

## CSC 64 Exercício 05

## CE 265 Exercício 05

### 1. Finalidade

Paralelizar MPI o cálculo de  $\pi$ .

### 2. Preparo no SDumont

Cada vez que entrar no SDumont, digite os comandos *module load python* e *module load openmpi/gnu*, nessa ordem. Esses comandos instalam o interpretador *python* e a biblioteca *openmpi*.

### 3. Cálculo de $\pi$

Um algoritmo interessante (embora pouco eficiente) para calcular  $\pi$  gera pontos aleatoriamente distribuídos no quadrado  $[-1:1] \times [-1:1]$  do plano cartesiano e verifica quantos pontos gerados são internos ao círculo inscrito nesse quadrado (círculo unitário). A razão entre as quantidades dos pontos no círculo e no quadrado é uma aproximação da razão entre as respectivas áreas. Logo,  $\pi$  pode ser aproximado por quatro vezes esta razão.

Sua tarefa é utilizar MPI para paralelizar versão sequencial desse cálculo.

### 4. Arquivo com os Programas deste Exercício

Encontre nesta atividade o arquivo *PiNoSDumont.tgz*. Leve-o para sua área no SDumont. Como sempre, desempacote o arquivo no diretório `$$SCRATCH`. A operação de desempacotar produz o diretório *Exercicio05* que contém quatro arquivos: os programas fonte *Pi.c* e *wall\_time.c*, *Makefile* e o script de submissão *Xmit.py*.

O programa fonte *Pi.c* calcula o valor de  $\pi$  gerando *points* pontos aleatórios no retângulo  $[-1:1, -1:1]$  do plano cartesiano e verificando quantos desses pontos encontram-se no círculo unitário. O retângulo é definido pelas variáveis *xStart*, *xRange*, *yStart*, *yRange*, ou seja, o retângulo é  $[xStart: xStart + xRange, yStart: yStart + yRange]$ . O tempo de execução do cálculo é medido da maneira usual.

O programa termina imprimindo o valor de  $\pi$  calculado, o erro relativo com relação ao valor “exato”, o número de aleatórios empregados, o número de tarefas MPI e o tempo de execução do cálculo.

O script *Xmit.py* requer, como único argumento, o número de tarefas MPI. Submissões corretas produzem o arquivo de saída *Out\_Pi\_<n>.txt*, onde *<n>* é o número de tarefas MPI.

### 5. Sua Tarefa

Após carregar os dois módulos, compile utilizando o *Makefile*. Observe que o programa fonte *Pi.c* já contém os comandos MPI de inicialização e término, além das invocações MPI que retornam o total de processos MPI e o identificador de cada processo MPI. Entretanto, o programa não está paralelizado MPI. Execuções com mais de um processo MPI replicam o trabalho sem reduzir o tempo de execução.

Para paralelizar o cálculo de  $\pi$  utilizando MPI, divida o intervalo do eixo *x*  $[-1:1]$  e o número de pontos aleatórios entre os processos MPI de tal forma que os processos formem uma partição do quadrado e que o número total de pontos gerados seja o mesmo para qualquer número de processos. Não particione o eixo *y*.

Cada processo MPI calcula o número de pontos dentro do círculo unitário na sua partição. Em seguida, o fluxo de execução diverge entre os processos. O processo zero totaliza

o número de pontos dentro do círculo unitário e calcula  $\pi$ , enquanto os demais processos fazem as operações necessárias para que o processo zero totalize o número de pontos dentro do círculo unitário. Meça o tempo de execução desse trecho do programa (cálculo do número de pontos no círculo unitário, troca de mensagens e cálculo de  $\pi$ ) em cada processo.

Em seguida, apenas o processo zero executa o *printf* original, imprimindo os resultados. Certifique-se que o cálculo está correto.

Execute o programa paralelizado utilizando *Xmit.py* para  $n=1,2,4,8,12,24,48,72,96$  ( $n$  é o número de processos MPI). Guarde os arquivos resultantes.

Monte arquivo .tgz com o programa paralelizado e entregue em conjunto com seu relatório.

Confeccione relatório em pdf contendo:

1. Breve descrição do paralelismo MPI implementado.
2. O arquivo *Pi.c* paralelizado (cut-and-paste);
3. Tabela com os tempos de execução e speed-up do cálculo de  $\pi$  em função do número de processos MPI para os valores de  $n$  definidos acima. Use os tempos de execução reportados na linha impressa pelo programa.
4. Avalie os speed-up reportados na tabela do item anterior. O que te chama a atenção?
5. Observe a variação do valor de  $\pi$  calculado com  $n$ . Porque isso ocorre?

Retorne o relatório e o arquivo .tgz até a meia noite imediatamente anterior à próxima aula.