

# Análise de Desempenho da Multiplicação de Matrizes em Diferentes Modelos de Raspberry Pi

Gabriella Osório Ribeiro<sup>1</sup>, Lucas Freire Sêmeler<sup>2</sup>, Wanderson Roger Azevedo Dias<sup>2</sup>

<sup>1</sup>Coordenadoria do Curso Técnico em Informática (CCTI)

<sup>2</sup>Coordenadoria do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (CCSTADS)  
Laboratório de Arquiteturas Computacionais e Computação Paralela (LACCP)  
Instituto Federal de Rondônia (IFRO)  
Ji-Paraná – RO – Brasil

{gabriellaosorioribeiro, lucasemeler, wradias}@gmail.com

**Resumo.** Este estudo avaliou o desempenho da computação paralela na execução do algoritmo de multiplicação de matrizes usando a biblioteca OpenMP em diferentes modelos da plataforma Raspberry Pi (2, 3, 4 e 5). Os resultados mostraram que a paralelização trouxe ganhos significativos em desempenho, com a Raspberry Pi 5 se destacando como sendo a mais eficiente em todas as execuções, apresentando um speedup máximo de 4,06x em relação à implementação do algoritmo na versão sequencial. As melhorias foram notáveis em matrizes maiores, executando na Raspberry Pi 5 sendo 93,96% mais eficiente em comparação à Raspberry Pi 2 na execução sequencial. O uso da biblioteca OpenMP provou ser eficaz, demonstrando a viabilidade do uso de plataformas de baixo custo para aplicações que exigem alto poder computacional. Além disso, os dados sugerem que a implementação de técnicas de paralelismo em ambientes educacionais pode facilitar a compreensão de conceitos avançados de computação de alto desempenho (HPC), tornando essas plataformas uma ferramenta valiosa para estudantes e pesquisadores.

## 1. Introdução

A computação paralela tem ganhado destaque como uma abordagem essencial para melhorar o desempenho de sistemas computacionais, especialmente com dispositivos de baixo custo como a Raspberry Pi (RPI). Essas plataformas têm sido amplamente utilizadas em ambientes educacionais e de pesquisa devido à sua acessibilidade e eficiência. Além disso, a RPI oferece uma excelente relação custo-benefício para experimentos com algoritmos paralelos, permitindo que pesquisadores e estudantes explorem técnicas de paralelismo, mesmo com recursos financeiros limitados, uma vez que a RPI custa em torno de 35 a 55 dólares, conforme o modelo da plataforma. Segundo Ignácio e Dias (2023), a RPI tem se mostrado uma ferramenta poderosa para a implementação de algoritmos paralelos em contextos de ensino e pesquisa.

Algoritmos, como o de multiplicação de matrizes, são cruciais em várias áreas da ciência e engenharia, dado que essa operação está presente em diversas aplicações, desde cálculos científicos até aprendizado de máquina. A multiplicação de grandes matrizes exige um processamento intensivo, tornando-a um caso de teste ideal para avaliar o desempenho de diferentes arquiteturas e abordagens paralelas. O uso de bibliotecas como a OpenMP, que permite a execução de código em múltiplos núcleos de um processador, facilita a paralelização de algoritmos, como o de multiplicação de matrizes, dividindo o trabalho a ser executado entre *threads* de forma eficiente.

Assim, neste artigo foi avaliado o desempenho de algoritmos de multiplicação de matrizes implementados de forma sequencial e paralela, executando em diferentes versões da plataforma Raspberry Pi, incluindo os modelos 2, 3, 4 e 5. A implementação paralela foi realizada utilizando a biblioteca OpenMP (versão 4.5), que possibilitou explorar o paralelismo em sistemas de memória compartilhada. A análise focou em comparar o tempo de execução e o *speedup* obtido com a versão paralela em relação à sequencial, utilizando matrizes de diferentes dimensões (1000×1000, 3000×3000 e 5000×5000) para testar o desempenho dos diversos modelos da plataforma Raspberry Pi. O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta uma breve contextualização, a fim, de ambientar o estudo corrente; a Seção 3 apresenta as análises do desempenho computacional das RPIs; e a Seção 4 finaliza com as conclusões, e ideias para trabalhos futuros.

