INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 26: Desempenho de memórias

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

Sumário

- Tipos de falhas de cache
- Prefetching
- Efeitos do hardware
- Efeitos do software
- Programação ciente da cache
- Exercícios
- Considerações finais

TIPOS DE FALHAS DE CACHE

Tipos de falhas de cache

- Falha de cache
  - *Cache fault*
  - Instrução ou dado não encontrado na memória cache
  - **Nem todas as falhas de cache são iguais**

Tipos de falhas de cache

- **Falhas compulsórias**
  - Primeiro acesso a um bloco qualquer sempre falha
    - Cada bloco só é trazido após ser requisitado!
  - Também chamados de *cold start misses*
  - São as falhas que aconteceriam mesmo em uma cache infinita

Tipos de falhas de cache

- Exemplo
  - Cache com quatro blocos
    - Mapeamento direto
    - Sequência de acesso a blocos: 0, 1

End. do bloco	**Acerto ou falha**	**Conteúdo dos blocos de cache após referência**			
		0	1	2	3
Bloco 0	**Falha**	**Mem[0]**			
Bloco 1	**Falha**	Mem[0]	**Mem[1]**		

Tipos de falhas de cache

- **Falhas de capacidade**
  - Falhas que acontecem quando o conjunto de trabalho não cabe na cache
    - O programa usa mais dados do que a cache consegue armazenar
    - Blocos são retirados e buscados novamente
    - Resolvidos com uma cache infinita
  - São as falhas que aconteceriam em uma cache de tamanho X

Tipos de falhas de cache

- Exemplo
  - Cache com dois blocos
    - Mapeamento totalmente associativo
    - Sequência de acesso a blocos: 0, 1, 2, 0

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência	
		0	1
Bloco 0	Falha		Mem[0]
Bloco 1	Falha	Mem[1]	Mem[0]
Bloco 2	Falha	Mem[1]	Mem[2]
Bloco 0	**Falha**	Mem[1]	**Mem[0]**

Tipos de falhas de cache

- **Falhas de conflito**
  - Falhas que acontecem quando múltiplos blocos são mapeados para a mesma posição
    - Blocos são retirados e buscados novamente
    - Acontece em mapeamento direto e associativo por conjunto
    - Resolvidos por uma cache totalmente associativa
  - São as falhas que aconteceriam em uma cache de tamanho X e associatividade Y

Tipos de falhas de cache

- Exemplo
  - Cache com quatro blocos
    - Mapeamento 2-associativo
    - Sequência de acesso a blocos: 0, 2, 4, 0

End. do bloco	**Acerto ou falha**	**Conteúdo dos blocos de cache após referência**			
		Conjunto 0		Conjunto 1	
		Posição 0	Posição 1	Posição 0	Posição 1
Bloco 0	Falha	Mem[0]			
Bloco 2	Falha	Mem[0]	Mem[2]		
Bloco 4	Falha	Mem[4]	Mem[2]		
Bloco 0	**Falha**	Mem[4]	**Mem[0]**		

PREFETCHING

Prefetching

- Voltando as falhas compulsórias...
  - E se eu pudesse **prever quais blocos serão usados num futuro próximo**?
    - Detectar que acessos à memória são sequenciais ou
    - Identificar um padrão de acesso saltando de 8 em 8 blocos...
  - *Prefetch*
    - Ato de buscar algo antes de ser necessário
    - Funciona para qualquer memória :D

Prefetching

- Exemplos
  - Trazer para a memória principal um programa que costuma ser executado pelo usuário
  - Baixar uma página subsequente da Internet

Prefetching

- Exemplo
  - Cache com quatro blocos
    - Mapeamento direto
    - Sequência de acesso a blocos: 0, 1, 2, 3
      - **Sem prefetching**

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0	**Falha**	**Mem[0]**			
Bloco 1	**Falha**	Mem[0]	**Mem[1]**		
Bloco 2	**Falha**	Mem[0]	Mem[1]	**Mem[2]**	
Bloco 3	**Falha**	Mem[0]	Mem[1]	Mem[2]	**Mem[3]**

Prefetching

- Exemplo
  - Cache com quatro blocos
    - Mapeamento direto
    - Sequência de acesso a blocos: 0, 1, 2, 3
      - **Com prefetching do próximo bloco**

