

X Simpósio Brasileiro de Arquitetura de Computadores

DESEMPENHO DE CACHES REMOTOS EM APLICAÇÕES CIENTÍFICAS COM E SEM OTIMIZAÇÕES

Edward D. Moreno
edmoreno@lsi.usp.br

Sergio T. Kofuji
kofuji@lsi.usp.br

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

RESUMO

A idéia principal neste artigo é mostrar os resultados de desempenho da utilização de caches remotos CARDS (Caches para Acesso Remoto a Dados), em sistemas multiprocessadores de alto desempenho, modernos e futuros, para aplicações científicas com e sem otimizações. As versões otimizadas organizam a estrutura de dados e aloca os dados para conseguir maior localidade e aproveitar melhor a memória local de cada processador.

ABSTRACT

This paper shows the benefits of remote data caches (CARDS) in modern and futures high-performance multiprocessors for scientific applications with and without optimizations. The optimized versions allow data structure to be organized and allocated contiguously in the local memory of processors, thus enhancing data locality properties. Using execution driven simulation, we show remote caches improve the overall performance in modern and future systems for optimized and non-optimized programs.

1. INTRODUÇÃO

Hoje está se tornando popular construir sistemas multiprocessadores de memória compartilhada usando nós SMP (*Symmetric Multiprocessing*). Esta nova tendência é atrativa pois consegue-se facilmente no mercado, módulos com 2 ou 4 microprocessadores *high-end*. Do ponto de vista de *hardware*, o fato de agrupar vários processadores formando um nó simétrico ("*clusterização*") é uma maneira de amortizar a complexidade nos controladores e nas interfaces de rede dos nós, pois quando o número de processadores no sistema aumenta consideravelmente, a *clusterização* resulta numa máquina com menor sobrecarga (*overhead*) *hardware*, quando comparada a sistemas formados por nós com um único processador por nó. Do ponto de vista do *software*, nós SMPs são muito atrativos para aqueles que desenvolvem aplicações, uma vez que o sistema operacional oferece uma grande confiança para trabalhar escalando desempenho e utilizando recursos na medida em que eles são adicionados. A aplicação não tem que mudar à medida em que os processadores aumentam. Além disso, o projetista da aplicação visualiza um espaço de endereçamento uniforme, simplificando a portabilidade do *software* em sistemas SMP.

A pesar dos benefícios destes caches terem sido objeto de pesquisas em vários centros acadêmicos e industriais do mundo, neste trabalho pretende-se responder às seguintes perguntas: (i) O uso de caches remotos em sistemas CC-NUMA futuros, trará benefícios significativos? e (ii) No caso da realização de mudanças nos programas aplicativos visando otimizar os acessos aos caches locais dos processadores e aproveitar melhor a hierarquia de memória dos atuais e futuros sistemas CC-NUMA, o uso de caches remotos ainda será de utilidade relevante para tolerar o problema dos acessos remotos?

Usando simulação dirigida à execução, e um modelo de tempo bem próximo da realidade, obtém-se um aumento no *speedup* de 39% e 11% para as versões não otimizadas dos programas LU e Water, para arquiteturas futuras. Similarmente, atingi-se um ganho de desempenho de 17% e 8%, respectivamente, para as versões otimizadas e futuras. Já para sistemas modernos, o *speedup* beneficia-se no máximo de LU: (28% e 66%), Water: (15% e 17%) e Ocean: (55% e 56%) para versões otimizadas e não otimizadas.

Nestes dados de desempenho pode-se observar que, ainda para versões otimizadas, os caches remotos apresentam benefícios razoáveis. Estes benefícios tornam-se mais significativos com um número maior de processadores nos sistemas.

2. ARQUITETURA CC-NUMA COM CACHES REMOTOS

Na figura 1, pode-se observar que cada aglomerado pode conter um ou vários processadores (desde 1 até 4 processadores) com os seus próprios caches locais (tanto de primeiro nível e/ou segundo nível: L1/L2), a sua memória principal local, unidades de entrada e saída e a interface de acesso à rede de interconexão. Além destes elementos chaves, o nó também possui um cache CARD (Cache de Dados para Acessos Remotos).

