



Laboratório 07
Matriz em Cache

Alunas: Jéssica Regina dos Santos e Myllena Correa
Disciplina: Organização de Computadores I
Professor: Marcelo Daniel Berejuck
Entrega: 04/06/25

Introdução

Este relatório apresenta a análise do desempenho de duas implementações de matrizes (desenvolvidas no Laboratório 05) por meio da ferramenta *Data Cache Simulator*.

Foram simulados cinco cenários distintos de configuração de cache, variando o número de blocos e o tamanho de cada bloco, com o objetivo de comparar a taxa de *cache miss* entre as implementações.

Questão 1

Como foi requisitado, utilizamos o *Data Cache Simulator* do MARS para avaliar o desempenho da cache em cada abordagem. A análise foi realizada com diferentes configurações de cache, variando o número de blocos e o tamanho de cada bloco. As configurações testadas foram 8 blocos de 4 palavras cada, 8 blocos de 8 palavras cada, 16 blocos de 8 palavras cada, 16 blocos de 16 palavras cada e 16 blocos de 32 palavras cada.

O foco da nossa avaliação foi a taxa de falhas de cache (*Cache Miss Rate*) para cada configuração citada.

Tabela de Taxa de Falhas (*Miss Rate*)

	8 blocos de 4 words	8 blocos de 8 words	16 blocos de 8 words	16 blocos de 16 words	16 blocos de 32 words
Exercício 1	25%	12,5%	12,5%	6,25%	3,13%
Exercício 2	100%	100%	100%	6,25%	3,13%

Interpretação dos Resultados

Exercício 1

Como o programa percorre a matriz linha por linha, acessando dados sequencialmente, isso resulta em acessos contínuos à mesma região da cache, exceto no início de cada bloco, onde ocorrem falhas (*misses*).

Para as configurações 8x4, 8x8 e 16x8, notamos que a taxa de falhas é inversamente proporcional ao tamanho do bloco. Por exemplo, em blocos de 4 palavras, $\frac{1}{4}$ dos acessos são falhas, resultando em uma taxa de 25% e em blocos de 8 palavras $\frac{1}{8}$ dos acessos são falhas, resultando em uma taxa de 12,5%.

Exercício 2

Dessa vez, o programa percorre a matriz coluna por coluna, saltando 16 endereços de memória por vez. Esse padrão de acesso não aproveita a localidade espacial, causando falhas em praticamente todos os acessos nas configurações 8x4, 8x8 e 16x8 (taxa de 100%).

Como nas configurações 16x16 e 16x32 a cache é grande o suficiente para armazenar toda a matriz ou grande parte dela, apenas o primeiro acesso a cada bloco resulta em falha, levando a taxas de 6,25% para blocos de 16 palavras ($1/16$) e 3,13% para blocos de 32 palavras ($1/32$).

Conclusões

Os resultados do Exercício 1 demonstraram a importância do acesso sequencial para aproveitar a localidade espacial, reduzindo significativamente a taxa de falhas e os do Exercício 2 ilustraram como padrões de acesso não sequenciais podem degradar o desempenho da cache, especialmente em configurações menores – configurações de cache maiores (16x16 e 16x32) mitigam esse problema, pois acomodam melhor os dados, reduzindo as falhas mesmo para padrões de acesso não ideais.