End. do bloco	Acerto ou falha	Conteúdo dos blocos de cache após referência			
		0	1	2	3
Bloco 0	**Falha**	**Mem[0]**	**Mem[1]**		
Bloco 1	**Acerto**	Mem[0]	**Mem[1]**	**Mem[2]**	
Bloco 2	**Acerto**	Mem[0]	Mem[1]	**Mem[2]**	**Mem[3]**
Bloco 3	**Acerto**	Mem[0]	Mem[1]	Mem[2]	**Mem[3]**

Prefetching

- Nem sempre resolve os problemas
  - Padrões de acesso difíceis de detectar
  - Códigos com pouca localidade espacial
    - Traz mais dados do que o necessário
    - Pode aumentar falhas de conflito

EFEITOS DO HARDWARE

Efeitos do hardware

- Características da hierarquia de cache que influenciam o desempenho
  - **Tempo de acesso**
  - **Taxa de falha**
  - **Penalidade de falha**

Efeitos do hardware

- Fatores que afetam o tempo de acesso
  - Tamanho da cache
  - Associatividade
  - Tecnologia de fabricação
- Fatores que afetam a taxa de falha
  - Tamanho da cache
  - Associatividade
  - Tamanho do bloco

Efeitos do hardware

- Exemplo de efeito sobre o tempo de acesso
  - **CACTI**
    - Ferramenta da IBM usada para estimar características de caches
    - http://quid.hpl.hp.com:9081/cacti/index.y?new

Tamanho da cache (B)	**4096**	**16384**	**4096**	**4096**	**4096**
Tamanho do bloco (B)	16	16	32	16	16
Associatividade	4	4	4	8	4
N de bancos	1	1	1	1	1
Tecnologia (nm)	32	32	32	32	90
Tempo de acesso (ns)	0.34	0.41	0.40	0.39	1.26

EFEITOS DO SOFTWARE

Efeitos do software

- Software influencia a taxa de falhas
  - **Estruturas de dados com pouca localidade espacial**
  - **Estruturas de dados muito grandes**
  - **Algoritmos com pouco reuso de dados**
  - Etc.

Efeitos do software

- Exemplo: Arrays vs Listas encadeadas
  - Array
    - Array[10], Array[11]
  - Lista encadeada
    - Lista.valor, Lista.próximo
      - Ocupa mais memória
      - Precisa de dois acessos para chegar no próximo valor
      - Menor localidade espacial

Efeitos do software

- Exemplo: Acesso a matrizes
  - Por linha
    - Matriz[i][j], Matriz[i][j+1]
  - Por coluna
    - Matriz[i][j], Matriz[i+1][j]
  - Como matrizes são alocadas em memória
    - Matrix[i][j] -> Matrix[i*#colunas+j]

Efeitos do software

- Exemplo: Acesso a matrizes
  - **Código C no Moodle**
  - Ferramenta perf do Linux
    - Pega informações de contadores de hardware
      - /usr/bin/perf stat -e L1-dcache-loads -e L1-dcache-load-misses -e LLC-load-misses -e cache-misses -e instructions

#lin	#col	Passando por linhas			Passando por colunas		
		L1 misses	% dos acesso	Tempo	L1 misses	% dos acesso	Tempo
10^3	10^3	113.681	1,21%	0,007 s	1.077.563	11,47%	0,010 s
10^4	10^4	12.716.208	1,41%	0,468 s	106.824.506	11,84%	2,451 s

PROGRAMAÇÃO CIENTE DA CACHE

Programação ciente da cache

- **O que o programador e o compilador podem fazer para reduzir os misses?**
- Instruções
  - Reordenar procedimentos em memória
    - Maior localidade espacial
  - Decompor procedimentos em blocos pequenos
    - Maior localidade espacial
  - Perfilar a aplicação para encontrar conflitos

Programação ciente da cache

- O que o programador e o compilador podem fazer para reduzir os misses?
- Dados
  - **Fundir arrays**
    - Maior localidade espacial

De: int valor[10], int chave[10]

Para: par arrays_fundidos[10], onde

typedef struct par_t{ int valor; int chave;} par;

Programação ciente da cache

- O que o programador e o compilador podem fazer para reduzir os misses?
- Dados
  - **Fundir arrays**
    - Maior localidade espacial

De: double real[10], double imaginario[10]

Para: complexo valores[10], onde

```
typedef struct comp_t{
double real; double imaginario} complexo;
```

