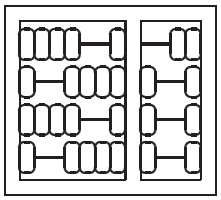
UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO



ESTUDO DA CACHE ATRAVÉS DE SIMULAÇÕES

Relatório do primeiro laboratório de MC723

Aluno: Tiago Chedraoui Silva

Resumo

O princípio de funcionamento da memória cache é duplicar parte dos dados contidos na memória principal (a memória lenta, neste caso) em um módulo menor (o cache) composto por dispositivos de memória mais rápidos. Quando o processador solicita um item de dado (gerando uma referência para seu endereço, que pode ser físico ou virtual), o gerenciador de memória requisita este item do cache. Duas situações podem ocorrer: cache hit: item está presente no cache, é retornado para o processador praticamente sem período de latência; cache miss: item não está presente no cache, processador deve aguardar item ser buscado da memó- ria principal. Nesse laboratório, estudaremos a melhor organização de uma memória cache para a execução de um determinado programa.

Sumário

1 Dinero

O Software Dinero é um simulador de cache para traces de memória (registro de execução de um programa).

Dentre as opções de configuração da memória cache, o dinero nos fornece possibilidades de alterar o tamanho da memória cache, o tamanho do bloco da memória cache, o tamanho do sub-bloco, a associatividade, a política de substituição (LRU[1], FIFO[2] ou aleatório), a política de escrita se ocorrer um hit (write-back[3], write-through[4]) e a política de escrita se ocorrer um miss (write-allocate[5], no-write-allocate[6], fetch on write[7], no fetch on write[8]).

Utilizamos dois tipos de traces, o F2B, que executa 2 bilhões de instruções, e o M2B, que executa 2 bilhões de instruções depois de pular 50 bilhões. Logo os traces F2B possuirão uma taxa de miss diferente do M2B, já que ambos acessam posições diferentes da memória durante a execução das instruções.

2 Simulação

2.1 Tamanho do bloco, associatividade, tamanho da cache

O tamanho total de uma cache pode ser calculado pela multiplicação do número de linhas, número de blocos por linha e tamanho dos blocos. Uma cache L1 possui geralmente um tamanho entre 16KB e 128KB, podendo chegar, em algumas caches, a 8MB.

Para um determinado tamanho de cache e blocos, alterou-se a sua associatividade, através do qual percebe-se que a cache totalmente associada é o que apresentou a melhor performance.

Posteriormente, aumentou-se o tamanho da memória cache, deixando-a totalmente associada. Percebeu-se que quanto maior a memória cache, menor é a taxa de miss. Contudo, vale ressaltar que em certos momentos há uma estabilidade, ou seja, por mais que a cache cresça, a taxa de miss fica quase constante.

Por outro lado, o tamanho do bloco é relativo. De acordo com a figura 1b, para cada valor de associatividade, um tamanho de bloco maior que 32 tende a ser melhor, porém após passar determinado valor ele tende a aumentar o valor de miss rate.

2.2 Políticas de substituição

Simulou-se a utilização das três políticas de substituição fornecida pelo Dinero (Random, LRU e FIFO). De acordo com a figura 1a, para uma mesma configuração a política LRU foi a que apresentou um melhor resultado.

[1] Least recently used: substitui a página na memória cuja última referência é a mais antiga.

[2] First-in, first-out: substitui a página mais antiga na memória

[3] Quando um ciclo de escrita ocorre para uma palavra, ela é atualizada apenas no cache.

[4] Quando um ciclo de escrita ocorre para uma palavra, ela é escrita no cache e na memória principal simultaneamente.

[5] Com alocação em escrita:Se ocorrer um miss é alocado na cache uma linha para o dado escrito. Usando, na maioria das vezes, com a política write-back

[6] Sem alocação em escrita:Se ocorrer um miss de escrita, uma linha não é alocada na cache para o dado escrito. Usado, na maioria das vezes, com a política write-through.

