Topology-aware process placement on multicore architectures

Towards Exascale High Performance Computing

François Tessier
University of Bordeaux - LaBRI - Inria

Contact Information:
Runtime Team - Inria Bordeaux
200, Avenue de la Vieille Tour
33405 Talence

Phone: +33 5 24 57 41 52
Email: francois.tessier@inria.fr

Abstract

Current generations of NUMA node clusters feature multicore or manycore processors. Programming such architectures efficiently is a challenge because numerous hardware characteristics have to be taken into account, especially the memory hierarchy. One appealing idea to improve the performance of parallel applications is to decrease their communication costs by matching the communication pattern to the underlying hardware architecture. In this poster, we detail the algorithm and techniques proposed to achieve such a result: first, we gather both the communication pattern information and the hardware details. Then we compute a relevant reordering of the various process ranks of the application. Finally, those new ranks are used to reduce the communication costs of the application.

General Overview

- High performance applications
  - Molecular dynamics
  - Climate simulation
  - Plane wing design

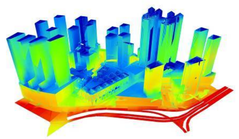
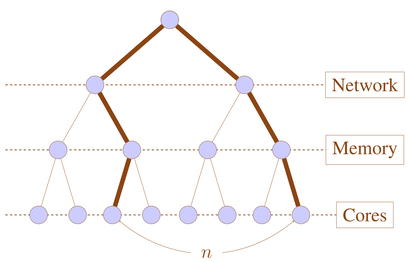


Figure 1: Acoustic simulation : the impact of urban noise

- Multi-node and multi-core architectures
  - Different levels of memory hierarchy
  - NUMA effects (Non Uniform Memory Access)
  - Data locality?



n : amount of communication

Process Placement

Why we should consider it ?

- Amount of data exchanged between processus not homogeneous (affinity)

Proc	0	1	2	3	4	5	6	7
0	0	**311811**	0	0	**157979**	0	0	0
1	**311811**	0	**311810**	0	0	**159986**	0	0
2	0	**311810**	0	**311811**	0	0	**153832**	0
3	0	0	**311811**	0	0	0	0	**151825**
4	**157979**	0	0	0	0	**311811**	0	0
5	0	**159986**	0	0	**311811**	0	**311810**	0
6	0	0	**153832**	0	0	**311810**	0	**311811**
7	0	0	0	**151825**	0	0	**311811**	0

Table 1: Communication matrix from a LU factorization carried out on 8 processes (metric : size)

- Communication speed : $cache > RAM > network$

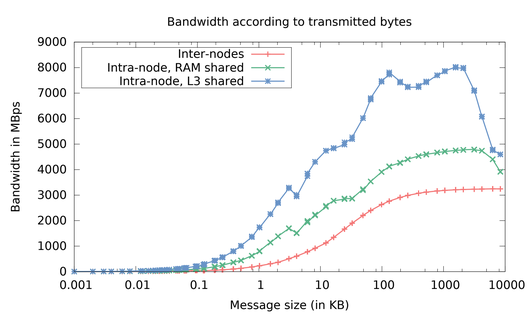


Figure 2: Bandwidth in MB/s of various memory levels according to the size of the sent data

- Performances improved by a relevant process placement [1]

Formalisation

Given :

- The application affinity pattern (e.g. communication)
- The underlying architecture

Map application processes to physical resources (cores) to reduce the communication cost.

Solution

Communication Pattern

- Communication matrix of size $p \times p$ with p the number of processes
- Metrics
  - *size* : amount of data exchanged (kB)
  - *msg* : number of messages exchanged
  - *avg* : average size of a message ($size/msg$)

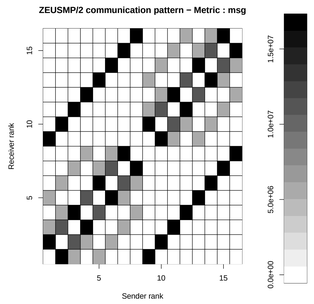


Figure 3: Communication matrix of the CFD application ZeusMP/2 carried out on 16 processes (metric : msg)

Hardware Topology

- hwloc : library mainly developed at Inria
- Gather the node topology of modern architectures

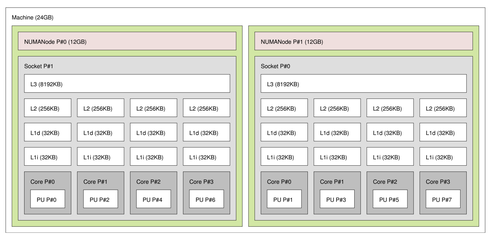


Figure 4: Hardware topology gathered by hwloc

TreeMatch

- Algorithm to compute process placement based on processes affinity and NUMA topology
  - p the number of processes
  - n the number of cores, $n \geq p$

σ_i the core on which we map process i,
$\sigma_i \in [1, n], i \in [1, p]$

Results

- CFD application ZeusMP/2
- Architecture
  - 8 nodes, 2 processors with 8 cores on each (Intel Xeon Nehalem X5550 (2,6 GHz))
  - InfiniBand network
  - Open MPI 1.5.4
- *msg* metric
- Compared to well-known methods

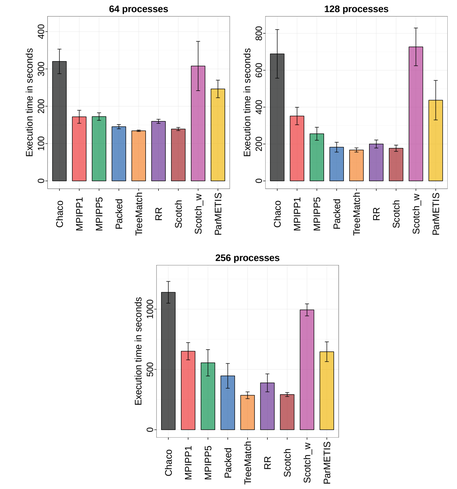


Figure 5: Impact of process placement on the execution time of the ZeusMP/2 CFD application

Conclusion

- Good results compared to the other strategies
- Improvement to consider the network ?
- Algorithm also used for dynamic process placement (e.g. load balancing)

References

[1] PRACE. The scientific case for high performance computing in Europe. report, 2012. P. 147, http://www.prace-ri.eu/IMG/pdf/prace_-_the_scientific_case_-_executive_s.pdf.

[2] Francois Tessier, Guillaume Mercier, and Emmanuel Jeannot. Process placement in multicore clusters:algorithmic issues and practical techniques. *IEEE Transactions on Parallel and Distributed Systems*, 25(4):993–1002, 2014.