Topology-Aware Resource Selection

Joint work with: Emmanuel Jeannot, Guillaume Mercier

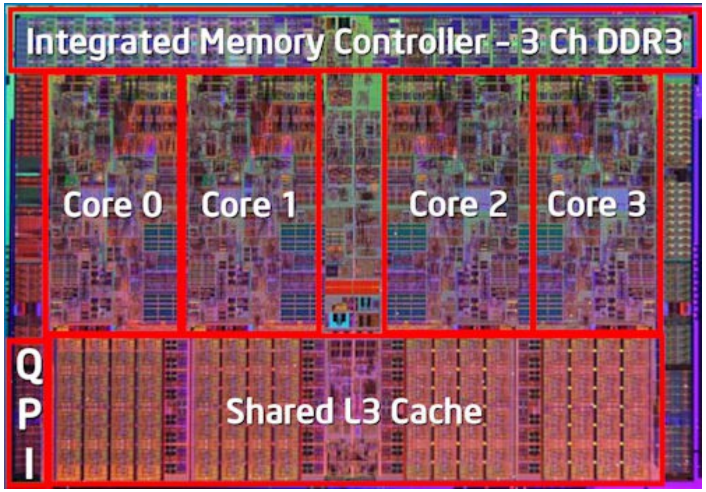
Adèle Villiermet
Runtime Team
Inria Bordeaux Sud-Ouest

September 24, 2014

1 Context

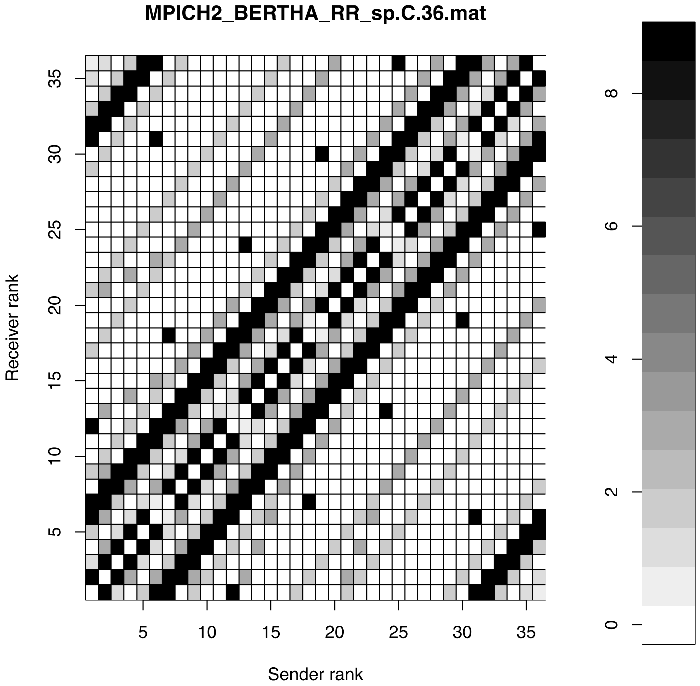
The topology is not flat

Due to multicore processors current and future parallel machines are hierarchical



Not all the processes exchange the same amount of data

The speed of the communications, and hence performance of the application depends on the way processes are mapped to resources.



Process Placement Problem

Given :

- Parallel machine **topology**
- Process **affinity** (communication pattern)

Map processes to resources (cores) to reduce communication cost: a nice algorithmic problem:

- Graph partitionning (Scotch, Metis)
- Application tuning [Aktulga et al. Euro-Par 12]
- Topology-to-pattern matching (TreeMatch)

