

Parallel and Distributed Algorithms and Programs

TP n°4 - Matrix multiplication 2

Hadrien Croubois
hadrien.croubois@ens-lyon.fr

Aurélien Cavelan
aurelien.cavelan@ens-lyon.fr

6/11/2015

All documents are available on my website : <http://hadriencroubois.com/#Teaching>

Part 1

Pipelining

We have already seen how to multiply two matrices using multiple processes with the dual diffusion algorithm. However, because it uses broadcasts, this algorithm is very communication intensive.

Another algorithm for matrix multiplication is Cannon's algorithm, which uses pipelined communication to distribute the load more evenly.

Our objective in this session is to implement this algorithm and compare it to our previous implementation.

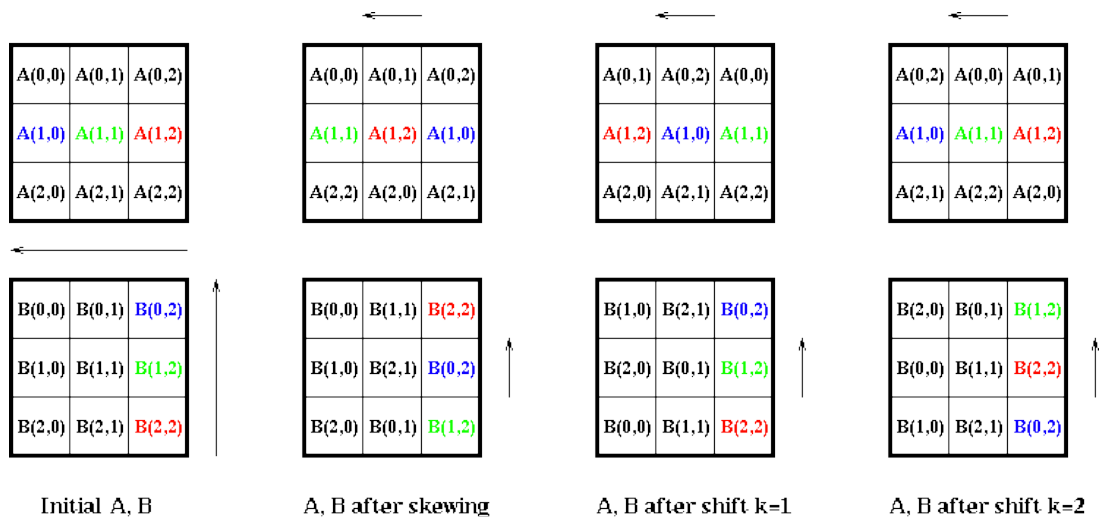


FIGURE 1 – Cannon's algorithm on a 3×3 grid of processors

Question

- How are communications orchestrated in Cannon's algorithm?
- Implement a skewing operation using MPI.
- Implement a shift operation using MPI.
- Using the last two questions, implement Cannon's algorithm using MPI.
- Why would Cannon's algorithm be faster than the diffusion method? What are the advantages of pipelining? How is the network topology involved?
- Suppose you can run your implementation on any platform of your choice. In order to measure the efficiency of your implementation of Cannon's algorithm against your previous implementation of the dual diffusion algorithm, what characteristics of the platform would you like to vary and what experiments would you like to run?
- Same question with both sorting algorithms from TP n°2.