TP - Performance optimisation

ING2-GSI-MI – Architecture et programmation parallèle

Academic year 2023–2024

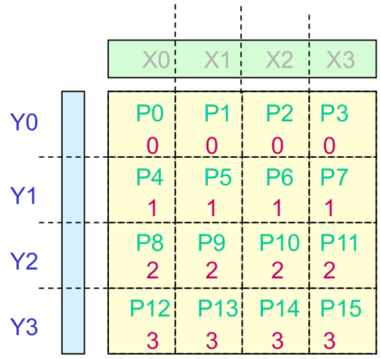
False sharing

1 Write an OpenMP program with nth threads and three arrays of `double`: `sum_local[nth]`, `x[N]` and `y[N]`, where $N = 1000$. Using a parallel loop, each thread will calculate a chunk of the product $x[i] * y[i]$ and will store the sum of the local products on its corresponding cell of `sum_local`. Measure the execution time for 1, 2, 4, 8 and 64 threads. Is the false sharing effect noticeable? Try to palliate this effect using different scheduling policies first, and using local variables later. □

Cannon's algorithm

2 Implement a matrix product of two square matrices in MPI using Cannon's algorithm. As a hint, you can follow these guidelines:

- Since Cannon's algorithm shifts data along rows and columns of processors, you could create a communicator for each row and column.
- For example, by row: $colour = myRank \, \% \sqrt{}(p)$, and then use MPI_Comm_split().
- Then, processes on each communicator will perform a circular shift on its own communicator.



□