

IMPACTO DAS VARIAÇÕES DOS PARÂMETROS ARQUITETURAIS NUM MULTIPROCESSADOR COM 2-NÍVEIS DE HIERARQUIA DE BUS/CACHE

Edward D. M. Ordoñez
edmoreno@lsi.usp.br

Martha X. T. Delgado
mxttd@lsi.usp.br

Mario D. Marino
mario@lsi.usp.br

Sergio Takeo Kofuji
kofuji@lsi.usp.br

*Laboratório de Sistemas Integráveis
Departamento de Engenharia Eletrônica
Escola Politécnica da Universidade de São Paulo*

RESUMO

Este artigo apresenta o impacto das variações de alguns parâmetros arquiteturais no desempenho de um sistema multiprocessador baseado na hierarquia de 2-níveis de cache/bus. Tais parâmetros são: o tamanho de bloco de cache, o tamanho do cache, a largura de banda e a latência da rede de interconexão (estes dois últimos com o fim de determinar o impacto dos acessos remotos). Os resultados são obtidos com simulações comandadas por execução usando o Mint e alguns dos programas aplicativos do conjunto Splash e considerando como medidas de desempenho o TDF (Taxa De Falhas) e o CMPR (Custo Médio Por Referência), que permitem obter faixas de valores que otimizam o desempenho do sistema.

ABSTRACT

This document describes the impact some parameters changes on the performance of a multiprocessor system based on two-level hierarchical cache/bus. The specific parameters are: cache block size, cache size, bandwidth and latency of the interconnection network (this two-last parameters with intention the study the effects of remote access). To evaluate the behavior of the system, a multiprocessor simulator was designed to work together with the Mint program, and to incite system are used to Splash application programs. Using the CMPR (Cost Mean Per Reference) as an implicit measure of the program execution time and TDF (Miss Ratio), acquires values that optimized of the system performance.

1. INTRODUÇÃO

Na figura 1 mostra-se a arquitetura do sistema em análise, um sistema multiprocessador baseado na hierarquia de 2-níveis de cache/bus. Este sistema constitui um super-conglomerado (*supercluster*) que está conformado por vários aglomerados (*clusters*) interconetados através de um barramento (2-nível da hierarquia; indicado na figura como REDE). Cada conglomerado é formado por um número relativamente pequeno de processadores ($N=4,8$) que são interconectados através de um outro barramento (nível 1 da hierarquia, indicado na figura como rede) similarmente à arquitetura de estudo em outros trabalhos [Wils87, Vern88, Ande92]. Em [More94a] mostra-se detalhadamente que a coerência de memórias cache nesta arquitetura, é mantida usando o nosso protocolo de coerência baseado em invalidações, que é adequado para esta configuração pois aproveita as vantagens dos esquemas "snoopy" com caches espíões [Papa84]. As simulações aqui feitas correspondem à execução dos programas considerando um sistema com 32 processadores organizados em 4 conglomerados (8 processadores por conglomerado, pois encontrou-se que é a melhor organização para os programas aplicativos aqui considerados -estes resultados não são mostrados neste artigo), caches finitos de 64 Kbytes com mapeamento direto e tamanho de bloco de 64 bytes (pois encontrou-se através de simulações que este tamanho é adequado, ótimo para as aplicações consideradas) e o tamanho da rede de interconexão é de 32 bits.

Com os avanços tecnológicos que seguramente aumentarão a largura de banda das redes, e a tendência dos sistemas a serem organizados em vários níveis que aumentarão a latência (maiores acessos a nós remotos), é interessante avaliar como o sistema se comporta ante tais variações. Portanto, neste artigo mostra-se como o desempenho do sistema é influenciado com as variações dos parâmetros arquiteturais tais como: tamanho de bloco de cache, tamanho de cache, largura de banda da rede, latência da rede. Tal comportamento é analisado considerando o TDF (Taxa de Falhas) e o tempo de execução dos programas aplicativos (medido em ciclos de processador de 100 MHZ) medido através do CMPR (Custo Médio Por Referência).

Na seção 2 apresenta-se a metodologia de avaliação de desempenho, feita através de simulação comandada por execução e fornece as características dos programas usados. Explica-se também como são definidas as medidas de desempenho usadas, fundamentalmente descrevendo como é calculado o CMPR. Na seção 3 descreve-se como as variações do tamanho de bloco de cache afetam o desempenho do sistema, e similarmente na seções 4, 5 e 6 mostra-se os efeitos das variações da largura de banda, da latência e do tamanho de cache respectivamente. Finalmente, na seção 7 discutem-se as conclusões do trabalho e enumeram-se algumas tendências em trabalhos futuros.

2. METODOLOGIA DE AVALIAÇÃO

2.1 Simulação Comandada por Programa

Neste estudo decidiu-se fazer a avaliação de desempenho com simulação comandada por programa, usando o Mint (MIPS Interpreter) que é uma ferramenta que permite realizar simulação de arquiteturas fortemente acopladas e facilita a construção de simuladores de hierarquia de memória de sistemas multiprocessadores [Veen94].

Geralmente um simulador comandado por programa pode ser dividido em duas partes principais: um gerador de eventos e um simulador do sistema alvo. O gerador de eventos modela a execução de um programa aplicativo com um número determinado de

processadores. Quando o programa realiza uma operação de interesse (por exemplo uma referência a memória), o gerador transmite o evento ao simulador do sistema alvo. O simulador do sistema alvo modela a rede de interconexão e o sistema de memória do sistema em estudo determinando quais processos podem continuar sua execução baseado no momento em que cada evento se completa. Processo mostrado na figura 2. Um terceiro componente (não visível na figura) são as bibliotecas de simulação que gerenciam os eventos e os processos do programa em execução.

