



Análise de Desempenho da Multiplicação de Matrizes por Strassen contra o Método Tradicional

Arthur M. Krause, Gabriel B. Moro, Lucas M. Schnorr
{amkrause, gbmoro, schnorr}@inf.ufrgs.br

Roteiro

- Introdução
- Métodos de Multiplicação de Matrizes
- Otimizações no Uso da Memória Cache
- Metodologia
- Resultados
- Considerações Finais
- Referências Utilizadas

Roteiro

- **Introdução**
- Métodos de Multiplicação de Matrizes
- Otimizações no Uso da Memória Cache
- Metodologia
- Resultados
- Considerações Finais
- Referências Utilizadas

Introdução

- **Motivação**
 - Ampla utilização da operação de multiplicação de matrizes em diversas áreas do conhecimento

Introdução

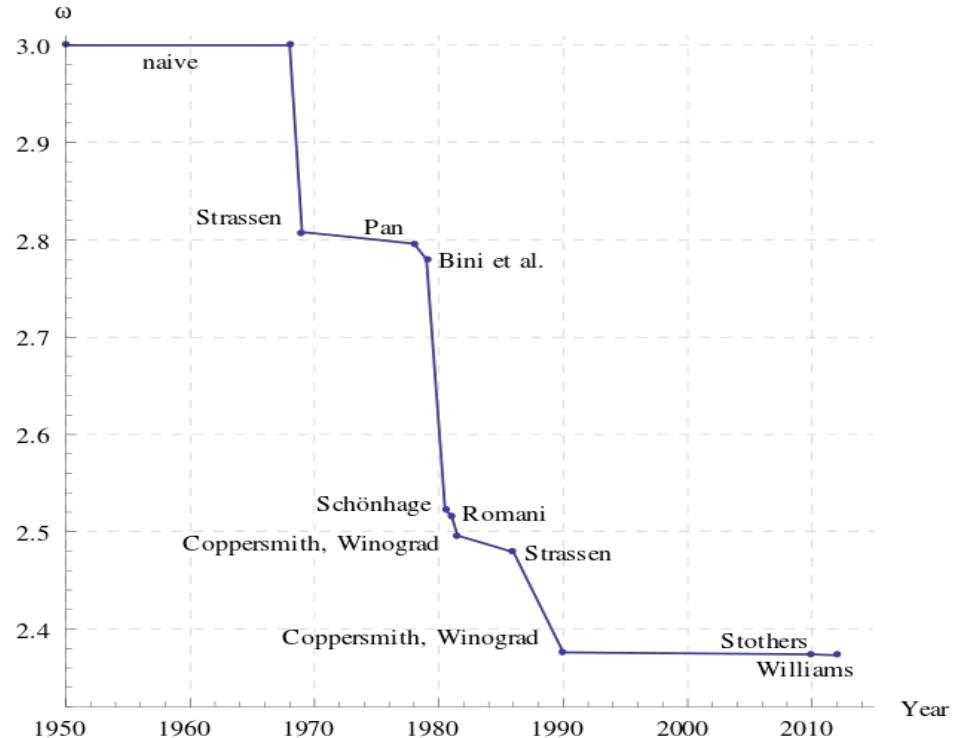
- **Motivação**
 - Ampla utilização da operação de multiplicação de matrizes em diversas áreas do conhecimento
 - Trabalhos recentes analisados:
 - Comparação de diferentes escalonadores OpenMP e uma implementação em Pthreads Andrade e Cera [1]
 - Avaliação de desempenho utilizando diversos níveis de otimização do GCC Silva et al. [2].

Introdução

- **Objetivo**
 - Apresentar uma análise de desempenho para dois algoritmos de multiplicação de matrizes, o **Tradicional** e o de **Strassen**, ambos com algumas otimizações no uso da cache.

