ALGORITMOS PARA DESCOBRIR O MÁXIMO EM UM MULTICOMPUTADOR BASEADO EM BARRAMENTOS HIERÁRQUICOS.

Alex Alves Freitas* e Claudio Kirner**

UFSCar (Universidade Federal de São Carlos)
Programa de Pós-graduação - Ciência da Computação
Caixa Postal 676
13560 São Carlos - SP
Fone: (0162) 74-8233 ou 74-8232
E-mail: UFSCarC@BRFAPESP.BITNET

RESUMO

Este artigo apresenta algoritmos para determinação do valor máximo dentre n valores distintos, em um multicomputador baseado em barramentos hierárquicos. É analisada a influência, no projeto e desempenho desses algoritmos, de características da arquitetura, como: número total de EPs (elementos de processamento), razão entre número de "clusters" e número de EPs por "cluster", e relacionamento entre tempo para transmissão de mensagens em um barramento e tempo para processamento em um EP.

ABSTRACT

This paper presents algorithms for finding the maximum value out of n distinct values, in a multicomputer based upon hierarchical buses. We analyze the influence, on the design and performance of these algorithms, of architecture features, such as: total number of PEs (processing elements), ratio between number of clusters and number of PEs per cluster, and relationship between time for transmitting messages in a bus and time for processing in a PE.

* Mestrando em Ciência da Computação/UFSCar; Tecnólogo em Processamento de Dados (FATEC-SP,1989), Especialização (pós-graduação "lacto sensu") em Informática (Univ. Mackenzie, 1990).
** Prof. Adj./DC-UFSCar; Eng. Eletricista (EESC/USP,1973), Mestre Eng. Eletrônica (ITA,1978), Dr. Eng. Sistemas e Computação (COPPE/UFRJ,1986).

1 INTRODUÇÃO.

Este artigo apresenta algoritmos para descobrir o valor máximo dentre um conjunto de n valores distintos, em um multicomputador (com memória distribuída) cuja rede de interconexão é constituída por barramentos hierárquicos.

Embora o problema de determinação do máximo seja relativamente simples, e talvez não seja muito interessante "por si só", o estudo realizado por este artigo é justificado por dois fatores:

a) Algoritmos para determinação do máximo estão intimamente relacionados com outros tipos de algoritmos "baseados em comparação", tais como ordenação e seleção (dos m maiores valores, dentre n valores), os quais têm uma vasta gama de aplicações. Assim, estudando-se o problema de determinação do máximo, pode-se descobrir princípios gerais de utilidade em algoritmos mais complexos e importantes.

b) Recentemente, o modelo de comunicação por barramento tem atraído a atenção dos projetistas de algoritmos paralelos, devido principalmente à implementação natural e eficiente de operações de difusão (transmissão de uma mensagem de um EP (elemento de processamento) para todos os outros EPs da máquina). Como exemplos de trabalhos nessa área, pode-se citar [2], [6], [7], [8] e [9]. Entretanto, essas referências assumem que todos os EPs estão conectados através de um único barramento, o que não é tecnologicamente viável, quando se deseja trabalhar com uma grande quantidade de EPs. A fim de preencher essa lacuna, este artigo é orientado para máquinas altamente paralelas (com mais de 1000 EPs) baseadas em barramentos hierárquicos.

2 O MODELO DE COMUNICAÇÃO POR BARRAMENTO.

Neste modelo, cada EP (elemento de processamento) está conectado a um barramento (às vezes denominado barramento de difusão - "broadcast bus"), e os EPs podem comunicar-se somente através do barramento. Se mais de um EP tentar difundir sua mensagem ao mesmo tempo, ocorrerá um conflito, e algum esquema de arbitragem será usado para escolher apenas uma mensagem para ser difundida a seguir (está sendo assumido que o barramento possui um único canal para transmissão de dados). Entretanto, todos os EPs podem ler simultaneamente a mensagem difundida pelo barramento.

O tempo de execução de um algoritmo é dado pela soma: tempo de comunicação + tempo de processamento. O tempo de comunicação, por sua vez, é dado pela soma: tempo de arbitragem + tempo de transmissão (difusão) da mensagem. Deve-se minimizar tanto o tempo de comunicação quanto o tempo de processamento. Assim, o esquema de comunicação deve ser projetado de forma que o único canal disponível para difusão seja completamente utilizado [7]. Em uma situação ideal, sempre que o barramento fosse utilizado, a mensagem difundida deveria ser útil para todos os EPs.