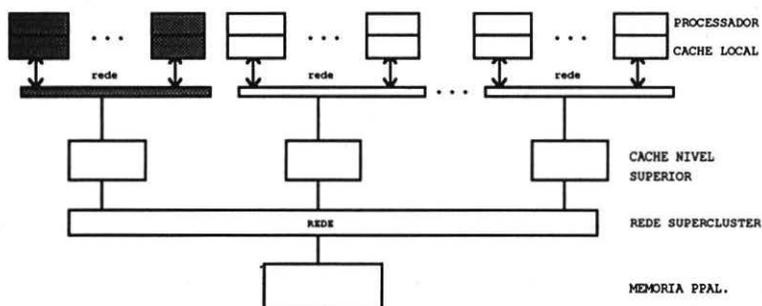


Figura 1. Sistema Multiprocessador Baseado em 2-Níveis de Hierarquia de Cache/Bus

Portanto, neste trabalho foi construído um simulador do sistema alvo -sistema multiprocessador de memória compartilhada baseado em hierarquia de barramentos e de memórias cache. O simulador foi escrito em linguagem C e executa conjuntamente com o Mint.



Figura 2 Análise por Simulação Comandada por Programa

2.2 Características dos Programas Usados

Neste trabalho usam-se um programa de multiplicação de matrizes do tipo inteiro (matmult: algoritmo simples) e três programas do conjunto Splash (códigos de aplicação que foram escritos para projetar e avaliar sistemas multiprocessadores de memória compartilhada): Barnes-Hut, Water e Mp3d -aplicações escritas em linguagem C [Sing91].

O programa BARNES-HUT, simula a evolução do comportamento de um sistema de corpos sob a influência de forças gravitacionais, ele é um problema clássico de simulação de N-corpos onde cada corpo é modelado como um ponto com massa que exerce forças sobre todos os outros corpos do sistema. A aplicação WATER avalia forças e potenciais num sistema de moléculas de água através da solução de equações de movimento usando um método predictor/corretor. MP3D simula fluxo rarefeito sobre um objeto usando a simulação com cinco graus de liberdade de moléculas individuais.

O programa Barnes-Hut foi simulado com os parâmetros típicos e considerando somente o comportamento de 128 corpos sob a influência de forças gravitacionais para 2 passos. O Mp3d foi simulado com 10000 partículas para 20 passos. O Water foi executado considerando um sistema com 512 moléculas para 2 passos de tempo e o

programa Matmult considera uma multiplicação de matrizes inteiras de 120x120. As características destes programas com 32 processadores são mostrados na tabela 1, onde as referências são classificadas como eventos de leitura e de escrita e em cada uma delas são consideradas referências a dados privados (existentes só num cache) e dados compartilhados. Os valores nas colunas correspondem ao número de referências e sua percentagem correspondente. Por exemplo, na coluna 3 do Barnes aparece 9.11 (44.09), isto significa que 9.11 Mbytes são referências a leituras compartilhadas o que corresponde a um 44.09 % das referências totais (20.66 Mbytes). Assim, pode-se observar que estes programas apresentam uma alta percentagem a dados compartilhados, o que faz com que sejam adequados em nossa avaliação.

Os tamanhos das entradas dos programas usados foram selecionados concordando com o tamanho do cache. Primeiramente foram determinados os tamanhos das entradas que pudessem ser simulados num tempo razoável e depois foram considerados diferentes tamanhos de caches. Finalmente o tamanho selecionado foi de 64Kbytes. Além das considerações anteriores foi considerado esse tamanho com o objetivo de não se enfatizar nas falhas produzidas por efeitos do tamanho do cache (substituições).

Tabela 1 Características das Referências a Memória dos Programas Usados

PROGRAMA APLICAÇÃO	Leitura Privada	Leitura Compartilhada	Escrita Privada	Escrita Compartilhada	Referências Total
Barnes-Hut	3.36(16.26)	9.11(44.09)	3.74(18.10)	4.31(20.81)	20.66 M
Mp3d	3.54(24.75)	5.82(40.69)	1.74(12.44)	3.15(22.02)	14.30 M
Water	10.61(21.14)	26.69(53.18)	5.78(11.51)	6.94(13.83)	50.18 M
Matmult	0.084	7.84(92.78)	0.034	0.015	8.45 M

2.3 Medidas de Desempenho

O impacto das variações dos parâmetros de interesse são visíveis considerando as variações na Taxa De Falhas e no Custo Médio Por Referência denominadas neste texto como TDF e CMPR respectivamente.

Taxa De Falhas (TDF): total de falhas/total de referências, é usada para quantificar o desempenho dos caches em sistemas monoprocessoadores e em sistemas multiprocessoadores. O TDF permite obter um limite superior da influência do tamanho de bloco de cache, da latência e da largura de banda da rede num determinado sistema de memória. Infelizmente, não é suficiente para caracterizar detalhadamente um determinado sistema, devido ao fato que em multiprocessoadores o tempo necessário para satisfazer uma falha não é constante. Isto é, o dado pedido pode encontrar-se em diferentes níveis da hierarquia de memória. Por exemplo, o dado pode encontrar-se num cache do mesmo conglomerado ou em algum outro cache de outro conglomerado e, obviamente, os tempos de acesso ao mesmo bloco são diferentes, afetando diferentemente ao sistema. Neste caso, a diferença no custo do acesso depende das características da rede de interconexão: latência e largura de banda.

Custo Médio Por Referência (CMPR): Esta medida implicitamente determina o tempo de execução da aplicação, pois é definida como o tempo (normalizado em unidades de ciclos de processador, considera-se um processador de 100 MHz) que o processador espera por um determinado dado no caso de não contenção. Por exemplo, acessos a uma referência privada satisfeitos pelo cache local (sucessos *-hit*) possuem um CMPR de 1 (toma um ciclo de processador para ser completado). No caso

de acesso a um dado compartilhado existente no cache local, é necessário estabelecer ações de coerência e o CMPR associado a esta operação será diferente de 1. No caso de falhas, o processador fica esperando até que todos os dados procurados sejam transmitidos. Neste caso, o CMPR considera o tempo de busca de dados, o tempo na transmissão pela rede de interconexão, o tempo necessário para estabelecer as operações de coerência, etc. Esta medida é importante porque depende dos acessos remotos, que por sua vez dependem das características da rede de interconexão (neste caso, dos barramentos): latência e da largura de banda. Considera-se que:

$$CMPR_{total} = CMPR_{hit} + CMPR_{substituição} + CMPR_{coerência}$$

O CMPR depende das ações de coerência (falha de leitura, escrita a dado compartilhado ou privado, etc). Por exemplo, o CMPR associado a uma falha de leitura a um dado compartilhado é:

$$CMPR_{coerência} \text{ (dado no mesmo conglomerado)} = \text{Latência da rede} + \text{Tamanho de bloco} * \text{Fator de Largura de Banda}^1$$

ou

$$CMPR_{coerência} \text{ (dado num outro conglomerado)} = 2 * CMPR_{mesmo conglomerado} + (\text{Latência da rede nível superior} + \text{Tamanho bloco} * \text{Fator banda superior})$$

No caso de uma operação de substituição,

$$CMPR_{substituição} = \text{Latência} + \text{Tamanho do bloco} * \text{Fator banda do barramento } i$$

Expressões similares podem ser obtidas para cada uma das transações de coerência [More94b].

