UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO - UFRPE
UNIDADE ACADÊMICA DE SERRA TALHADA – UAST

Projeto de Sistema Distribuídos – SI – 2021.2
Prof. Celso Brennand

Implementar algoritmo de multiplicação de matrizes distribuída com o objetivo de acelerar o tempo de resposta.

A multiplicação é feita como mostrada abaixo:

M1

A	B
C	D

M2

E	F
G	H

=

M1 x M2

AxE + BxG	AxF + BxH
CxE + DxG	CxF + DxH

Como a multiplicação de matrizes pode ser feita dividindo em partes menores, essas partes podem ser computadas em threads/processos/máquinas diferentes.

A matriz seria dividida pelo número de processadores. Por exemplo, suponhamos que temos 4 máquinas cada uma com um core. Vamos então dividir a multiplicação em 4 partes onde cada parte vai ser multiplicada em um computador diferente. Para isto será necessário dividir as linha e colunas e passar para a função que vai calcular a multiplicação para que ela calcule apenas a parte incumbida a ela. Por exemplo o computador C1 vai calcular a linha 1 da M1 x coluna 1 da M2. O computador 2 vai multiplicar a linha 1 da M1 vezes a coluna 2 da M2, o computador 3: linha 2 de M1 x coluna 1 de M2 e computador 4 vai multiplicar linha 2 do M1 x coluna 2 de M2. Fazer o programa tendo como entrada duas matizes de tamanho genérico (verificar a compatibilidade na multiplicação da matriz, Se A é uma matriz m×n e B é uma matriz n×p, então seu produto é uma matriz m×p. Note que o número de linha da matriz A tem que ser igual ao número de colunas da matriz B). [Produto de matrizes – Wikipédia, a enciclopédia livre (wikipedia.org)](https://pt.wikipedia.org/wiki/Produto_de_matrizes)

M1 x M2

AxE + BxG	AxF + BxH
CxE + DxG	CxF + DxH

Será necessária uma função para fazer a divisão e a junção da matriz.

O programa usa como entrada dois arquivos de texto que serão passados como parâmetros na chamada do programa.

O arquivo tem a seguinte estrutura.

- Primeira linha contém o tamanho da matriz linha x coluna
- Da segunda linha em diante os valores da matriz com os números separados por espaço.

Matriz M1 Matriz M2

2 2 A B C D

2 2 E F G H

Exemplo da entrada do programa com a matriz a ser calculada.

3 3

2 2 2

2 2 2

2 2 2

Exemplo 2:

10 10
41.321922 88.20721 99.041435 47.00141 25.225544 6.949991 70.56636 30.643867 7.3438406 95.62614
53.60645 15.720427 41.019314 14.689255 52.315266 32.15262 89.67744 1.5518546 67.7406 75.78506
44.466274 89.20498 11.219853 34.134567 12.641155 99.26454 62.556236 81.21789 31.135803 26.41433
9.508938 16.483337 16.268093 6.660181 95.5636 82.230225 64.79549 52.47012 73.80851 4.0704308
61.159412 41.896397 63.78109 15.14408 84.90016 9.614241 22.184616 59.935528 16.606634 19.237572
18.285793 23.445612 2.6735365 7.3674736 90.53103 99.77698 2.4202764 9.987604 6.4673243 37.304646
26.190853 85.750465 46.488266 69.77438 1.8753529 83.418106 40.376503 40.94559 28.405905 2.5630832
50.272022 39.638412 88.281456 28.244394 74.81458 99.93096 73.726166 87.70591 46.105545 63.271217
10.908145 56.471188 67.18758 3.7194073 48.52476 4.110098 45.391674 50.53427 67.343765 8.324469
82.65411 2.7860224 98.88826 17.048573 93.326965 75.358475 48.838806 9.505426 2.724123 84.14336

