

X Simpósio Brasileiro de Arquitetura de Computadores

*Avaliação de uma Estratégia Cooperativa para Gerência de Memória em Ambientes de Alto Desempenho com Multiprogramação**

Edson Toshimi Midorikawa
emidorik@lsi.usp.br
Laboratório de Sistemas Integráveis
Departamento de Engenharia Eletrônica
Escola Politécnica da
Universidade de São Paulo
São Paulo, SP 05508-900

Liria Matsumoto Sato
liria@pcs.usp.br
Departamento de Engenharia de Computação
e Sistemas Digitais
Escola Politécnica da
Universidade de São Paulo
São Paulo, SP 05508-900

RESUMO

Nenhuma das soluções propostas para o problema dos sistemas de memória dos modernos computadores de alto desempenho apresenta um enfoque integrado. São técnicas que resolvem eficientemente apenas algum aspecto particular do problema geral. Atualmente, os fabricantes têm começado a oferecer suporte a multiprogramação temporal e espacial em suas máquinas paralelas. Este trabalho procura apresentar nossas pesquisas em uma estratégia que procura abordar, de uma maneira integrada, o problema dos acessos à memória em ambientes de alto desempenho com suporte a multiprogramação. Os resultados obtidos mostram que é possível fazer uso mais eficiente da memória que as estratégias tradicionais.

ABSTRACT

None of the proposed solutions presents an integrated approach to the memory system problem of modern high-performance computers. They are techniques that provide only an efficient solution to a specific aspect of the general problem. Nowadays parallel machine manufacturers are starting to support temporal and spatial multiprogramming. This paper presents an integrated strategy to the memory access problem in high-performance environments with multiprogramming. The obtained results show that it is possible to use the available memory in a more efficient way than traditional strategies.

1. Introdução

O problema dos sistemas de memória nos computadores de alto desempenho tem merecido uma série de trabalhos de pesquisa. A literatura vem mostrando um conjunto muito grande de alternativas que foram surgindo como resultado das pesquisas dos diversos grupos em universidades e centros de pesquisa do mundo inteiro. De uma maneira geral, a maioria delas trabalha o aspecto da *latência da memória*, ou seja, do longo tempo necessário para o acesso aos dados residentes na memória principal [WULF95].

Os métodos propostos seguem uma das duas técnicas: a primeira procura *reduzir a latência* [CARR94] e a segunda, *tolerar a latência* [HEXS94]. De uma maneira ou de outra, estas soluções têm melhorado o desempenho dos modernos sistemas de memória. Infelizmente, devido a algumas dificuldades, ainda não há uma solução definitiva para o problema. Este trabalho apresenta nossa contribuição na busca desta solução. A nossa estratégia aborda três aspectos complementares: a otimização do padrão de acessos à memória, a participação conjunta dos programas de sistema e, finalmente, um sistema de execução que leva em consideração o sistema de memória.

Este trabalho procura apresentar uma estratégia alternativa e complementar àquelas tradicionalmente propostas na literatura. São levantados aspectos relativos ao desenvolvimento de programas nos modernos computadores de alto desempenho. Este trabalho é estruturado da seguinte forma: apresentamos inicialmente a nossa proposta para uma gerência de memória cooperativa de alto desempenho na seção 2. A seção seguinte apresenta o estudo realizado e os principais resultados obtidos. Finalmente, colocamos nossas conclusões na seção 4.

* Este trabalho de pesquisa foi financiado em parte pela Finep (no contexto do Projeto Hipersistemas).

X Simpósio Brasileiro de Arquitetura de Computadores

2. Gerência Cooperativa de Memória

Os atuais sistemas de computação utilizam, de uma maneira generalizada, um enfoque seqüencial para a gerência da memória, onde cada componente do sistema de programação executa sua tarefa de maneira independente dos demais. Embora existam diversas possibilidades para a interação, elas não são exploradas. Nossas pesquisas neste aspecto nos levaram à estratégia *Communion*, cujo propósito é investigar tal enfoque interativo para os *programas de sistema*, os quais “trabalham em comum” com o intuito de alcançar um desempenho otimizado [MIDO97]. A figura 1 mostra um diagrama do enfoque *Communion*.

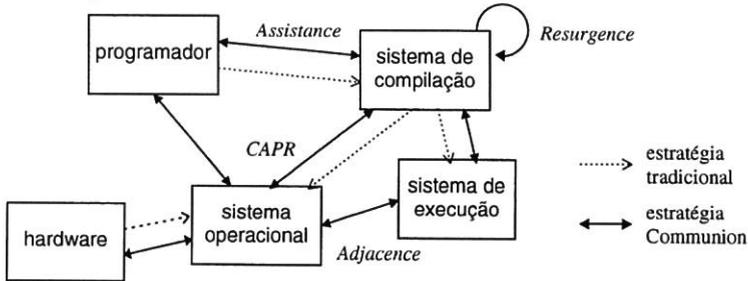


Figura 1 - Interação entre os componentes do sistema.