3. EFEITO DA VARIAÇÃO DO TAMANHO DO BLOCO DE CACHE

Os resultados aqui mostrados foram obtidos através de simulações considerando uma latência da rede constante (50 ciclos de processador) e uma largura de banda infinita com o objetivo de por considerar que o mínimo valor no TDF representa um limite superior no tamanho do bloco do cache. Com estas considerações, este valor é o tamanho ótimo. Blocos maiores podem possuir um maior TDF e um maior CMPR enquanto que blocos menores possuem um maior custo na transmissão da mesma quantidade de dados. As figuras 3 a-d apresentam as variações do TDF de cada uma das aplicações em função do tamanho de bloco de cache. Nestas figuras, o TDF é a soma das percentagens de falhas classificadas como: partida a frio, substituições, falso e real compartilhamento e por pedidos exclusivos. Na figura 3, aparecem como cold, subs, false, true e excl respectivamente. Esta classificação é similar aos trabalhos de Bianchini e Dubois [Bian94, Dubo92].

A figura 3.a mostra o comportamento das falhas para a aplicação Barnes-Hut. O menor valor de TDF ocorre para um tamanho de bloco de 64 bytes (TDF=0.15%). Blocos maiores produzem um maior número de falhas produzidas essencialmente por substituições e por falso compartilhamento; é importante observar que as outras categorias sofrem uma pequena diminuição com o aumento do tamanho do bloco do cache. Com um tamanho de bloco pequeno e igual a 8 bytes, o TDF é de 2.97% (maior valor) onde as categorias de maior peso são as partida a frio e as exclusivas.

¹ Estes últimos valores são obtidos com a seguinte expressão:

Largura de Banda [Mbytes/s] = Tamanho da rede (bytes)*Frequência Processador [Mhz]/Fator de Banda

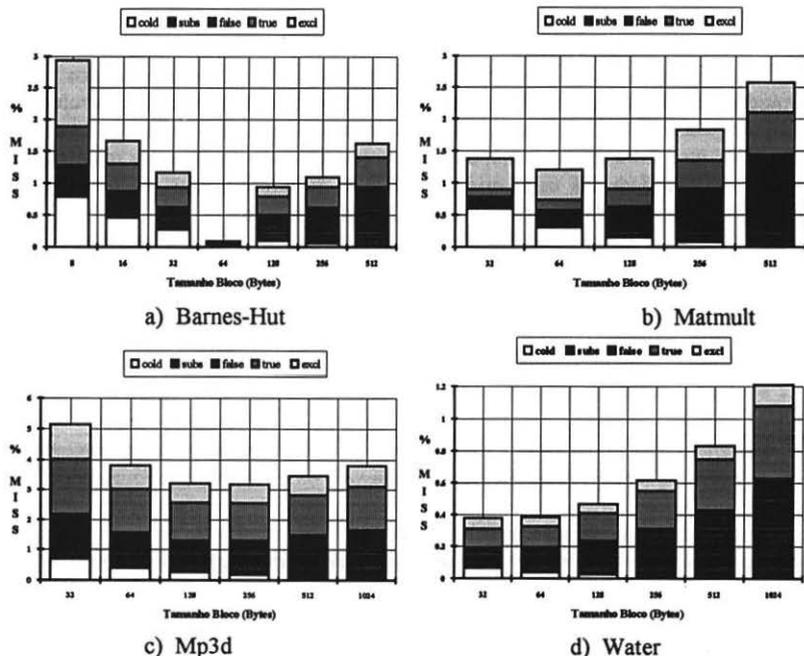


Figura 3 Comportamento da Taxa de Falhas

A figura 3.b mostra o comportamento do TDF da aplicação Matmult. O mínimo valor de TDF acontece com um bloco de 64 bytes (TDF = 1.38%). Existe uma certa similaridade do comportamento para os valores 32-128 bytes (TDF varia entre 1.38 e 1.44%) devido a boa localidade espacial e de processador deste programa aplicação. Esta característica também é manifestada no fato de possuir poucas falhas por partilha falsa (ainda para blocos grandes) e ao aumento das falhas por partilha real. A maior contribuição ao TDF para blocos maiores a 128 bytes é pelas substituições.

Na figura 3.c mostram-se as variações de TDF para o programa Mp3d. O menor TDF é obtido com blocos de 128 bytes e muito próximo também para 256 bytes. Nesta aplicação a maior contribuição ao TDF é produzida pelas falhas de verdadeiro compartilhamento (a qual é mantida quase constante para todos os valores de tamanho de bloco) o que mostra como a sobrecarga destas falhas (relacionados ao padrão de compartilhamento) é o seu maior limitante. As falhas por substituições não aumentam consideravelmente mostrando assim que existe uma boa localidade de referências (espacial e temporal) neste programa.

A figura 3.d mostra a distribuição das falhas no programa Water, onde é mais visível o fato de que o falso compartilhamento aumenta com o tamanho de bloco de cache. Neste programa, as falhas relacionadas com o compartilhamento e com as substituições devidas à pouca localidade temporal dominam o comportamento do TDF. O melhor tamanho de bloco corresponde a um bloco de 32 bytes e possivelmente a um bloco de 64 bytes.