[7] No caso de um miss para escrita, item é trazido para o cache após atualizado, se foi alocado espaço para ele na cache.

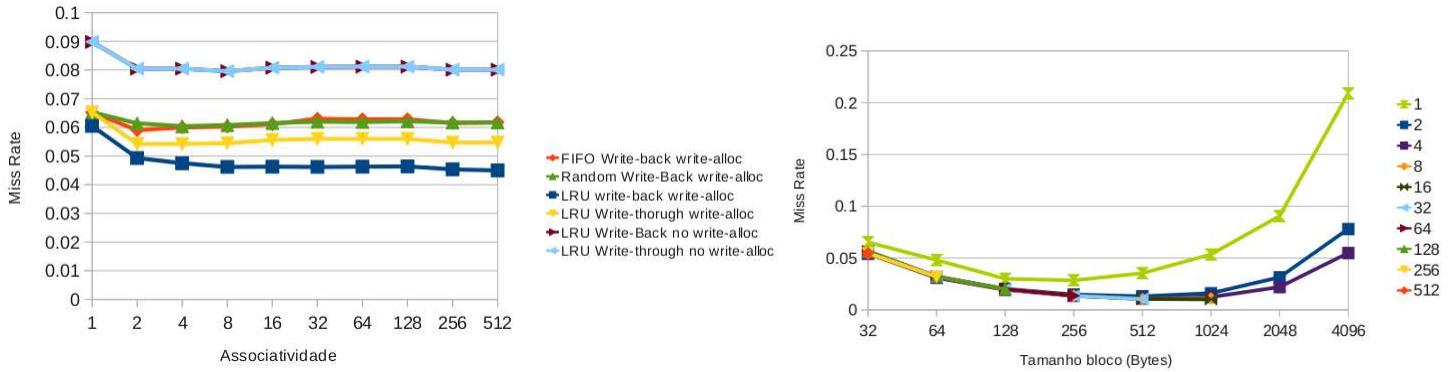
[8] No caso de um miss para a escrita, item é atualizado apenas na memória principal, não sendo trazido para o cache.

2.3 Políticas de escrita se ocorrer um hit

Simulou-se a utilização das duas políticas de escrita se ocorrer um hit fornecida pelo Dinero. De acordo com a figura 1a, para uma mesma configuração a política write-back foi a que apresentou um melhor resultado.

2.4 Políticas de escrita se ocorrer um miss

Simulou-se a utilização das duas políticas de escrita se ocorrer um miss fornecida pelo Dinero. Para uma mesma configuração, a política padrão, sempre usar write allocate policy foi a que apresentou um melhor resultado.



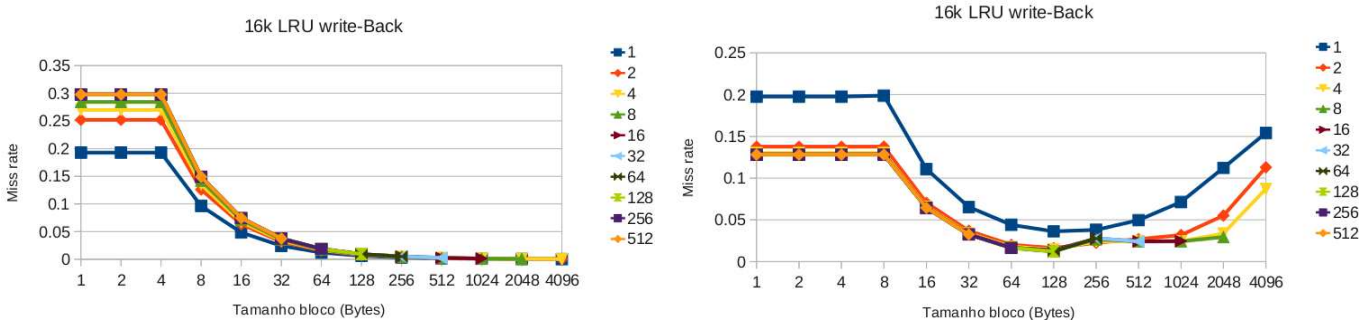
(a) Alteração de associatividade para diferentes políticas