## 2. Contextualização

### 2.1. Plataforma Raspberry Pi

A Raspberry Pi é um microcomputador de baixo custo, do tamanho de um cartão de crédito, desenvolvido pela *Raspberry Pi Foundation* com o objetivo de promover a educação em informática e facilitar o acesso à tecnologia. Equipado com um processador, memória RAM e interfaces para conexão com periféricos como monitores, teclados e dispositivos de armazenamento, a RPi é capaz de executar sistemas operacionais baseados em Linux e outras plataformas (Upton e Halfacree, 2013). Muito popular em projetos educacionais, de pesquisa e em aplicações como automação, Internet das Coisas (IoT) e computação paralela, a Raspberry Pi oferece uma solução acessível e versátil para desenvolvedores e entusiastas.

### 2.2. Biblioteca OpenMP

OpenMP (*Open Multi-Processing*) é uma interface de programação de aplicação (API - *Application Program Interface*) que suporta a execução paralela de código em sistemas de memória compartilhada, permitindo o desenvolvimento de programas que utilizam múltiplos processadores ou núcleos de forma eficiente. Implementada em linguagens como C, C++ e Fortran, a OpenMP utiliza diretrizes simples e *pragmas* para facilitar a paralelização de tarefas, permitindo que o programador divida o trabalho entre várias *threads*. Muito usada em computação de alto desempenho, OpenMP é amplamente adotada para otimizar o desempenho de algoritmos em ambientes de múltiplos núcleos, sendo uma solução prática e escalável para paralelismo (Chapman *et al.*, 2007).

### 2.3. Algoritmo de Multiplicação de Matrizes

O algoritmo de Multiplicação de Matrizes, implementado tanto de forma sequencial quanto paralela, com a biblioteca OpenMP, realiza o produto de duas matrizes, o qual é possível apenas quando o número de colunas da primeira matriz é igual ao número de linhas da segunda. Se a matriz  $A$  tem dimensões  $m \times n$  (denotada como  $A_{m,n}$ ) e a matriz  $B$  tem dimensões  $n \times p$ , o produto resultante,  $C$ , será uma matriz  $m \times p$ , representada por  $AB$  (ou  $A \times B$ ). Cada elemento  $C_{ij}$  da matriz  $C$  é calculado como o produto escalar da  $i$ -ésima linha da matriz  $A$  com a  $j$ -ésima coluna da matriz  $B$ , para todo  $i$  e  $j$ , onde  $1 \leq i \leq m$  e  $1 \leq j \leq p$ .

## 3. Resultados e Discussão

Para a realização dos testes, foram utilizados os modelos 2, 3, 4 e 5 da plataforma Raspberry Pi. Cada teste foi executado 5 vezes, e a média dos tempos de execução foi considerada tanto para as versões sequenciais quanto para as versões paralelizadas utilizando a biblioteca OpenMP. Os parâmetros de entrada para execução foram as dimensões das matrizes  $1000 \times 1000$ ,  $3000 \times 3000$  e  $5000 \times 5000$ . Os resultados foram analisados considerando o tempo médio de execução e o *speedup* obtido pela implementação paralela em OpenMP em comparação com a versão sequencial. O algoritmo paralelizado usou 4 *threads* em todas as execuções. Para o paralelismo, a matriz  $A$  foi particionada igualmente entre as *threads*, enquanto a matriz  $B$  foi compartilhada entre elas.

A Tabela 1 apresenta os resultados do tempo médio de execução em segundos e o *speedup* obtido pela versão paralela em OpenMP em relação à sequencial, nas diferentes versões da Raspberry Pi (modelos 2, 3, 4 e 5).

Tabela 1. Execução do algoritmo de Multiplicação de Matrizes

Modelo de RPi	Dimensões 1000 x 1000			Dimensões 3000 x 3000			Dimensões 5000 x 5000		
	Serial	OpenMP 4 threads	Speedup	Serial	OpenMP 4 threads	Speedup	Serial	OpenMP 4 threads	Speedup
2	133,37	39,25	3,40	2913,92	1755,30	1,66	31646,26	9132,80	3,47
3	74,90	19,47	3,85	2579,08	1432,28	1,80	20425,50	11292,68	1,81
4	22,89	6,56	3,49	742,13	419,67	1,77	7334,140	4200,50	1,75
5	9,17	2,26	4,06	331,44	85,13	3,89	1910,83	566,950	3,37

Conforme observado na Tabela 1, para o parâmetro de  $1000 \times 1000$ , as RPi 2, 3, 4 e 5 executaram o algoritmo sequencialmente em média nos tempos de 133,37s, 74,90s, 22,89s e 9,17s, respectivamente, enquanto na versão paralela com OpenMP os tempos foram de 39,25s, 19,47s, 6,56s e 2,26s. Os *speedups* máximos atingidos pelas Raspberries Pi 2, 3, 4 e 5 foram, respectivamente, de 3,40x, 3,85x, 3,49x e 4,06x. A Raspberry Pi 5 foi a mais rápida, sendo 93,12% mais eficiente na versão sequencial e 94,24% na versão paralela em comparação à Raspberry Pi 2.