Métodos de Multiplicação de Matrizes

- Método Tradicional: $O(n^3)$
- Algoritmo de Strassen: $O(n^{2.81})$



Métodos de Multiplicação de Matrizes

- **Algoritmo Tradicional**

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \times \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 + b_1c_2 & a_1b_2 + b_1d_2 \\ c_1a_2 + d_1c_2 & c_1b_2 + d_1d_2 \end{bmatrix}$$

Métodos de Multiplicação de Matrizes

- **Algoritmo Tradicional**

$$\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \times \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1a_2 + b_1c_2 & a_1b_2 + b_1d_2 \\ c_1a_2 + d_1c_2 & c_1b_2 + d_1d_2 \end{bmatrix}$$

Data: Matrizes de entrada: A e B. Matriz resultado: R.

```
for i a N do
  for j a N do
    for k a N do
      R[i][j] += A[i][k] * B[k][j]
    end
  end
end
```

Algorithm 1: Algoritmo Tradicional para Multiplicação de Matrizes.

Métodos de Multiplicação de Matrizes

- **Algoritmo de Strassen**

- Eficiente para matrizes grandes
- Menos multiplicações

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix}$$

Métodos de Multiplicação de Matrizes

- Algoritmo de Strassen

- Eficiente para matrizes grandes;

- Menos multiplicações [3]. $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix}$

$$\mathbf{M}_1 := (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2})$$

$$\mathbf{M}_2 := (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1}$$

$$\mathbf{M}_3 := \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2})$$

$$\mathbf{M}_4 := \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1})$$

$$\mathbf{M}_5 := (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2}$$

$$\mathbf{M}_6 := (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2})$$

$$\mathbf{M}_7 := (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2})$$

$$\mathbf{C}_{1,1} = \mathbf{M}_1 + \mathbf{M}_4 - \mathbf{M}_5 + \mathbf{M}_7$$

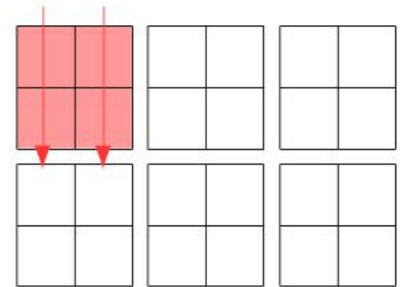
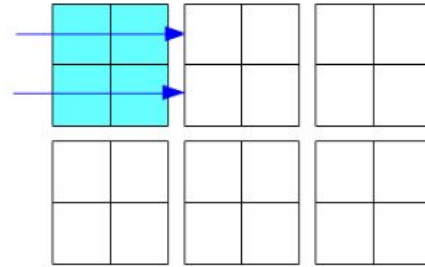
$$\mathbf{C}_{1,2} = \mathbf{M}_3 + \mathbf{M}_5$$

$$\mathbf{C}_{2,1} = \mathbf{M}_2 + \mathbf{M}_4$$

$$\mathbf{C}_{2,2} = \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_6$$

Otimizações de Cache

- **Blocking**
 - Aumenta localidade temporal [4].
- **Matriz B Transposta**
 - Aumenta localidade de dados.
- **Vetorização**
 - Alocação contínua em memória.