4. EFEITO DA VARIAÇÃO DA LARGURA DE BANDA DA REDE

Uma boa análise não somente deve-se limitar a considerar como melhor tamanho de bloco aquele obtido no menor valor do TDF. Senão, deve também relacionar como o custo dos acessos a nós remotos produzidos nos casos de detecção de falhas influenciam na seleção. Esta análise é importante porque considera o custo de resolver uma falha, especificamente relacionando o efeito das características da rede de interconexão que influenciam na penalidade das falhas (*miss penalty*)².

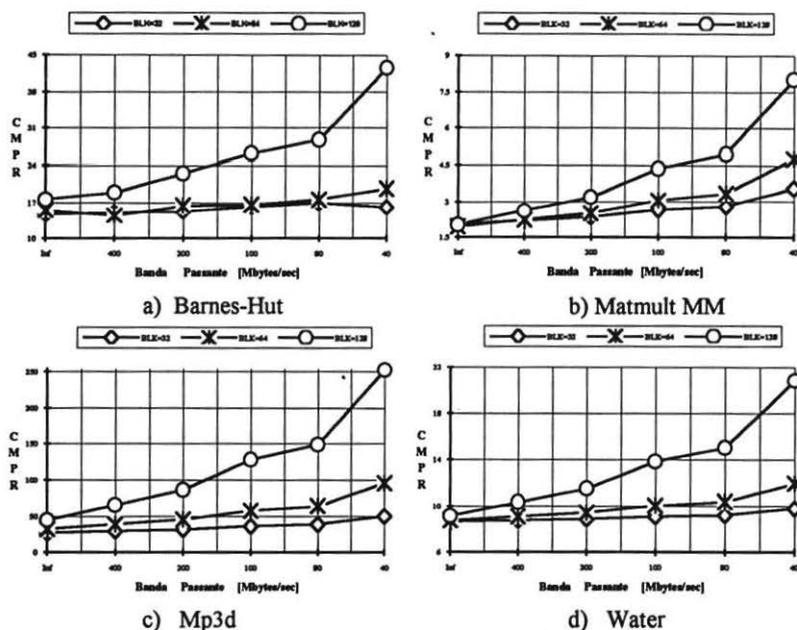


Figura 4 Comportamento do CMPR com variações na Largura de Banda

As figuras 4 a-d apresentam os resultados das simulações onde a latência foi mantida constante (50 ciclos) e se gráfica o CMPR para cada uma das aplicações como uma função da largura de banda da rede que varia entre Infinita-40 Mbytes/s. Foram feitas simulações para diferentes valores de tamanho de bloco 16-1024 bytes, mas só foram graficados aqueles que permitiram obter um bom TDF e um bom CMPR (blocos de 32-128 bytes).

A figura 4.a. ilustra o comportamento do programa Barnes-Hut. Independentemente da largura de banda, tamanhos de 32 e 64 bytes apresentam um menor CMPR que caracteriza-se por ser quase constante. Isto significa, que um bom tamanho de bloco de cache em termos do CMPR poderia ser também 32 bytes (mas deve-se ter em mente existe uma diferença de 1.12% ao 0.16% no TDF que corresponde

² **Penalidades de Falhas:** tempo para substituir um bloco do cache ativo com o bloco trazido do sistema de memória mais o tempo de transferência do bloco ao cache solicitante que depende logicamente do tempo de acesso à primeira palavra (relacionado com latência) e com o tempo de transferência (tempo adicional para transferir as palavras restantes do bloco) relacionado com a largura de banda.

a uma diferença da ordem do 86.66%. Para blocos maiores a 128 bytes, existe uma diferença de 82.22% respeito ao melhor (blk=64 bytes), mas com o CMPR mostra-se que este tamanho produz um maior CMPR e o seu comportamento é ainda mais crítico com redes lentas. Por exemplo, com bandas de 40 Mbytes/s aparece uma degradação de 51.19% no CMPR o que leva a pensar na utilização de redes rápidas com um tamanho razoável de bloco de cache. Neste programa, bandas altas (400-Infinita) podem favorecer a transmissão de grandes tamanhos de blocos (o que outros trabalhos afirmam). Acredita-se que não adianta usar redes (barramentos) com estes blocos grandes, pois a diferença não é muito significativa. Por exemplo, com blk=128 bytes a diferença obtida nestes dois valores de bandas foi de 6.45%.

A figura 4.b mostra as variações do CMPR para o programa Matmult. Determina-se o melhor tamanho de bloco é 32 ou 64 bytes, independentemente da largura de banda. Para valores de banda muito baixos, o melhor tamanho seria 32 bytes (apesar que este tamanho não foi o melhor segundo o critério do TDF pois existe uma diferença do 13.04%). TDF mostrou que o comportamento era similar para blocos de 32-128 bytes. Bandas lentas e blocos grandes não oferecem uma boa escolha, pois ainda com b=128 bytes e bandas de 40 Mbytes/s a diferença em CMPR é de 57.51%. Nestes casos a largura de banda consegue um maior impacto no CMPR, assim uma melhoria na banda num fator de 10 (400 a 40 Mbytes/s) produz uma melhoria do CMPR em 76.02% (e portanto no tempo de execução).

Na figura 4.c mostram-se as tendências do CMPR para o programa Mp3d. Uma banda infinita favorece o tamanho de 32 ou 64 (que possui o melhor TDF). Não obstante, para bloco de 128 tenha sido o melhor segundo o critério do TDF, com o CMPR preferiria-se um valor menor que poderia ser de 32 ou 64 apesar da diferença observada no TDF seja de 38.23% e 17.12% respectivamente.

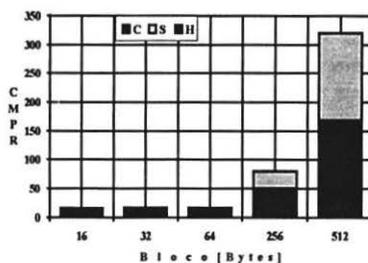
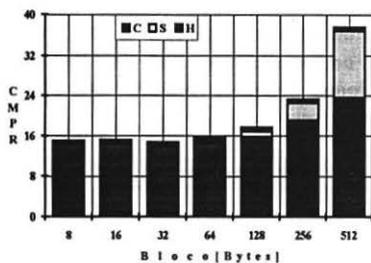
Na figura 4.d descreve as variações no CMPR no programa Water. Nesta figura visualiza-se que os melhores tamanhos são 32 e 64 bytes (quase em todas as bandas possíveis). Neste caso, a diferença não é muita pois para bandas lentas (40 Mbytes/s) é de 16.66%. O uso de tamanhos maiores é razoável até bandas da ordem de 200 Mbytes/s onde a diferença é de 17.79%.

