
Arquiteturas de Computadores

Otimização da cache

Slides baseados em slides de COMP 740:
Computer Architecture and Implementation,
Montek Singh, Sep 14, 2016

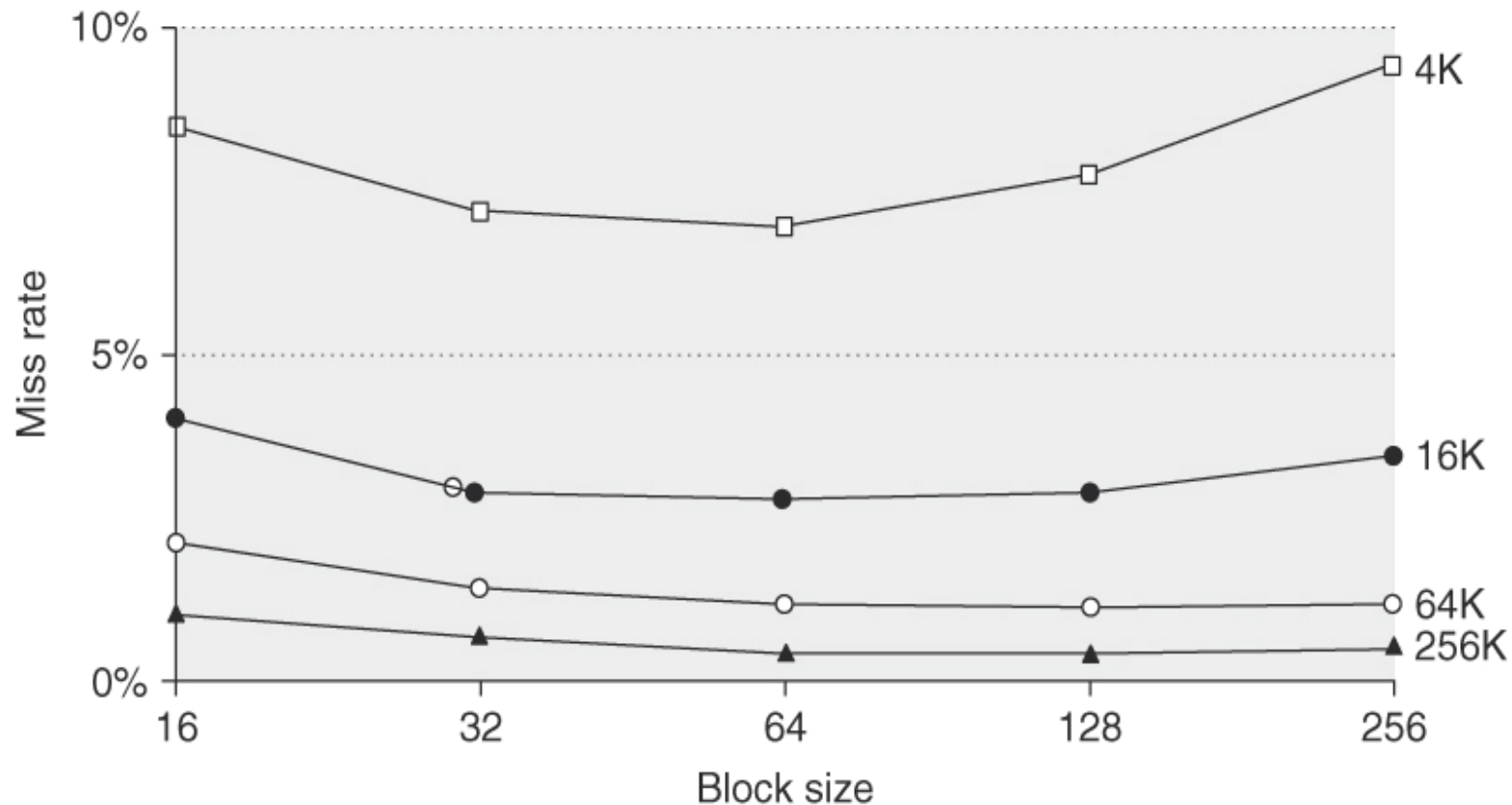
Noções básicas de hierarquia de memória

$$\frac{\text{Faltas}}{\text{Instrução}} = \frac{\text{Taxa de falta} \times \text{Acessos à memória}}{\text{Número de instruções}} = \text{Taxa de falta} \times \frac{\text{Acessos à memória}}{\text{Instrução}}$$

$$\text{Tempo médio de acesso à memória} = \text{Tempo de acerto} + \text{Taxa de falta} \times \text{Penalidade por falta}$$

