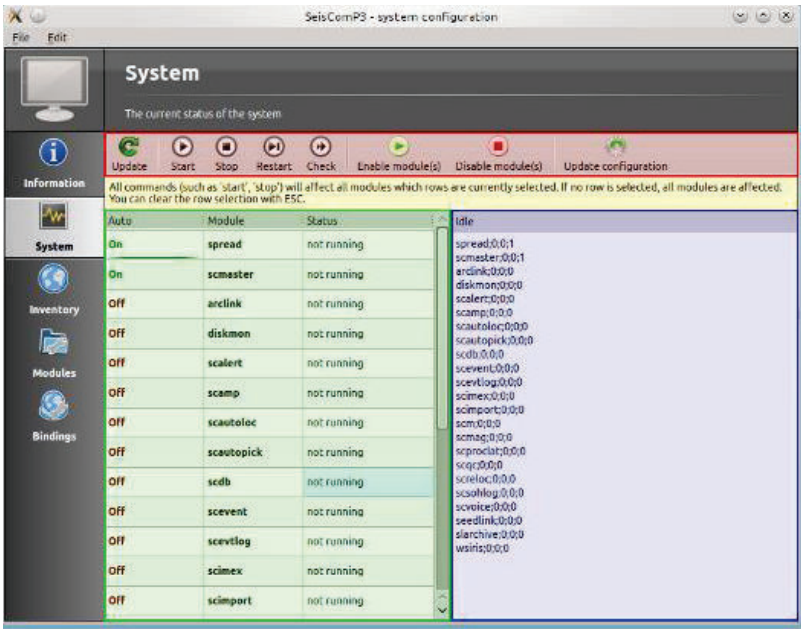

Figura 1.20 Panel System

1.2.2.3 Inventory

El panel *Inventory* se presenta en la Figura 1.21, este panel permite importar y sincronizar la lista de sensores sísmicos. En la parte central se muestra una lista de archivos XML que contienen la información de los sensores, tal como tipo de sensor, sensibilidad, latitud, longitud, etc. Estos archivos XML se encuentran en la carpeta `$SEISCOMP_ROOT/etc/inventory`.

La barra de herramientas de esta sección permite realizar operaciones como importar un inventario, realizar una sincronización de prueba, sincronizar el archivo XML con la base de datos de SeisComP3, etc.

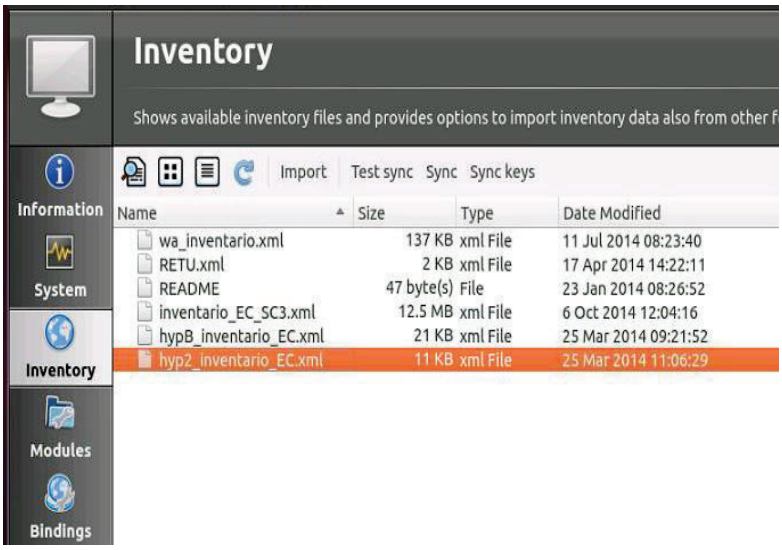

Figura 1.21 Panel Inventory

1.2.2.4 Modules

El panel *Modules* se presenta en la Figura 1.22, este panel permite realizar la configuración de los módulos que forman SeisComP3.

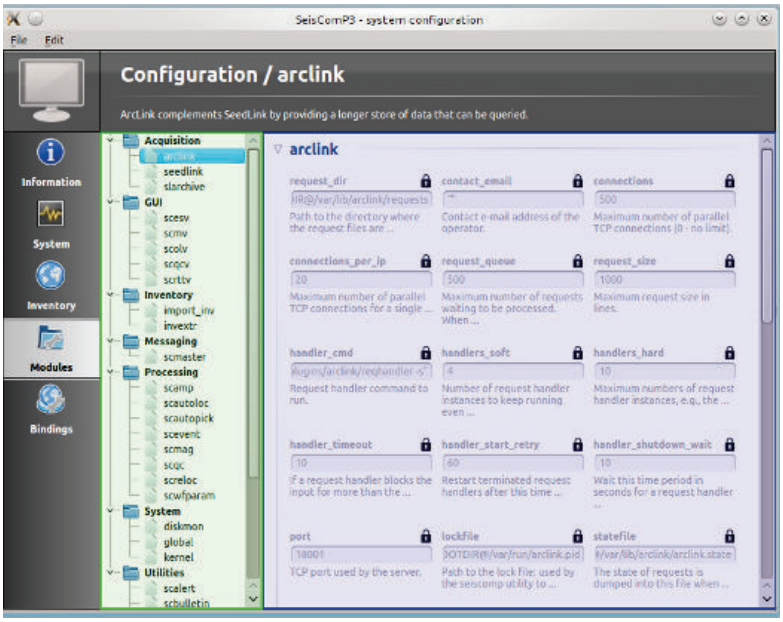

Figura 1.22 Sección Modules

La parte coloreada de verde indica la lista de los módulos agrupados por categorías, mientras que en la parte azul indica los parámetros que pueden configurarse para el modulo seleccionado.

1.2.2.5 Bindings

Como se indica en la Figura 1.23, el panel *Bindings* se divide en tres secciones principales, la sección de estaciones (color rojo y naranja), el contenido del *binding* (color verde) y la parte de módulos (en color azul y violeta).

