

Trabalho 3: entregar até 29/Jun/23 (grupos: 2 alunos)

Multiplicação de matrizes com paralelismo em MPI – motivação e idéias iniciais

Neste trabalho faremos uma implementação simples de um algoritmo de multiplicação paralela de matrizes em MPI.

A implementação DEVE seguir as idéias algoritmo descritas aqui.

Para simplificar, o paralelismo deve usar MPI apenas.

As operações coletivas como MPI_Bcast e MPI_Scatter (e/ou MPI_Gather) devem ser usadas para distribuir as matrizes entre os nodos de processamento, e para obter, no nodo 0, a matriz resultante da multiplicação de duas outras.

Dadas duas matrizes quadradas A e B,
de ordem $[N_{la} \times M]$ e $[M \times N_{cb}]$ respectivamente.
Queremos obter a matriz C de ordem $[N_{la} \times N_{cb}]$,
resultante da multiplicação de $A * B$.

As matrizes A, B e C devem ser de números DOUBLE.

Para simplificar, e evitar leitura de disco, faremos
A inicialização das matrizes A e B no nodo 0.

A partir de parametros na da linha de comando o programa deve rodar os processos MPI que farão a execução do multiplicação paralela.

Se for informada a opção -v na linha de comando, o programa deve rodar uma versão sequencial do algoritmo e comparar o resultado com a versao paralela (comparar SE multiplicou corretamente).

Vamos colocar o nome “mmul.c” no nosso programa.
E “mmul” será o programa executável.

(continua...)

Para rodar, fazer (MODO NORMAL, rodar e medir APENAS):

```
mpirun -np n Nla M Ncb
```

Isso roda o a versão paralela com n processos MPI, inicializando as duas matrizes quadradas A e B, de ordem $[Nla \times M]$ e $[M \times Ncb]$ Respectivamente e fazendo a multiplicação paralela.

Nessa opção, o programa deve imprimir:

- a) CADA processo MPI deve imprimir:
 - o seu rank e o nome do host que rodou.
- b) o tempo gasto na multiplicação paralela apenas (reportar em segundos)
- c) a VAZAO obtida no cálculo (em GFLOPS)
Note que GFLOPS = Giga Floating Point Operations per Second
(ou seja, o S de GIGAFLOPS significa “por segundo”)

OBS: nesse modo, se rodar com -np 1, o MPI deve rodar apenas no nodo 0, a versão SEQUENCIAL e reportar o tempo, e GFLOPS sequenciais.

Para rodar, fazer (MODO VERIFICADOR, rodar e verificar correção do resultado APENAS, sem medir tempo.)

```
mpirun -np n Nla M Ncb -v
```

Isso roda o a versão paralela com n processos MPI, inicializando as duas matrizes quadradas A e B, de ordem $[Nla \times M]$ e $[M \times Ncb]$ Respectivamente e fazendo a multiplicação paralela.

Ao final, no nodo 0, roda uma versão SEQUENCIAL do algoritmo e compara a correção do resultado paralelo em relação ao serial.

Nessa opção -v, o programa deve imprimir:

- a) Se o resultado paralelo foi considerado correto.
- b) alguma medida de erro (se houver)

(continua...)

Medições:

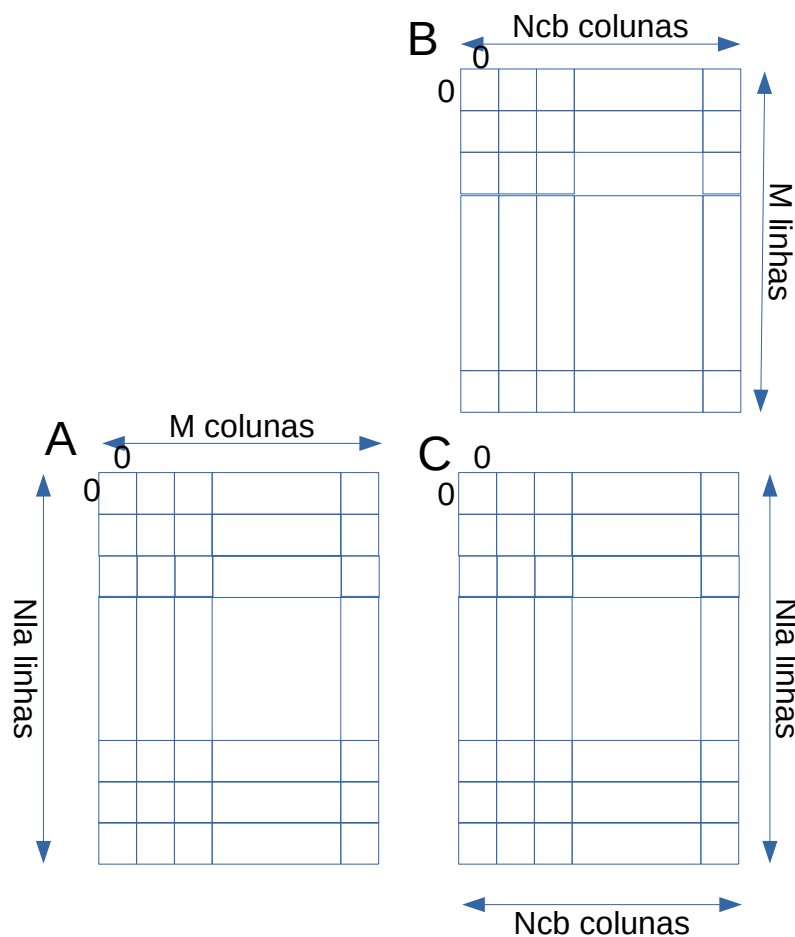
Vamos medir com processos EXCLUSIVOS, no cluster Xeon, na fila de processos (comando sbatch). Ou seja, CADA processo MPI deve rodar em um nodo separado e rodar sozinho no nodo (modo EXCLUSIVO)

Medir:

- Em principio, vamos medir para matrizes A de [2000x1000] e B de [1000x800] doubles. Se esse tamanho demorar muito ou for muito rápido, vamos mudar.
 - Vamos medir 10 x cada experimento e reportar a média das medidas
 - Vamos medir para 1 nodo (sequencial), 2 nodos, 3 nodos e 4 nodos.
 - Fazer a planilha como nos trabalhos anteriores, incluindo todas as infos, medidas, médias, graficos ADEQUADOS e tabelas de tempo e aceleração.
 - fazer e entregar relatório nos moldes dos trabalhos anteriores, com conclusões.
- Entrega:
Entrega será na UFPR virtual em método similar aos trabalhos anteriores.

Data da entrega: entregar até 29/Jun/23 (grupos: 2 alunos)

OBS: O prof. poderá discutir em sala como pode ser feita a distribuição das matrizes para os processos MPI com, via operações coletivas de MPI descritas no início dessa descrição.



n-
1

Envio das matrizes e processamento nos nodos

Exemplo para 3 processos MPI

- Processo 0 preenche matrizes A e B
- Processo 0 faz MPI_Scatter de 3 faixas horizontais (de linhas) de A, cada processo recebe uma faixa
- Processo 0 faz MPI_Broadcast de B para os outros nodos
- Cada processo MPI multiplica sua faixa horizontal de A com suas colunas necessárias de B. Assim cada processo obtém sua faixa horizontal da multiplicação em C
- Os processos fazem MPI_Gather da matriz C para destino no nodo 0
- Ao final o processo 0 tem as 3 matrizes completas, sendo $C = A * B$