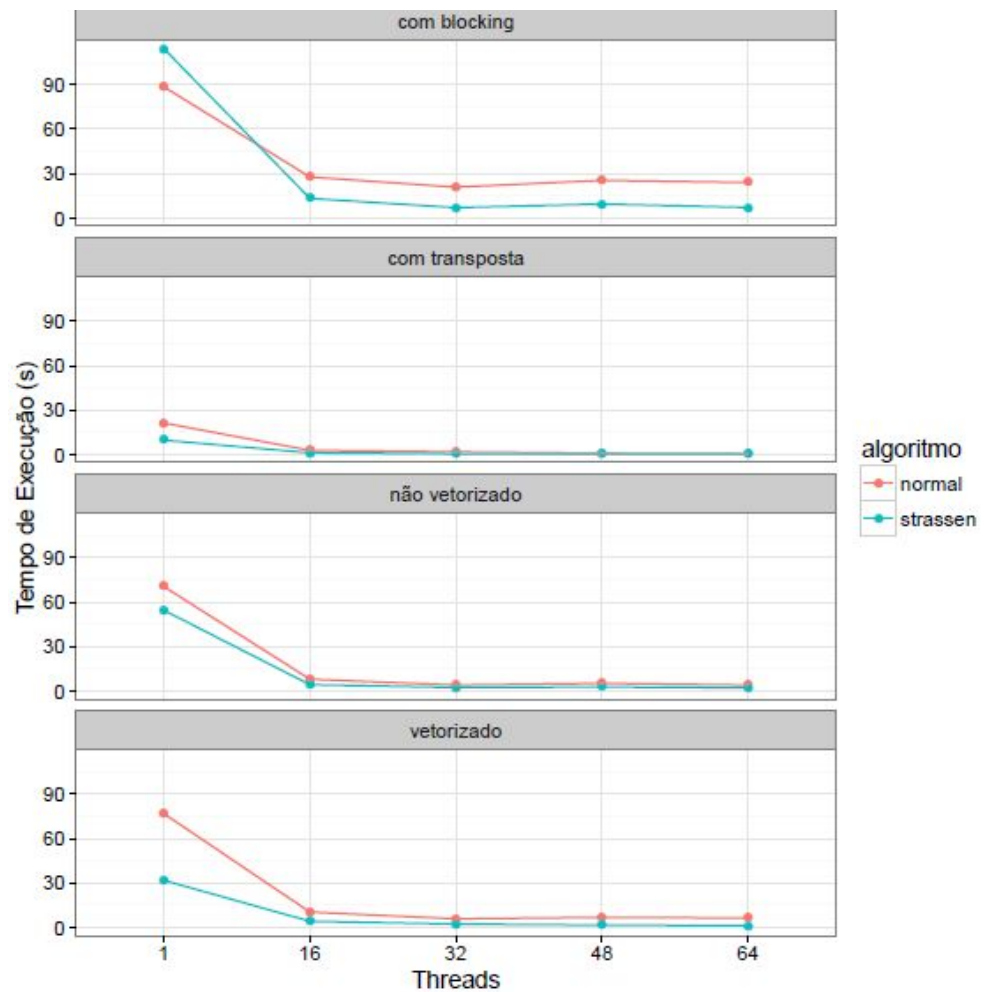
Metodologia

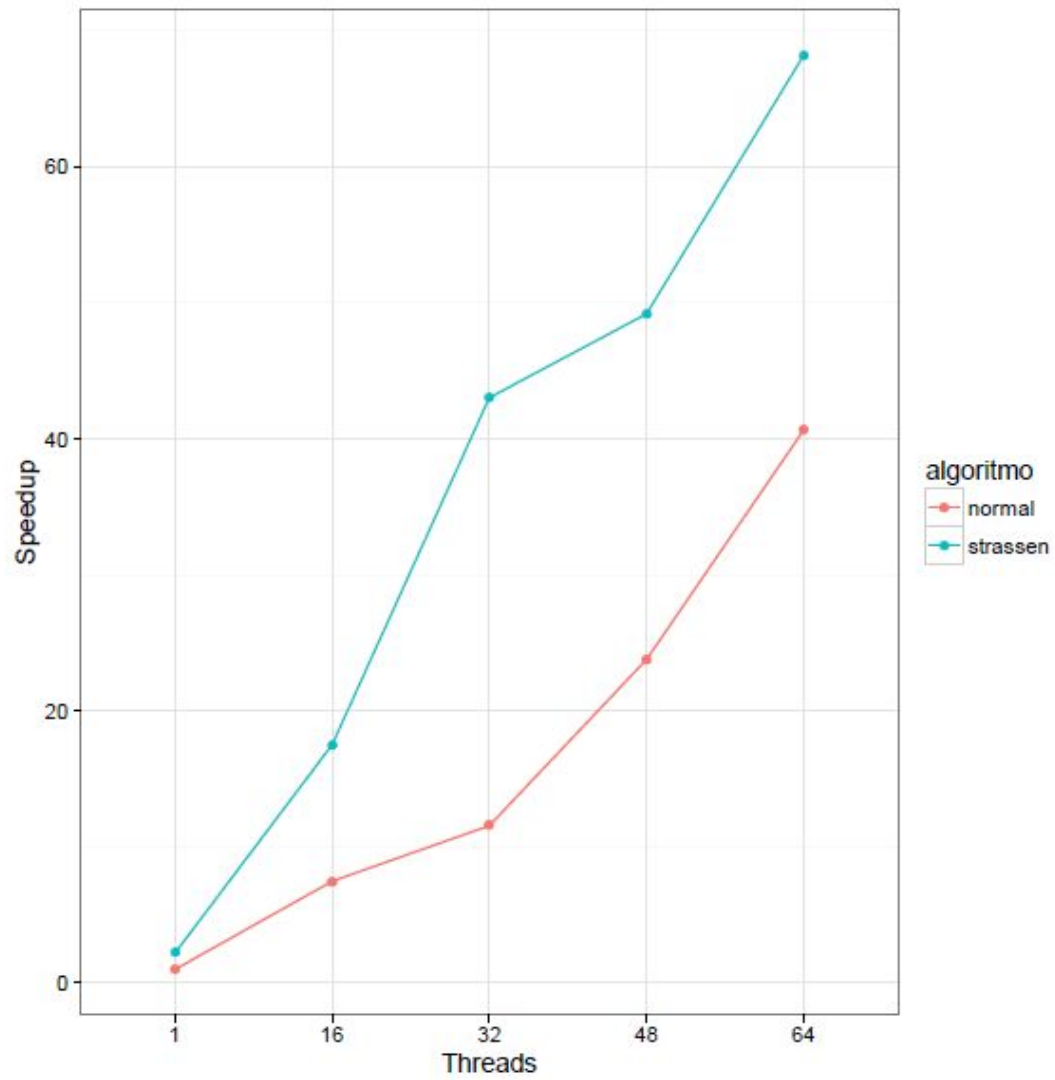
- **Experimento:**
 - Uso da técnica *Design of Experiments* [5];
 - Dimensão das matrizes: 256, 1024 e 2048;
 - Threads: 1, 16, 32, 48 e 64;
 - 15 execuções para cada configuração;
 - Aleatoriedade no experimento.

