AVALIAÇÃO DE DESEMPENHO PARA SISTEMAS PARALELOS

Ricardo Zelenovsky, Rubén D. Léon V.
Escuela Politécnica del Ejército - ESPE

Resumen:
É proposto uma metodologia para a análise de sistemas paralelos. O desempenho é calculado com relação a decomposição e alocação dos subproblemas às unidades de processamento. Também são computados o tamanho do grão, eficiência do "cache" e a presença de trechos seriais.

Abstract:
A method for the analysis of parallel systems is proposed. The performance is done respectively with the decomposition and allocation of subproblems to the processing units. The grain size, hit-ratio of cache and serial sections are also computed.

1. Introdução.

A avaliação de desempenho de sistemas multiprocessadores não é uma tarefa simples. A além da grande quantidade de parâmetros a serem analisados, o inter-relacionamento entre eles é complexo e difícil de ser quantizado.

Segundo Cvetanovic[1] os parâmetros de influência mais significativas no desempenho de um sistema paralelo são:
a) quantidade de paralelismo intrínseco do problema,
b) método utilizado na decomposição do problema em subproblemas,
c) método utilizado para alocar os subproblemas aos processadores,
d) tamanho do grão,
e) método de acesso aos dados,
f) estrutura de interconexão (acesso às memórias),
g) velocidade de processadores, memória e rede de interconexão.

Esse trabalho expõe uma metodologia para a análise da adequabilidade de algoritmos ao processamento paralelo.

2. Parâmetros da Modelagem

Para uma modelagem eficaz é necessário selecionar parâmetros que modelem o "software" e o "hardware", e que permitam, através de cálculos simples, a previsão do desempenho de um determinado "software" quando executado em um particular "hardware"

A seguir é desenvolvida uma modelagem com objetivo de verificar o desempenho de sistemas multiprocessadores, com memória compartilhada, quando sujeitos a variação da banda passante da memória, eficiência do cache e carga de trabalho do algoritmo a ser executado.

A carga de trabalho ("software") é dividida em processamento e comunicação. Aqui comunicação tem o significado de acesso à memória através da estrutura de interconexão. O processamento é caracterizado por uma **"Unidade de Processamento"**, que pode variar para cada caso; em alguns casos utiliza-se como unidade uma operação de ponto flutuante, e em outros um conjunto de operações de ponto flutuante. A comunicação é caracterisada pela **"Unidade de Comunicação"**, que também não possue uma definição rígida, significando acesso a um byte na memória ou acesso a uma palavra.

Com esses dois conceitos define-se uma série de parâmetros que modelam o "hardware" e o "software":
Tp -> quantidade de unidades de processamento do algoritmo,
tp -> tempo para executar uma unidaue de processamento,
Tc -> quantidade de unidades de comunicação do algoritmo,
tc -> tempo para executar uma unidade de comunicação,
N -> número de processadores,
BW -> banda passante da memória, dada pelo número de processadores que ela atende por ciclo,
HR -> taxa de acertos ("hit-ratio") do cache.

Análogo à definição dado por Vrsalovic[2] define-se **"Função Decomposição de Processamento (Dp)"** como a razão entre o tempo de processamento para um sistema uniprocessador e o tempo de processamento para cada processador em un sistema multiprocessador e **"Função Decomposição de Comunicação (Dc)"**, como a razão entre o tempo de acesso a dados em um sistema uniprocessador e o tempo de acesso a dados para cada processador em um sistema multiprocessador. Os parâmetros tc, tp, N, BW dependem das características do sistema, já Dc, Dp, Tc, Tp dizem respeito ao algoritmo. A adequabilidade de um algoritmo ao processamento paralelo é representado por suas funções Dp e Dc.

$$Dp(N) = \frac{\text{tempo de processamento em sistema uniprocessador}}{\text{tempo de processamento para cada processador em sistema multiprocessador}} \quad (1)$$

$$Dc(N) = \frac{\text{tempo de acesso a dados em sistema uniprocessador}}{\text{tempo de acesso a dados para cada processador em sistema multiprocessador}} \quad (2)$$

O modelamento é complexo, pois não existem apenas os parâmetros físicos do "hardware", mas também os parâmetros do algoritmo:
tc, tp, N, BW ==> relativos ao "hardware",
Dc, Dp, Tc, Tp ==> relativos ao algoritmo.