Os algoritmos apresentados neste artigo são baseados nas seguintes considerações [2]:

a) É utilizado um esquema de arbitragem "ideal", ou seja, qualquer conflito é resolvido em um tempo constante (independente do número de EPs disputando o barramento).
b) Os EPs computam em paralelo sincronizadamente, de forma que as análises dos algoritmos não levarão em conta variações no tempo de computação entre os EPs.
c) Em cada passo, cada EP pode ler a mensagem recém difundida, fazer alguma computação, e submeter uma mensagem para ser difundida no próximo passo.
d) Uma operação de comparação é considerada como o passo básico de computação (já que os algoritmos para descobrir o valor máximo são baseados em comparações), e a difusão de uma mensagem é considerada como o passo básico de comunicação.

Ao longo de todo este artigo, é assumido que há n EPs e n valores distintos. Cada EP armazena um valor. Os logaritmos representados pelo símbolo "log" estão na base 2.

3 ALGORITMO MAX.

Nesta seção, será apresentado um algoritmo simples, denominado MAX, para descobrir o valor máximo no modelo de comunicação por barramento. MAX é apresentado também em [2], [5] e [6]. Nessa última referência, porém, o algoritmo é apresentado em sua forma dual, denominada MIN, que descobre o valor mínimo.

Nas seções 5 e 6, MAX será adaptado para um máquina baseada em barramentos hierárquicos.

No algoritmo MAX, cada EP pode assumir um dentre dois estados:

ATIVO: O EP armazena um valor maior do que o maior valor difundido até o momento, de forma que o EP ainda está tentando difundir seu valor.

INATIVO: O EP já difundiu seu valor ou já "desistiu" (em virtude de ter recebido um valor maior).

O algoritmo MAX é apresentado na Figura 1. Quando é detectado um silêncio no barramento, todos os EPs ficam sabendo que o algoritmo terminou. O último valor difundido é o valor máximo.

```
inicio
  todos EPs são rotulados como ATIVO;
  enquanto ( não (todos os EPs estão INATIVO) ):
    todos EPs ATIVO tentam acessar barramento;
    um esquema de arbitragem decide qual EP ganha o acesso ao
     barramento;
    o EP vencedor difunde seu valor e é rotulado como INATIVO;
    todos EPs ATIVO comparam seu valor com valor recém difundido;
    se (valor do EP) ≤ (valor recém difundido)
      então EP é rotulado como INATIVO;
    fimse;
  fimenquanto;
fim.
```

Figura 1 - Algoritmo MAX.

3.1 Análise do algoritmo MAX.

Note-se que, no algoritmo MAX, cada difusão é seguida por uma operação de comparação, executada em paralelo por alguns EPs. Portanto, é suficiente contar apenas o número de difusões, como medida de complexidade.

Seja C(A,B) a complexidade de tempo do algoritmo A no caso B, onde B pode ser: melhor, pior ou médio.

O melhor caso ocorre quando o valor máximo é o primeiro a ser difundido. Portanto, C(MAX,melhor) = 1.

O pior caso ocorre quando em cada passo k, k=1,2,...,n, o k-ésimo menor valor é difundido. Portanto, C(MAX,pior) = n.

No caso médio (sobre todas as instâncias do problema), o número de passos necessários para descobrir o valor máximo pode ser deduzido da seguinte forma [6]:

Durante a execução de MAX, a probabilidade do i-ésimo menor valor ser difundido é 1/i. Por exemplo, o terceiro menor valor só será difundido se, por acaso, ele for escolhido para ser difundido antes dos dois menores valores. Esse evento tem probabilidade 1/3. Portanto,

$$C(MAX,\text{médio}) = \sum_{i=1}^{n} 1/i$$

Esse resultado é denominado série harmônica, e seu valor, H(n), pode ser expresso por:

$$H(n) = \ln n + g + (1/2n) - (1/12n^2) + (1/120n^4) \ldots ,$$

onde g é a constante de Euler, g = 0,577. Para $n \geq 5$, H(n) é menor do que log n. Logo, C(MAX,médio) = O(log n). Outras formas de analisar o desempenho de MAX no caso médio, que conduzem ao mesmo resultado, são apresentadas em [2] e [5].

4 UM MULTICOMPUTADOR BASEADO EM BARRAMENTOS HIERÁRQUICOS.

As propriedades físicas de um barramento determinam um limite superior para o número de EPs que pode ser conectado àquele barramento. Quando um grande número de EPs deve ser conectado, um modo de contornar essa limitação é agrupar os EPs em "clusters" (grupos).