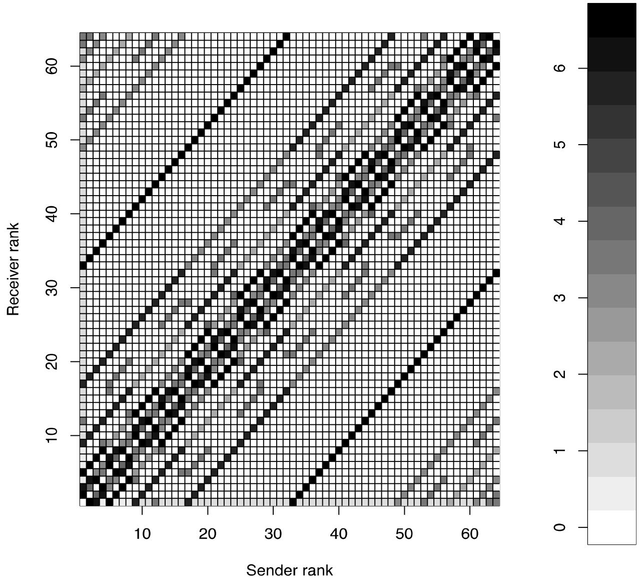
2

TreeMatch, a Process Placement Solution

Building the communication pattern

We need affinity between processing elements: communication pattern

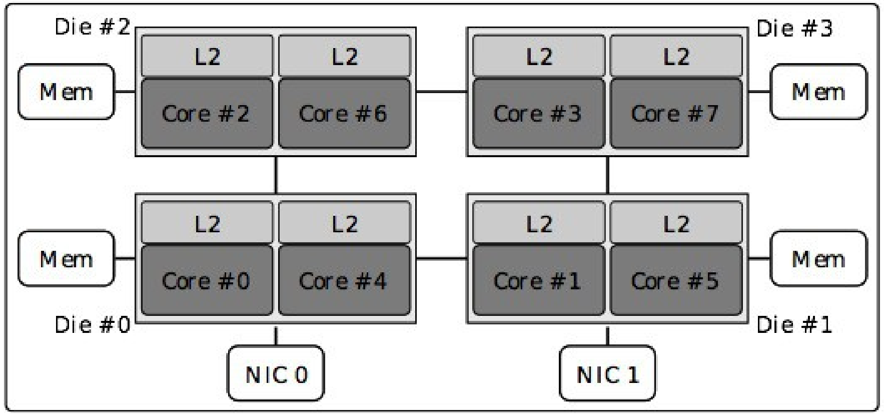
- Statically (thanks to compiler)
- Dynamic Monitoring
- Blank execution and tracing
- After data partitioning (e.g. Scotch)

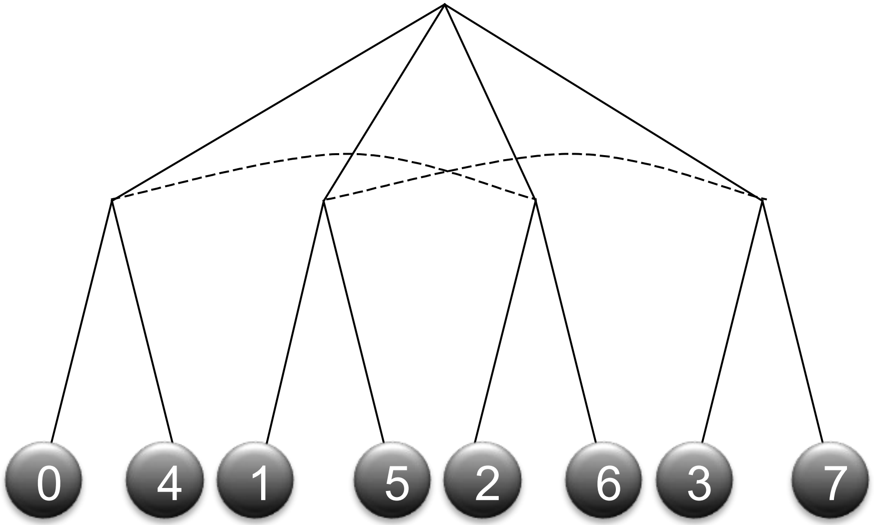


Obtaining the topology

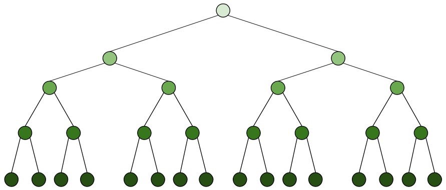
Abstract the topology with a tree

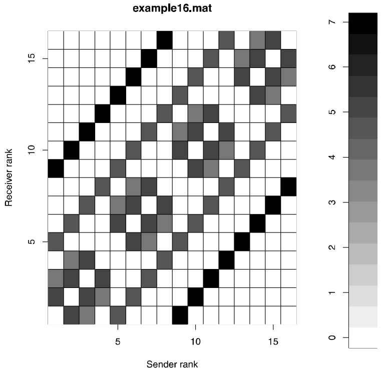
Assume communication always cost more when you need to reach higher levels





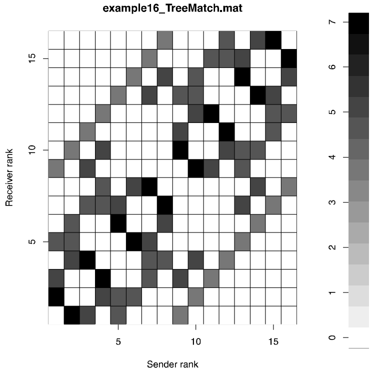
Putting everything together: Process Placement with TreeMatch