Exemplo:
Um determinado algoritmo "A" quando executado em um sistema uniprocessador gasta 360 seg em processamento e 180 seg em comunicação:
Tp * tp = 360 seg e Tc * tc = 180 seg.
Esse mesmo algoritmo executado de forma balanceada em um sistema com 9 processadores gasta 40 seg em processamento e 20 seg em comunicação.

$$Dp = \frac{360}{40} = 9 \quad e \quad Dc = \frac{180}{20} = 9$$

O presente algoritmo adapta-se muito bem aos sistemas multiprocessadores já que a divisão da carga de processamento e comunicação não apresenta nenhum custo adicional.

Por outro lado, um algoritmo "B", semelhante ao algoritmo "A" (mesmos Tp e Tc), ao ser executado de forma balanceada em um sistema com 9 processadores gasta 40 seg em processamento e 60 seg em comunicação.

$$Dp = \frac{360}{40} = 9 \quad e \quad Dc = \frac{180}{60} = 3$$

O algoritmo "B" apresenta uma penalidade na divisão da carga de comunicação. Casos piores podem existir, onde Dc=1, ou seja , não adianta aumentar o número de processadores que a carga de comunicação nunca é reduzida.

O **"Ganho de Velocidade (G)"** é dado pela relação entre o tempo gasto em um sistema uniprocessador e o tempo gasto em um sistema multiprocessador:

$$G = \frac{\text{tempo de execução em sistema uniprocessador}}{\text{tempo de execução em sistema multiprocessador}} \qquad (3)$$

O tempo de execução serial para um certo algoritmo é dado pela soma do tempo gasto em processamento (Tp*tp) com o tempo gasto em comunicação (Tc*tc):

$$Es = Tp*tp + Tc*tc \qquad (4)$$

Esse mesmo algoritmo quando executado em um sistema multiprocessador gasta em processamento:

$$Ep = \frac{Tp}{Dp}\, tp \qquad (5)$$

Se os dados a serem acessados estiverem em uma única memória compartilhada, de banda passante BW, acessada por todos os processadores, é gasto em comunicação:

$$Ec = \frac{Tc}{Dc} * \frac{N}{BW} * tc \qquad (6)$$

Na última equação, Tc/Dc é igual ao número de acessos para cada processador e N/BW reflete a disputa pela memória e os problemas de contenção.

O tempo total (Et) gasto na execução paralela é dado pela soma de Ep e Ec:

$$Et = Ep + Ec \qquad (7)$$

$$Et = \frac{Tp}{Dp} * tp + \frac{Tc}{Dc} * \frac{N}{BW} * tc \qquad (8)$$

O ganho de velocidade do sistema multiprocessador, em relação ao monoprocessador é:

$$G = \frac{Es}{Et} = \frac{Tp*tp + Tc*tc}{Ep + Ec} \qquad (9)$$

$$G = \frac{Tp * tp + Tc * tc}{\frac{Tp}{Dp} * tp + \frac{Tc}{Dc} * \frac{N}{BW} * tc} \qquad (10)$$

A equação (10) também pode ser escrita da seguinte forma:

$$G = \frac{\frac{Tp * tp}{Tc * tc}}{\frac{TP*tp}{Tc*tc} * \frac{1}{Dp} + \frac{1}{Dc} * \frac{N}{BW}} \qquad (11)$$

Nessa equação, temos:
Tp * tp ==> quantidade de tempo gasto em processamento,
Tc * tc ==> quantidade de tempo gasto em comunicação.
Define-se "K" como a relação entre o tempo gasto em processamento e o tempo gasto em comunicação:

