

# Simulador de Cache

## Organização de Computadores I

Juliana Miranda Bosio, Vinícios Rosa Buzzi

<sup>1</sup>Departamento de Informática e Estatística – Universidade Federal de Santa Catarina  
Caixa Postal 5094 – 88035-972 – Florianópolis – SC – Brazil

### 1. Objetivo

O objetivo deste relatório é analisar o desempenho do simulador de cache utilizando o MIPS, através da execução de dois programas implementados no Laboratório 04 - Manipulação de Matrizes em MIPS: Preenchimento Linha a Linha e Coluna a Coluna.

### 2. Procedimento de Simulação

O procedimento de simulação para ambos os programas envolve os seguintes passos:

1. **Configuração da Cache:**
  - Especifique a configuração inicial da cache (por exemplo, tamanho do bloco de 4 palavras, 16 blocos, mapeamento direto).
2. **Ajuste o controle deslizante *Run Speed* para 30 instruções por segundo:**
  - Isso torna a execução mais lenta para que você possa observar a animação do desempenho da cache.
3. **Execute o programa:**
  - Clique no botão *Go* na barra de ferramentas ou pressione *F5*.
  - Observe a animação do desempenho da cache conforme ela é atualizada com cada acesso à memória.
4. **Registre os Resultados:**
  - Anote a taxa final de acertos do cache.
  - Modifique o tamanho do bloco para 8 palavras e execute novamente o programa para anotar a nova taxa de acertos.
  - Modifique o tamanho do bloco para 2 palavras e execute novamente o programa para anotar a nova taxa de acertos.
5. **Modifique a configuração da cache (quando necessário):**
  - Crie uma segunda instância do *Data Cache Simulator* selecionando novamente *Data Cache Simulator* no menu *Tools*.
  - Na nova instância, ajuste o tamanho do bloco e o número de blocos conforme necessário.
  - Conecte a nova instância ao MIPS e execute novamente o programa.
  - Registre o desempenho do cache para a primeira e segunda instância.

### 3. Preenchimento Linha a Linha

#### 3.1. Resultados

Os resultados obtidos foram:

- Taxa de acertos com 8 blocos de 16 palavras: 94%
- Taxa de acertos com 8 blocos de 8 palavras: 88%
- Taxa de acertos com 8 blocos de 4 palavras: 75%
- Taxa de acertos com 8 blocos de 2 palavras: 50%

## 4. Preenchimento Coluna a Coluna

### 4.1. Resultados

Os resultados obtidos foram:

- Taxa de acertos com 8 blocos de 16 palavras: 50%
- Taxa de acertos com 8 blocos de 8 palavras: 46%
- Taxa de acertos com 8 blocos de 4 palavras: 40%
- Taxa de acertos com 8 blocos de 2 palavras: 26%

Criando uma segunda instância do Simulador de Cache e executando novamente o programa utilizando 16 blocos de 16 palavras, obtivemos uma taxa de acertos de 97%. Tal procetagem, quando comparada com a primeira definida em 8 blocos de 16 palavras, é nitidamente maior, pois na primeira configuração o desempenho é de 46%. Contudo, quando a comparação é feita apenas na parte pura do código, sem o segundo acesso a memória para o print dos resultados, a diferença é ainda maior, uma vez que para 8 blocos de 16 palavras a taxa de acertos é de 0%

pois em cada acesso a memória, a configuração implementada faz o carregamento da matriz na cache. Se observado apenas a parte pura do c

## 5. Conclusão

Em suma, a implementação que obteve maior desempenho em relação ao uso de memória foi a primeira, pois ela tira melhor proveito da localidade espacial, ou seja, como a matriz é armazenada de forma linear, acessar linha a linha implica acessar posições contíguas de memória maximizando o uso da cache. De modo contrário, a segunda implementação ao saltar de 16 em 16 palavras, acaba por mapear frequentemente para diferentes blocos da cache, e, como o mapeamento só permite armazenar um endereço por linha, a maioria dos acessos resulta em `miss`.

