

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 memória principal. Nesse laboratório, estudaremos a melhor organização de uma memória cache para a execução de um determinado programa.

Sumário

1	Dinero	1
2	Simulação	1
2.1	Tamanho do bloco, associatividade, tamanho da cache	1
2.2	Duas caches: dados e instrução	1
2.3	Políticas de substituição	1
2.4	Políticas de escrita se ocorrer um hit	2
2.5	Políticas de escrita se ocorrer um miss	2
3	Conclusão	2

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 as seguintes possibilidades: tamanho da memória cache, tamanho do bloco da memória cache, tamanho do sub-bloco, associatividade, política de substituição (LRU¹, FIFO² ou aleatório), política de escrita se ocorrer um hit (write-back³, write-through⁴), política de escrita se ocorrer um miss (write-allocate, no-write-allocate, fetch on write⁵, no fetch on write⁶).

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. No qual percebe-se que o totalmente associado é o que apresentou a melhor performance.

Posteriormente, aumentou-se o tamanho da memória cache total, 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.

Já 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 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 análise de caches separadas, o tempo é o fator de escolha e não o miss rate.

2.3 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.

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

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

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

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

⁵Com alocação em escrita: no caso de um miss para escrita, item é trazido para o cache após atualizado;

⁶Sem alocação em escrita: no caso de um miss para a escrita, item é atualizado apenas na memória principal, não sendo trazido para o cache.

2.4 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.

2.5 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 no fetch on write foi a que apresentou um melhor resultado.

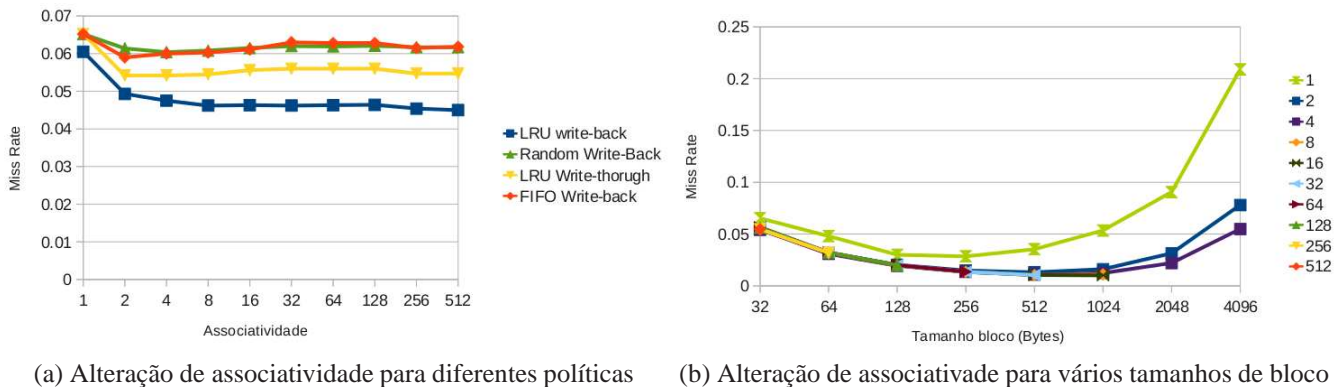


Figura 1: Caches unificadas de 16KB com diversas configurações

3 Conclusão

De acordo com nossas simulações, se considerarmos como fator de escolha a taxa de miss rate e o custo de uma memória cache⁷, escolheremos uma chace de 16KB, tamanho de bloco 1024B, associatividade 16, política de substituição LRU. o que nos proporcionou um valor de miss rate próximo a 1,01%

Por outro lado, se quisermos um tempo de acesso menor, dividiremos nossa cache em uma de dados e outra de instrução.

Referências

- [1] J. Kurose e K. Ross. *Software DineroIV*. Disponível em <http://www.ic.unicamp.br/~ducatte/>, [Último acesso: 27/02/2011].
- [2] Descrição do projeto. Disponível em <http://www.ic.unicamp.br/~ducatte/>, [Último acesso: 27/02/2011].
- [3] Harvard configuration. Configuração de cache Harvard. Online, Fevereiro 2011. Disponível em <http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel>. [Último acesso: 16/11/2009].
- [4] J. Kurose e K. Ross. *Organization and Computer design..* Disponível em <http://www.ic.unicamp.br/~ducatte/>, [Último acesso: 27/02/2011].

⁷Queremos otimizar uma cache pequena em vez de possuir uma muito grande, o que teria uma eficiência melhor de acordo com a seção 2.1