$$K = \frac{Tp * tp}{Tc * tc} \qquad (12)$$

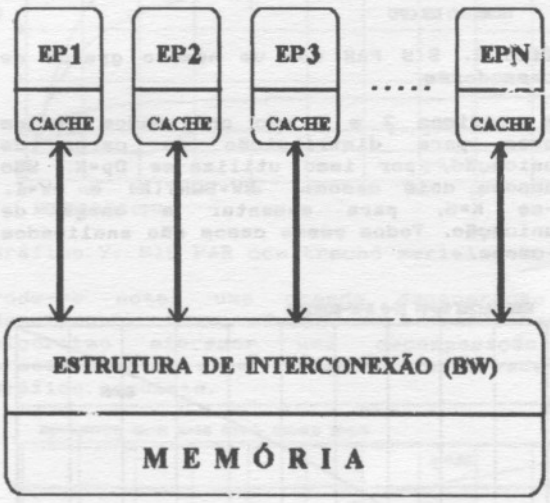
Agora o ganho de velocidade pode ser definido como:

$$G = \frac{1 + K}{\frac{K}{Dp} + \frac{1}{Dc} * \frac{N}{BW}} \qquad (13)$$

Nota-se que à medida que K aumenta, o ganho de velocidade também aumenta. Valores de K menores que 1 implicam em mais comunicação que processamento, o que gera uma grande quantidade de conflitos, retardando assim todo o sistema. Em geral a quantidade de processamento é maior que a quantidade de comunicação. Para valores muito alto de K, o ganho de velocidade aproxima-se de Dp.

3. Análise de Desempenho

Com os conceitos expostos faz-se facilmente a análise de desempenho do conjunto "hardware" e "software". Como ilustração utiliza-se a arquitetura da figura 1, denominada de "SISPAR".



EP = elemento processador (CPU)
Figura 1. SIS PAR com N processadores.

No gráfico 1 é feito uma análise onde o principal objetivo é o resultado da distribuição da carga de processamento. Para evitar que a carga de comunicação influencie, usa-se Dc=N. E também suposto uma carga de processamento 10 vezes maior que a comunicação (K=10).

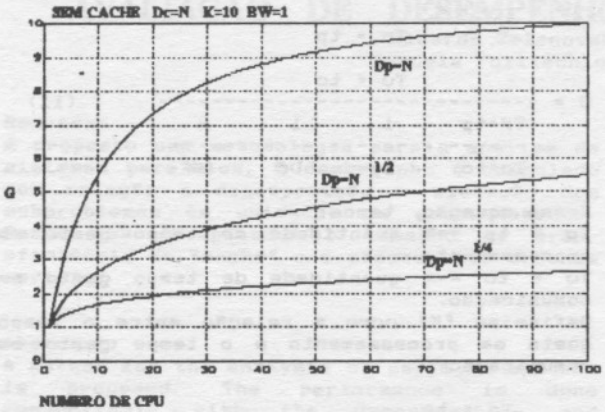


Gráfico 1. SIS PAR com diversos Dp.

Nota-se que para valores grandes de N, o ganho tende a estabilizar-se em um certo valor. De fato prova-se facilmente que para Dc=N e grandes valores de Dp, o ganho tende para (1+K)*BW. Isso pode ser observado no gráfico 2 onde existem curvas para diversos valores de BW.

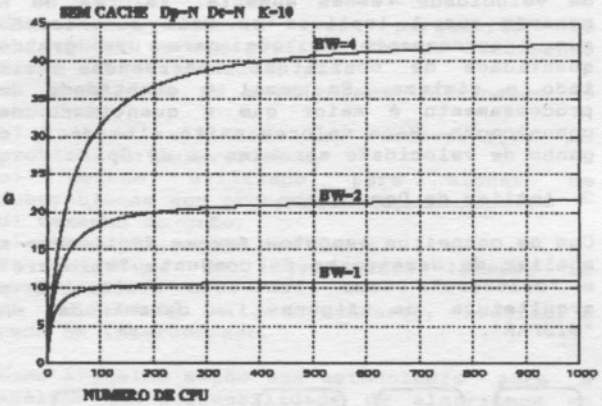


Gráfico 2. SIS PAR com um número grande de processadores.

Nos gráficos 2 e 3 são analisados vários valores para distribuição da carga de comunicação, por isso utiliza-se Dp=N. São estudados dois casos: BW=SQRT(N) e BW=1. Usa-se K=5, para aumentar a carga de comunicação. Todos esses casos são analisados sem cache.