Uma máquina baseada em "clusters" envolve uma rede de interconexão hierárquica (RIH). Neste artigo, assumir-se-á uma RIH de dois níveis. No nível mais baixo, cada "cluster" possui um barramento "intracluster". No nível mais alto, os "clusters" estão conectados através de um barramento "intercluster". O projeto de um multicomputador baseado nessas idéias é descrito em [3] e [4].

O esquema baseado em "clusters" explora a localidade de comunicação [1], pois todos os barramentos "intracluster" podem estar sendo usados ao mesmo tempo.

Os algoritmos apresentados nas seções seguintes deste artigo são baseados nas seguintes considerações:

a) Cada "cluster" tem um "gateway" de barramento (GB), para interconexão com outros "clusters". O GB também é um EP, e pode armazenar um valor (veja item d). O uso de um EP no papel de um GB não aumenta significativamente o custo da máquina, desde que o número de EPs por "cluster" seja "suficientemente grande". Por exemplo, se a máquina tivesse 32 EPs por "cluster" e 32 "clusters" (1024 EPs no total), seriam necessários apenas 32 EPs adicionais no papel de GBs (o custo extra seria aproximadamente 3%).

b) Há uma linha de sinalização ligando cada EP ao GB do seu respectivo "cluster". Essas linhas têm a função de apoiar a comunicação hierárquica, através da sinalização bidirecional entre o GB e os EPs do seu "cluster". Há também uma linha de sinalização ligando cada GB ao hospedeiro. Essa estrutura é ilustrada na Figura 2.

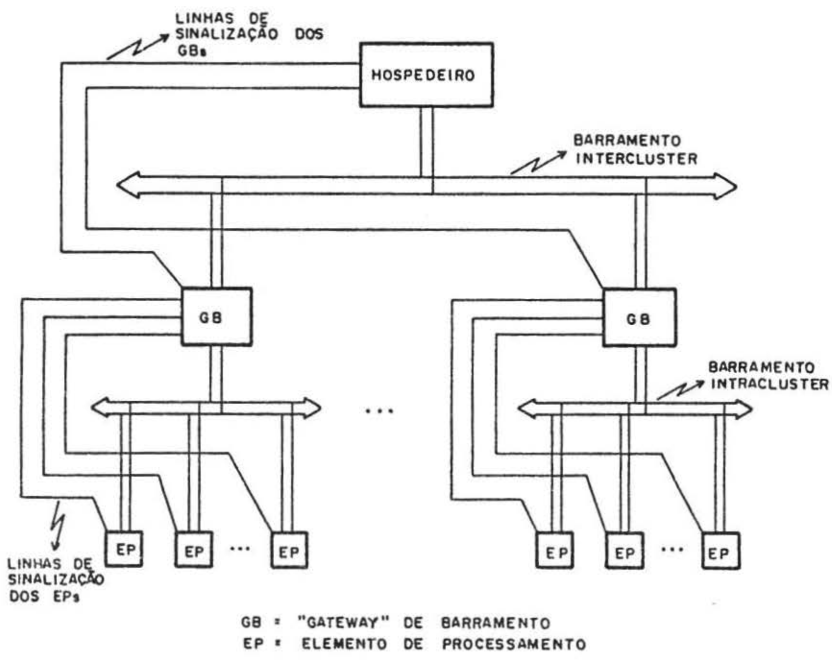


Figura 2 - Estrutura geral de um multicomputador baseado em barramentos hierárquicos.

c) Um valor difundido por um barramento "intracluster" pode ser lido apenas pelos EPs dentro do respectivo "cluster". Um valor difundido pelo barramento "intercluster" pode ser lido apenas pelos GBs. Um valor difundido pelo barramento "intercluster" e por todos os barramentos "intracluster" pode ser lido por todos EPs e GBs, e nesse caso a difusão é denominada global.

d) O número de EPs por "cluster" é o mesmo para todos os "clusters".
e) Sempre que ocorrer uma difusão "intracluster", o GB do respectivo "cluster" armazenará o valor recém difundido. Para economizar memória do GB, esse armazenamento pode destruir o valor previamente armazenado no GB, sem prejudicar a execução do algoritmo (já que o valor previamente armazenado sempre será menor do que o valor recém difundido).
f) A difusão global tem prioridade sobre as difusões "intercluster" e "intracluster". Em outras palavras, difusões "intercluster" e "intracluster" só podem ocorrer quando nenhum GB deseja fazer uma difusão global.

