MPI Problems

Karim Hasnaoui

Institut du Développement et des Ressources en Informatique Scientifique, UAR851 - Orsay
Maison de la Simulation UAR3441 - Saclay

karim.hasnaoui@idris.fr

List of MPI problems

- The ring algorithm
- π evaluation with the Wallis formula
- π evaluation with a Monte Carlo method
- Ping-pong & Communication Modes
- Matrix Addition/Substraction

The ring algorithm

- Let's assume $n > 1$ process
- The process 0 sends a value initialized to 1000 to the process 1
- When a process k $<$ n - 1 receives an integer, it tramists it incremented by 1 to the process k + 1
- When the n - 1 receives an integer, it tramists it incremented by 1 to the process 0
- The process stops when a final value is reached
- Write a program performing this ring communication

π evaluation with the Wallis formula

- π can be evaluated with the Wallis formula such as :

$$\pi = 2 \prod_{n=1}^{+\infty} \frac{4n^2}{4n^2 - 1} \tag{1}$$

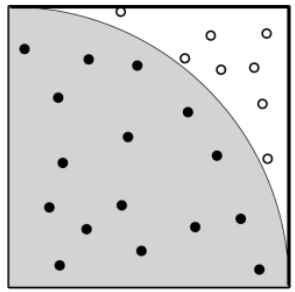
- By using MPI, write a parallel program that evaluates π trough the Wallis formula

- Tips : Each MPI process does a part of the matrix product

- Tips : Final result is obtained by the master process by using a reduction

π evaluation with a Monte Carlo method

π can be calculated as an area, by a stochastic approximation such as $\pi/4$ is the area of a quarter circle of radius 1



- N points can be randomly drawn in the square $[0,1]^2$
- Let N_r the number of points (x,y) randomly drawn such $x^2 + y^2 \leq 1$
- π can be appriximated such as $\pi \simeq 4\frac{N_r}{N}$ for large N
- Write the MPI code calculating the approximation of π :
  - The process 0 requests an unsigned long integer **n** from the keyboard and broadcasts it to other processes
  - Each process initializes a random generator **srand** differently
  - Each process performs **n** random draws and calculates its approximation of π
  - Local calculations of π are averaged to get an approximation, and disseminated to all processes

Ping-pong & Communication Modes

- Take the ping_pong_3.c code developped during the MPI Hands On - Exercise 2
- Adapte the code in order to create **synchronous**, **buffered** and **non blocking** communications
- With the Ruche supercomputer, try to do the ping pong within the processes in the same node and with two different nodes
- In order to get proper measurement for the bandwidths, the nodes must be allocated exclusively (use the directive $\#SBATCH --$exclusive)
- What can you conclude about the different modes of communication ?

Matrix Addition/Substraction

- Write a parallel code that does the matrix addition/substraction
- The matrixes are square and their sizes are a multiple of the number of processes
- The matrixes A and B are defined on process 0
- The process 0 can send horizontal or vertical slices of the matrixes A and B
- At the end, process 0 gathers and verifies the results