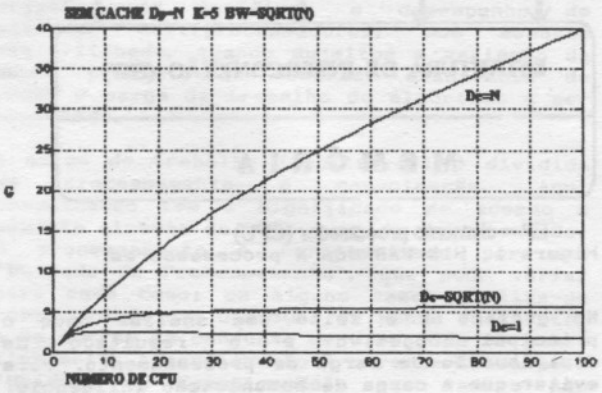


Gráfico 3. SIS PAR com diversos Dc e BW=SQRT(N).

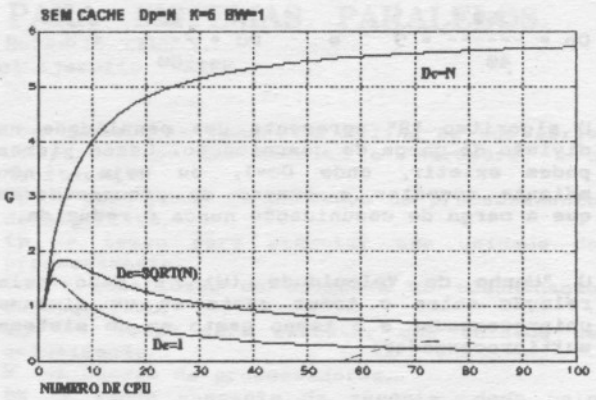


Gráfico 4. SIS PAR com diversos Dc e BW=1.

Nos dois casos anteriores, ve-se o grande custo dos acessos a memória. Dependendo da situação, o desempenho pode até ser pior que o de um sistema monoprocessado.

3.1. Influência do Cache

A memória cache pode ajudar muito o desempenho de um sistema multiprocessado. O cache é caracterisado pelo seu "hit ratio (HR)", sendo este dependente da estrutura do algoritmo e do tamanho do cache. Os Tc acessos realizados são divididos em (HR*Tc) acessos ao cache e ((1-HR)*Tc) acessos fora do cache. A equação do ganho de velocidade toma então a seguinte forma:

$$G = \frac{1 + K}{\frac{K}{Dp} + \frac{(1-HR)}{Dc} \cdot \frac{N}{BW} + \frac{HR}{Dc} X} \qquad (14)$$

Onde X=tch/tc, ou seja, a relação entre a velocidade do cache e a velocidade da memória. No presente caso considera-se o cache 3 vezes mais veloz (X=1/3). O gráfico a seguir ilustra a melhora obtida pela presença do cache.

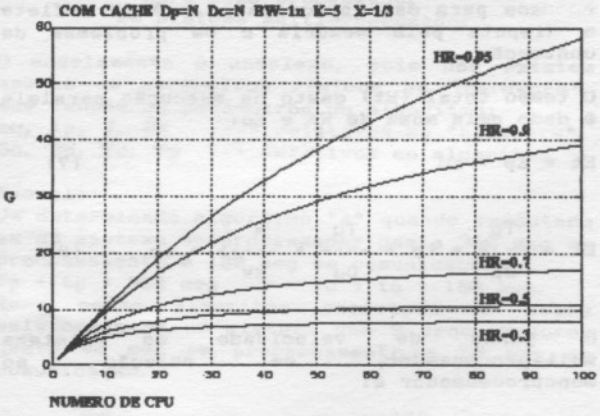


Gráfico 5. SIS PAR com cache.

Nota-se a grande melhoria provocada pelo cache e também que o desempenho do sistema é muito sensível para grandes valores de HR.

3.2. Banda Passante Hipotética

Para analisar a influência do cache, utiliza-se um sistema hipotético, sem cache, cuja banda passante BWh é aumentada até o sistema atingir o mesmo ganho de velocidade que o sistema com cache. Cria-se assim um sistema hipotético onde a variação da banda

passante é usada para capturar o efeito do cache.