Finalmente, devemos salientar o impacto da largura de banda no desempenho do sistema através da figura 5, onde a parte 5.a mostra os relativamente pequenos valores de CMPR para banda infinita e a parte 5.b quando a banda é mais real, de um sistema com banda de 80 Mbytes/s. Nestas figuras, observam-se três categorias do CMPR: pelas ações de coerência (C), pelas substituições (S) e pelos sucessos (H); e claramente visualiza-se que o CMPR por coerência é dominante (aproximadamente o 80%) e que as substituições começam a ser significativas quando consideram-se tamanhos de bloco maiores ao tamanho do ponto de "poluição" da memória.

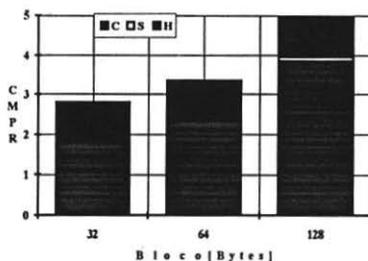
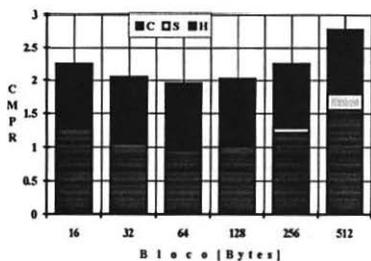
4. EFEITO DA VARIAÇÃO DA LATÊNCIA DA REDE

Os resultados anteriores mostraram como o desempenho de uma certa organização da máquina varia com o tamanho de bloco e com a largura de banda da rede de interconexão mantendo a latência constante e de valor 50 ciclos. Não obstante, este valor seja representativo em multiprocessadores escaláveis, considera-se que é importante mostrar como os resultados obtidos podem mudar com as suas variações. Os resultados aqui apresentados referem-se a valores de latência maiores (50,100 e 150) devido ao fato deles ser mais reais nos casos de máquinas com um maior número de nós organizados em vários níveis (sistemas de maior porte).

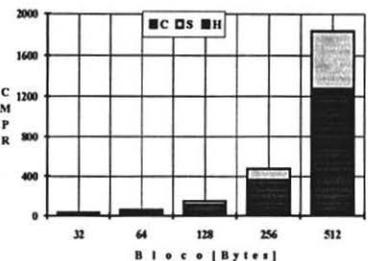
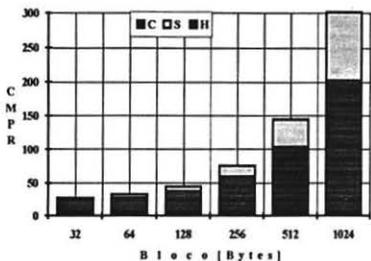
BARNES-HUT



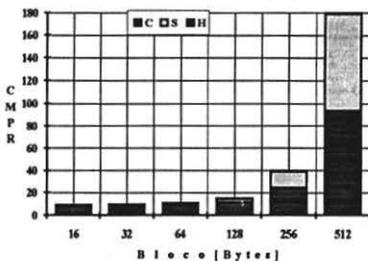
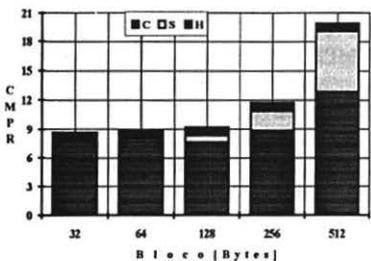
MATMULT



MP3D



WATER



a)

b)

Figura 5 Distribuição do CMPR, Cache=64 Kbytes
a) Banda Infinita b) Banda 80 Mbytes/s

O fato de considerar maiores latências faz com que seja mais importante fazer reduções na taxa de falhas, já que existe um maior custo associado com cada falha. Blocos maiores podem ser mais úteis quando temos maiores latências já que uma pequena diminuição no TDF pode-se tornar mais importante, pois uma transmissão pode-se fazer com uma latência só, enquanto blocos pequenos precisariam várias transmissões equivalentes a se ter uma latência maior.

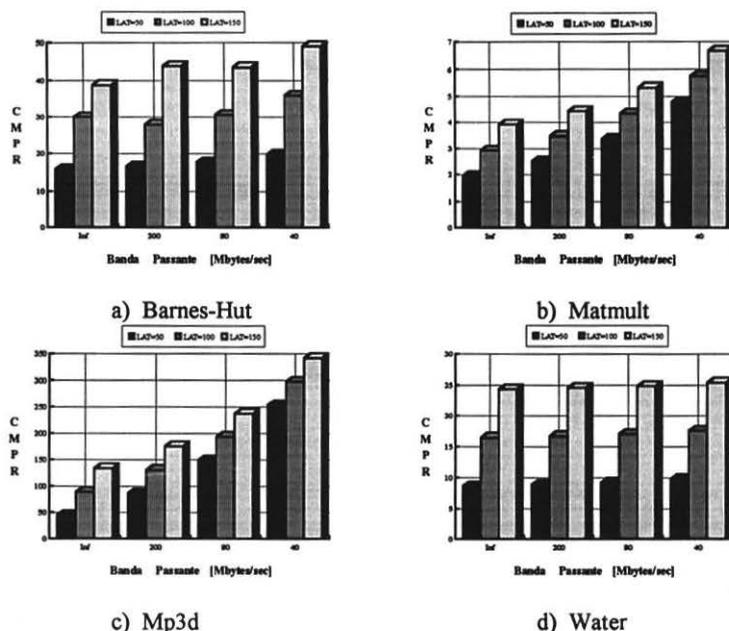


Figura 6 Efeito das Variações da Latência da Rede de Interconexão

As figuras 6a-d mostra os resultados obtidos caso de um sistema com $N=32$ (N : Número de processadores no sistema) e $NC=4$ (NC : Número de Conglomerados) e tamanho de bloco ótimo para cada aplicação (encontrado segundo o TDF da figura 3) e latência variável 50,100 e 150 ciclos (indicado nas figuras como $LAT=valor$) em função da largura de banda.