5 ADAPTAÇÃO DE MAX A BARRAMENTOS HIERÁRQUICOS.

A fim de explorar as possibilidades de paralelismo oferecidas por uma estrutura baseada em "clusters", será descrito, a seguir, MAX-BH, uma versão do algoritmo MAX adaptada para um multicomputador baseado em barramentos hierárquicos. MAX-BH consiste de duas fases. Na primeira fase, um algoritmo MAX é executado dentro de cada "cluster", até que cada "cluster" tenha descoberto seu máximo local. Na segunda fase, um algoritmo MAX é executado no nível "intercluster", usando os valores armazenados em cada GB (o máximo local do respectivo "cluster").

Em MAX-BH, cada GB pode assumir um dentre três estados:

ESPERANDO: O GB está esperando o término de MAX no seu "cluster".

ATIVO: O GB já conhece o máximo local do seu respectivo "cluster", e estará tentando difundir seu valor, se todos os outros "clusters" também já descobriram seu máximo local.

INATIVO: O GB já difundiu seu valor ou já "desistiu" (em virtude de ter recebido um valor maior).

Cada EP pode assumir um dentre dois estados, ATIVO ou INATIVO, como explicado na seção 2.

Inicialmente, todas as linhas de sinalização (tanto as dos EPs quanto as dos GBs) estão desligadas. Quando um EP tornar-se INATIVO, ele ligará sua linha de sinalização. Assim, quando todos os EPs de um "cluster" tiverem suas linhas de sinalização ligadas, o GB daquele "cluster" ficará imediatamente ciente desse fato, e se tornará ATIVO, ligando sua linha de sinalização.

Quando todos os GBs tiverem suas linhas de sinalização ligadas, o hospedeiro ficará imediatamente ciente desse fato e desligará a linha de sinalização de todos os GBs. Isso será entendido, pelos GBs, como uma ordem para iniciar a execução de um algoritmo MAX (no nível "intercluster"). A partir de então, o término de MAX-BH ocorre quando todos os GBs estiverem INATIVO, o que pode ser percebido de duas formas: detecta-se um silêncio no barramento "intercluster", ou o hospedeiro detecta que todas as linhas de sinalização dos GBs foram novamente ligadas.

O último valor difundido no nível "intercluster" é o valor máximo. MAX-BH é apresentado na figura 3.

```
inicio
  todos GBs são rotulados como ESPERANDO;
  todos EPs são rotulados como ATIVO;
  enquanto ( não (todos GBs estão ATIVO) ):
    em cada cluster, executar um algoritmo MAX entre seus EPs, e,
      quando MAX terminar, rotular o GB do cluster como ATIVO;
  fimenquanto;
  enquanto ( não (todos GBs estão INATIVO) ):
    executar um algoritmo MAX entre GBs, em vez de entre EPs;
  fimenquanto;
fim.
```

Figura 3 - Algoritmo MAX-BH.

Sejam #C e #EPC o número de "clusters" e número de EPs por "cluster", respectivamente. Obviamente, n = #C * #EPC.

O pior caso de MAX-BH ocorre quando:

a) Na fase "intracluster", pelo menos um "cluster" leva #EPC passos para descobrir seu máximo local; e

b) Na fase "intercluster", #C passos são necessários para descobrir o máximo global.

Seja T(A,B) o tempo de execução (medido pelo número de difusões executadas) de um algoritmo A no caso B (melhor, pior ou médio). Note-se que MAX-BH reduz consideravelmente o tempo de execução no pior caso, em comparação com MAX, pois T(MAX-BH,pior) = #EPC + #C, enquanto que T(MAX,pior) = #EPC * #C.

A complexidade de tempo de MAX-BH, em função de n, depende dos valores de #EPC e #C. Por exemplo, se #EPC = n/2 e #C = 2, então C(MAX-BH,pior) = n/2 + 2 = O(n). Por outro lado, fazendo-se #EPC = #C = $\sqrt{n}$, C(MAX-BH,pior) = O($\sqrt{n}$).

Porém, MAX-BH tem uma desvantagem: o número de difusões necessárias para o término da fase "intracluster" é o número de difusões executadas pelo "cluster" "mais lento", o qual é um gargalo. Isso tende a fazer com que T(MAX-BH,médio) seja maior do que T(MAX,médio). Esse algoritmo pode ser aperfeiçoado, conforme descrito na próxima seção.