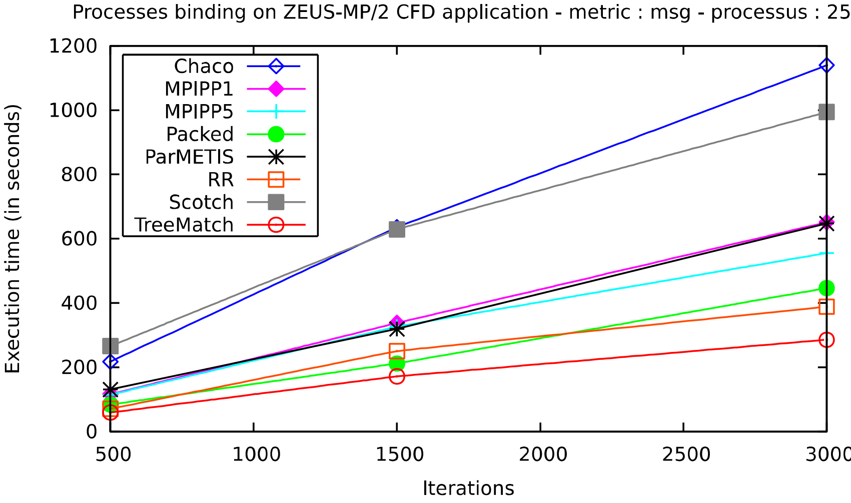




$\sigma = (0,2,8,10,4,$
$6,12,14,1,3,9,$
$11,5,7,13,15)$

$\Longrightarrow$

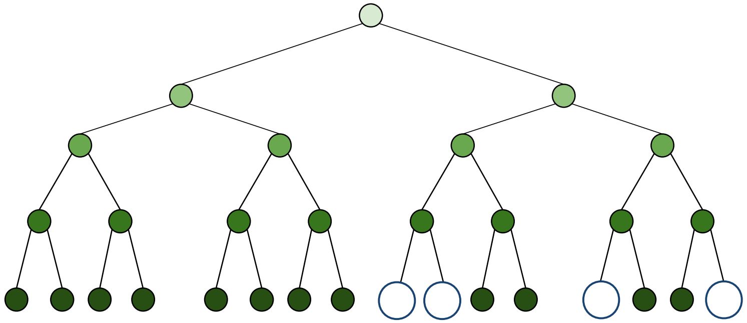


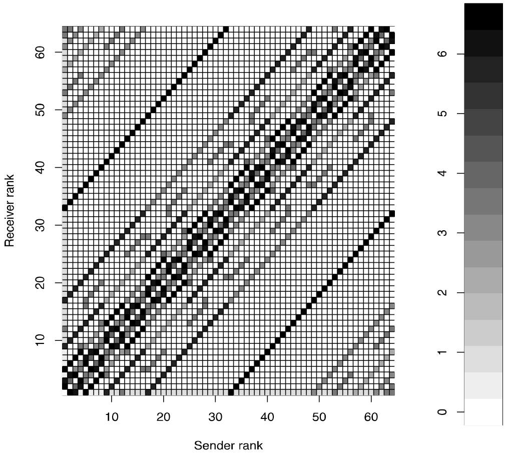


Communication matrix + Tree Topology = Process permutation

3
Resource selection

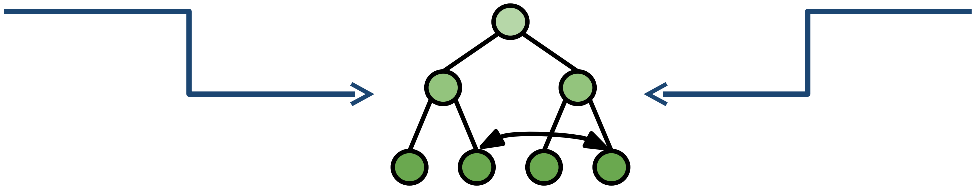
Selecting Resources



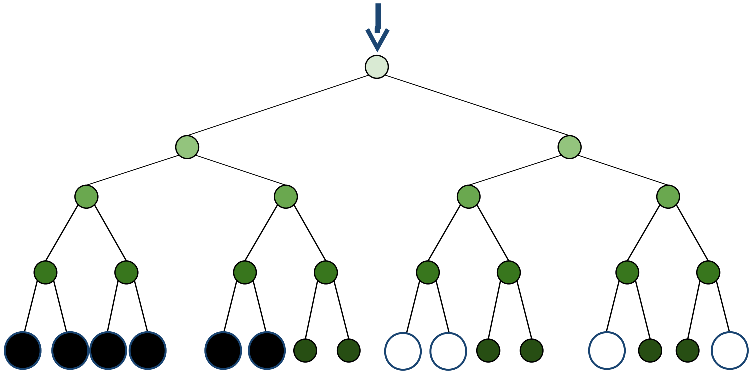


Model of the machine

Model of the application