Nós temos desenvolvido o simulador SIM-SMP em C++; um simulador modular, escalável e de fácil uso. Neste simulador SIM-SMP, assume-se que a procura por instruções e as referências a dados privados sempre

X Simpósio Brasileiro de Arquitetura de Computadores

são realizados nos caches locais dos processadores. Todos os dados compartilhados e os diferentes eventos de sincronização são detalhadamente estudados.

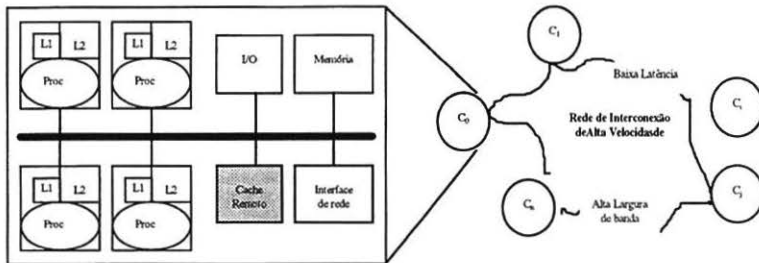


Figura 1. Organização de Multiprocessadores Baseados em Aglomerados SMP

Motivados pelos contínuos e rápidos avanços na microeletrônica, na construção de circuitos integrados, nos sistemas digitais e na arquitetura de sistemas computacionais, espera-se que num futuro muito próximo existam arquiteturas CC-NUMA com 2 ou 4 processadores de alto desempenho executando em 500 MHz. Inclusive, é possível a existência comercial e de "custo baixo" de chips-multiprocessadores com 2 ou 4 processadores.

Apesar dos excelentes trabalhos estudando caches remotos [2, 3, 4, 8, 10], ao nosso entender nenhum deles mostra o efeito de aumentar o número de processadores no sistema, e não consideram os efeitos de mudar as velocidades dos principais parâmetros do nó SMP, como a velocidade dos processadores, das memórias e as redes de interconexão. Além disso, nenhum deles enfatiza na importância dos efeitos nas mudanças dos mesmos programas que executam no sistema. Aqui nós focalizamos nesses pontos e consideramos uma análise detalhada do efeito dos acessos aos caches em função da sua implementação.

3. PROGRAMAS APLICATIVOS - SPLASH-2

As aplicações consideradas neste estudo são três programas do conjunto SPLASH-2 (*benchmark* para sistemas multiprocessadores de memória compartilhada). Este benchmark, desenvolvido na Universidade de Stanford, é bem conhecido na comunidade acadêmica da área de arquitetura de computadores de alto desempenho e é de domínio público¹. O conjunto SPLASH-2 consiste de 12 programas, 8 aplicações completas e 4 *kernels* que representam uma variedade de problemas científicos, de engenharia e de visualização.

Neste artigo, somente consideram-se três aplicações (LU, WATER-SPATIAL e OCEAN) e as suas versões não-otimizadas "LU-non-contiguous", "Ocean-non-contiguous" e "Water-nquared" (A partir desse momento serão chamadas de LU-NOP, WATER-NOP e OCEAN-NOP, respectivamente). Não obstante, somente são mostrados resultados para a programa LU, tanto na sua versão otimizada e a não otimizada.

4. RESULTADOS E ANÁLISE DE DESEMPENHO

Esta seção descreve e analisa os resultados obtidos via simulação, usando-se das seguintes métricas de desempenho: *Speedup* e o NET (Tempo de Execução Normalizado). Em todas as figuras, aparece a indicação 1, 2, e 4; o que significa que o sistema com N processadores tem um nível de "clusterização" de 1, 2 ou 4, i.e., cada nó no sistema possui 1, 2 ou 4 processadores. Neste caso, pode-se visualizar facilmente o efeito de considerar vários processadores num único nó.. Quando a indicação é 1C, 2C e 4C quer dizer que o sistema com 1, 2 ou 4 processadores/nó está considerando a utilização de caches remotos CARDS.

Uma observação geral, é que os caches remotos sempre produzem um aumento significativo no *speedup* da aplicação. Este aumento vai tornando-se mais significativo à medida em que aumenta-se o número de processadores no sistema. Assim, extrapolando tais pontos, poder-se-ia dizer que a utilização de caches remotos torna-se mais significativa em grandes sistemas. Outra observação, válida para todas as figuras, é que não existe muita diferença entre os níveis 2 e 4 de "clusterização" para sistemas com menos de 32 P.

¹ Disponível no <http://www-flash.stanford.edu/pub/splash2/>.