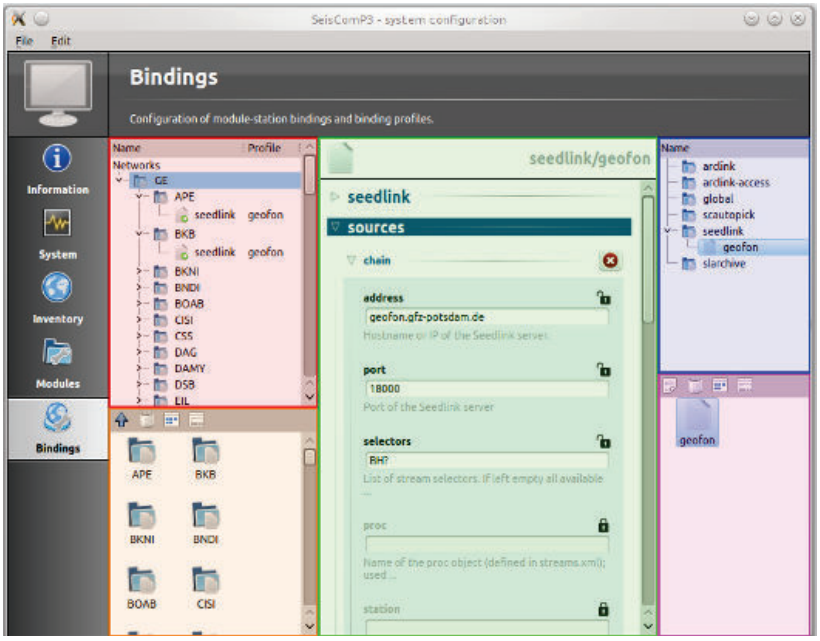

Figura 1.23 Sección Bindings

La sección de estaciones lista todas las redes y estaciones a las que SeisComP3 se conecta para obtener las formas de onda, al hacer clic en el nombre de una red, la sección naranja presenta todas las estaciones de esa red, al hacer clic en el nombre de la estación la sección naranja presenta las vinculaciones que la estación tiene con determinados módulos.

El contenido de la sección *binding* permite configurar mediante algunos parámetros la forma en que la estación se relaciona con el módulo con la que está enlazada.

La sección de módulos (azul) contiene todos los módulos con los que una estación puede realizar vinculaciones, la parte inferior (morado) muestra los perfiles de vinculación para el módulo seleccionado, en esta sección se puede agregar nuevos perfiles o eliminar perfiles existentes.

1.3 PERFILES DE SEISCOMP3

SeisComP3 permite realizar la configuración de las estaciones de forma gráfica mediante el uso de perfiles, de esta forma se facilita el agregar nuevas estaciones o cambiar los parámetros de las mismas.

Un perfil es un archivo en texto plano al que se asigna un nombre y que contiene un conjunto de parámetros que puede aplicarse a una o varias estaciones, lo que facilita configurar un gran número de estaciones. Los módulos que permiten la creación de perfiles son:

- `Arclink`
- `Arclink-access`
- `Seedlink`
- `Slarchive`
- `Scautopick`
- `Global`

El perfil `Global` no está ligado a ningún módulo sino que permiten configurar el canal de datos que `SeedLink` debe detectar.

Por ejemplo un perfil para el módulo `scautopick` denominado `perfilVolcan` se asociaría con las estaciones volcánicas, mientras que otro perfil `perfilSismico` se asociaría a las estaciones sísmicas.

Los perfiles se crean gráficamente utilizando la sección *Bindings* del programa `scconfig`, es necesario primero configurar los módulos de SeisComP3 antes de crear cualquier perfil.

1.4 CONFIGURACIÓN DE LOS MÓDULOS DE COMUNICACIÓN

Para que el sistema de comunicación de SeisComP3 funcione correctamente deben configurarse los módulos que se muestran a continuación.

1.4.1 SCMASTER

Los parámetros más importantes de este módulo están relacionados con la base de datos que SeisComP3 utiliza y se presentan en la Tabla 1.2.

Parámetro	Valor por defecto
`plugins`	`dbplugin`
`Core.plugins`	`dbmysql`
`plugins.dbPlugin.dbDriver`	`mysql`
`plugins.dbPlugin.readConnection`	`usuario:clave@localhost/seiscomp3`
`plugins.dbPlugin.writeConnection`	`usuario:clave@localhost/seiscomp3`

Tabla 1.2 Parámetros principales del módulo scmaster

La configuración de este módulo se realiza con los comandos que se presentan en la Línea de Comandos 1.5.

1.4.2 SPREAD

Este módulo no tiene parámetros para configurar, sin embargo el puerto TCP 4803 que el módulo utiliza debe estar disponible y abierto en el *firewall*.

1.5 CONFIGURACIÓN DE LOS MÓDULOS DE ADQUISICIÓN

En esta sección se presenta la configuración de los módulos `seedlink`, `arclink` y `slarchive`, necesarios para que el servidor SeisComP3 realice la adquisición y almacenamiento de las formas de onda que se obtienen de los diferentes sensores sísmicos.

```
$ echo " plugins = dbplugin" >>
$SEISCOMP_ROOT/etc/scmaster.cfg

$ echo "core.plugins = dbmysql " >>
$SEISCOMP_ROOT/etc/scmaster.cfg

$ echo "plugins.dbPlugin.dbDriver = mysql" >>
$SEISCOMP_ROOT/etc/scmaster.cfg

$ echo "plugins.dbPlugin.readConnection =
test:test@localhost/seiscomp3 " >>
$SEISCOMP_ROOT/etc/scmaster.cfg

$ echo "plugins.dbPlugin.writeConnection=
test:test@localhost/seiscomp3 " >>
$SEISCOMP_ROOT/etc/scmaster.cfg

$ seiscomp update-config

$ seiscomp restart
```

Línea de Comandos 1.5 Configuración del módulo scmaster

1.5.1 SEEDLINK

Los parámetros más importantes de este módulo son:

- *Port*: define el puerto en el que el módulo acepta solicitudes de datos, si se cambia el valor por defecto es necesario cambiar este valor en el resto de módulos.

- *Filebase*: es el directorio en el que se almacenarán los datos de forma temporal, el módulo crea automáticamente tantas carpetas como estaciones ingresen al sistema.

- *Msrtsimul*: esta opción se utiliza para alimentar al sistema Seiscomp3 con formas de onda antiguas y no con las formas de onda que ingresan en tiempo real.

- *Inventory_connection*: define la conexión con la base de datos para obtener la descripción de las estaciones.

- *Connections*: este parámetro especifica el número de conexiones TCP/IP permitidas.

En la Tabla 1.3 se presentan estos parámetros y sus valores por defecto.