TreeMatch Algorithm in the batch scheduler

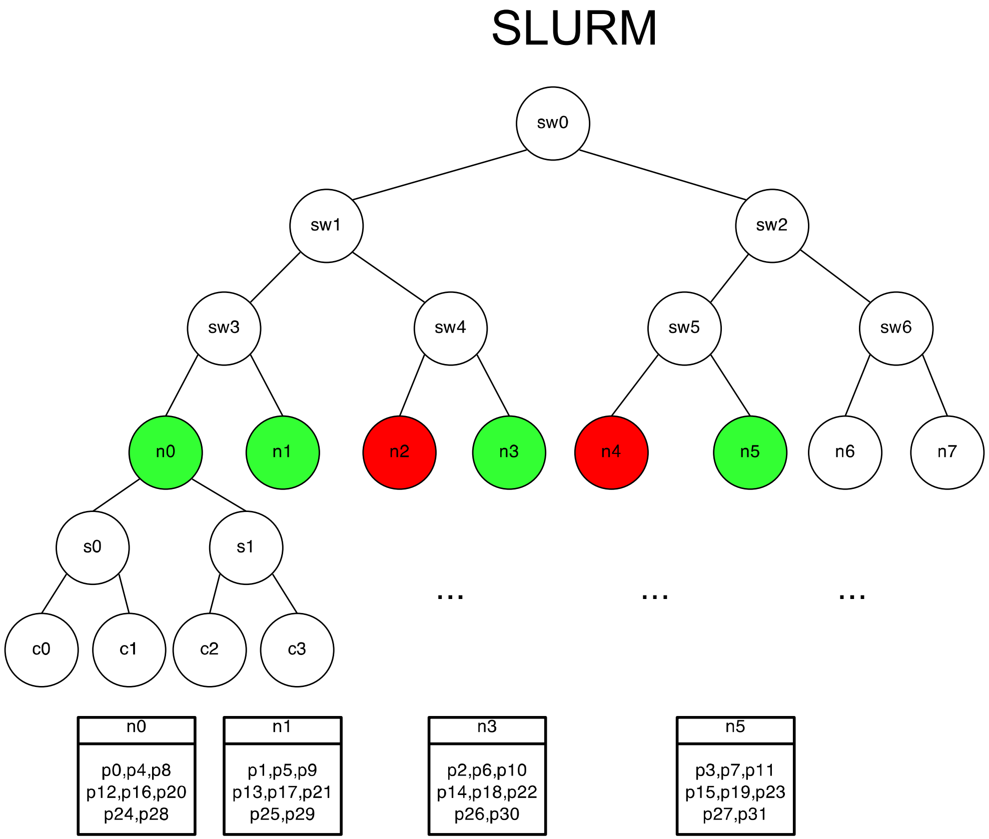


Implementation

- Within SLURM 2.6.5
- Only with select/cons_res plugin
- srun
- Binding possibility with cgroup
- Resource selection and process placement at the same time

Why topology-aware resource selection could work?

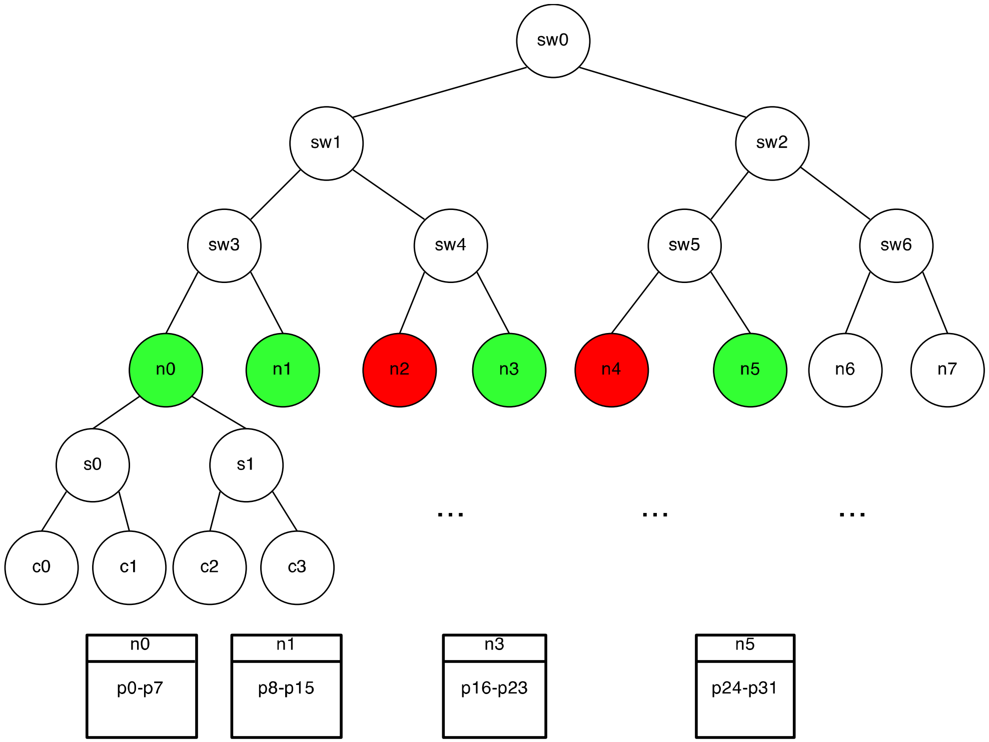
0-7	0	1000	0	20
8-15	1000	0	10	0
16-23	0	10	0	1000
24-31	20	0	1000	0



Why topology-aware resource selection could work?