O ganho de velocidade desse sistema hipotético é:

$$G = \frac{1 + K}{\frac{K}{Dp} + \frac{1}{Dc} + \frac{N}{BWh}} \qquad (15)$$

Igualando-se as equações (14) e (15) obtêm-se BWh em função do HR, N, BW e X.

$$BHh = \frac{n * BW}{(1-HR)*N + HR * BW * X} \qquad (16)$$

No gráfico a seguir está a banda hipotética, onde se nota a grande sensibilidade para com valores elevador de HR.

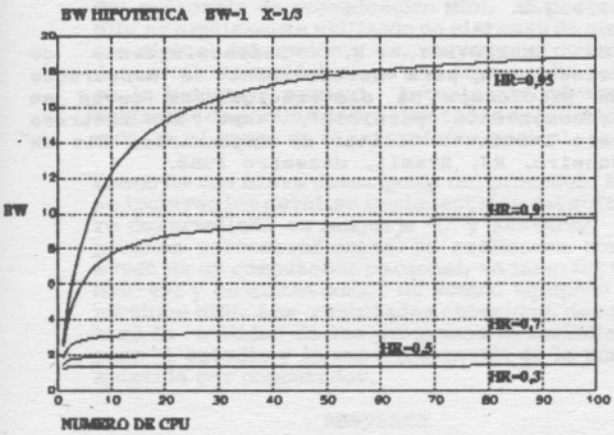


Gráfico 6. BW hipotética para SIS PAR com cache.

Esse método de avaliar o peso de uma alteração através de sua tradução em banda passante de um sistema hipotética, pode ser usado em outras situações, por exemplo, quando se faz alterações na arquitetura.

3.3. Algoritmos com trechos seriais

Algumas vezes o processamento não pode ser totalmente decomposto entre os processadores, pois parte dele precisa ser executado de forma serial por um único processador. Neste caso divide-se a quantidade de processamento em uma porção paralela (Tpp) e em uma porção serial (Tps).

$$Tp = Tpp + Tps, \text{ se } q = \frac{Tps}{Tp}, \text{ então}$$

$$1 - q = \frac{Tpp}{Tp} \qquad (17)$$

Supondo que a carga paralela "Tpp" apresenta uma decomposição igual a "N", calcula-se uma nova função decomposição de processamento, como está na equação (18).

$$Dps = \frac{Tp}{\frac{Tpp}{N} + Tps} = \frac{1}{\frac{(1-q)}{N} + q} \qquad (18)$$

A equação de Amdahl permite calcular o máximo ganho de velocidade teórico na paralelização de programas seriais. Por exemplo seja "T" o tempo para execução serial do programa, "N" o número de processadores e "q" a fração do tempo de execução de "T" que deve permanecer sequencial. Amdahl formula que o tempo gasto na porção serial é igual a "q*T" e que o tempo mínimo para execução da parte paralela é ((1-q)*T/N). Aplicando-se a definição da Função Decomposição do Processamento, obtem-se uma expressão igual a equação (18):

$$Dps = \frac{T}{qT + (1-q)*\frac{T}{N}} = \frac{1}{\frac{(1-q)}{N} + q} \qquad (19)$$

A equação desenvolvida para Dps supõe que a decomposição da carga paralela seja igual a "N", porém isso nem sempre acontece. Uma expressão mais geral, com "Dpp" indicando a decomposição do trecho paralelo é apresentada.

$$Dps = \frac{1}{\frac{(1-q)}{Dpp} + q} \qquad (20)$$

Uma equação global para o ganho de velocidade, abrangendo todos os parâmetros, inclusa o cache é apresentado a seguir.

$$G = \frac{1 + K}{\frac{(1-q)*K}{Dpp} + q*K + \frac{(1-HR)*N}{Dc*BW} + \frac{HR}{Dc} X} \qquad (21)$$

