Impacto da Localidade de Dados na Performance

Localidade Temporal e Espacial de Cache

Werbert Arles de Souza Barradas

Universidade Federal do Rio Grande do Norte (UFRN)
Disciplina de Programação Paralela - DCA3703

20 de agosto de 2025

Introdução

Objetivo do Estudo

Demonstrar e quantificar o impacto da localidade de dados no desempenho de algoritmos computacionais.

Questão Central

Identificar o ponto a partir do qual o padrão de acesso à memória sequencial (linha a linha) versus não sequencial (coluna a coluna) causa uma divergência significativa no tempo de execução.

Mecanismo Investigado

O fenômeno está diretamente ligado à arquitetura e ao funcionamento da hierarquia de memória cache do processador.

Metodologia do Experimento

- **Implementação:** Duas versões da multiplicação de matriz por vetor ($y = A \cdot x$) foram desenvolvidas em C.
- **Diferença Chave:** A única alteração entre as versões é a ordem dos laços de repetição, forçando diferentes padrões de acesso à memória.
- **Ambiente:** O código foi compilado com GCC, usando o nível de otimização `-O2` e a flag `-fopenmp`.
- **Coleta de Dados:**
  - Foram testadas matrizes quadradas ($N \times N$) com N variando de 32 a 16384.
  - O tempo de execução medido foi a **mediana** de múltiplas execuções para garantir robustez.

Padrões de Acesso à Matriz

Função 1: Acesso por Linhas (Cache-Friendly)

O laço interno percorre as colunas, resultando em acessos a endereços de memória contíguos.

```
void multiply_matrix_vector(
    int rows, int cols,
    double **A, double *x, double *y) {
  for (int i = 0; i < rows; i++) {
    y[i] = 0.0;
    // Laço interno percorre colunas (j)
    for (int j = 0; j < cols; j++) {
      y[i] += A[i][j] * x[j];
    }
  }
}
```

Padrões de Acesso à Matriz

Função 2: Acesso por Colunas (Cache-Unfriendly)

O laço interno percorre as linhas, resultando em acessos não sequenciais e distantes na memória.

```
void multiply_matrix_vector_cols_outer(
    int rows, int cols,
    double **A, double *x, double *y) {
  for (int i = 0; i < rows; i++) {
    y[i] = 0.0;
  }
  // Laço externo percorre colunas (j)
  for (int j = 0; j < cols; j++) {
    // Laço interno percorre linhas (i)
    for (int i = 0; i < rows; i++) {
      y[i] += A[i][j] * x[j];
    }
  }
}
```

Hipótese: O Papel da Cache

Hipótese Principal

O acesso por linhas terá um desempenho superior devido ao princípio da **localidade espacial**.

Mecanismo de Cache

A CPU não lê dados da RAM byte a byte. Ela carrega blocos contíguos chamados *cache lines* (tipicamente 64 bytes).

- **Acesso por Linhas (Cache Hit):** Ao acessar `A[i][0]`, os elementos vizinhos (`A[i][1]`, `A[i][2]`, etc.) são carregados juntos na cache. As próximas iterações encontram os dados já disponíveis na memória ultrarrápida.
- **Acesso por Colunas (Cache Miss):** O acesso a `A[i+1][j]` após `A[i][j]` requer um bloco de memória completamente diferente , forçando a CPU a buscar uma nova *cache line* da RAM e causando degradação de performance.

Ambiente de Teste: Hierarquia de Cache

Especificações da Máquina de Teste

Os dados de cache do sistema onde o experimento foi executado são:

- **Cache L1:** 96 KB
- **Cache L2:** 2.5 MB
- **Cache L3:** 6 MB

Estimativa de Uso de Memória por Matriz

- **N=512 (2 MB):** Cabe confortavelmente na cache L2 (2.5 MB).
- **N=1024 (8 MB):** Excede a capacidade total da cache L3 (6 MB).
- **N=2048 (32 MB):** Excede em muito a capacidade de todos os níveis de cache.

Impacto do Padrão de Acesso à Memória (Escala Logarítmica)