Para o parâmetro de  $3000 \times 3000$ , os tempos médios sequenciais nas RPi 2, 3, 4 e 5 foram de 2913,92s, 2579,08s, 742,13s e 331,44s, respectivamente, enquanto na versão paralela os tempos foram de 1755,30s, 1432,28s, 419,67s e 85,13s. Novamente, a Raspberry Pi 5 foi a mais eficiente, sendo 88,62% mais rápida na versão sequencial e 95,15% na paralela em relação à Raspberry Pi modelo 2.

Para o parâmetro de  $5000 \times 5000$ , os tempos sequenciais foram de 31646,26s, 20425,50s, 7334,14s e 1910,83s para as RPi 2, 3, 4 e 5, respectivamente, enquanto na versão paralela com OpenMP os tempos foram de 9132,80s, 11292,68s, 4200,50s e 566,95s. O *speedup* máximo foi de 3,47x. A Raspberry Pi 5 apresentou 93,96% de eficiência superior à Raspberry Pi 2 implementado na versão sequencial e 93,79% na versão paralela.

Os resultados confirmam que as execuções paralelas com a biblioteca OpenMP apresentaram ganhos de desempenho significativos em relação às sequenciais, de acordo com as expectativas. Além disso, os modelos mais recentes da Raspberry Pi mostraram-se progressivamente mais eficientes que as versões anteriores. Um ponto notável foi que, na dimensão  $5000 \times 5000$ , a RPi 2 apresentou um desempenho 23,65% mais rápido que a Raspberry Pi 3 na execução paralela com OpenMP. No geral, o uso de OpenMP demonstrou ser eficiente, proporcionando ganhos expressivos de desempenho computacional, com a Raspberry Pi 5 se destacando em todas as execuções.

#### 4. Conclusões e Trabalhos Futuros

Os resultados deste estudo demonstram que a computação paralela, com o uso da biblioteca OpenMP, proporcionou ganhos de desempenho significativos na execução do algoritmo de multiplicação de matrizes em diferentes modelos da plataforma Raspberry Pi. Entre os modelos testados, a RPi 5 apresentou o melhor desempenho, superando de maneira expressiva as versões anteriores, tanto nas execuções sequenciais quanto nas paralelas. Para matrizes de dimensões maiores ( $5000 \times 5000$ ), a RPi 5 foi 93,96% mais eficiente na execução sequencial e 93,79% mais eficiente na paralela, em relação à Raspberry Pi 2, comprovando sua superioridade em ambientes de alto processamento. Além disso, os *speedups* obtidos com a paralelização variaram de acordo com o modelo da plataforma, sendo que os mais recentes obtiveram os maiores ganhos, como o caso da RPi 5, que alcançou um *speedup* de até 4,06x.

Para trabalhos futuros, sugere-se: (i) explorar técnicas de paralelização alternativas, como GPUs ou arquiteturas heterogêneas, para comparar o desempenho com o OpenMP em plataformas de baixo custo; (ii) incluir algoritmos com alta demanda computacional, como os de aprendizado de máquina, para avaliar a aplicabilidade da Raspberry Pi em diferentes cenários, e (iii) analisar o impacto da comunicação entre dispositivos em um *cluster* de Raspberry Pi pode oferecer *insights* sobre a escalabilidade para aplicações distribuídas.

#### Agradecimentos

Os autores agradecem ao Instituto Federal de Rondônia (IFRO), pelo apoio financeiro concedido através dos Editais N° 13/2023/REIT - PROPESP/IFRO e N° 14/2023/REIT - PROPESP/IFRO, ambos PIBITI Ciclo 2023-2024, que proporcionaram a execução desta pesquisa.

#### Referências

- Chapman, B.; Jost, G.; Pas, R. V. D. (2007) "Using OpenMP: Portable Shared Memory Parallel Programming". Cambridge, Massachusetts, EUA: MIT Press, 1<sup>st</sup> edition, 353p.
- Ignácio, A. L. J.; Dias, W. R. A. (2023) "Análise do Desempenho Computacional de Algoritmos Paralelizados com OpenMP e MPI Executados em Raspberry Pi". In *Workshop de Iniciação Científica - Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD)*, Porto Alegre, RS, Brasil, pp. 41-48.
- Upton, E., Halfacree, G. (2013) "Raspberry Pi Manual do Usuário". São Paulo, Brasil: Novatec, 269p.