6 UMA VERSÃO APERFEIÇOADA DE MAX PARA BARRAMENTOS HIERÁRQUICOS.

Nesta seção, será apresentado o algoritmo MAX-BH-A, (MAX para Barramentos Hierárquicos Aperfeiçoado), cujo princípio básico é executar fases alternadas de difusões "intracluster" e global. Na fase "intracluster", cada "cluster" trabalha apenas com valores locais, de modo que em cada barramento "intracluster" será difundido um valor diferente. Na fase global, em todos os barramentos "intracluster" será difundido o mesmo valor.

```
inicio
  todos GBs são rotulados como ESPERANDO;
  todos EPs são rotulados como ATIVO;
  enquanto ( não (todos GBs estão INATIVO) ):
    se (nenhum GB está ATIVO)
      então (* fase de difusão "intracluster" *)
        cada "cluster" cujo GB está ESPERANDO executa mais um
         passo do algoritmo MAX;
        em cada "cluster", se MAX terminar, rotular o GB do
         "cluster" como ATIVO;
      senão (* fase de difusão global *)
        todos GBs ATIVO tentam acessar barramento;
        um esquema de arbitragem decide qual GB ganha;
        o GB vencedor difunde globalmente (nos barramentos
         "intercluster" e "intracluster") seu valor;
        cada GB ATIVO:
          compara seu valor ao valor recém difundido;
          se (valor do GB) ≤ (valor recém difundido)
            então GB é rotulado como INATIVO;
          fimse;
        cada EP ATIVO:
          compara seu valor ao valor recém difundido;
          se (valor do EP) ≤ (valor recém difundido)
            então EP é rotulado como INATIVO;
          fimse;
        cada GB ESPERANDO:
          se (todos EPs do "cluster" estão INATIVO)
            então GB é rotulado como INATIVO;
          fimse;
    fimse;
  fimenquanto;
fim.
```

Figura 4 - Algoritmo MAX-BH-A.

Em MAX-BH-A, logo que algum "cluster" descobrir seu máximo local, o seu GB tentará fazer uma difusão global daquele máximo. A idéia chave deste algoritmo é usar o máximo local de um "cluster" para ajudar os outros "clusters" a descobrirem mais rapidamente seus máximos locais.

Em MAX-BH-A, da mesma forma que em MAX-BH, cada GB pode assumir um dentre três estados: ESPERANDO, ATIVO ou INATIVO; enquanto que cada EP pode assumir um dentre dois estados: ATIVO ou INATIVO. Em cada passo do algoritmo, é realizada uma difusão global ou "intracluster", dependendo de haver ou não, respectivamente, pelo menos um GB ativo. O algoritmo MAX-BH-A é apresentado na figura 4.

6.1 Análise de MAX-BH-A.

Em cada passo de MAX-BH-A, é executada, alternativamente, uma fase de difusão global ou uma fase de difusão "intracluster". Portanto, o tempo de execução desse algoritmo depende da soma: número de fases globais + número de fases "intracluster". Porém, deve-se levar em conta que a fase de difusão global é mais demorada do que a fase de difusão "intracluster".

O tempo para executar uma fase de difusão global é dado por:

$Tfase(global) = TA_{inter} + TD_{inter} + TC_{GB} + TD_{intra} + TC_{EP}$, onde

TA_{inter} = tempo para arbitragem no nível "intercluster";

TD_{inter} = tempo para realizar uma difusão "intercluster";

TC_{GB} = tempo para executar a computação (comparação) nos GBs;

TD_{intra} = tempo para realizar uma difusão "intracluster";

TC_{EP} = tempo para executar a computação (comparação) nos EPs.

Note-se que não há perda de tempo com uma arbitragem no nível "intracluster", pois a prioridade máxima da difusão global força imediatamente a difusão do valor recém difundido no nível "intercluster". Note-se também que, como o GB é um EP, $TC_{GB} = TC_{EP}$. Assumindo-se que a computação nos GBs pode ser feita em paralelo com a difusão "intracluster" e computação nos EPs, a computação nos GBs sempre terminará antes que a computação nos EPs. Assim, a equação anterior fica:

$Tfase(global) = TA_{inter} + TD_{inter} + TD_{intra} + TC_{EP}$.

O tempo para executar uma fase de difusão "intracluster" é dado por:

$Tfase(intra) = TA_{intra} + TD_{intra} + TC_{EP}$, onde

TA_{intra} = tempo para arbitragem no nível "intracluster";

TD_{intra} = tempo para realizar uma difusão "intracluster";

TC_{EP} = tempo de computação dos EPs.