(b) Alteração de associatividade para vários tamanhos de bloco

Figura 1: Caches unificadas de 16KB com diversas configurações

2.5 Duas caches: dados e instrução

A memória cache pode ser separada em cache de instruções e para cache de dados, de forma que dados e instruções poderiam ser acessados em paralelo. Vale ressaltar que o valor da taxa de miss será maior nas caches separadas, contudo o tempo de acesso é menor. Portanto, em uma comparação entre caches separadas e unificada, o tempo deve o fator de escolha e não o miss rate. No entanto, para determinar-se a melhor configuração da cache, o miss rate é essencial.



(a) Cache de dados

(b) Cache de instrução

Figura 2: Caches separadas: Alteração de associatividade para vários tamanhos de bloco

3 Conclusão

De acordo com as simulações, considerando como fator de escolha a taxa de miss rate e o custo de uma memória cache[9], escolher-se-á uma cache unificada de 16KB, tamanho de bloco 1024B, associatividade 16 e política de substituição LRU. Tal configuração proporcionou um valor de miss rate próximo a 1,01% para o trace m2b e 0.84% para o trace f2b.

Por outro lado, para otimizar o uso de uma cache, deve-se dividi-la em uma de dados e outra de instrução, o que possibilitaria o acesso em paralelo à ambas caches durante a execução de um programa, diminuindo o tempo de execução do programa.

Dessa maneira, com base nas simulações e considerando a taxa de miss rate e o custo de uma memória cache[10], escolher-se-á uma cache de instrução de 16KB, tamanho de bloco 4096B, associatividade 4 ou 2[11], política de substituição LRU e uma cache de dados de 16KB, tamanho de bloco 128B, associatividade 128 e política de substituição LRU. Tais configuração proporcionou para a cache de instrução um valor de miss rate próximo a 0.02% para o trace m2b e 0.04% para o trace f2b e para a cache de dados um valor de miss rate próximo a 4.48% para o trace m2b e 1.22% para o trace f2b.

Porém, considerando somente possuir espaço para um total de 16KB de cache, escolher-se-á uma cache de instrução de 8KB, tamanho de bloco 4096B, associatividade 2, política de substituição LRU e uma cache de dados de 8KB, tamanho de bloco 512B, associatividade 8 e política de substituição LRU. tais configurações proporcionou para a cache de instrução um valor de miss rate próximo a 0.03% para o trace m2b e 0.05% para o trace f2b e para a cache de dados um valor de miss rate próximo a 3.85% para o trace m2b e 3.12% para o trace f2b.

Referências

[1] Jan Edler e Mark D. Hill *Software DineroIV*. Disponível em *http://www.cs.wisc.edu/ markhill/DineroIV/*, [Último acesso: 27/02/2011].

[2] Descrição do projeto. Disponível em *http://www.ic.unicamp.br/ ducatte/mc723/1s2011/exercicio1.htm*, [Último acesso: 27/02/2011].

[3] Politicas de write-miss. Disponível em *people.engr.ncsu.edu/efg/521/f02/common/lectures/notes/lec6.pdf*.

[4] David A. Patterson e John L. Hennessy. *Computer Organization and Design, Fourth Edition: The Hardware/Software Interface*.

[9] Queremos otimizar uma cache pequena em vez de possuir uma muito grande, o que teria uma eficiência melhor de acordo com a seção 2.1

[10] Queremos otimizar uma cache pequena em vez de possuir uma muito grande, o que teria uma eficiência melhor de acordo com a seção 2.1

[11] Ambos valores de associatividade apresentaram mesma taxa de miss rate