Parámetro	Valor por defecto
port	1800
filebase	@ROOTDIR@/var/lib/seedlink/buffer
msrtsimul	uncheck
Inventory_connection	mysql://usuario:clave@localhost/seiscomp3
connections	500

Tabla 1.3 Parámetros principales del módulo seedlink

1.5.1.1 Configuración

El módulo `SeedLink` utiliza la configuración por defecto, con excepción del parámetro `filebase` que se configura con el comando que indica la Línea de Comandos 1.6.

```
$ echo "filebase = /data/seedlink/buffer" >>
$SEISCOMP_ROOT/etc/seedlink.cfg

$ seiscomp update-config

$ seiscomp restart seedlink
```

Línea de Comandos 1.6 Configuración del módulo seedlink

1.5.2 SLARCHIVE

Los parámetros más importantes de este módulo son:

- *Address*: este parámetro indica la dirección IP del servidor en el que se ejecuta el módulo `seedlink`.
- *Port*: indica el puerto en el que el servidor `seedlink` atiende solicitudes de datos y al que `slarchive` se conectará para solicitar esos datos.
- *Archive*: con este parámetro se configura la carpeta en la que se almacenarán los archivos `mseed` de las formas de onda de todos los sensores sísmicos que ingresan al sistema.

En la Tabla 1.4 se presentan estos parámetros y sus valores por defecto.

Parámetro	Valor por defecto
Address	127.0.0.1
port	18000
archive	/var/lib/archive

Tabla 1.4 Parámetros principales del módulo slarchive

1.5.2.1 Configuración

Si no se realizó ningún cambio en la configuración del módulo `seedlink`, el módulo `slarchive` utilizará la configuración por defecto, con excepción del parámetro `archive`, cuya configuración se indica en la Línea de Comandos 1.7.

```
$ echo "archive = /data/archive/" >>
$SEISCOMP_ROOT/etc/slarchive.cfg

$ seiscomp update-config

$ seiscomp restart slarchive
```

Línea de Comandos 1.7 Configuración del módulo slarchive

1.5.3 ARCLINK

Los parámetros más importantes de este módulo son:

- *Port*: indica el puerto en el que el módulo `arclink` atiende solicitudes de datos del resto de módulos.
- *Nrtdir*: este parámetro configura la carpeta en donde se almacenan las formas de onda de las estaciones y que `arclink` lee para compartirlas con el resto de módulos.

En la Tabla 1.5 se presentan estos parámetros y sus valores por defecto.

Parámetro	Valor por defecto
port	18001
nrtdir	@ROOTDIR@/var/lib/archive

Tabla 1.5 Parámetros principales del módulo arclink

1.5.3.1 Configuración

El módulo `arclink` utilizará la configuración por defecto, con excepción del parámetro `nrtdir` cuya configuración se indica en la Línea de Comandos 1.8.

```
$ echo " nrtdir=/var/lib/archive " >>
$SEISCOMP_ROOT/etc/arclink.cfg

$ seiscomp update-config

$ seiscomp restart arclink
```

Línea de Comandos 1.8 Configuración del módulo arclink

1.6 CONFIGURACIÓN DE LOS MÓDULOS DE PROCESAMIENTO

Los módulos de procesamiento requieren configurar parámetros relacionados con filtros, magnitudes sísmicas, ecuaciones, etc. Estos valores fueron cuidadosamente elegidos por un equipo de sismólogos, y su explicación está fuera del alcance del presente Proyecto de Titulación, por lo que se indicará únicamente los comandos usados para configurar dichos módulos.

1.6.1 SCAUTOPICK

La configuración de los parámetros de este módulo se presenta en la Línea de Comandos 1.9

```
$ echo "amplitudes = MLv,ML,mb,mB,Mwp" >>
$SEISCOMP_ROOT/etc/scautopick.cfg

$ echo "picker = AIC" >> $SEISCOMP_ROOT/etc/scautopick.cfg

$ echo "spicker = S-L2" >> $SEISCOMP_ROOT/etc/scautopick.cfg

$ seiscomp update-config

$ seiscomp restart scautopick
```

Línea de Comandos 1.9 Configuración del módulo scautopick

1.6.2 SCAUTOLOC

La configuración de los parámetros de este módulo se presenta en la Línea de Comandos 1.10

```
$ echo "autoloc.grid =
@CONFIGDIR@/scautoloc/grid_local.conf" >>
$SEISCOMP_ROOT/etc/scautoloc.cfg

$ echo "autoloc.stationConfig =
@CONFIGDIR@/scautoloc/station_local.conf " >>
$SEISCOMP_ROOT/etc/scautoloc.cfg

$ echo "autoloc.xxl.minAmplitude = 10 " >>
$SEISCOMP_ROOT/etc/scautoloc.cfg

$ echo "autoloc.xxl.minSNR = 3" >>
$SEISCOMP_ROOT/etc/scautoloc.cfg

$ seiscomp update-config

$ seiscomp restart scautoloc
```

Línea de Comandos 1.10 Configuración del módulo scautoloc

1.6.3 SCAMP

La configuración de los parámetros de este módulo se presenta en la Línea de Comandos 1.11

```
$ echo “connection.subscriptions =
PICK,PICK2,AMPLITUDE,AMPLITUDE2,LOCATION” >>
$SEISCOMP_ROOT/etc/scamp.cfg

$ echo “amplitudes = MLv,mb,mB,Mwp,ML,Mjma” >>
$SEISCOMP_ROOT/etc/scamp.cfg

$ seiscomp update-config

$ seiscomp restart scamp
```

Línea de Comandos 1.11 Configuración del módulo scamp

1.6.4 SCEVENT

La configuración de los parámetros de este módulo se presenta en la Línea de Comandos 1.12.

```
$ echo “eventIDPrefix = "igepn"” >>
$SEISCOMP_ROOT/etc/scevent.cfg

$ echo “eventAssociation.maximumMatchingArrivalTimeDiff = 3”
>> $SEISCOMP_ROOT/etc/scevent.cfg

$ echo “eventAssociation.minimumDefiningPhases = 4” >>
$SEISCOMP_ROOT/etc/scevent.cfg

$ echo “eventAssociation.maximumTimeSpan = 30” >>
$SEISCOMP_ROOT/etc/scevent.cfg

$ seiscomp update-config

$ seiscomp restart scevent
```

Línea de Comandos 1.12 Configuración del módulo scevent

1.6.5 SCMAG

La configuración de los parámetros de este módulo se presenta en la Línea de Comandos 1.13.

```
$ echo "connection.subscriptions = PICK,PICK2,AMPLITUDE" >>
$SEISCOMP_ROOT/etc/scmag.cfg

$ echo "magnitudes = MLv,mb,mB,Mwp,ML,Mjma" >>
$SEISCOMP_ROOT/etc/scmag.cfg

$ seiscomp update-config

$ seiscomp restart scmag
```

Línea de Comandos 1.13 Configuración del módulo scmag

1.7 CREACIÓN DE PERFILES

Para que la información que un sensor captura pueda adquirirse, almacenarse y procesarse es necesario crear algunos perfiles que relacionen el sensor sísmico con los módulos correspondientes.