$TA_{intra} = TA_{inter}$, já que está sendo assumido que a arbitragem é feita em tempo constante. Obviamente, TD_{intra} e TC_{EP} da fase "intracluster" são iguais a TD_{intra} e TC_{EP} da fase global.

Portanto, o tempo necessário para executar uma fase de difusão global é dado por:

$$Tfase(global) = Tfase(intra) + TD_{inter} .$$

Assim, para determinar-se o quanto a fase global é mais demorada que a fase "intracluster", deve-se determinar a taxa $X = TD_{inter}/Tfase(intra)$, cujo valor depende da máquina utilizada. Uma vez determinado X, o tempo da fase global é dado por:

$$Tfase(global) = Tfase(intra) + (X * Tfase(intra)),$$ ou seja

$$Tfase(global) = (1 + X) * Tfase(intra) .$$

A análise de MAX-BH-A será feita considerando-se quatro valores de X, a saber: 0,2; 0,4; 0,6; 0,8. Portanto, o tempo de execução de MAX-BH-A será dado por:

$$T(MAX\text{-}BH\text{-}A) = ((1+X) * \#fg) + \#fi,$$ onde

#fg = número de fases globais;

#fi = número de fases "intraclusters".

Também serão consideradas variações na taxa Y = #C/#EPC, a fim de estudar o valor ótimo de Y, para uma determinada quantidade total de EPs, n. Naturalmente, quanto maior for o valor de X, mais vantajoso parece ser um Y pequeno, para diminuir o número de fases globais. Por outro lado, um menor número de difusões globais implica em menor "ajuda" aos "clusters", que levariam mais tempo para determinarem seus máximos locais, reduzindo os benefícios da estrutura hierárquica.

Para analisar o desempenho de MAX-BH-A no caso médio, executaram-se simulações numéricas para vários valores de n e Y. Os resultados são mostrados na tabela 1. Para cada par de valores n e Y da tabela, os valores de #fg e #fi foram determinados executando-se 5.000 simulações do algoritmo, e calculando-se a média dos valores de #fg e #fi encontrados em cada simulação. As simulações foram executadas em um microcomputador 386.

Estudando-se a tabela 1, observa-se que:

a) Fixando-se Y e X, quanto maior for n, melhor o desempenho de MAX-BH-A, em comparação com MAX, conforme figura 5. Note-se que, a partir de um certo valor de n = P, a estrutura hierárquica possibilita um desempenho até mesmo melhor do que a estrutura "ideal" formada por um único barramento. Por exemplo, sejam Y = 1/4 e X = 0,2. Para n = 16, 64, 256, 1024 e 4096, o tempo de execução de MAX-BH-A em relação à de MAX é, respectivamente, 129% (4,36/3,38), 114%, 104%, 96%, 91%.

Tabela 1 - Desempenho de MAX e MAX-BH-A no caso médio.

n	T(MAX, médio)	Y	#fg	#fi	T(MAX-BH-A,médio)			
					X=0,2	X=0,4	X=0,6	X=0,8
16	3,38	1/4	1,49	2,57	4,36	4,66	4,95	5,25
		1	2,11	1,95	4,48	4,90	5,33	5,75
		4	2,70	1,49	4,73	5,27	5,81	6,35
64	4,74	1/16	1,50	3,71	5,51	5,81	6,11	6,41
		1/4	2,09	2,89	5,40	5,82	6,23	6,65
		1	2,75	2,21	5,51	6,06	6,61	7,16
		4	3,38	1,79	5,85	6,52	7,20	7,87
		16	4,03	1,50	6,34	7,14	7,95	8,75
256	6,12	1/64	1,49	4,92	6,71	7,01	7,30	7,60
		1/16	2,13	3,88	6,44	6,86	7,29	7,71
		1/4	2,77	3,02	6,34	6,90	7,45	8,01
		1	3,43	2,39	6,51	7,19	7,88	8,56
		4	4,09	1,97	6,88	7,70	8,51	9,33
		16	4,72	1,76	7,42	8,37	9,31	10,26
		64	5,42	1,48	7,98	9,07	10,15	11,24
1024	7,51	1/64	2,12	4,94	7,48	7,91	8,33	8,76
		1/16	2,74	3,93	7,22	7,77	8,31	8,86
		1/4	3,45	3,08	7,22	7,91	8,60	9,29
		1	4,15	2,47	7,45	8,28	9,11	9,94
		4	4,82	2,05	7,83	8,80	9,76	10,73
		16	5,43	1,90	8,42	9,50	10,59	11,67
		64	6,10	1,74	9,06	10,28	11,50	12,72
4096	8,90	1/16	3,48	3,89	8,07	8,76	9,46	10,15
		1/4	4,13	3,12	8,08	8,90	9,73	10,55
		1	4,84	2,50	8,31	9,28	10,24	11,21
		4	5,50	2,10	8,70	9,80	10,90	12,00
		16	6,13	1,96	9,32	10,54	11,77	12,99