0-7	0	1000	0	20
8-15	1000	0	10	0
16-23	0	10	0	1000
24-31	20	0	1000	0

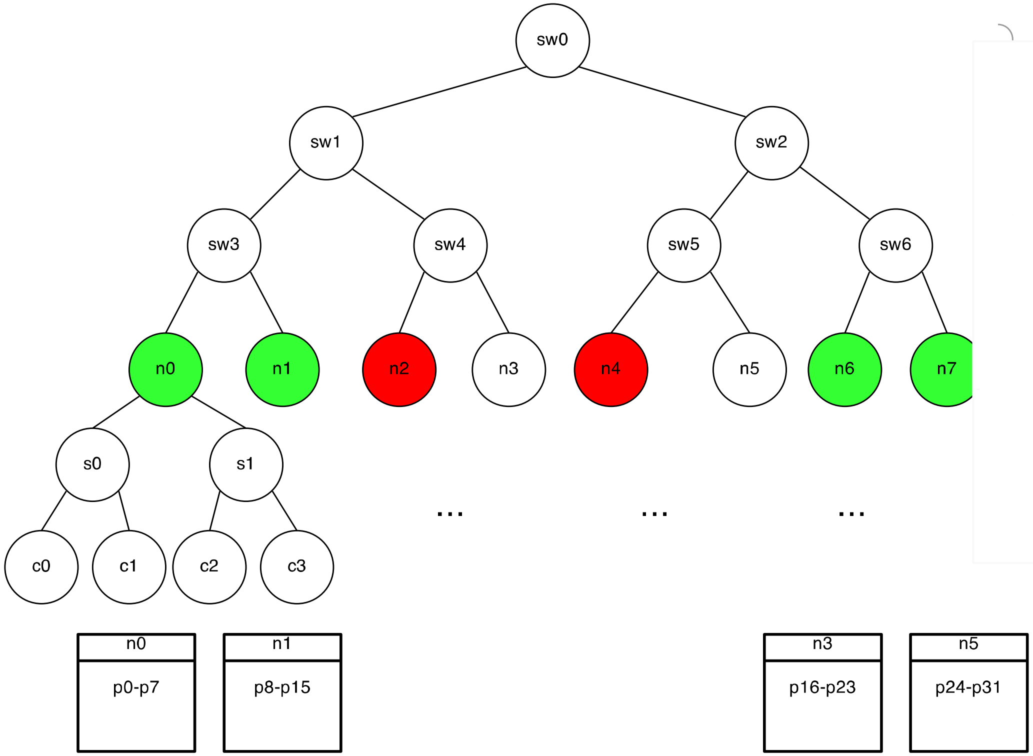
SLURM Then TreeMatch



Why topology-aware resource selection could work?