Dentre as várias possibilidades existentes, quatro interações já foram especificadas: [MIDO97] [MIDO98a]:

- a interação entre os componentes do sistema de compilação, cuja estratégia tem a denominação *Resurgence*;
- a interação entre o compilador e o sistema operacional, cuja estratégia foi chamada *CAPR* (*Compiler-Aided Page Replacement*);
- a interação entre o sistema operacional e o sistema de execução, nomeada *Adjacence* e
- a interação entre o programador e o sistema de compilação, nomeada *Assistance*.

2.1. CAPR: Compiler-Aided Page Replacement

A estratégia *CAPR* compõe uma nova técnica de gerência de memória baseada na interação do compilador e o sistema operacional. Sob esta estratégia ambos os programas de sistema trocam entre si informações relativas com os requisitos e a utilização de memória dos programas [MIDO95]. Após uma análise do programa-fonte, o compilador detecta as possíveis fontes de localidade e insere diretivas para informar ao sistema operacional. Por exemplo, as diretivas podem especificar:

- mudança de localidade,
- mudança no requisito de memória,
- mudança no parâmetro de controle do algoritmo e
- troca de algoritmo de substituição.

Por outro lado, após a coleta de dados da execução dos programas, o sistema operacional pode fornecer informações ao compilador sobre o real padrão de acessos à memória. Com isto, o compilador pode alterar a forma de armazenamento dos dados no espaço de endereçamento virtual de forma que seja otimizada a localidade dos dados verificada durante a execução.

Com isto é possível realizar uma gerência de memória específica a cada programa, ou seja, que seja a mais adequada à estrutura daquele programa. Em comparação com os sistemas atuais, onde todos os programas são administrados da mesma forma (enfoque do “caso médio”), a estratégia *CAPR* deverá apresentar desempenhos maiores, devido ao enfoque do tipo “feito sob medida” para a gerência da memória de cada programa.

X Simpósio Brasileiro de Arquitetura de Computadores

3. Comportamento da Estratégia em Ambientes Multiprogramados

Devido aos bons resultados obtidos anteriormente em sistemas monoprogramados [MIDO95], desenvolvemos um estudo onde verificamos os efeitos do *CAPR* em um ambiente multiprogramado, onde nosso objetivo principal foi reduzir a ocupação de memória durante a execução de um programa. Foram adotados dois programas para o estudo realizado:

- *local*: programa sintético que apresenta um alto grau de localidade em seus acessos à memória. Representa as aplicações com um padrão de acessos adequado ao sistema de memória. Apresenta um tamanho total de 492 páginas;
- *seq*: programa sintético com um padrão de acessos sequencial à memória. Procura avaliar casos extremos de aplicações como o *SOR*. Compõe um tamanho total de 54 páginas.

Foram avaliadas duas estratégias para a gerência das páginas dos programas em um ambiente multiprogramado. Na primeira delas é adotada um único algoritmo para todos os programas (estratégia global) e na segunda, um algoritmo específico para cada programa (estratégia local). Para a estratégia local, foram adotados os algoritmo *LRU* para o programa *local* e o *MRU* para o *seq*. Avaliamos também a política de particionamento de memória entre os programas. Os casos foram nomeados “x/y” e representam um particionamento em que x% da memória é reservada ao programa *seq* e y%, ao programa *local*.

Devido a restrições de espaço apresentamos aqui apenas alguns resultados¹. Executando-se a avaliação aos dois programas, obtivemos os gráficos 1 e 2. Com relação ao número total de faltas de páginas, o gráfico 1 nos mostra que, de uma maneira geral, a maioria dos casos da estratégia local apresentaram um desempenho melhor para os vários tamanhos de memória.

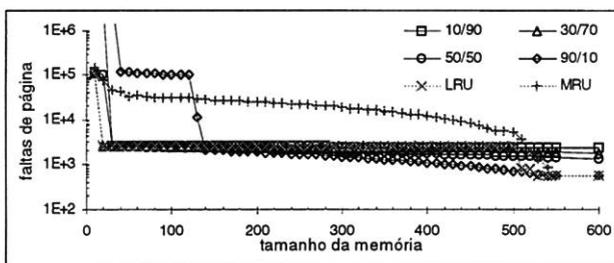


Gráfico 1 - Comparação entre as estratégias global e local com relação às faltas de página.

Notamos também, para os tamanhos inferiores a 140 páginas, uma alta taxa de faltas de página do caso 90/10. Este número excessivo de faltas se deve ao comportamento interno do programa *local*. Uma análise individual de cada programa nos mostra que este fato não ocorre com o programa *seq* [MIDO98b]. Por outro lado, para tamanhos superiores a 140 páginas, o caso 90/10 apresenta a menor taxa de faltas de página entre todos os casos estudados.