Y = #C/#EPC
#fg = número médio de fases globais (p/ MAX-BH-A)
#fi = número médio de fases "intraclusters" (p/ MAX-BH-A)
X = tempo da difusão "intercluster"/tempo da fase "intracluster"

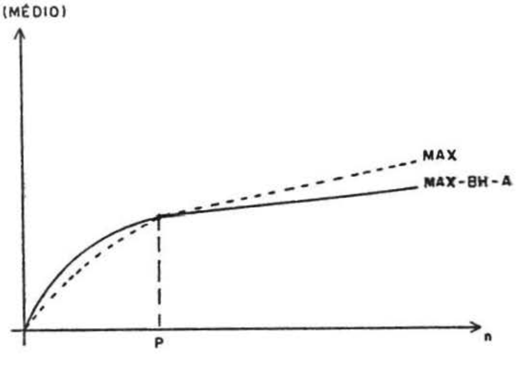
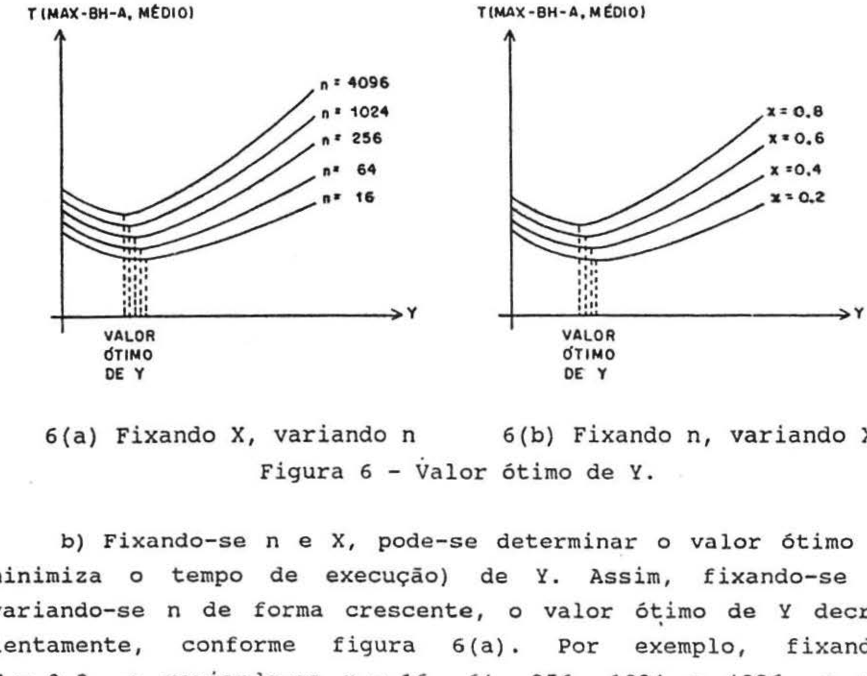


Figura 5 - Comparação entre o tempo de execução, no caso médio, de MAX e MAX-BH-A.



6(a) Fixando X, variando n   6(b) Fixando n, variando X

Figura 6 - Valor ótimo de Y.

b) Fixando-se n e X, pode-se determinar o valor ótimo (que minimiza o tempo de execução) de Y. Assim, fixando-se X e variando-se n de forma crescente, o valor ótimo de Y decresce lentamente, conforme figura 6(a). Por exemplo, fixando-se X = 0,2, e variando-se n = 16, 64, 256, 1024 e 4096, o valor ótimo de Y mantém-se aproximadamente igual a 1/4 para n = 16, 64 e 256. Para n = 1024, há dois valores ótimos: 1/4 e 1/16. Para n = 4096, o valor ótimo passa a ser 1/16. Analogamente, fixando-se n e variando-se X de forma crescente, o valor ótimo de Y também diminui, conforme ilustrado na figura 6(b). Por exemplo, fixando-se n = 256 e variando-se X = 0,2; 0,4; 0,6 e 0,8; o valor ótimo de X é, respectivamente, 1/4, 1/16, 1/16 e 1/64.