0-7	0	1000	0	20
8-15	1000	0	10	0
16-23	0	10	0	1000
24-31	20	0	1000	0

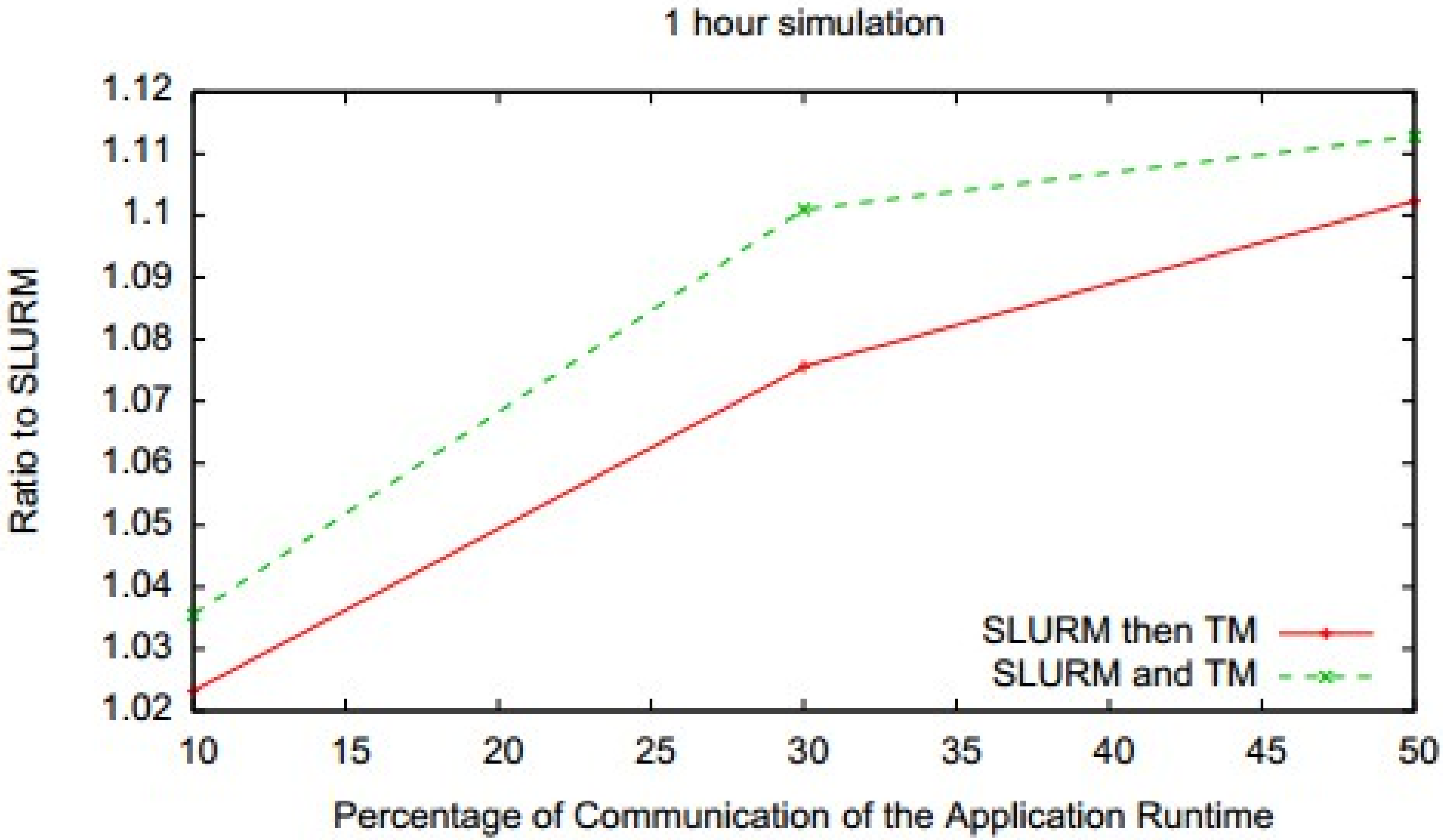
SLURM and TreeMatch



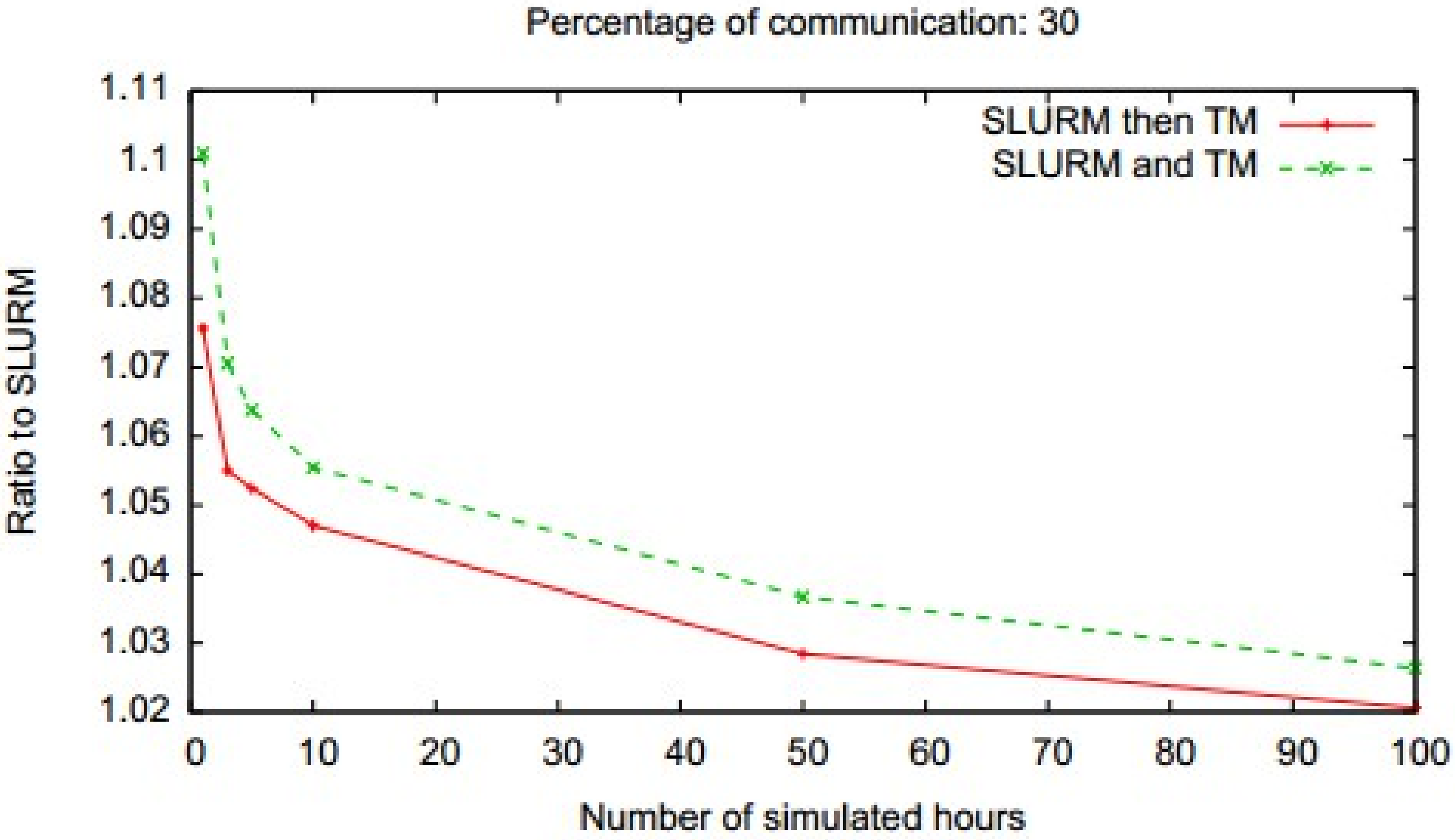
Early experiments

- Same protocol as SLURM/Bull team.
- Simulation using real traces of the Curie CEA machine: 80640 cores.
- Model of performance gain of TreeMatch depending on the amount of communication performed by application (10%, 30%, 50%).
- Randomly generated communication matrices.
- Same starting workflow:
  - 130 running jobs
  - 26 queued jobs
- Submitted jobs from 372 (1 hour) to 14171 (100 hours).
- Evaluation on the difference of the submitted jobs.