Todavia, quando utilizamos 16 blocos com 16 palavras ambos os programas apresentam resultados similares, pois o tamanho do bloco da cache permite que elementos de uma mesma coluna caibam no mesmo bloco, por exemplo, na segunda implementação.

## 6. Repositório da Atividade

Confira os códigos implementados acessando o *QRCode* abaixo ou pelo link:

<https://github.com/buzziologia/UFSC/tree/main/OrganizacaoDeComputadores/Laboratorio06>



Figure 1. Repositório Github

## Anexo

Nesta seção, estão apresentados os resultados obtidos durante as simulações da cache utilizando os códigos implementados no Laboratório 05.

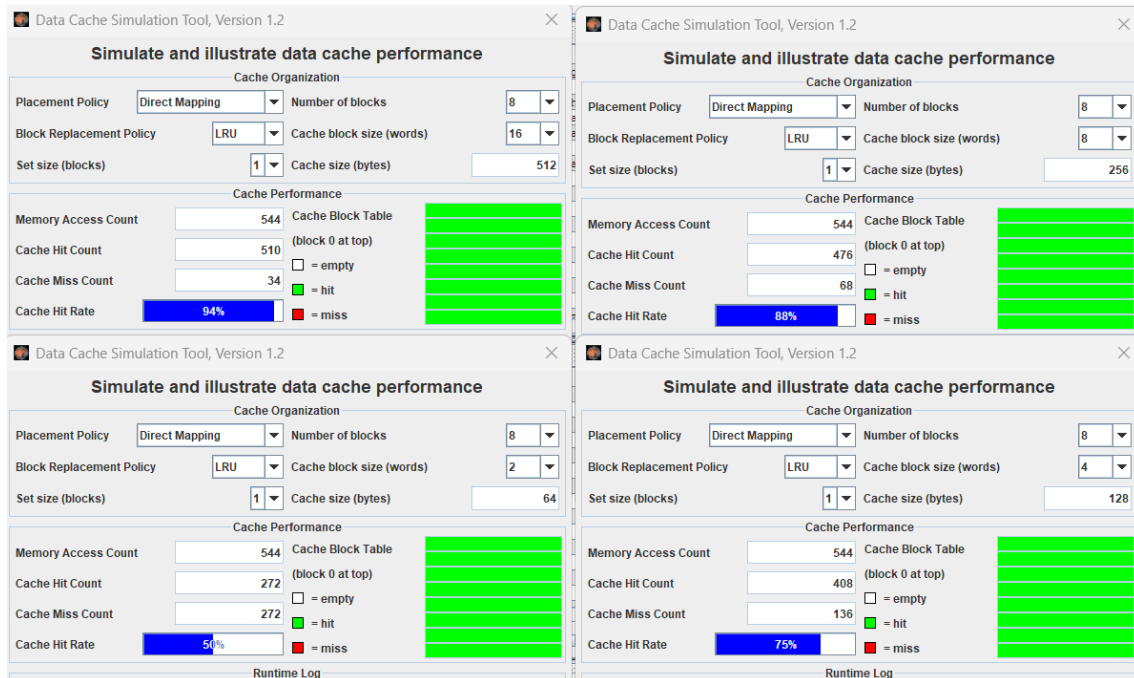
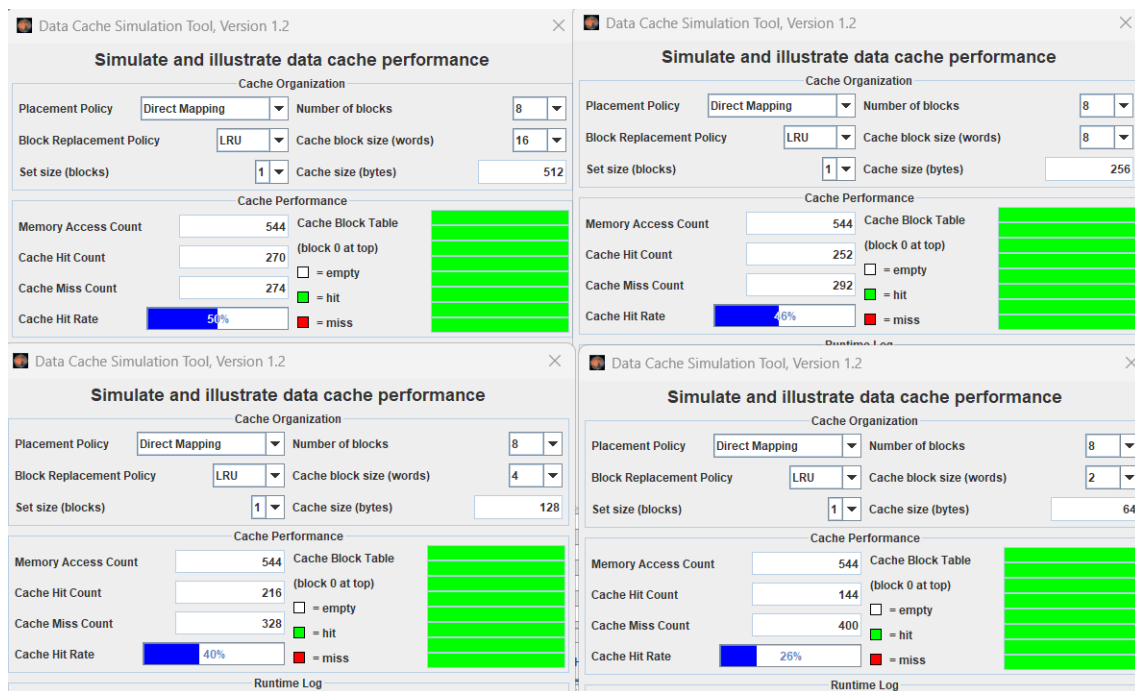