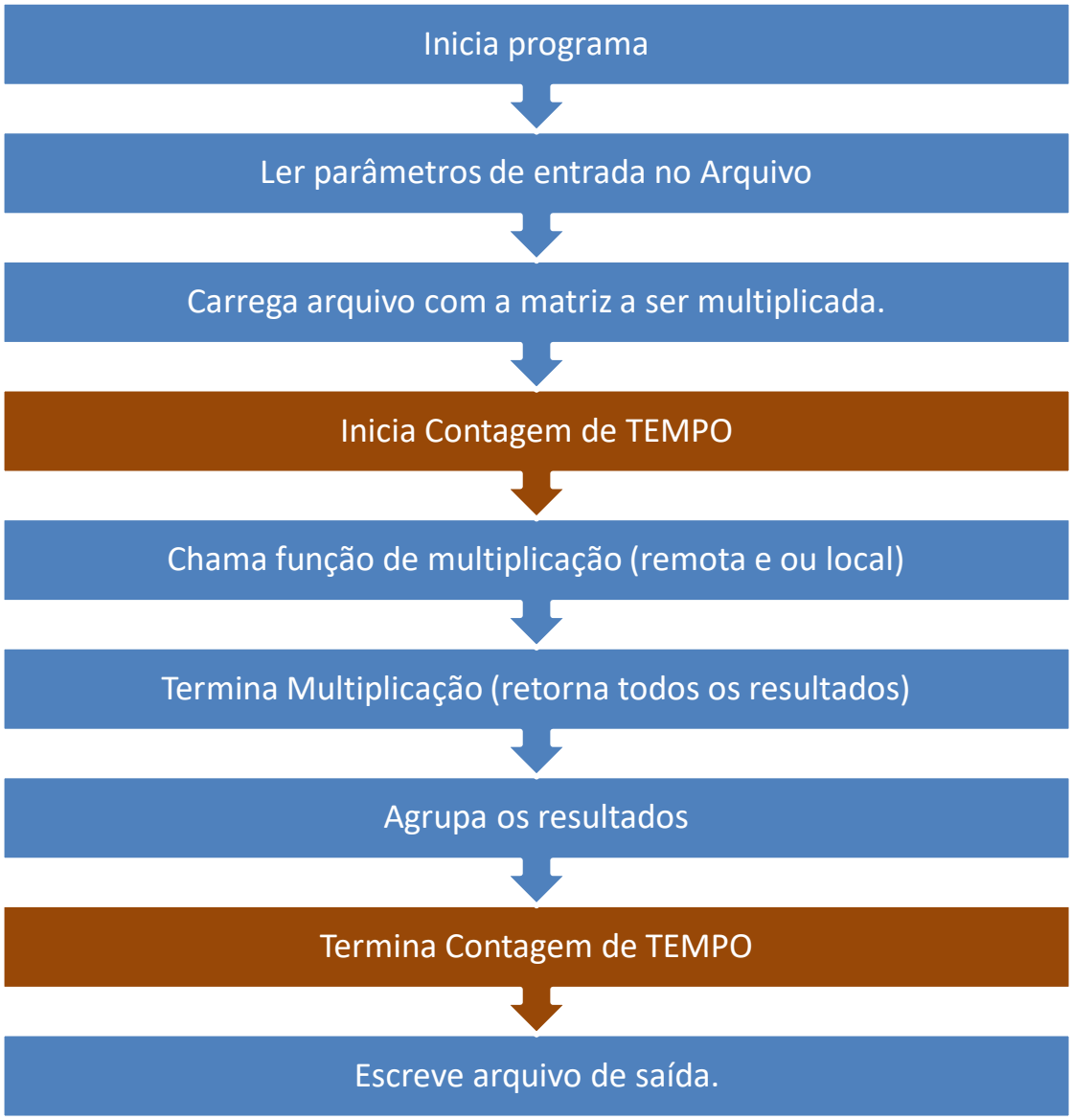
Tempo de excursão

Speedup pode ser definido como a relação entre o tempo gasto para executar uma tarefa com um único processador e o tempo gasto com N processadores, ou seja, Speedup é a Medida do ganho em tempo.

$$S = \frac{T(1)}{T(N)}$$

Onde 'S' é o speedup e 'T'(N) é o tempo gasto para 'N' processadores

Para medir o ganho será necessário medir o tempo de execução do programa.
Para medir o tempo terá que ser colocado um contador de tempo quando a **multiplicação inicia** até quando a **multiplicação termina**. O tempo de carregamento da entrada e da escrita da saída no arquivo tem que estar fora da medição.



Variações da implantação

No projeto quero 5 variações desta implementação:

Variação	Descrição
P1	Programa local sem thread, logo sem divisão da matriz
P2	Programa local com Thread, número de threads deve ser igual ao número de cores da máquina
P3	Programa local com Thread, número de threads deve 2 vezes o número de cores da máquina.
P4	Programa local com Thread, número de threads deve metade do número de cores da máquina.
P5	Programa remoto usando RMI-RPC com thread em todas as máquinas. O número de thread por máquina será igual ao número de cores da máquina que está rodando. Para isso é necessário ter uma máquina central que irá se conectar a outras máquina conhecidas para enviar uma parte da matriz para que ela calcule e responda a máquina principal. A máquina principal tem que esperar todas as máquinas remotas calcularem sua parte da multiplicação e ao receber todas as respostas efetuar a junção.

Saída do programa

Como saída do programa quero um arquivo contendo os seguintes dados na seguinte estrutura:

Linha 1	Variação do programa (P1...P5)
Linha 2	Número de Cores
Linha 3	Número de computadores Remotos
Linha 4	Número de linhas da matriz
Linha 5	Número de colunas da matriz
Linha 6	Tempo de processamento
Linha 7	Linha em branco
Linha 8...n	Matriz de Resposta

Relatório

Como documentação entregar:

- Descrição do programa.
- Gráfico comparando o tempo das diferentes variações do programa.
- Gráfico com o Speedup.
- Descrição do gráfico comparativo explicando o porquê uma variação é mais rápida que outra.

- Conclusão.

Arquivos de entrada

Arquivos de teste e para gerar os resultados de desempenho estão anexos no google sala de aula. São eles:

Arquivos de Teste (como são dois arquivos de entrada para a multiplicação pode usar o mesmo, ex. 4_int.txt **X** 4_int.txt):


4_int.txt


10_int.txt


10_float.txt

Arquivos para gerar os resultados:


128.txt


512.txt


1024.txt


2048.txt