1.7.1 PERFIL GLOBAL

El procedimiento para crear un perfil global para un sensor se presenta en la Línea de Comandos 1.14.

```
$ mkdir -p seiscomp3/etc/key/global/

$ echo "detecStream = HN" >>
$SEISCOMP_ROOT/etc/key/global/profile__HN

$ seiscomp update-config
```

Línea de Comandos 1.14 Configuración de un perfil global

1.7.2 PERFIL SEEDLINK.

El siguiente perfil a crear será el del módulo `seedlink`, existen diferentes tipos de perfiles dependiendo del tipo de sensor sísmico, en la Línea de Comandos 1.15 se muestra un perfil que solicita datos a un servidor `seedlink` existente cuya dirección IP es 192.168.1.16 y que acepta peticiones de datos en el puerto 18013

```
$ mkdir -p seiscomp3/etc/key/seedlink/

$ echo "sources = scauto2:chain">>
seiscomp3/etc/key/seedlink/profile_scauto2

$ echo “sources.scauto2.address= 192.168.1.16” >>
$SEISCOMP_ROOT/etc/key/seedlink/profile_scauto2

$ echo “sources.scauto2.port = 18013” >>
$SEISCOMP_ROOT/etc/key/seedlink/profile_scauto2

$ seiscomp update-config
```

Línea de Comandos 1.15 Configuración de un perfil seedlink

1.7.3 PERFIL SLARCHIVE

Si se desea almacenar las formas de onda que `SeedLink` adquiere es necesario crear un perfil `slarchive`, como se indica en la Línea de Comandos 1.16.

```
$ echo “keep = 120” >>
$SEISCOMP_ROOT/etc/key/slarchive/profile_archive_EC

$ seiscomp update-config
```

Línea de Comandos 1.16 Configuración de un perfil slarchive

1.7.4 PERFIL SCAUTOPICK

En la Línea de Comandos 1.17 se indica el procedimiento para crear un perfil de este tipo.

```
$ echo "picker.AIC.filter = BW(4,4,20)" >>
$SEISCOMP_ROOT/etc/key/scautopick_local/profile_local_100

$ echo "spicker.L2.filter = BW(4,4,20)" >>
$SEISCOMP_ROOT/etc/key/scautopick_local/profile_local_100

$ echo "spicker.L2.detector = STALTA(0.2,10)" >>
$SEISCOMP_ROOT/etc/key/scautopick_local/profile_local_100

$ seiscomp update-config
```

Línea de Comandos 1.17 Configuración de un perfil scautopick

1.8 AGREGAR UNA ESTACIÓN EN SEISCOMP3

Una vez configurados los módulos y creados los perfiles ya es posible agregar un sensor sísmico o estación al sistema, con el procedimiento que se indica en la Línea de Comandos 1.18, la estación `AAM1` estará vinculada con los perfiles que acabamos de crear.

```
##Vincular el perfil global##

$ echo “global:_HN” >> $SEISCOMP_ROOT/etc/key/station_EC_AAM1

##Vincular el perfil seedlink##

$ echo “seedlink:scauto2” >> $SEISCOMP_ROOT/etc/key/station_EC_AAM1

##Vincular el perfil slarchive##

$ echo “slarchive:archive_EC” >> $SEISCOMP_ROOT/etc/key/station_EC_AAM1

##Vincular el perfil scautopick##

$ echo “scautopick_local:local_100” >>

$SEISCOMP_ROOT/etc/key/station_EC_AAM1

$ seiscomp update-config

$ seiscomp restart
```

Línea de Comandos 1.18 Creación de la estación AAM1

ANEXO 2

1. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA EARTHWORM

En este Anexo se presenta el procedimiento a seguir para instalar y configurar el sistema de adquisición y procesamiento Earthworm.

1.1 INSTALACIÓN

1.1.1 REQUISITOS

El sistema Earthworm puede instalarse en servidores físicos o virtuales con sistemas operativos Linux y Windows de 32 o 64 bits, para este manual se utiliza el instalador correspondiente al sistema operativo CentOS 7 de 64 bits.

Antes de instalar el sistema Earthworm es necesario instalar algunas librerías como se indica en la Línea de Comandos 1.1

```
# yum install glibc.i686

# yum install libgcc-4.8.2-16.el7.i686
```

Línea de Comandos 1.1 Instalación de librearías adicionales

El archivo de instalación se descarga de la página web del proyecto Earthworm, como se indica en la Línea de Comandos 1.2 .

```
$ wget
http://www.earthwormcentral.org/distribution/earthworm_7.8-
centos7.1-64bit-bin.tar.gz  -P $HOME
```

Línea de Comandos 1.2 Descarga del instalador de Earthworm

1.1.2 INSTALACIÓN DE EARTHWORM

Una vez descargado el archivo de instalación se lo descomprime con el procedimiento que se indica en la Línea de Comandos 1.3.

```
$ cd $HOME

$ tar zxfv earthworm_7.7.1-centos6.4-64bit-bin.tgz

$ mv Earthworm_7.7 earthworm
```

Línea de Comandos 1.3 Descomprimir el instalador

A continuación es necesario agregar el contenido del Archivo de configuración 1.1 al archivo `$HOME/.bashrc`.

```
export PATH=$HOME/earthworm/bin:$PATH

export EW_PARAMS=$HOME/earthworm/params/

export EW_LOG=$HOME/earthworm/log/

export EW_HOME=$HOME/earthworm/

export EW_INSTALLATION=INST_MEMPHIS

export EW_VERSION=v7.7
```

Archivo de configuración 1.1 Variables de entorno para Earthworm

El siguiente paso es realizar la configuración del sistema Earthworm, se presenta la configuración de un sistema básico que realiza adquisición de formas de onda de algunas estaciones y realiza un gráfica de las formas de onda en el dominio de la frecuencia.