No pior caso, o tempo de execução de MAX-BH-A é semelhante ao de MAX-BH (#C + #EPC); bastando, para obter o valor exato de T(MAX-BH-A,pior), considerar o fator de correção X, da forma vista anteriormente.

7 CONCLUSÃO.

Os algoritmos apresentados neste artigo, para determinação do máximo, exploram as facilidades de difusão oferecidas pelo modelo de comunicação por barramento. Uma restrição desse modelo, entretanto, é a limitação do número de EPs que podem ser conectados a um barramento. Quando se deseja trabalhar com uma máquina altamente paralela, na qual uma grande quantidade de EPs são conectados, uma forma de contornar essa limitação é agrupar os EPs em "clusters", utilizando-se níveis hierárquicos de barramento.

Estudou-se a influência dessa estrutura hierárquica no projeto e desempenho de algoritmos para determinação do máximo. Como resultado desse estudo, apresentou-se o algoritmo MAX-BH-A, que explora vantajosamente os recursos oferecidos pela estrutura hierárquica de barramentos. MAX-BH-A explora o paralelismo de duas formas, permitindo que em alguns momentos os "clusters" trabalhem de forma independente entre si (explorando a localidade de comunicação), e, quando conveniente, os "clusters" compartilhem informações globais.

Mostrou-se, através de simulações numéricas, que, assintoticamente (para uma quantidade muito grande de EPs), desde que o tempo para difusão no barramento "intercluster" não seja muito longo (em comparação aos tempos para difusão "intracluster" e processamento nos EPs), o tempo de execução de MAX-BH-A no caso médio é até mesmo menor do que o tempo de execução de MAX (que considera a existência de um "enorme barramento" conectando todos os EPs, o que é tecnologicamente inviável). Quanto maior a quantidade de EPs da máquina, maior a eficiência de MAX-BH-A, em comparação a MAX. Além disso, MAX-BH-A reduz consideravelmente o tempo de execução no pior caso, em relação a MAX.

Estudou-se também a influência da variação da taxa #C/#EPC no desempenho de MAX-BH-A, tendo-se concluído que, desde que o tempo para difusão no barramento "intercluster" não seja muito longo, a taxa ótima é aproximadamente 1/4. Mostrou-se também que o valor ótimo dessa taxa diminui lentamente com um aumento no número de EPs da máquina.

REFERÊNCIAS BIBLIOGRÁFICAS.

[1] DANDAMUDI,S.P. & EAGER,D.L. Hierarchical interconnection networks for multicomputer systems. IEEE Trans. Comp., v. 39, n. 6, p. 786-797, Jun. 1990.

[2] DECHTER,R. & KLEINROCK,L. Broadcast communications and distributed algorithms. IEEE Trans. Comp., v. C-35, n. 3, p. 210-219, Mar. 1986.

[3] KIRNER,C. Design of a recursively structured parallel computer. In: PROCEEDINGS OF THE ACM-COMPUTER SCIENCE CONFERENCE - CSC, Louisville, Kentucky, USA, Feb. 1989.

[4] KIRNER,C. Arquitetura de Sistemas Avançados de Computação. In: XI CONGRESSO NACIONAL DA SBC - X JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA. Santos - SP, ago. 1991.

[5] LEVITAN,S.P. & FOSTER,C.C. Finding an extremum in a network. In: PROC. 9TH ANN. SYMP. COMP. ARCHITECT., Austin, Texas, USA, 1982. p. 321-325.

[6] YANG,C.-B; LEE,R.C.T.; CHEN,W.-T. Finding minimum spanning trees based upon single-channel broadcast communications. In: PROC. INT'L COMPUTER SYMPOSIUM, Taipei, Taiwan, 1988, p. 1451-1456.

[7] YANG,C-B; LEE,R.C.T.; CHEN,W-T. Parallel graph algorithms based upon broadcast communications. IEEE Trans. Comp., v. 39, n. 12, p. 1468-1472, Dec. 1990.

[8] YANG,C.-B. Reducing conflict resolution time for solving graph problems in broadcast communications. In: PROC. IASTED 9TH INT'L SYMP. APPLIED INFORMATICS, Innsbruck, Austria, Feb. 1991. p. 445-448.

[9] YANG,C-B; LEE,R.C.T.; CHEN,W-T. Conflict-free sorting algorithms under single-channel and multi-channel broadcast communication models. In: PROC. INT'L CONF. COMPUTING AND INFORMATION, Ottawa, Canada, May 1991. p. 350-359.