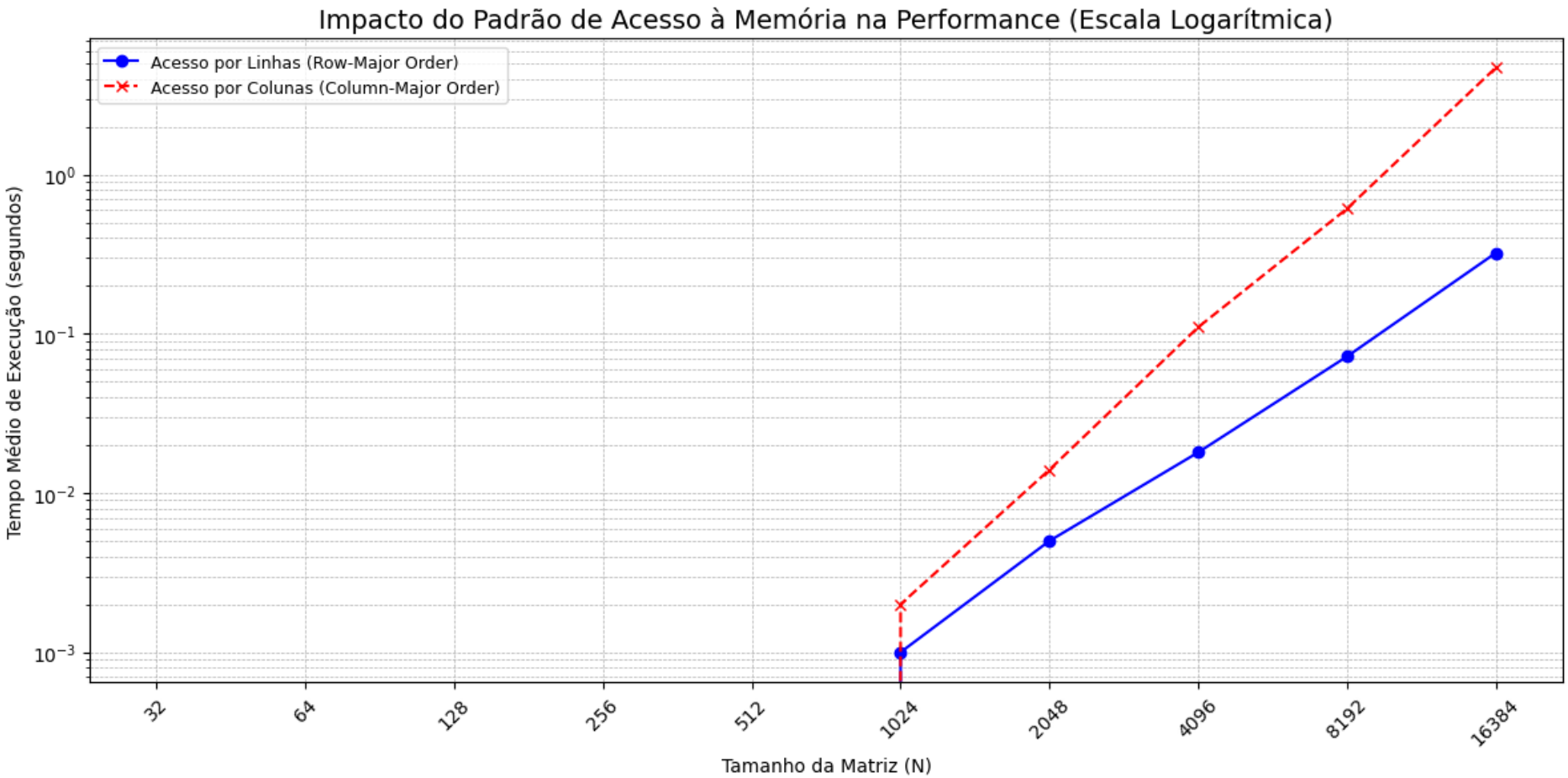

Figure: Comparação do tempo de execução mediano para acesso por linhas e colunas.

Crescimento do Fator de Lentidão vs. Tamanho da Matriz

Métrica: Fator de Lentidão

Uma forma de normalizar os resultados para análise, calculada como:

$$\text{Fator de Lentidão} = \frac{\text{Tempo do método por Colunas}}{\text{Tempo do método por Linhas}}$$

Um fator de 5.0, por exemplo, significa que o acesso por colunas foi 5 vezes mais lento.

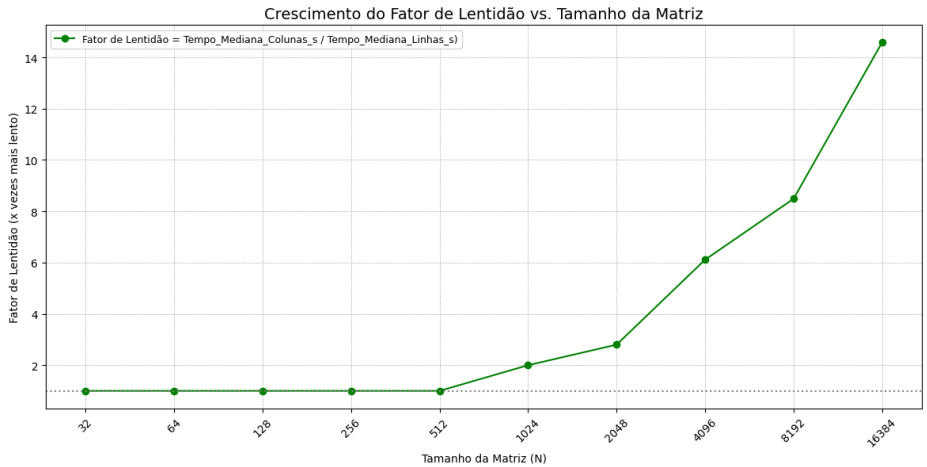

Figure: O fator de lentidão demonstra o custo crescente do acesso não otimizado à memória.

Conclusão

Hipótese Validada

O estudo validou experimentalmente que o desempenho é profundamente influenciado pela maneira como os dados são acessados na memória.

Correlação Direta com a Hierarquia de Cache

A análise dos dados revelou a causa da divergência de performance:

- **Início da Divergência:** Ocorre de forma mensurável quando a matriz excede a capacidade do **cache L2**.
- **Agravamento Drástico:** A degradação se agrava drasticamente quando o volume de dados ultrapassa a capacidade do **cache L3**.
- **Colapso da Performance:** Neste ponto, o acesso não sequencial resulta em falhas de cache constantes, tornando a alta latência da **memória RAM** o principal gargalo de velocidade.