Programação ciente da cache

- O que o programador e o compilador podem fazer para reduzir os misses?
- Dados
  - **Permuta de loops**
    - Trocar a ordem/aninhamento de laços para melhorar padrão de acesso
    - Maior localidade espacial

```
for ( j=0; j<columns; j++ )
    for ( i=0 ; i<rows ; i++ )
        matrix[i][j] = 0;
```

```
for ( i=0 ; i<rows ; i++ )
    for ( j=0; j<columns; j++ )
        matrix[i][j] = 0;
```

Programação ciente da cache

- O que o programador e o compilador podem fazer para reduzir os misses?
- Dados
  - **Fusão de loops**
    - Fundir múltiplos laços de mesmo número de iterações

```
for ( i=0 ; i<rows ; i++ )
      for ( j=0; j<columns; j++ )
            a[i][j] = b[i][j] + c[i][j];
for ( i=0 ; i<rows ; i++ )
      for ( j=0; j<columns; j++ )
            d[i][j] = a[i][j] * a[i][j];
```

```
for ( i=0 ; i<rows ; i++ )
      for ( j=0; j<columns; j++ ){
            a[i][j] = b[i][j] + c[i][j];
            d[i][j] = a[i][j] * a[i][j];
      }
```

Programação ciente da cache

- Dados
  - ***Blocking***
    - Acesso aos dados em blocos ao invés de linhas ou colunas
    - Maior localidade espacial e temporal
    - Exemplo: algoritmo de Smith-Waterman

$$H(i,0) = 0,\ 0 \le i \le m$$
$$H(0,j) = 0,\ 0 \le j \le n$$
$$H(i,j) = \max \begin{Bmatrix} 0 & \\ H(i-1,j-1) + s(a_i,b_j) & \text{Match/Mismatch} \\ \max_{k \ge 1}\{H(i-k,j) + W_k\} & \text{Deletion} \\ \max_{l \ge 1}\{H(i,j-l) + W_l\} & \text{Insertion} \end{Bmatrix},\ 1 \le i \le m, 1 \le j \le n$$

Programação ciente da cache

- Dados
  - ***Blocking***
    - Acesso aos dados em blocos ao invés de linhas ou colunas
    - Maior localidade espacial e temporal
    - Exemplo: algoritmo de Smith-Waterman

	-	A	C	G	A
-	0	0	0	0	0
C	0	0	2	1	0
G	0	0	1	4	3
C	0	0	2	3	2
A	0	2	1	2	5

	-	A	C	G	A
-	0	0	0	0	0
C	0	←	↖	←	←
G	0	←	↑	↖	←
C	0	←	↖	↑	←
A	0	↖	←	↑	↖

Programação ciente da cache

- Dados
  - ***Blocking***
    - Acesso aos dados em blocos ao invés de linhas ou colunas
    - Maior localidade espacial e temporal
    - Exemplo: algoritmo de Smith-Waterman

Acesso por linhas

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Acesso por colunas

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Acesso por blocos

1	2	5	6
3	4	7	8
9	10	13	14
11	12	15	16

EXERCÍCIOS

Exercícios

- Dada a sequência de instruções MIPS-32 ao lado, aponte o número de cache misses para as seguintes configurações de caches:
  - Cache de 32B com blocos de 8B
    - Com mapeamento direto
    - Totalmente associativa
    - 2-associativa

```
lui  $s0, 0xAAAA
lw $t0, 0($s0)
lw $t2, 8($s0)
lw $t1, 4($s0)
lw $t3, 16($s0)
lw $t4, 24($s0)
sw $t0, 12($s0)
sw $t1, 20($s0)
sw $t2, 36($s0)
sw $t3, 28($s0)
sw $t4, 0($s0)
```

Exercícios

- Quais são os tamanhos das tags e o tamanho efetivo das seguintes configurações de cache?
  - Cache de 32B com blocos de 8B
    - Com mapeamento direto
    - Totalmente associativa
    - 2-associativa

```
lui  $s0, 0xAAAA
lw $t0, 0($s0)
lw $t2, 8($s0)
lw $t1, 4($s0)
lw $t3, 16($s0)
lw $t4, 24($s0)
sw $t0, 12($s0)
sw $t1, 20($s0)
sw $t2, 36($s0)
sw $t3, 28($s0)
sw $t4, 0($s0)
```

CONSIDERAÇÕES FINAIS

Considerações finais

- Tipos de falhas
- Prefetching
- Como o hardware e o software podem influenciar o desempenho das caches

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 26: Desempenho de memórias

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br