1.2 CONFIGURACIÓN

La configuración del sistema se basa solamente en archivos de configuración que se localizan en la carpeta `$HOME/earthworm/params/`.

Es posible usar archivos de configuración de prueba y modificarlos según nuestras necesidades. Estos archivos pueden descargarse y copiarse en la carpeta `$HOME/earthworm/params/` con los comandos que se indica en la Línea de Comandos 1.4

```
$ wget http://folkworm.ceri.memphis.edu/ew-
dist/v7.8/earthworm_7.5and_up_test.memphis.tar.gz -P $HOME

$ cd $HOME

$ tar zxfv earthworm_7.5and_up_test.memphis.tar.gz

$ cp -pr $HOME/memphis/params $HOME/earthworm/
```

Línea de Comandos 1.4 Descomprimir el instalador

El primer módulo que debe configurarse es `startstop`, como se presenta a continuación.

1.2.1 MÓDULO STARTSTOP

Este módulo se configura mediante el archivo `$HOME/earthworm/params/startstop_unix.d`. El contenido de este archivo se presenta en el Archivo de configuración 1.2 y determina los módulos de Earthworm que arrancarán junto con el módulo `startstop`, el grupo o anillo de mensajes que se crearán, el nombre de cada anillo, etc.

Como puede verse en el Archivo de configuración 1.2 los módulos que `startstop` arrancará son: `slink2ew`, `wave_serverV` y `sgram`. La configuración de estos módulos se presenta a continuación.

```
nRing          1

Ring SCNL      10240

MyModuleId    MOD_STARTSTOP

HeartbeatInt  50

MyClassName   OTHER

MyPriority     0

LogFile        1

KillDelay      30

HardKillDelay  5

Process          "slink2ew slink2ew.d"

Class/Priority   OTHER 0

Process           "wave_serverV wave_serverV.d"

Class/Priority    OTHER 0

 Process           "sgram sgram.d"

Class/Priority    OTHER 0
```

Archivo de configuración 1.2 Parámetro de configuración del módulo startstop

1.2.2 SLINK2EW

Este módulo se utiliza para que el sistema Earthworm pueda tomar datos de un servidor SeedLink, guardarlos en memoria y ponerlos a disposición del módulo `wave_serverV`, el módulo se configura mediante el archivo `$HOME/earthworm/params/slink2ew.d`, cuyo contenido se presenta en el Archivo de configuración 1.3. En el archivo se indica el anillo en el que el módulo

escribe los datos, la dirección IP y el puerto del servidor al que se solicitarán datos, así como las estaciones de las que se solicitarán los datos.

```
MyModuleId       MOD_SLINK2EW

RingName         SCNL

HeartBeatInterval     30

LogFile               1

Verbosity      0

SLhost    192.168.1.113

SLport         18013

StateFile

##Configuracion de estaciones volcan Cotopaxi

Stream EC_BMOR "BH?.D"

Stream EC_BTAM "BH?.D"

Stream EC_CAMI "SH?.D"

Stream EC_PITA "SH?.D"

Stream EC_SUCR "BH?.D"

Stream EC_SRAM "BH?.D"

Stream EC_BREF "BH?.D BDF.D"
```

Archivo de configuración 1.3 Parámetro de configuración del módulo slink2ew

1.2.3 WAVE_SERVERD

Este programa recolecta los datos de uno de los anillos de mensajes de Earthworm y los almacena de forma temporal en disco y los pone a disposición de los módulos de procesamiento de Earthworm. La configuración de este módulo se

realiza mediante el archivo `$HOME/earthworm/params/wave_serverV.d`, cuyo contenido se presenta en el Archivo de configuración 1.4.

```
MyModuleId    MOD_WAVESERVER

RingName      SCNL

LogFile       1

HeartBeatInt  30

ServerIPAdr   192.168.1.36

ServerPort    16025

IndexUpdate 1

TankStructUpdate 1

TankStructFile   tnk/ig_1.str

###Definicion de las estaciones que ingresaran

Tank BREF  BHZ EC --  4096  INST_WILDCARD  MOD_SLINK2EW  100  4096
tnk/BREF_BHZ_EC_00.tnk

Tank SUCR  BHZ EC --  4096  INST_WILDCARD  MOD_SLINK2EW  100  4096
tnk/SUCR_SHZ_EC_00.tnk

###Archivo de respaldo del estado de adquisicion

RedundantTankStructFiles 1

RedundantIndexFiles      1

TankStructFile2   tnk/ig_2.str

InputQueueLen 300

MaxMsgSize 4096

Debug 1

SocketDebug 0
```

Archivo de configuración 1.4 Parámetro de configuración del módulo wave_serverV

En el archivo se define el anillo al que solicitará los datos, la dirección IP y el puerto en el que el módulo aceptará peticiones, el directorio en el que se almacenarán los archivos temporales, etc.

1.2.4 SGRAM

Este módulo se conecta al módulo wave_`serverV` para solicitar formas de onda y graficar esas formas de onda en el dominio de la frecuencia. Estos gráficos se almacenan en formato GIF y se puede acceder a ellos a través de un navegador web.

El archivo de configuración del módulo `sgram` se encuentra en la ruta `$HOME/earthworm/params/sgram.d`, y su contenido se presenta en el Archivo de configuración 1.5.