Metodologia

- **Plataforma de Execução**
 - Máquina Turing;
 - 4 x Intel® Xeon® Processor X7550 @2.00GHz;
 - 8 cores físicos por CPU (2 threads por core);
 - 8 x 32 KB L1;
 - 8 x 256 KB L2;
 - 18 MB L3.

Resultados





Roteiro

- **Introdução**
- **Métodos de Multiplicação de Matrizes**
- **Otimizações no Uso da Memória Cache**
- **Metodologia**
- **Resultados**
- Considerações Finais
- Referências Utilizadas

Roteiro

- **Introdução**
- **Métodos de Multiplicação de Matrizes**
- **Otimizações no Uso da Memória Cache**
- **Metodologia**
- **Resultados**
- **Considerações Finais**
- Referências Utilizadas

Considerações Finais

- **Melhor Combinação**
 - Strassen com transposta
- **Aprendizado**
 - A otimização será eficiente ou não dependendo do tipo de algoritmo utilizado
- **Contribuição principal**
 - Conhecimento do impacto das diferentes otimizações de cache para os dois algoritmos de multiplicação de matrizes

Considerações Finais

- **Trabalho Futuro**

- Rastrear a taxa de cache misses das diferentes versões implementadas com as otimizações de cache para compreender o momento da aplicação que a otimização vale a pena.

Referências Utilizadas

- [1] G. L. Andrade and M. C. Cera, **“Paralelização de uma multiplicação de matrizes utilizando openmp,”** Anais do Salão Internacional de Ensino, Pesquisa e Extensão, vol. 7, no. 2, 2016.
- [2] S. A. da Silva and C. Schepke, **“Análise de desempenho de aplicações paralelas em arquiteturas multi-core e many-core,”** Anais da XVI Escola Regional de Alto Desempenho, 2016.
- [3] T. Cormen, Introduction to Algorithms. MIT Press, 2009.
- [4] M. Kowarschik and C. Weiß, **“An Overview of Cache Optimization Techniques and Cache-Aware Numerical Algorithms,”** Algorithms for Memory Hierarchies, vol. 2625, pp. 213–232, 2003.
- [5] R. Jain, **Art of Computer Systems Performance Analysis: Techniques For Experimental Design Measurements Simulation and Modeling.** Wiley, 1991.

Obrigado a todos!

Esta investigação recebe fundos do programa H2020 da União Europeia e do MCTI/RNP-Brasil através do projeto HPC4E com o código 689772, do projeto FAPERGS/Inria ExaSE, do projeto universal do CNPq 447311/2014-0, e do laboratório internacional CNRS/LICIA.



Análise de Desempenho da Multiplicação de Matrizes por Strassen contra o Método Tradicional

Arthur M. Krause, Gabriel B. Moro, Lucas M. Schnorr
{amkrause, gbmoro, schnorr}@inf.ufrgs.br