9ª Jornada Científica e Tecnológica do IFSULDEMINAS
6º Simpósio da Pós-Graduação
ISSN 2319-0124

ANÁLISE DE DESEMPENHO DE DIFERENTES LINGUAGENS DE PROGRAMAÇÃO NA PARALELIZAÇÃO DE ALGORITMOS DE MULTIPLICAÇÃO DE MATRIZES

Alysson Eduardo ESTEVAM[1]; Luiz Felipe de Aragão PASSOS[2]; Hugo RESENDE[3]

RESUMO

A multiplicação de matrizes pode ser utilizada como uma tarefa intermediária ou finalística na construção de soluções para problemas da computação. Entretanto, tal tarefa possui um alto custo computacional, o que pode levar à geração de limitações quando é necessário aplicá-la a grandes instâncias de dados. Atualmente, com o advento das arquiteturas *multicore*, algoritmos com custo computacional elevado são aplicáveis à quantidade de dados cada vez maiores, principalmente quando tais algoritmos são eficientemente paralelizados e executados em arquiteturas robustas. Neste sentido, este trabalho investiga a viabilidade de paralelização dos algoritmos de multiplicação de matrizes tradicional e de Strassen nas linguagens de programação C, Fortran e Java, de modo a apresentar qual ou quais dessas linguagens melhor lidam com a paralelização de algoritmos que multiplicam matrizes.

Palavras-chave: Linguagens de Programação, Multiplicação de Matrizes, Paralelismo, Desempenho.

1. INTRODUÇÃO

A multiplicação de matrizes pode ser considerada como uma técnica de estimada importância em vários contextos do cenário computacional. Por exemplo, em algoritmos de criptografia, tal tarefa é utilizada na criação de códigos criptografados (ZATTI, 2009). Embora seja amplamente utilizada, a multiplicação de matrizes demanda um grande esforço computacional, mesmo quando implementada por algoritmos eficientes e, para se multiplicar instâncias de matrizes com milhares de linhas e colunas, são necessários altos tempos de processamento.

Nas últimas décadas, o recurso de programação paralela tem surgido com o intuito de agilizar e/ou amenizar o tempo demandado em tarefas que requerem altos tempos de processamento, por exemplo, na multiplicação de matrizes. Algumas linguagens de programação mais conhecidas na computação, tais como C, Fortran e Java fornecem recursos para se programar de forma paralela (PACHECO, 2011).

Diante do exposto, ao revisar a literatura, identificou-se que não há um estudo específico que descreva qual linguagem de programação, dentre as três linguagens mencionadas, consegue lidar com maior eficiência na paralelização de algoritmos que multiplicam matrizes, principalmente

1 IFSULDEMINAS – Câmpus Passos – alysson.estevam@alunos.ifsuldeminas.edu.br
2 IFSULDEMINAS – Câmpus Passos – luiz.passos@alunos.ifsuldeminas.edu.br
3 IFSULDEMINAS – Câmpus Passos – hugo.resende@ifsuldeminas.edu.br

ISSN 2319-0124

quando executados em arquiteturas *multicore*. Após uma vasta pesquisa na literatura computacional, optou-se por paralelizar os algoritmos tradicional e de Strassen, principalmente pela viabilidade e facilidade de entendimento dos pseudocódigos de tais algoritmos (MARQUEZAN, 2002; COPPERSMITH, 1990; WILLIAMS, 2014).

Nesse sentido, este documento apresenta os principais resultados da aplicação dos algoritmos paralelos tradicional e de Strassen, utilizados para multiplicar matrizes com vários quantitativos de linhas e colunas. Espera-se que tais resultados possam servir como elementos norteadores na determinação de quais linguagens se utilizar para tarefas de alto desempenho similares à multiplicação de matrizes.

2. MATERIAIS E MÉTODOS

Inicialmente, foi realizada uma pesquisa sobre os algoritmos de multiplicação de matrizes mais conhecidos na literatura, de modo a identificar quais algoritmos seriam trabalhados. Após essa etapa, foram escolhidos o algoritmo de multiplicação de matrizes tradicional (algoritmo *ijk*), cujo custo computacional é $\Theta(N^3)$, por ser o algoritmo primitivo para a solução do problema, e o algoritmo de Strassen, com custo $O(N^{2.8074})$, caracterizado por seu bom desempenho se comparado ao algoritmo tradicional (CORMEN, 2002). Após a definição dos algoritmos, estes foram implementados nas linguagens procedurais C e Fortran e na linguagem orientada a objetos Java, tendo como base os pseudocódigos tradicionais encontrados na literatura.