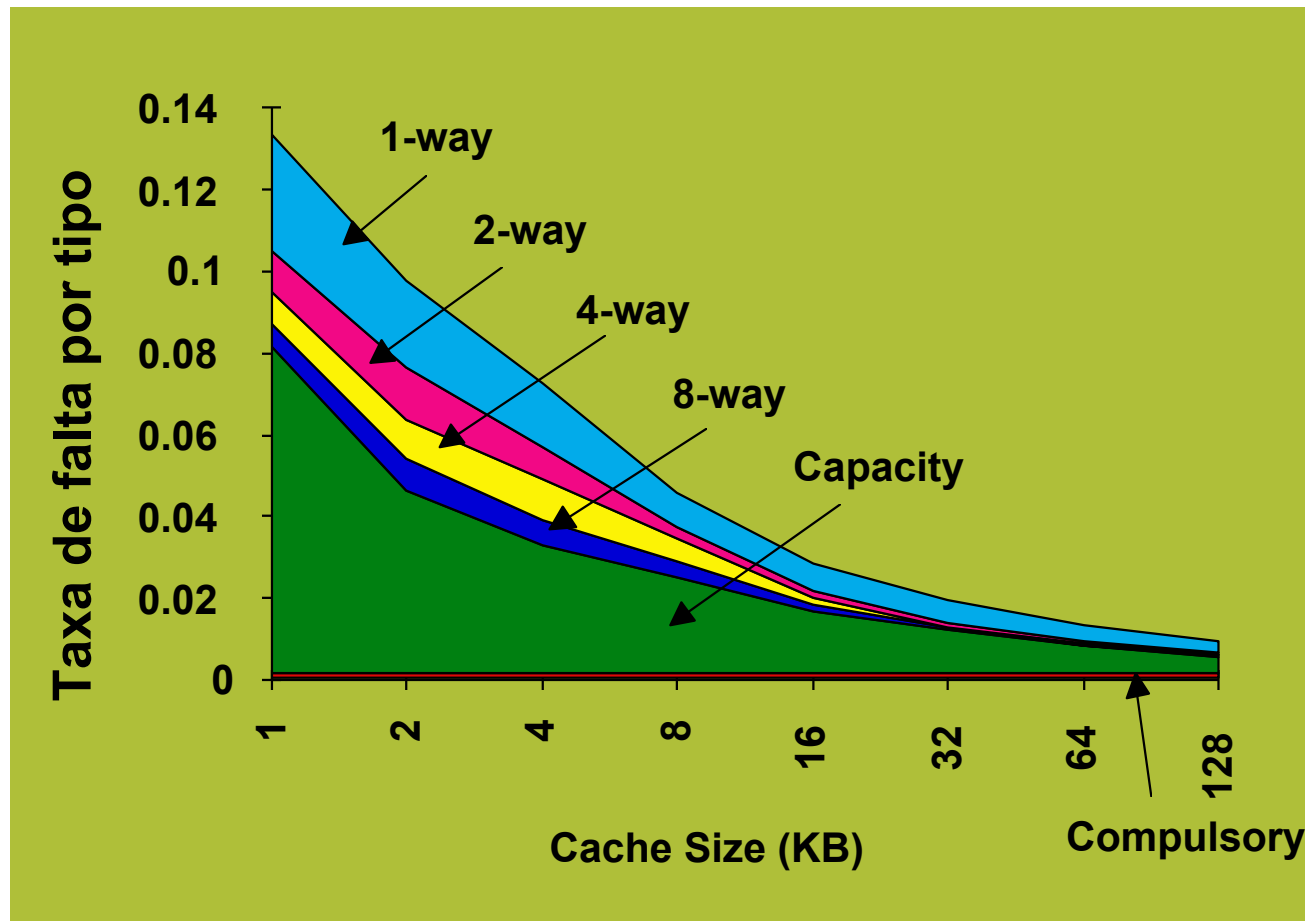
- Como reduzir tempo médio de acesso à memória
 - ❑ Reduzir taxa de falta
 - ❑ Reduzir penalidade por falta
 - ❑ Reduzir tempo de acerto