Na figura 6.a se visualizam as variações do CMPR do programa Barnes-Hut com bloco de cache de 64 bytes. Com banda infinita, aumentando a latência a 100 e 150 se produz um aumento do CMPR de 48.38% e 59.5% respectivamente. Esta diferença é mantida quase constante em todas as variações do valor da largura de banda.

Na figura 6.b se mostram as mudanças do CMPR para o programa Matmult com bloco de cache de 64 bytes. Nesta caso, as variações respeito à latência de 50 são da ordem de 33.33% e 48.75% para as latências de 100 e 150 respectivamente, considerando uma banda infinita.

Na figura 6.c aparecem as variações no CMPR para o programa Mp3d com bloco de 128 bytes. Nesta aplicação a maior diferença corresponde a uma banda de 40 Mbytes/s, onde as diferenças do CMPR são 16.66% e 28.57% para latências de 100 e 50

respectivamente. Não obstante, o gráfico da figura 6.2.c mostre que a tendência do CMPR para o bloco de 128 bytes seja aumentar com as diminuições da banda, neste 6.4.c mostram-se que as variações do CMPR com a latência não mudam muito, pois este tamanho de bloco relativamente alto é beneficiado com latências maiores já que um maior número de blocos podem ser transferidos com o mesmo valor da latência.

Finalmente, na figura 6.d aparecem as variações do programa Water para um tamanho de bloco de 32 bytes, onde pode-se observar que o CMPR não varia muito com os diferentes valores da largura de banda e que os valores de CMPR mudam num 45.45 e 64 % para latências de 100 e 150 respeito à latência de 50 ciclos de processador. Este comportamento explica-se pelo fato que a inclinação do gráfico 6.d para este tamanho de bloco é quase de 0 (paralela com o eixo das abcisas), idem para o programa Barnes-Hut (com a figura 6.a).

Na tabela 3 mostram-se as variações do CMPR considerando os tamanhos de na faixa de 32-128 bytes, com diferentes valores banda (Infinita, 200, 80 e 40 Mbytes/s) para vários valores de latência (50,100 e 150 ciclos).

Tabela 3 Dados CMPR para Vários Valores de Latência, Tamanhos de Blocos e Largura de Banda

Programa Barnes-Hut					Programa Matmult					
Banda\ Latência	Infinita	200	80	40	Infinita	200	80	40		
50	(32)	14.78	15.35	16.98	16.18	(32)	2.04	2.35	2.79	3.58
	(64)	15.66	16.44	17.59	19.73	(64)	1.95	2.51	3.35	4.78
	(128)	17.63	22.53	28.75	42.62	(128)	2.04	3.20	4.95	8.03
100	(64)	29.95	27.87	30.58	35.85	3.09	3.39	3.84	4.60	
	(128)	31.80	37.80	43.93	57.74	2.92	3.47	4.33	5.74	
						3.06	4.22	6.03	9.08	
150	(64)	38.76	43.79	43.46	49.02	4.13	4.43	4.90	5.67	
	(128)	51.88	54.84	65.52	76.46	3.90	4.42	5.32	6.66	
						4.09	5.25	7.00	9.95	

Programa MP3D					Programa WATER					
Banda\ Latência	Infinita	200	80	40	Infinita	200	80	40		
50	(64)	32.74	45.32	64.15	96.07	(32)	8.66	8.89	9.23	9.79
	(128)	45.04	86.52	149.09	252.84	(64)	8.79	9.42	10.37	11.94
	(256)	75.40	155.38	475.30	878.04	(128)	9.16	11.50	15.02	20.86
100	(128)	89.40	130.94			16.47	16.69	17.03	17.59	
	(256)	149.35	310.18			16.72	17.35	18.30	19.21	
						17.45	19.80	23.31	28.59	
150	(128)	133.87	175.13	237.04	341.82	24.27	24.50	24.83	25.40	
	(256)	224.60	385.36	627.58	931.14	24.65	25.28	26.43	29.72	
						25.74	28.07	31.64	38.86	

5. EFEITO DA VARIAÇÃO DO TAMANHO DE CACHE

Aumentar o tamanho de cache é uma técnica importante para melhorar a utilização dos barramentos, é melhor que aumentar o tamanho de bloco ou a associatividade; mas deve-se ter em mente que os tempos de acesso podem ser maiores e portanto não ajudar significativamente no desempenho do sistema [Przy88].

Assim, nas figuras 7a-d mostram-se as variações no TDF das aplicações executadas com caches locais, na hierarquia de memória, de tamanho 128 Kbytes. Pode-se visualizar como o TDF é diminuído respeito aos valores com cache de tamanho

64 Kbytes, especialmente a categoria das falhas por substituições e consequentemente o CMPR destes programas. Estas mudanças são mostradas na figura 7 e correspondem a uma largura de banda infinita, latência de 50 ciclos para o melhor tamanho de bloco em cada aplicação. O valor total do CMPR é classificado similarmente à da figura 3, só que nesta figura as divisões são mostradas como C, S, F, T e E, similar ao figura 3, só que para caches de tamanho 128 Kbytes.