Una vez configurados los módulos ya es posible iniciar el módulo `startstop`, como se indica en la Línea de Comandos 1.5. El módulo inicia sin inconvenientes y presenta información del servidor en el que se ejecuta, los grupos o anillos de mensajes creados, los directorios donde se almacenan los ejecutables, archivos de log, etc.

```
LogSwitch     1

MyModuleId    MOD_SGRAM

RingName      SCNL

HeartBeatInt  15

StandAlone

wsTimeout 40

WaveServer           192.168.1.36   16025

LocalTarget /var/www/html/gif/

GifDir /var/www/html/gif/

Prefix uw

###Estaciones a plotear

Plot BREF  BHZ EC    24 72  -5  TLE  1  1   600  1   1  64  10  0.2   2
150000   2   "COTOPAXI (BANDA ANCHA)"

Plot BNAS  BHZ EC    24 72  -5  TLE  1  1   600  1   1  64  10  0.2   2
150000   2   "COTOPAXI (BANDA ANCHA)"

Plot BTAM  BHZ EC    24 72  -5  TLE  1  1   600  1   1  64  10  0.2   2
150000   2   "COTOPAXI (BANDA ANCHNAS"

Plot BMOR  BHZ EC    24 72  -5  TLE  1  1   600  1   1  64  10  0.2   2
150000   2   "COTOPAXI (BANDA ANCHNAS"

Days2Save     7

UpdateInt     5

RetryCount    2

Logo    pnsn_logo2.gif

SaveDrifts

PlotDown

Make_HTML
```

Archivo de configuración 1.5 Configuración del módulo sgram

```
 [earth@centos1n ~]$ startstop

using default config file startstop_unix.d

                    EARTHWORM SYSTEM STATUS

        Hostname-OS:            earth - Linux 3.10.0-
123.el7.x86_64

        Start time (UTC):       Tue Mar  3 15:35:00 2015

        Current time (UTC):     Tue Mar  3 15:35:04 2015

        Disk space avail:       1146224 kb

        Ring  1 name/key/size:  SCNL / 1045 / 128 kb

        Startstop's Log Dir:    /home/earth/earthworm/log/

        Startstop's Params Dir:
/home/earth/earthworm/params/

        Startstop's Bin Dir:    /home/earth/earthworm/bin/

        Startstop Version:      v7.7 2012-08-13

         Process  Process           Class/    CPU

          Name      Id     Status  Priority   Used  Argument

         -------  -------  ------  --------   ----  --------

       startstop   30391   Alive    ??/  00:00:00  -

        slink2ew   30393   Alive    ??/  00:00:00
slink2ew.d

   wave_serverV    30394   Alive    ??/  00:00:00
wave_serverV.d

      sgram        30396   Alive      ??/ 0 00:00:00
sgram.d
```

Línea de Comandos 1.5 Arranque exitoso del sistema Earthworm

ANEXO 3

1. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA SHAKEMAP

En este Anexo se presenta el procedimiento a seguir para instalar y configurar el sistema de generación de mapas de movimiento sísmico ShakeMap.

1.1 INSTALACIÓN

1.1.1 REQUISITOS

1.1.1.1 Sistema operativo

Es posible utilizar solamente sistemas operativos Linux de 32 o 64 bits, en este manual se utilizará el sistema operativo Ubuntu.

1.1.1.2 Programas adicionales

El sistema necesita de algunos programas externos para funcionar, en la Figura 1.1 se muestran los principales requisitos de ShakeMap.

- Subversion: es un programa para controlar versiones y revisión de software, permite conseguir la última versión del sistema ShakeMap.
- Make: compila programas y librerías a partir del código fuente y archivos de configuración.
- GMT (*Generic Mapping Tools*): es una colección de comandos y *scripts* que permiten manejar y convertir datos de coordenadas geográficas (x, y, z) en archivos PostScript que se usan en mapas de contorno, mapas de superficie, etc.

- NetCDF (*Network Common Data Form*): es un conjunto de librerías que permiten acceder a datos ordenados en forma de arreglos, como son los datos de tipo geográfico.

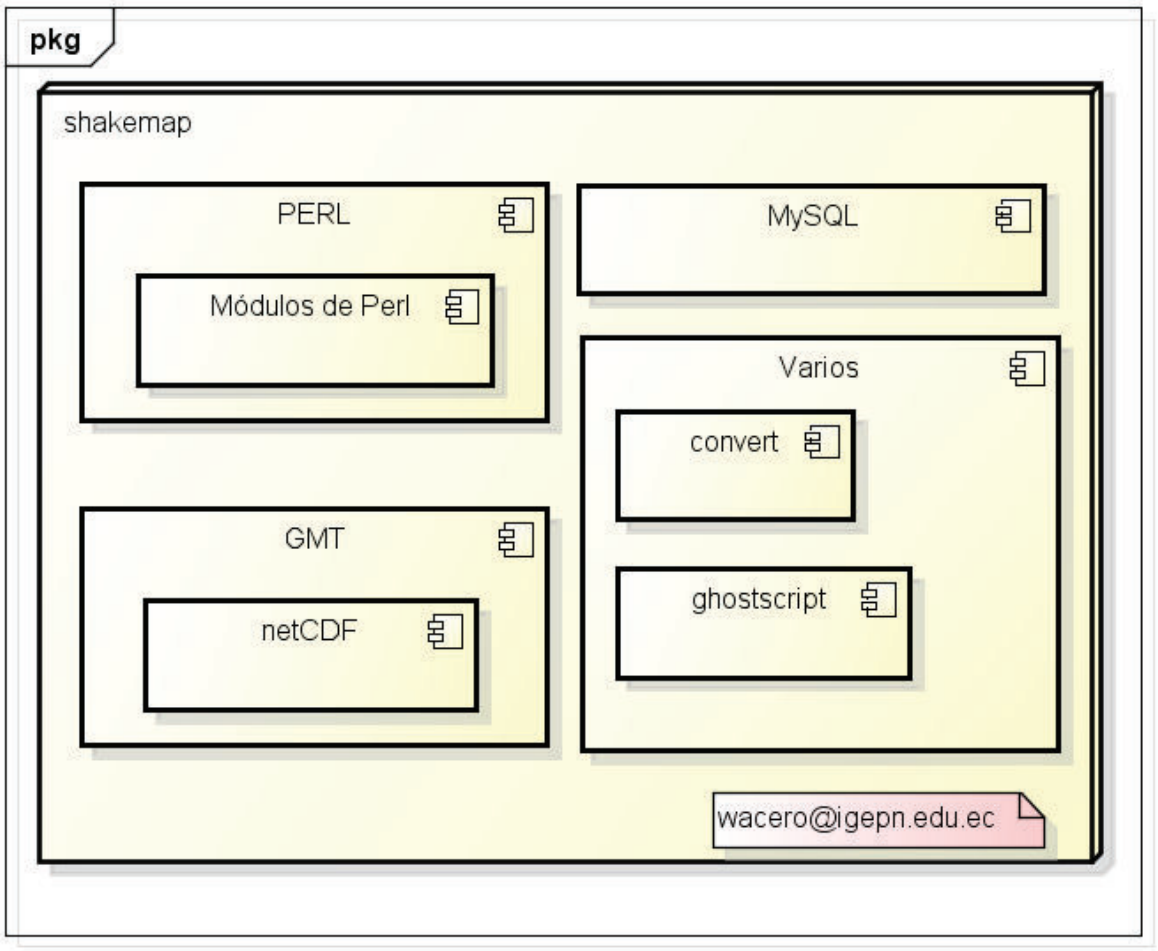

Figura 1.1 Programas requeridos por ShakeMap

- Ghostscript: es un conjunto de programas que permite crear y procesar archivos PostScript, que se convierten en imágenes mediante otro programa.