Dentre os outros critérios de partilha, verificamos que, excetuando-se tamanhos de memória superiores a 500 páginas, a estratégia local apresentou um melhor desempenho que a estratégia global.

A melhora é mais acentuada com relação à ocupação de memória. O gráfico 2 apresenta o produto espaço-tempo e nos mostra o melhor aproveitamento da estratégia local em relação às estratégias globais. Excetuando-se o caso 90/10 que apresentou a maior ocupação de memória de todos os casos, as estratégias locais apresentaram um bom desempenho. Isto pode ser facilmente explicado pelo comportamento local a cada processo. Como cada processo é gerenciado de forma mais adequada localmente, há uma melhor utilização da memória alocada ao processo. Para maiores detalhes veja [MIDO98b]. Para memórias com tamanhos superiores a 60 páginas, o caso 10/90 se resalta, apresentando a menor ocupação de memória. A diferença de ocupação é cerca de quatro vezes menor.

¹ Uma versão mais detalhada deste trabalho pode ser obtida no endereço <http://www.lsi.usp.br/~emidorik>.

X Simpósio Brasileiro de Arquitetura de Computadores

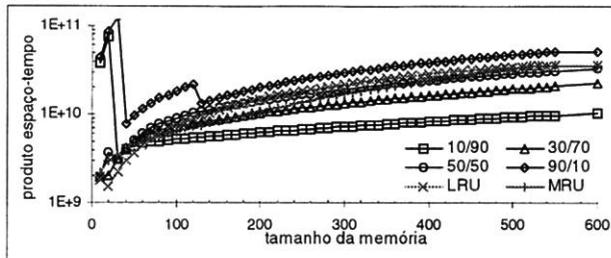


Gráfico 2 - Comparação entre as estratégias com relação ao produto espaço-tempo.

Para ambas as métricas estudadas, verificamos que para as situações em que os casos extremos de particionamento apresentavam resultados piores que a estratégia global, verificamos que casos intermediários (30/70 e 50/50, por exemplo) apresentaram desempenhos mais próximos aos apresentados pela estratégia global. Concluímos então que a adoção de uma estratégia de particionamento fixo não é um enfoque que leva ao melhor desempenho em todas as situações.

4. Conclusão e Trabalhos Futuros

Este trabalho apresentou nossas pesquisas no desenvolvimento em métodos de gerência de memória para sistemas de alto desempenho. De uma maneira geral, verificamos que a aplicação de diretivas com a especificação do algoritmo de gerência têm grande influência no desempenho global do sistema (produto espaço-tempo) como no desempenho individual das aplicações (taxa de faltas de página). Apesar dos resultados preliminares demonstrarem uma vantagem da adoção de uma estratégia local em ambientes multiprogramados, várias questões permanecem em aberto e merecem um estudo mais aprofundado: a partilha dinâmica da memória entre os processos em execução, a escolha do melhor algoritmo para cada processo [KRUE94] e a possibilidade de especificar um algoritmo para cada seção do programa.

Referências Bibliográficas

- [CARR94] CARR, S.; MCKINLEY, K. S.; TSENG, C.-W. *Compiler optimizations for improving data locality*. In: ASPLOS, 6, San Jose, CA. **Proceedings**. p.252-262. October, 1994.
- [HEXS94] HEXSEL, R. A. *A quantitative performance evaluation of SCI memory hierarchies*. PhD Thesis. Department of Computer Science, The University of Edinburgh. 1994.
- [KRUE94] KRUEGER, K. A. *Tools and strategies for the development of application-specific virtual memory management*. Technical report UCB/CSD-94-822, Computer Science Division, University of California at Berkeley. May 1994.
- [MIDO95] MIDORIKAWA, E. T. & ZUFFO, J. A. *Uma técnica para gerência de memória auxiliada por compilador*. In: Conferência Latino-Americana de Informática, 21 (PANEL 95), Canela, RS. **Anais**. p.49-60. Julho de 1995.
- [MIDO97] MIDORIKAWA, E. T. & SATO, L. M. *Communion: towards a cooperative strategy for high-performance memory management*. In: Simpósio Brasileiro de Arquitetura de Computadores - Processamento de Alto Desempenho, 9, Campos do Jordão, SP. **Anais**. p.509-524, Outubro de 1997.
- [MIDO98a] MIDORIKAWA, E. T. *Otimização de localidade de dados em aplicações reais e estratégias de alocação dinâmica de memória*. In: Simpósio Brasileiro de Arquitetura de Computadores - Processamento de Alto Desempenho, 10, Búzios, RJ. **Anais**. 1998.
- [MIDO98b] MIDORIKAWA, E. T. *Evaluation of a local strategy for high-performance memory management*. Submetido para publicação. 1998.
- [WULF95] WULF, W. A. & MCKEE, S. A. *Hitting the memory wall: implications of the obvious*. **Computer Architecture News**, v.23, n.1, p.20-24. March 1995.