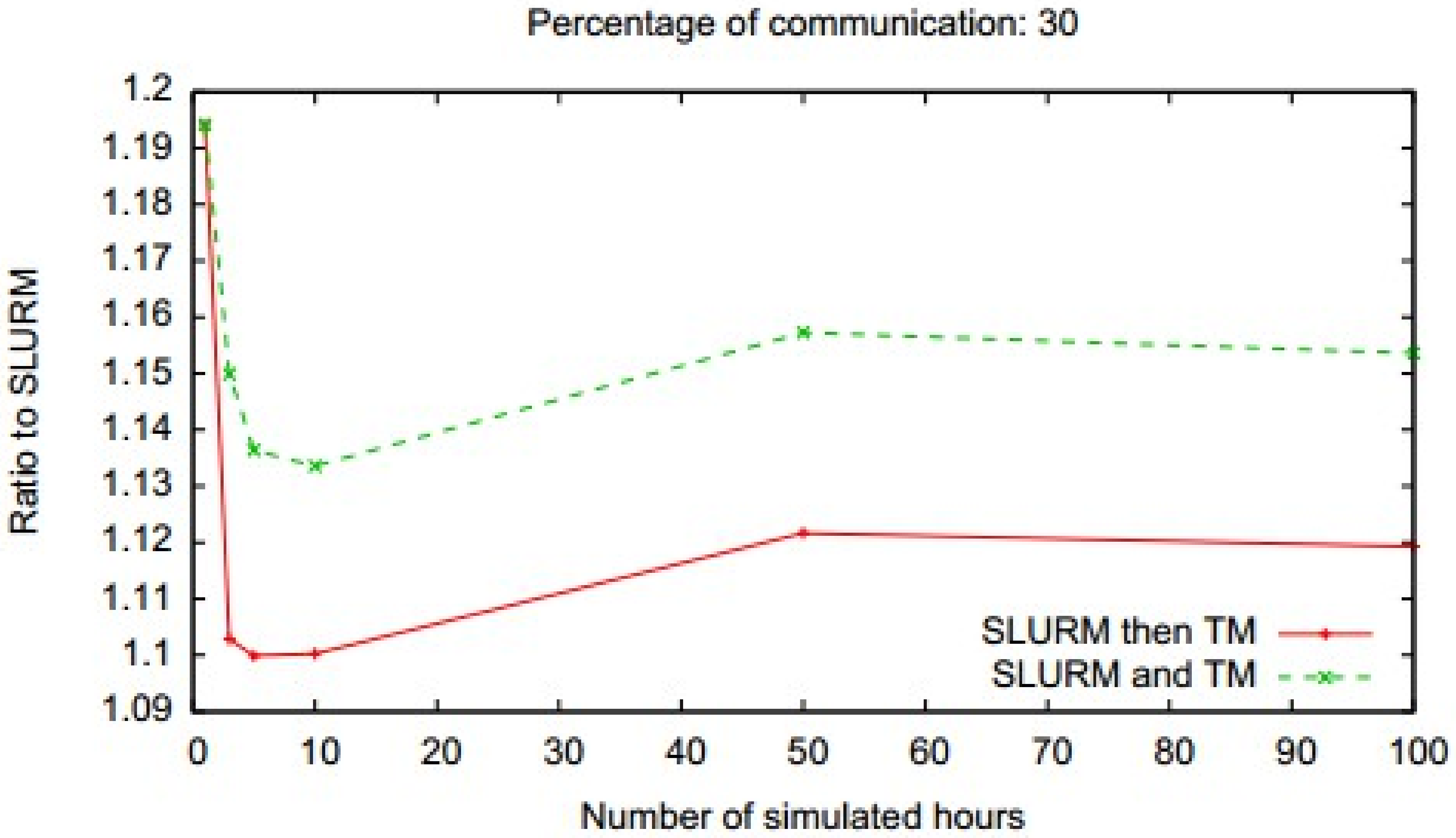
Simulation: makespan



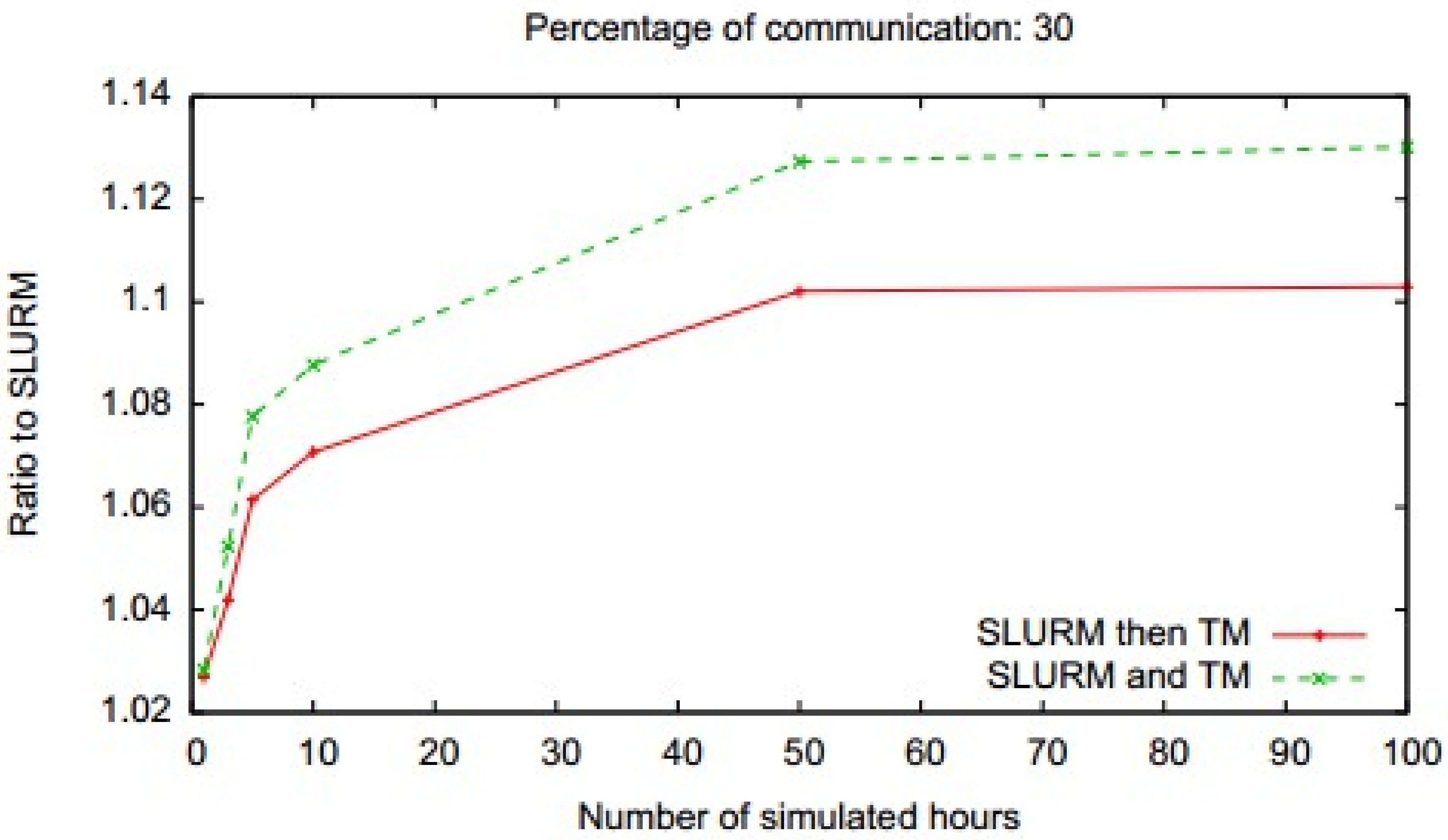
Simulation: makespan



Simulation: average stretch



Simulation: average flow



Conclusion

- Simulation results encouraging

- Start a PhD to continue

- Future works :

    Emulation

    Complete the implementation

- Improvement ideas

    Build a usual communication matrix list

    Improve algorithmic of resource selection

Thanks!

informatics mathematics

www.inria.fr