- Imagemagick: es una colección de programas que se usa para convertir archivos PostScript en imágenes JPEG y PNG.

- MySQL: es un servidor de base de datos de código abierto. El sistema ShakeMap es compatible únicamente con esta base de datos.

- Perl: es un lenguaje interpretado de alto nivel. Los módulos de ShakeMap están escritos en este lenguaje.

En la Línea de Comandos 1.1 se presentan los comandos usados para instalar los programas necesarios, a excepción de los módulos de Perl, GMT y NetCDF, que por ser más complejos se presentan en la siguiente sección.

```
$ sudo apt-get install subversion
$ sudo apt-get install make
$ sudo apt-get install imagemagick
$ sudo apt-get install ghostscript
$ sudo apt-get install mysql-server
$ sudo apt-get install perl
```

Línea de Comandos 1.1 Instalación de software necesario para ShakeMap

1.1.1.3 Módulos Perl

En las Línea de Comandos 1.2 y Línea de Comandos 1.3 se presenta la instalación de los módulos de Perl que el sistema necesita, algunos de ellos se instalaron usando el módulo de instalación CPAN[146], configurado con las opciones por defecto, mientras que otras librerías solamente fue posible instalarlas mediante el comando `apt-get`.

```
$ sudo apt-get install libwww-perl
$ sudo apt-get install libmysqlclient-dev
$ sudo apt-get install libexpat1-dev
$ sudo apt-get install libdatetime-perl
```

Línea de Comandos 1.2 Instalación de librerías Perl usando apt-get

[146] Comprehensive Perl Archive Network permite la instalación de librerías y módulos para Perl.

```
cpan> install Bundle::LWP

Do you want to modify/update your configuration (y|n) ? no

cpan > install DBD::mysql

cpan> install HTML::Template

cpan>  install XML::Parser

cpan> install XML::Writer

cpan> install enum

cpan> install Time::CTime

cpan> install Event

cpan> install Mail::Sender

Specify defaults for Mail::Sender? (y/N) N;

cpan> install Config::General

cpan> install XML::Simple

cpan> install Time::y2038
```

Línea de Comandos 1.3 Instalación de librerías Perl usando cpan

1.1.1.4 Instalación de GMT

A fin de facilitar la instalación de GMT se comprimieron los paquetes y librerías necesarios en un archivo `tar`, que se instala con el procedimiento presentado en la Línea de Comandos 1.4.

```
$ su -

# cd /usr/local

# tar zxvf $HOME/sacgmt.tar.gz
```

Línea de Comandos 1.4 Instalación del software GMT

Es necesario modificar la variable de entorno `$PATH`, como se indica en la Línea de Comandos 1.5.

```
$ echo "export GMTHOME=/usr/local/GMT" >> $HOME/.bashrc

$ export PATH=$GMTHOME/bin:$PATH >> $HOME/.bashrc
```

Línea de Comandos 1.5 Incluir a GMT en la variable $PATH

1.1.1.5 Servidor MYSQL

El siguiente paso es instalar el servidor de base de datos MYSQL y crear la base de datos que el sistema utilizará, como se indica en la Línea de Comandos 1.6 y la Línea de Comandos 1.7.

```
# yum install mariadb-server

# systemctl enable mariadb

# systemctl start mariadb
```

Línea de Comandos 1.6 Instalación del servidor de base de datos MySQL

```
$ mysql –u root -p

mysql> create database shakemap;

mysql> grant select,insert,update,delete,create,drop,alter
on shakemap.* to shake@localhost identified by 'shakeXX';
```

Línea de Comandos 1.7 Procedimiento para crear la base de datos shakemap

1.1.1.6 Archivos DEM

Para algunos comandos de ShakeMap son necesarios archivos DEM (*Digital Elevation Model*) que permitan crear imágenes de un área geográfica en tres dimensiones. En la Línea de Comandos 1.8 se muestra el procedimiento para descargar y utilizar estos datos.

```
$ cd $HOME

$ tar zxvf DEM.tar.gz
```

Línea de Comandos 1.8 Descarga de los archivos DEM

1.1.2 COMPILACIÓN E INSTALACIÓN DE SHAKEMAP

Una vez completados los pasos anteriores ya es posible descargar y compilar el programa, con el procedimiento que se indica en la Línea de Comandos 1.9.

```
$ svn checkout
https://vault.gps.caltech.edu/repos/products/shakemap/tags/re
lease-3.5/ $HOME/shake

$ cd $HOME/shake/install

$ ./make
```

Línea de Comandos 1.9 Descarga y compilación del software ShakeMap

El comando anterior genera el archivo de configuración `$HOME/shake/include/macros`, cuyas variables más importantes y los valores correspondientes se presentan en el Archivo de Configuración 1.1. Es muy importante que estas librerías existan, ya que de lo contrario el programa no compilará.

```
GMTLIB  =  /usr/local/gmt/lib/

GMTINC  = /usr/local/gmt/include/

CDFLIB  = /usr/local/netcdf/lib

CDFINC  = /usr/local/netcdf/include

CONVERT  = /usr/bin/

GMT_VERSION = 4.5

DEMDIR =$HOME/DEM/
```

Archivo de Configuración 1.1 Contenido del archivo `$HOME/shake/include/macros`

Una vez modificado el archivo de configuración ya es posible terminar la instalación de ShakeMap, con el comando que se presenta en la Línea de Comandos 1.10. Si no existen errores en la carpeta `$HOME/shake/bin` se habrán creado los programas de ShakeMap

```
$ cd $HOME/shake

$ make all
```

Línea de Comandos 1.10 Compilación de ShakeMap

1.2 CONFIGURACIÓN DE SHAKEMAP

Por defecto la configuración incluida en el software ShakeMap ya permite generar los mapas de movimiento, pero para el área de la bahía de San Francisco, EEUU. Para que los mapas de movimiento generados correspondan al área geográfica ecuatoriana es necesario configurar ecuaciones y tablas de datos, tarea que está a cargo del área de sismología del IG-EPN, razón por la cual no se incluye esa información en este documento.