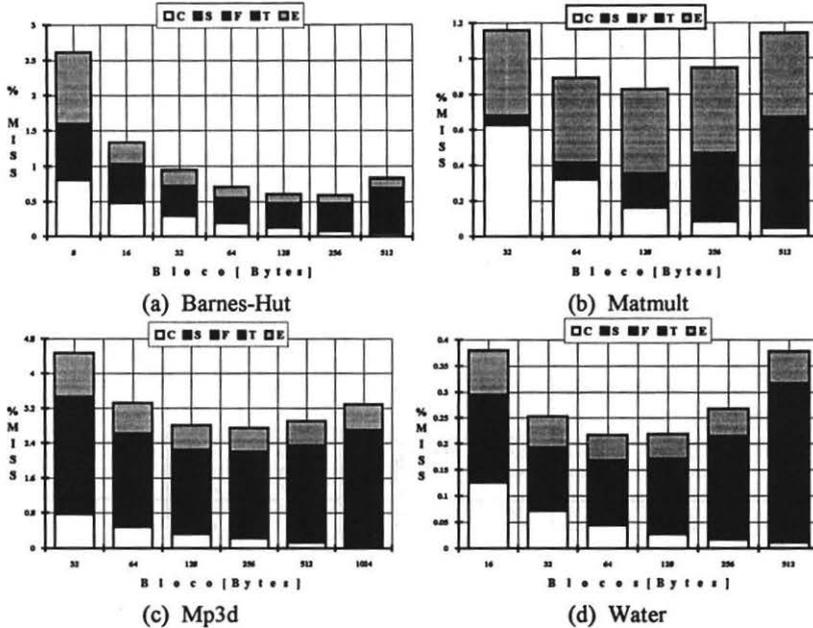
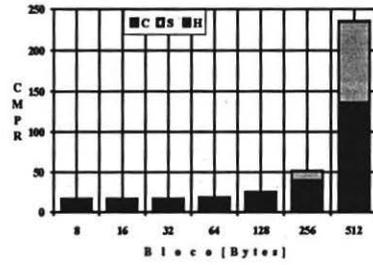
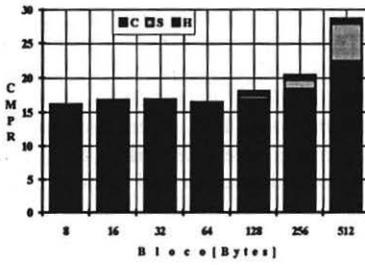


Figura 7 Taxa de Falhas dos Programas com Caches de 128 Kbytes

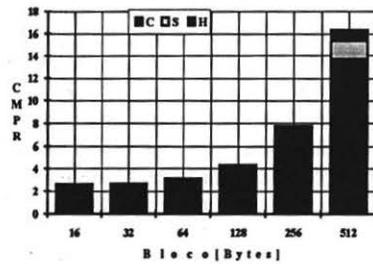
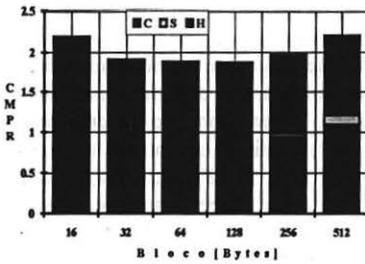
Na figura 7 pode-se visualizar que o TDF segue um comportamento similar ao de sistemas monoprocessadores, onde aumentando o tamanho de cache se reduz a percentagem de falhas até o ponto de poluição da memória que é deslocado a um bloco de tamanho maior. Assim obtém-se que o melhor tamanho de bloco é de 128 ou 256, 128, 256 e 64 bytes para o programa Barnes-Hut, Matmult, Mp3d e Water respectivamente.

Finalmente, é bom lembrar que não necessariamente as diminuições do TDF com o aumento do tamanho de cache são suficientes para decidir se esta variação produz uma melhoria no desempenho do sistema, pois é necessário observar também o comportamento através do CMPR (figura 8); onde pode-se visualizar que existe um ganho total 12.85%, 18.09%, 26.52%, 9.60% para banda infinita e 7.25%, 5.97%, 9.75% e 3.65 % para banda 80 Mbytes/s com blocos na faixa de 32-128 bytes para os programas Barnes, Matmult, Mp3d e Water (segundo as figuras 5 e 8).

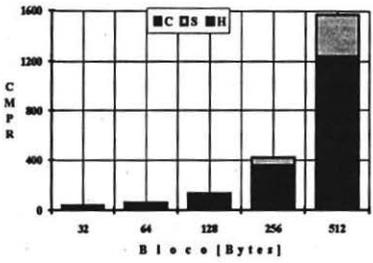
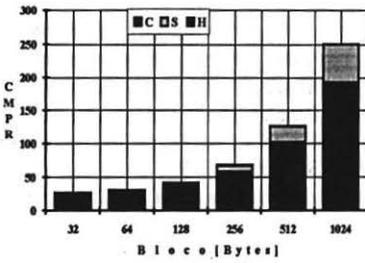
BARNES-HUT



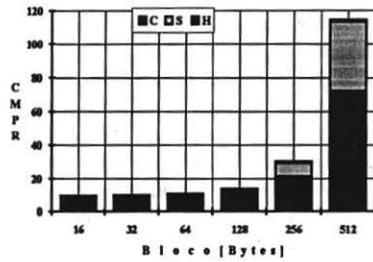
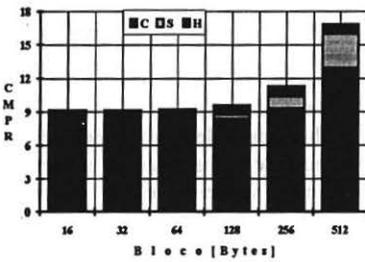
MATMULT



MP3D



WATER



a) Banda Infinita

b) Banda 80 Mbytes/s

Figura 8 Distribuição do CMPR, Cache=128 Kbytes

6. CONCLUSÕES

Finalmente e resumindo pode-se mencionar que o melhor tamanho de bloco não é único e que depende da aplicação. Existe uma faixa valores razoáveis que apresentam um baixo valor de TDF, blocos de 32-128 bytes. Também pode-se dizer que não existe uma categoria específica de falhas que domine o tamanho de bloco já que todas oferecem contribuições significativas no valor total na taxa de falhas e que dependendo da aplicação uma ou várias delas dominaram o comportamento do TDF. Além disso, pode-se salientar que grandes tamanhos de blocos (maiores que 128) não são muito convenientes devido ao fato de produzir um aumento considerável do TDF total e que as categorias dominantes nestes casos são relacionados ao compartilhamento (falso e real) e ao tamanho do cache (substituições). Assim os nossos resultados discordam com os de outros trabalhos existentes onde para algumas aplicações é justificado o uso de grandes tamanhos de bloco de caches. Evidenciando assim que neste sistema (baseado em hierarquia de barramentos) não é recomendável o uso de grandes tamanhos de blocos.

