Melhorando o desempenho da cache

Marcos Monteiro Junior

Medindo o desempenho da cache

- O tempo de CPU pode ser dividido entre os ciclos de clock que a CPU gasta executando o programa e os ciclos de clock que gasta esperando o sistema de memória:
  - Tempo de cpu = (ciclos de clock de execução da CPU + ciclos de clock de stall de memória) x (tempo de ciclo de clock)
  - Stalls de memória tem origem nas falhas de cache
  - Ciclos de clock de stall de memória = ciclos de stall de leitura + ciclos de stall de escrita
  - Ciclos de stall de leitura podem ser definidos em função do número de acesso de leitura por programa, a penalidade de falha nos ciclos de clock e a taxa de falhas de leitura:
  - Ciclos de stall de leitura = leituras/programa x (taxa de falhas de leitura) x (penalidade de falha de leitura)
  - As escritas são mais complicadas, pois temos duas origens de stalls: as falhas de escrita, que exigem a busca do bloco antes de continuar a escrita e os stalls do buffer de escrita, que ocorrem quando o buffer de escrita está cheio:
  - Ciclos de stall de escrita = (escritas/programa) x (taxa de falhas de escrita) x (penalidade de falha de escrita) + (stalls do buffer de escrita)

Medindo o desempenho da cache

- Na maioria das organizações *write-back* as penalidades de falha de leitura e escrita são iguais (equivalentes ao tempo para buscar o bloco da memória)
  - Considerando que os *stalls* no buffer de escrita insignificantes:
    - Ciclos de clock de stall de memória = acessos a memória/programa x taxa de falhas x penalidade de falha
    - Ciclos de clock de stall de memória = instruções /programa x falhas/instrução x penalidade de falha

Medindo o desempenho da cache

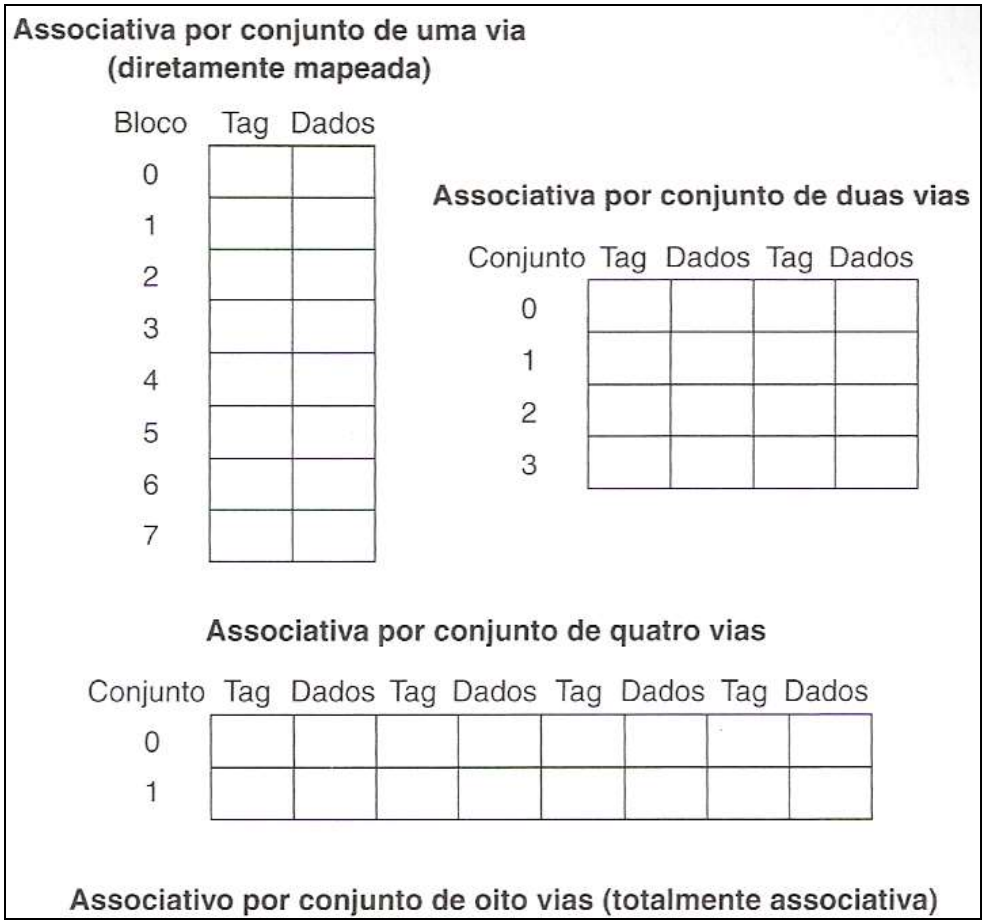
- Exemplo
  - Suponha que uma taxa de falhas de cache de instruções para um programa seja de 2% e que uma taxa de falhas de cache de dados seja de 4%. Se um processador tem um CPI igual a 2 sem qualquer stall de memória e a penalidade de falha é de 100 ciclos para todas as falhas, determine o quanto mais rápido um processador executaria com uma cache perfeita que nunca falhasse (utilizando as frequências de instruções do SPECint2000: 36% load/store)
    - Ciclos de falha de instruções = I x 2% x 100 = 2,00 x I
    - Ciclos de falha de dados = I x 36% x 4%x 100 = 1,44 x I
    - Número total de stalls de memória: 2,00 + 1,44 = 3,44 I
    - Cpi com stalls de memória: 2,00 + 3,44 = 5,44
    - Cpi sem stalls de memória: 2,00
    - O desempenho da cache perfeira: 5,44 / 2 = 2,72