1.2.1.1 Configuración de la base de datos

En la Línea de Comandos 1.11 se muestran los comandos para configurar la información del usuario y clave de la base de datos ShakeMap.

```
$ mkdir $HOME/shake/pw

$ echo "shakemap usuario clave" >> $HOME/shake/pw /passwords

$ echo "database : mysql database=shakemap usuario" >>
 $HOME/shake/config/mydb.conf
```

Línea de Comandos 1.11 Configuración de acceso a la base de datos shakemap

El siguiente paso es crear las tablas que utilizará ShakeMap con el comando de la Línea de Comandos 1.12, el comando no genera ninguna respuesta, y si resulta exitoso se crean las tablas `earthquake`, `server`, `shake_lock`, `shake_runs` y `shake_version`.

```
$ $HOME/shake/bin/mktables

 ## Verificar que se crearon las tablas

$ mysql  -u root –p shakemap

mysql> show tables;

earthquake

server

shake_lock

shake_runs

shake_version
```

Línea de Comandos 1.12 Crear las tablas necesarias para shakemap

1.2.2 GENERACIÓN DEL MAPA DE MOVIMIENTO

Una vez instalados todos los componentes del sistema ShakeMap ya es posible generar el primer mapa de movimiento de un evento sísmico.

ShakeMap incluye datos de prueba, es decir la información de un eventos sísmico necesaria para crear los respectivos productos, esta información se encuentra en la carpeta `$HOME/shake/data/9583161/input/`, y con el comando que indica en la Línea de Comandos 1.13 es posible crear el ShakeMap de este evento.

```
$HOME/shake/bin/shake -event 9583161
```

Línea de Comandos 1.13 Comando para crear el mapa de movimiento de un evento sísmico

Si el comando tuvo éxito se genera un mensaje como el que se muestra en la Figura 1.2.

```
ubudj@ubudj:~$
ubudj@ubudj:~$ shake/bin/shake -event 9583161
shake started event 9583161 at Mon May 19 15:39:12 2014
2014-05-19 15:39:13 : transfer: ----- Starting Transfer for 9583161 at 05/19/2014 15:39:13 -----
2014-05-19 15:39:13 : transfer: ----- Transfer finished 9583161 at 05/19/2014 15:39:13 -----
GMT_grd_is_global: no!
GMT_boundcond_param_prep determined edgeinfo: gn = 0, gs = 0, nxp = 0, nyp = 0
GMT_grd_is_global: no!
2014-05-19 15:39:18 : mapping: WARNING: no topography data found; will continue without topo.
2014-05-19 15:39:18 : mapping: WARNING: can't find topo data: making mi without topography
2014-05-19 15:39:18 : mapping: WARNING: can't find topo data: making mi without topography
2014-05-19 15:39:18 : mapping: WARNING: couldn't find topo and intensity data; will continue without topo.
Printing grade - (-0.000)
2014-05-19 15:39:27 : transfer: ----- Starting Transfer for 9583161 at 05/19/2014 15:39:27 -----
2014-05-19 15:39:27 : transfer: ----- Transfer finished 9583161 at 05/19/2014 15:39:27 -----
shake completed event 9583161 at Mon May 19 15:39:27 2014
ubudj@ubudj:~$
```

Figura 1.2 Generación exitosa de un mapa de movimiento

Mientras tanto en la carpeta `$HOME/shake/data/9583161/` se han creado las carpetas que contienen las imágenes, archivos HTML, archivos de mapas, etc. El producto final es una página web, que puede verse en la Figura 1.3.

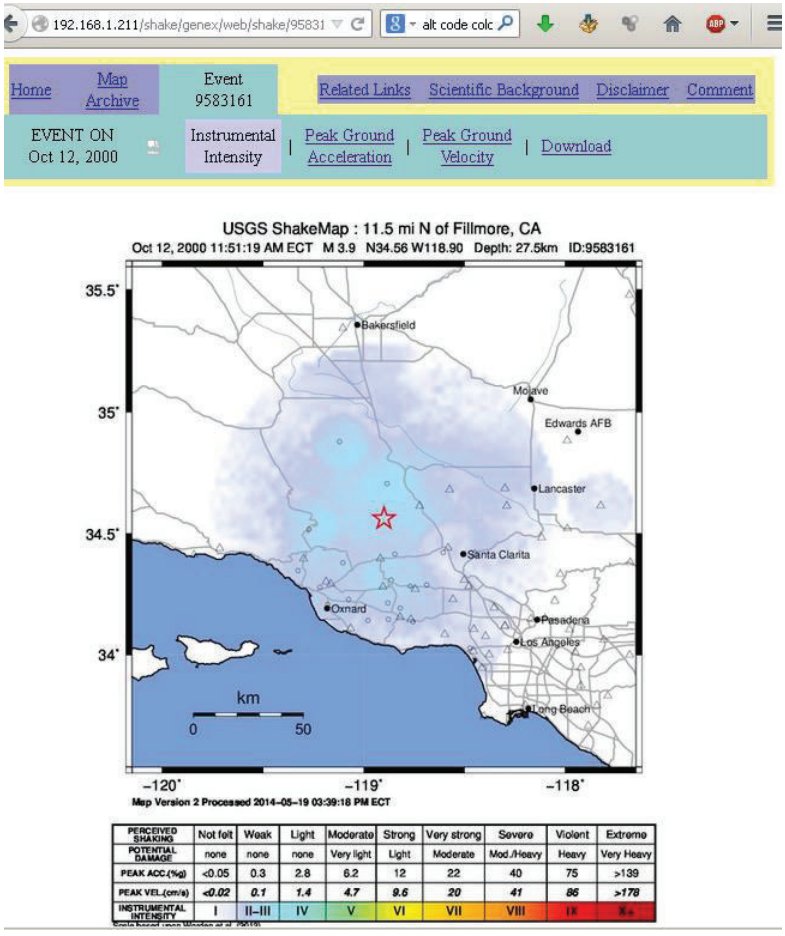

Figura 1.3 ShakeMap del evento 9583161