X Simpósio Brasileiro de Arquitetura de Computadores

PROGRAMA LU: Nas figuras 2.a-b (*speedup* para o programa LU) é possível observar que os caches remotos CARDS tornam o sistema paralelo mais eficiente em até 28%, para a versão otimizada, (para 16 processadores e "clusterização" 2-way). Apesar de na figura não serem muito visíveis tais diferenças, os resultados da tabela 1.a permitem identificar mais claramente o efeito destes caches no *speedup*. Esta observação é também válida para a aplicação não otimizada (LU-NOP). Neste caso, a maior diferença é alcançada para um sistema com 32 processadores/2-way onde a diferença é bastante alta, da ordem de 66% (ver tabela 1.b).

No caso desta aplicação pode-se dizer que as otimizações, sem caches remotos no sistema, produzem um aumento máximo de desempenho² de 22%, 36%, 23%, 22% e 25% para 4, 8, 16, 32 e 64 processadores/2-way, respectivamente. Quando os caches remotos são inseridos no sistema, as otimizações conseguem ter um melhoramento máximo de 6%, 27%, 12%, -16% e -5% para 2, 4, 8, 16, 32 e 64 processadores/4-way. O valor negativo significa que para sistemas com mais de 32 processadores, a utilização de caches remotos é mais notável para a aplicação sem otimizações.

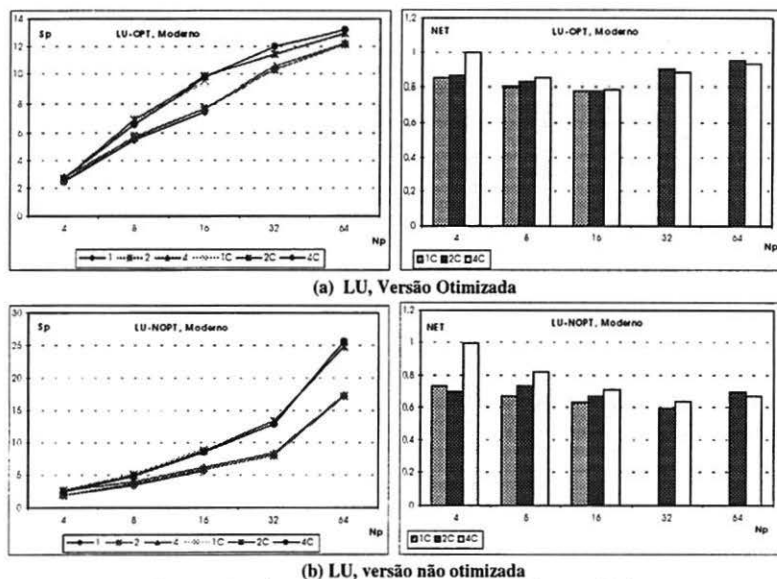


Figura 2 Speedup e NET na aplicação LU, para Sistemas Modernos

Tabela 1 Benefício³ dos CARDS no speedup (%) – Programa LU – Sistema Moderno
(a) LU, versão otimizada

LU	4P	8P	16P	32P	64P
1C	9.2	23.73	27.65	-	-
2C	7.9	20.52	28.44	10.74	18.60
4C	-3.23	17.73	26.64	13.11	17.23

(b) LU, versão não otimizada

LU-NOP	4P	8P	16P	32P	64P
1C	30.39	48.99	57.64	-	-
2C	34.01	36.68	47.79	66.54	60.50
4C	-4.5	21.51	40.03	55.36	53.28

² Estes dados, representando a percentagem do ganho da otimização, são calculados seguindo a expressão: $(\text{speedup OPT} - \text{speedup NOP}) / \text{speedup NOP} * 100$. Isto é válido para sistemas com e sem caches CARDS.

³ Este benefício em percentagem, válido para as versões otimizadas e não-otimizadas, calcula-se com a seguinte expressão: $(\text{speedup CARD} - \text{speedup normal}) / (\text{speedup CARD}) * 100$.

X Simpósio Brasileiro de Arquitetura de Computadores

Quando visualizam-se as diferenças no desempenho (através do *speedup*) da aplicação otimizada e a não-otimizada (i.e. LU e LU-NOP), observa-se que a otimização permite tirar maior vantagem da "clusterização" (ver os maiores valores do máximo desempenho, anteriormente analisados), pois a localidade torna-se maior. Isto explica o porquê do desempenho dos caches remotos ser menor em tais programas otimizados. Este efeito da otimização na "clusterização" é da ordem de 36%, e quando usa-se CARDS é da ordem de 27%.