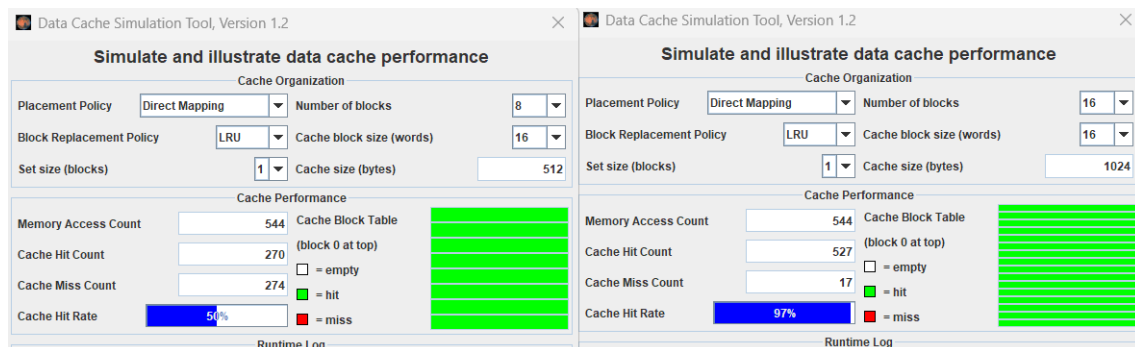
Figure 2. Simulação da Cache: Preenchimento Linha a Linha.

A Figura 2 ilustra o desempenho da cache ao executar o programa que preenche a matriz linha a linha. Essa abordagem aproveita melhor a localidade espacial dos dados, resultando em maior taxa de acertos, especialmente com blocos maiores.



**Figure 3. Simulação da Cache: Preenchimento Coluna a Coluna.**

A Figura 3 mostra o desempenho da cache durante o preenchimento da matriz coluna a coluna. Devido ao salto entre linhas no acesso, o aproveitamento da localidade espacial é menor, resultando em uma redução significativa na taxa de acertos.



**Figure 4. Comparação entre Configurações de Cache: 8x16 e 16x16.**

A Figura 4 apresenta a comparação entre as taxas de acertos das configurações de cache 8x16 e 16x16 para ambas as implementações. Essa análise demonstra como o aumento no número de blocos melhora o desempenho, reduzindo os misses, especialmente no preenchimento coluna a coluna.