1. Tamanho do bloco maior



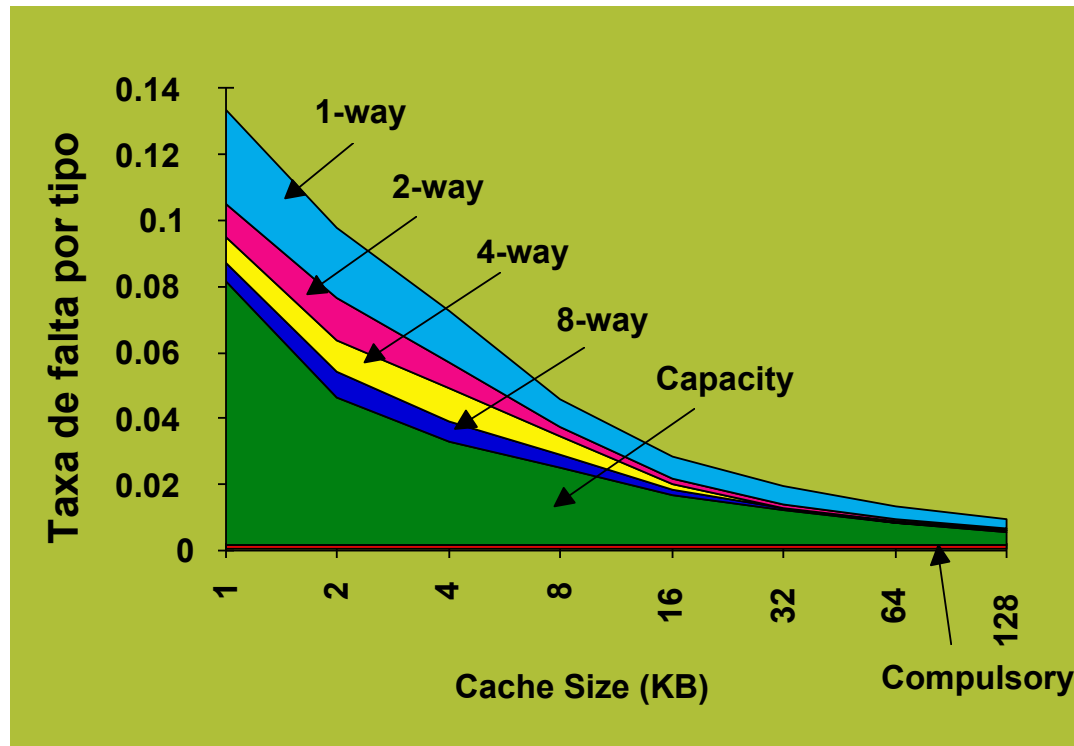
- ❑ Reduz faltas obrigatórias
- ❑ Aumenta as faltas de conflito e capacidade, aumenta a penalidade por falta

2. Aumentar a capacidade total da cache para reduzir a taxa de faltas



- Aumenta o tempo de acerto, aumenta o consumo de energia

3. Aumentar a associatividade para diminuir taxa de faltas



- Reduz faltas de conflito
- Aumenta o tempo de acerto, aumenta o consumo de energia

4. Reduzir a penalidade por falta com mais níveis de cache

■ Equações para L2:

- Tempo de acesso médio à memória = Tempo de acerto_{L1} + Taxa de falta_{L1} penalidade por falta_{L1}
- Penalidade por falta_{L1} = Tempo de acerto_{L2} + Taxa de falta_{L2} × Penalidade por falta_{L2}
- Tempo de acesso médio à memória = Tempo de acerto_{L1} + Taxa de falta_{L1} × (Tempo de acerto_{L2} + Taxa de falta_{L2} × Penalidade por falta_{L2})
- Definições:
- Taxa local de falta— número de faltas em uma cache dividido pelo número total de acessos à memória nesta cache (Em L2= taxa de falta_{L2})
- Taxa global de falta— número de faltas em uma cache dividido pelo número total de acessos à memória gerados pela CPU (Taxa de falta_{L1} × Taxa de falta_{L2})

5. Prioridade para faltas de leitura sobre faltas de escritas para reduzir penalidade por falta

- Executa atendimento de falta de leitura com prioridade sobre falta de escrita: Utiliza buffer de escrita
- Problemas:
 - Caches Write-through :
 - Pode acontecer que a falta de leitura tenha conflito com o buffer de escrita
 - Pode esperar o buffer de escrita esvaziar antes de tratar a falta de leitura que pode aumentar a penalidade por falta na leitura em 50%
 - Verifica se existe conflito entre o buffer de escrita e o que deve ser lido na falta de leitura; Se não existir conflito, continua com o acesso à memória para atender a falta da leitura
 - Caches Write-back :
 - Suponha que falta de leitura vai fazer com que um bloco modificado na cache seja utilizado
 - Procedimento normal: Escreve o bloco modificado na memória, e depois trata a falta de leitura
 - Ao invés disso, copia o bloco modificado para um buffer de escrita, trata a falta de leitura trazendo o bloco da memória, e depois escreve na memória o conteúdo do buffer
 - CPU para menos porque volta a executar a instruções assim que a operação de leitura é completada

Seis otimizações da cache

- Seis otimizações básicas de cache:
 - Tamanho de bloco maior
 - Reduz faltas obrigatórias
 - Aumenta as faltas de conflito e capacidade, aumenta a penalidade por falta
 - Maior capacidade total da cache para reduzir a taxa de faltas
 - Aumenta o tempo de acerto, aumenta o consumo de energia
 - Maior associatividade
 - Reduz erros de conflito
 - Aumenta o tempo de acerto, aumenta o consumo de energia
 - Maior número de níveis de cache
 - Reduz o tempo total de acesso à memória
 - Dar prioridade para faltas de leitura sobre faltas de escritas
 - Reduz a penalidade de falta
 - Evitar a conversão de endereços na indexação de cache
 - Reduz o tempo de acerto