Estes resultados são interessantes, pois estabelecem que o efeito de inserir caches remotos nos sistemas paralelos de alto desempenho é mais significativo em sistemas executando aplicações não otimizadas. Também permite concluir que, apesar de otimizar as aplicações, a utilização de caches remotos continua sendo muito útil.

LU-futuro: As observações relacionadas com a otimização e ao uso de caches CARDS continuam sendo válidas para sistemas futuros. Aqui é bom salientar que o efeito dos caches CARDS é menor que nos sistemas atuais, mas continuarão produzindo ganhos de desempenho. Esta diminuição no seu efeito deve-se ao fato de que num futuro próximo a rede de interconexão terá melhores características, isto é, maior largura de banda e menor latência. Ainda com estes melhoramentos, é bom salientar que estes caches remotos continuarão sendo de grande utilidade. A diferença maior encontrada aparece para um sistema com 16P/2-way, da ordem de 17%. Idem para o caso não otimizado, onde a diferença é da ordem de 39% para um sistema com 32P/2-way. Novamente, observa-se um maior ganho com a versão não otimizada.

Tabela 2 Benefício futuro dos CARDS no speedup (%) – Programa LU Sistema Futuro
(a) LU, versão otimizada

LU (fut)	4P	8P	16P	32P	64P
1C	5.32	15.24	17.24	-	-
2C	4.47	14.14	17.39	6.7	8.12
4C	-2.0	10.81	14.30	7.33	9.05

(b) LU, Versão não otimizada

LU-NOP (fut)	4P	8P	16P	32P	64P
1C	16.95	34.91	38.25	-	-
2C	2.29	26.17	31.10	39.86	42.20
4C	-3.34	20.55	27.52	37.11	46.72

No futuro, vide tabela 2, as diferenças máximas detectadas no desempenho entre as versões otimizadas e não otimizadas, em favor da versão otimizada, serão da ordem de -6%, 29%, 18%, 6% e 7% para 4, 8, 16, 32 e 64 processadores em uma organização de 2 processadores/nó. No sistema com caches remotos, tais diferenças são de 3%, 23%, 9%, -24% e -15%.

Um ponto chave de mencionar, é que as diferenças entre os sistemas atuais e os futuros serão menores para sistemas com mais do que 32 processadores (menores a 3%). Este resultado indica que, no futuro, caches remotos continuarão sendo de bastante utilidade, uma vez que será possível ter sistemas com um número maior de processadores.

5. CONCLUSÕES

Os resultados obtidos via simulação comandada a execução permitem dizer que caches remotos são de grande utilidade tanto em programas com boa localidade (versão otimizada dos programas) quanto nas suas versões não otimizadas. Esta afirmação continuará sendo válida para sistemas futuros, onde a velocidade das memórias/cache/redes de interconexão não acompanhará as velocidades dos futuros processadores.

Assim, caches remotos são e serão de grande benefício como mecanismo de *hardware* para reduzir as grandes latências ocasionadas pelos acessos remotos. Este resultado mantém-se ainda em programas otimizados (seja pelo programador ou por compiladores paralelizantes) que tiram maior proveito da hierarquia de memória dos modernos e futuros sistemas.

Finalmente, pode-se concluir também que os benefícios de caches remotos aumentam com o número de processadores no sistema, o que faz pensar que em grandes sistemas (mais de 64 processadores) estes caches sejam de muita utilidade em sistemas escaláveis. Além disso, para sistemas com 64 processadores, notou-se um ganho maior para sistemas organizados em 4 processadores/nó. Para um número inferior a 32 processadores, um benefício maior é obtido para uma organização de 2P/nó.

6. REFERÊNCIAS

- [1] Moreno, Edward D. Caches Remotos e Prefetching em Sistemas Multiprocessadores de Alto Desempenho: Considerações Arquiteturais. *Tese de Doutorado, Universidade de São Paulo*, Agosto, 1998.
- [2] Moreno, E.D. et Al. A Preliminary Study of Network Caches on the NUMAchine Multiprocessor. *Technical Report CSRI-350, University of Toronto*, Aug, 1996.