Finalmente é importante salientar que ainda com grandes valores de largura de banda não é justificável o uso de grandes tamanhos de blocos de caches (superiores a 128 bytes). Acreditamos que com os avanços tecnológicos que poderiam melhorar as características dos barramentos poder-se-á melhorar o desempenho do sistema pois a maior diferença entre uma banda alta (400 Mbytes/sec) e infinita encontrada em nossas simulações para um tamanho de bloco de 128 bytes foi de 10.52%, 22.02%, 23.8% e 10.78% para os programas Barnes, MM, Mp3d e Water respectivamente. Assim, concluímos que poderia-se pensar em usar redes na faixa de 80 a 400 Mbytes/s pois o melhoramento máximo é da ordem de 30% no desempenho total do sistema, pois em nossas considerações obtimos valores de 30.9%, 56.1%, 53.33% e 30% para nossos programas.

Também deve-se analisar como a latência da rede de interconexão influencia no casamento entre uma aplicação determinada e o tamanho ótimo de bloco de cache. Programas diferentes preferem tamanhos de caches diferentes e o tamanho favorável em cada aplicação muda com o valor da largura de banda. Para grandes bandas passantes da ordem de 400 Mbytes/s então tamanhos até de 128 bytes e para valores médios da ordem de 80 Mbytes/s então pequenos tamanhos de caches (32 bytes). Mas, em geral, nesta classe de sistemas é preferível usar sempre um tamanho de bloco na faixa de 32-64 bytes, pois apresentam sempre um menor CMPR.

Observando as figuras 3 e 7 pode-se dizer que as aplicações apresentam um compartilhamento de grão fino, pois o TDF aumenta com as diminuições do tamanho de bloco e os seus valores são compostos pelo aumento do tráfego no barramento produzidos pelo fato de transferir grandes unidades o que produz um aumento no tráfego pela rede de interconexão.

Aumentando o tamanho de bloco, a percentagem de falhas produzidas por substituições diminui. Mas, os dados compartilhados podem ficar nos caches por maiores períodos de tempo, tendo assim uma maior probabilidade de serem invalidados e portanto algumas das falhas que antes eram por capacidade (substituições) podem-se tornar falhas por invalidações (razão pela qual é mais visível o efeito do falso compartilhamento com um maior tamanho de cache). Por exemplo, considerando banda infinita e tamanho de bloco de 512 bytes (onde o número de falhas por substituições foram altas em ambos os valores do caches -64 e 128 Kbytes), observa-se que aconteceu uma diminuição num 8.09%, 24.69%, 9.61% e 11.05% nas falhas por substituições mas uma diferença não tão marcada nas invalidações refletidas nas poucas diferenças dos

valores pertencentes às ações de coerência (mostradas na figura 8).

O aumento no custo de compartilhamento com tamanhos de caches maiores é um problema associado aos protocolos de coerência baseados em atualizações [Egge89] e neste trabalho observa-se que tal problema acontece também com protocolos baseados em invalidações.

7. REFERÊNCIAS

- [Ande92] ANDERSON, C.; BAER, J.L. A Multi-Level Hierarchical Cache Coherence Protocol for Multiprocessors. **Technical Report**, University of Washintong, 1992
- [Bian94] BIANCHINI, R.; LeBLANC T. A Preliminary Evaluation of Cache-Miss Initiated Prefetching Techniques in Scalable Multiprocessors. **Technical Report No. 515**, University of Rochester, p. 25, May, 1994.
- [Dubn93] DUBNICKY, C. The Effects of Block Size on the Performance of Coherent Caches in Shared-Memory Multiprocessors. **PhD. Thesis**, DCS, University of Rochester, New York, 1993.
- [More94a] MORENO, O. E.D. Protocolo Snoopy Adaptativo para Sistemas Multiprocessadores Baseados em Hierarquia de Barramentos. **VI SBAC-PAD**, Simpósio Brasileiro de Arquitetura de Computadores - Processamento de Alto Desempenho, Caxambú - M.G., p. 173-184, Ago 1994.
- [More94b] MORENO, O. E.D. Efeito da Coerência de Cache e da Pré-Busca num Sistema Multiprocessador de Memória Compartilhada. **Dissertação de Mestrado**, Laboratório de Sistemas Integráveis da Escola Politécnica da Universidade de São Paulo USP, Nov, 1994.
- [Papa84] PAPAMARCOS, M.; PATEL, J. A Low Overhead Coherence Solution for Multiprocessors with Private Cache Memories. In: Ann. Int. Symp. on Computer Architecture, IEEE, 11th. **Proceedings**, p. 348-54, 1984.
- [Pryz90] PRYZBYLSKI, S. The Performance Impact of Block Sizes and Fetch Strategies. In: Ann. Int. Symp. on Computer Architecture, IEEE, 17th. **Proceedings**, p. 160-69, 1990
- [Sing91] SINGH, J.P.;WEBER, W.D.; GUPTA, A. Splash: Stanford Parallel Applications for Shared Memory. **Technical Report CSL-TR-91-469**, Stanford University, Apr. 1991.
- [Veen94] VEENSTRA, J.E; FOWLER, R.J. MINT Tutorial and User Manual. **Technical Report 452**, University of Rochester, Computer Science Department, New York, p. 52, June 1993.
- [Vern88] VERNON, M.K.; LAZOWSKA, E.D. An Accurate and Efficient Performance Analysis Technique for Multiprocessor Snooping Cache Consistency Protocols. In: Ann. Int. Symp. on Computer Architecture, IEEE, 18th. **Proceedings**, p. 308-15, 1988.
- [Wils87] WILSON, A.W. Hierarchical cache/Bus Architecture for Shared Memory Multiprocessors. In: Ann. Int. Symp. on Computer Architecture, IEEE, 14th. **Proceedings**, p. 244-52, 1987.