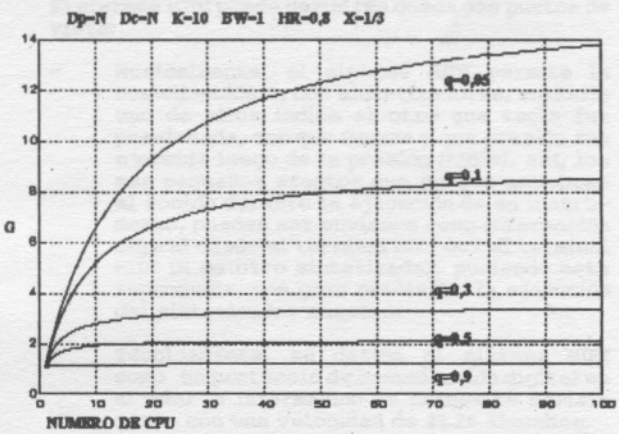


Gráfico 7. SIS PAR com trecho serial e Dc=N.

Pode-se notar uma grande degeneração no desempenho. Esse efeito será maior se o algoritmo oferecer uma decomposição de processamento ruim. Isso é observado no gráfico seguinte.

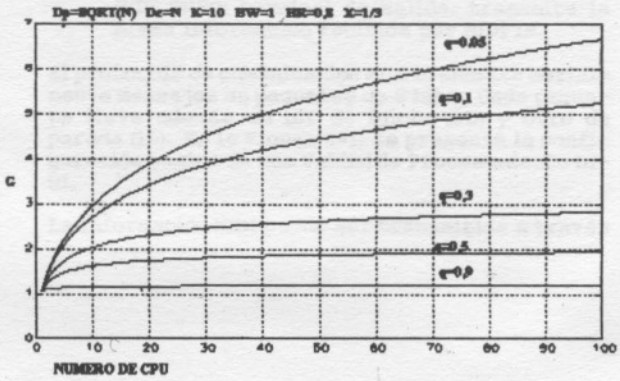


Gráfico 8. SIS PAR com trecho serial e Dc=SQRT(N).

4. Conclusões

No presente trabalho é apresentada uma metodologia para cálculo do desempenho de sistemas paralelos. Essa metologia considera não apenas a arquitetura, mas também a adequabilidade dos algoritmos para processamento paralelo.

Observa-se que o cache é muito importante, principalmente para os casos onde a função decomposição de comunicação apresenta um baixo valor. Nota-se uma grande sensibilidade para valores elevados de HR. Ou seja são proveitosos os esforços para elevar o HR de 90% para 95%, enquanto que a elevação de 10% para 20% deverá trazer pouca incremento no desempenho. Para melhor apreciar a influência do cache é sugerido sua conversão em banda passante equivalente.

Algoritmos com trechos seriais são danosos para o desempenho. Eles também não apresentam um comportamento linear. Uma variação de "q" de 5% para 10% provoca uma queda de desempenho pior que uma variação de 50% para 70%.

Referências :

(1) CVETANOVIC, Zarka, "The effects of problem partitioning, allocation, and granularity on the performance of multiple-processor systems", IEEE Transactions on Computers, Vol c-36, Nr 4, april 1987, pag 421-432.

(2) VRSLOVIC, D. & GEHRINGER, E.F. & SEGALL, Z.Z. & SIEWIOREK, D.P., "The influence of parallel decomposition strategies on the performance of multiprocessor systems", Proc. 12th Ann. Int. Comput. Architect., Boston, MA, IEEE Comput. Soc. and ACM, june 1985, pag 396-405.

(3) HWANG, K. & BRIGGS, F.A., "Computer architecture and parallel processing", Mc Grall-Hill, 1984.

(4) HWANG, K., "Advanced parallel processing with supercomputer architecture" Proc. IEEE, Vol 75, Nr 10, october 1987.

(5) ZELENOVSKY, R., "Avaliação de desempenho de um sistema multiprocessador", II Simpósio Brasileiro de Arquitetura de Computadores - Processamento Paralelo, Aguas de Lindóia, SP, Brasil, setembro 1988, pag 11.b.3.1 - 11.b.3.5.

(6) ZELENOVSKY, R. "Desenvolvimento de metodologia para balanceamento e mensuração da eficiência da distribuição de carga em processamento paralelo", tese de mestrado pelo Instituto Militar de Engenharia, Rio de Janeiro, RJ, Brasil, dezembro 1988.