Após a implementação das versões seriais, foi realizada uma pesquisa sobre a forma de paralelização de algoritmos nas linguagens propostas, com o objetivo principal de otimizar o desempenho dos algoritmos, buscando diminuir os seus tempos de execução. Tendo como base os algoritmos seriais implementados nas três linguagens objeto deste estudo, implementou-se uma versão paralela para os algoritmos nas linguagens C e Fortran, utilizando um dos recursos mais conhecidos no cenário da computação, que são as diretivas de compilação *OpenMP*. Para a linguagem Java fez-se o uso da classe *Thread*.

Com os algoritmos paralelizados, foram geradas matrizes densas artificiais de ordem (quantitativos de linhas e colunas) 1024, 2048 e 4096, a partir de um algoritmo que gera essas matrizes contendo elementos aleatórios dentro de uma faixa numérica, e utilizando-as como entrada, foram executados vários experimentos, tanto com os algoritmos seriais quanto os paralelos. Maiores detalhes serão apresentados na seção seguinte.

ISSN 2319-0124

3. RESULTADOS E DISCUSSÕES

Os experimentos foram executados em uma máquina com sistema operacional Linux Ubuntu, versão 16.04, processador Intel Xenon E5-2630 com frequência de 2,20 Ghz e 64 Gb de memória principal. Para possibilitar uma melhor análise dos resultados, os algoritmos paralelos foram executados com 1, 2, 4, 8 e 16 *threads*, de forma a avaliar o desempenho de cada um deles com maiores quantidades de núcleos do processador. As Tabelas 1 e 2 a seguir, apresentam os resultados obtidos (em segundos) pelos algoritmos tradicional e de Strassen na multiplicação de matrizes artificiais de diferentes ordens.

Tabela 1. Tempos de execução do algoritmo tradicional (em segundos)

			Quantidade de Threads				
Linguagem	**Ordem**	**Serial**	**1**	**2**	**4**	**8**	**16**
Fortran	1024	14	17	9	4	2	1
C		8	11	6	3	2	1
Java		2	3	1	0	0	0
Fortran	2048	98	106	64	35	18	10
C		96	101	51	27	15	9
Java		44	51	25	12	7	3
Fortran	4096	3396	3467	2235	1100	557	317
C		1189	1216	734	397	211	132
Java		340	367	183	95	49	26

Tabela 2. Tempos de execução do algoritmo de Strassen (em segundos)

			Quantidade de Threads				
Linguagem	**Ordem**	**Serial**	**1**	**2**	**4**	**8**	**16**
Fortran	1024	6	9	6	2	1	2
C		3	4	1	0	0	0
Java		2	3	1	0	0	0
Fortran	2048	48	53	28	15	7	5
C		29	32	18	11	5	3
Java		16	19	10	7	3	1
Fortran	4096	346	358	181	92	48	25
C		218	224	114	58	31	16
Java		116	124	64	30	17	9

ISSN 2319-0124

Por meio dos tempos de processamento obtidos percebe-se que a linguagem Java se mostrou superior as outras linguagens em ambos os algoritmos e ordens de matrizes. Percebe-se também que o algoritmo tradicional implementado na linguagem Fortran requer uma quantidade de tempo maior para efetuar a multiplicação de matrizes, principalmente as de maior ordem.

4. CONCLUSÕES

Neste trabalho foram analisadas versões paralelas e seriais de algoritmos de multiplicação de matrizes nas três linguagens de programação objetos de estudo. Dessa forma, foi possível recolher dados e informações relevantes, para identificar qual dessas linguagens é a mais propensa a ser utilizada na resolução de problemas que envolvam a multiplicação de matrizes, ou tarefas similares.

Após uma análise acerca dos resultados obtidos, chegou-se à conclusão que a linguagem Java é a mais indicada para a resolução do problema proposto, em relação as outras linguagens trabalhadas, uma vez que apresentou o maior desempenho na paralelização dos algoritmos implementados.

Como trabalhos futuros, pode-se analisar linguagens diferentes das investigadas neste trabalho, assim como outros algoritmos de alto custo que resolvem vários tipos de problemas, por exemplo, algoritmos de criptografia de dados.

AGRADECIMENTOS

Agradecemos ao IFSULDEMINAS pelo apoio financeiro para o desenvolvimento desta pesquisa.

REFERÊNCIAS

COPPERSMITH, D, WINOGRAD S. **Matrix multiplication via arithmetic progressions**. Journal of symbolic computation 9.3: 251-280. 1990.

CORMEN, T.; LEISERSON, C. **Algoritmos**: teoria e prática. **Editora Campus**, 2002.

MARQUEZAN, C.; CONTESSA D.; ALVES, R., DIVÉRIO, T. **Análise de Complexidade e Desempenho de Algoritmos para Multiplicação de Matrizes**. Escola Regional de Alto Desempenho, 2002.

PACHECO, P. **An introduction to parallel programming**. Elsevier, 2011.

ZATTI, S. B.; BELTRAME, A. M. **A presença da álgebra linear e da teoria dos números na criptografia**. 2009.