Reduzindo as falhas de cache com um posicionamento de blocos mais flexível

- No mapeamento direto, existe apenas uma posição onde o bloco pode ser encontrado na cache.
- No outro extremo, está um esquema onde um bloco pode ocupar qualquer lugar na cache: **totalmente associativo**
  - Todas a entradas devem ser pesquisadas, o que torna o hardware mais caro: viável apenas para caches com pequenos números de blocos
- Faixa intermediária: **mapeamento por conjunto**
  - Existe um número fixo de locais onde cada bloco pode ser colocado
  - Uma cache associativa por conjuntos com **n** locais para um bloco é chamada de cache associativa por conjunto de **n** vias
  - Uma cache associativa por conjunto de **n** vias consiste em diversos conjuntos, cada um consistindo em **n** blocos
  - Cada bloco é mapeado para um conjunto único na cache, determinado pelo campo **índice**, e um bloco pode ser colocado em qualquer elemento deste conjunto
  - O posicionamento associativo por conjunto combina o mapeamento direto com o mapeamento totalmente associativo, de forma que um bloco é diretamente mapeado para um conjunto e, então, uma correspondência é pesquisada em todos os blocos dos conjuntos

Reduzindo as falhas de cache com um posicionamento de blocos mais flexível

- Forma de mapeamento
  - Mapeamento direto
    - (número do bloco) módulo (número de blocos na cache)
  - Mapeamento associativo por conjunto
    - (número do bloco) módulo (número de conjuntos na cache)
  - Como o bloco pode ser posicionado em qualquer elemento do conjunto, *todas as tags de todos os elementos do conjunto devem ser pesquisadas*
  - A vantagem de aumentar a associatividade é diminuir a taxa de falhas
  - A desvantagem é a redução do tempo de acerto



Falhas e associatividade nas caches

- Considere três caches pequenas, cada uma consistindo em quatro blocos de 1 word cada
  - Uma cache totalmente associativa
  - Uma cache associativa por conjunto de 2 vias
  - Uma diretamente mapeada
  - Encontre o número de falhas para cada organização de cache, dada a seguinte sequência de endereços de bloco: 0, 8, 0, 6, 8
  - Diretamente mapeado:

Endereço do bloco	Bloco de cache
0	(0 módulo 4) = 0
6	(6 módulo 4) = 2
8	(8 módulo 4) = 0

Falhas e associatividade nas caches

- Diretamente mapeada gera cinco falhas para cinco acessos:

Endereço do bloco de memória associado	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
0	falha	Memória[0]			
8	falha	Memória[8]			
0	falha	Memória[0]			
6	falha	Memória[0]		Memória[6]	
8	falha	Memória[8]		Memória[6]	

- Cache associativa por conjunto tem dois conjuntos, com índices de 0 e 1, com dois elementos por conjunto:

Endereço do bloco	Bloco de cache
0	(0 módulo 2) = 0
6	(6 módulo 2) = 0
8	(8 módulo 2) = 0

Falhas e associatividade nas caches

- Já que existe a possibilidade de escolha entre qual bloco será substituído, uma regra deve ser utilizada
  - Bloco menos recentemente usado: o bloco usado a mais tempo será substituído

Endereço do bloco de memória associado	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		Conjunto 0	Conjunto 0	Conjunto 1	Conjunto 1
0	falha	Memória[0]			
8	falha	Memória[0]	Memória[8]		
0	acerto	Memória[0]	Memória[8]		
6	falha	Memória[0]	Memória[6]		
8	falha	Memória[8]	Memória[6]		

  - Apresenta 4 falhas

Falhas e associatividade na cache

- A cache totalmente associativa tem 4 blocos na cache, em um único conjunto
  - Apresenta o melhor desempenho, com 3 falhas:

Endereço do bloco de memória associado	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		Bloco 0	Bloco 1	Bloco 2	Bloco 3
0	falha	Memória[0]			
8	falha	Memória[0]	Memória[8]		
0	acerto	Memória[0]	Memória[8]		
6	falha	Memória[0]	Memória[8]	Memória[6]	
8	acerto	Memória[0]	Memória[8]		

  - Quanta redução na taxa de falhas é obtida pela associatividade? Considerando o SPEC2000 pra uma cache de dados de 64K com um bloco de 16 words e mostra a associatividade mudando do mapeamento direto para oito vias:

Associatividade	Taxa de falhas de dados
1	10,3%
2	8,6%
4	8,3%
8	8,1%

Localizando um bloco no cache

- Cada bloco em um cache associativa por conjunto tem um campo *tag* que fornece o endereço do bloco
  - Tal valor é comparado com o endereço de origem contido em uma requisição do processador