Methodology for efficient Execution of SPMD applications on Multicore Clusters.

Ronal Muresano*
Computer Architecture and Operating System Department (CAOS)
Universitat Autònoma de Barcelona, Barcelona, SPAIN
PhD Thesis in High Performance Computing
Advisor: Emilio Luque**
rmuresano@caos.uab.es*, emilio.luque@uab.es**

Nowadays, the scientific applications are developed with more complexity and accuracy and these precisions need high computational resources to be executed. Also, the current trend in high performance computing (HPC) is to find clusters composed of Multicore nodes, and these nodes include heterogeneity levels which have to be handled carefully if we want to improve the performance metrics. The integration of these Multicore nodes in HPC (High Performance Computing) has allowed the inclusion of more parallelism within nodes, but this parallelism generates challenges that have to be managed considering some troubles present in these environments that affect the application efficiency and speedup. Aspects associated to number of cores, data locality, shared cache, communications link inside the node, etc are considered relevant when our goal is to improve the performance [1].

We consider Multicore clusters as heterogeneous due to the different communication links in which the communication processes can be performed [2]. These communication heterogeneities generate idle time due to the different communication links which have distinct speeds and bandwidths and they may cause degradations in the parallel application performance (Fig 1). For this reason, these communication heterogeneities and an inadequate workload distribution to each core have to be managed with the aim of improving the application performance.

Therefore, a huge challenge for parallel programmers is to execute applications using message passing libraries with high synchronicity through tile dependencies and communication volumes such as SPMD applications (Single Program Multiple Data) on Multicore clusters. This SPMD paradigm has been selected due to its behavior, which is to execute the same program in all processes but with a different set of tasks. The SPMD parallel applications that we have chosen have three main characteristics: static, where parallel application defines the communication process and is maintained during all the execution, local, where applications do not have collective communications and regular, because communications are repeated for several iterations (Fig 1).

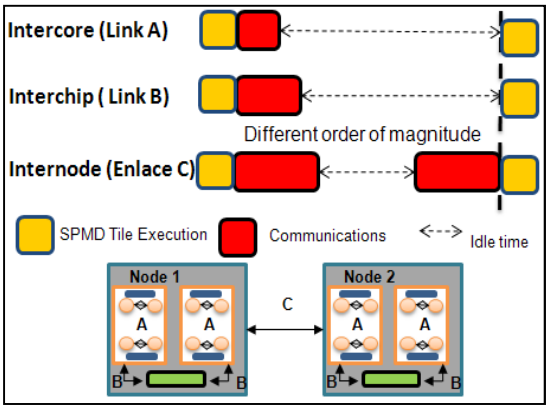

Fig 1.Multicore Cluster and SPMD tile communications

Additionally, in SPMD applications, the tile are executed with similar computational time and communication volume, but the communication process among neighbor are performed by different communication links depending on the location of SPMD processes. These communication links can vary the communication time in an order of magnitude according to the links and these variations are limiting factor to improve the performance [4].

From the problem defined above, our target is focused on describing a methodology which is based on achieving a suitable application execution with a maximum speedup achievable while the efficiency is maintained over the defined threshold [5]. To obtain this suitable execution, our methodology calculate the number of tiles that have to be assigned to each core and these tiles are divides in two groups internal and edge tiles.

This tiles division enables us to execute the overlapping process in which the internal computation and edge communication are overlapped. This methodology has been designed in four phases which allow us to handle the latencies and the communication imbalances created by the different communication links. These phases are included in the methodology as follow: the characterization (application and environment), a tile distribution (we determine the number of tiles and node necessary to execute efficiently and with the maximum speedup) a mapping strategy (distribution of SPMD tiles over cores, and the scheduling policy (define the execution order of SPMD tasks assigned) (Fig 2).

The characterization phase objective is to gather the inputs necessary of the SPMD applications and environment with the aim of calculating the tasks number which have to be assigned to each core. To calculate the number the tiles we have developed a tile distribution model which is calculated through an analytical model defined in [5]. This characterization is executed using a tool where different communications paths are tested and the computation and communication ratio is found [6].

The objective of next phase (tile distribution model) is to calculate number of tiles assigned to each core in order to achieve the maximum speedup and the desired efficiency is determined. This phase is calculated with the values obtained in

characterization phase. The results of the model enable us to determine the minimum values for Ki which are assigned to each core with the aim of maintaining the application efficiency[7].

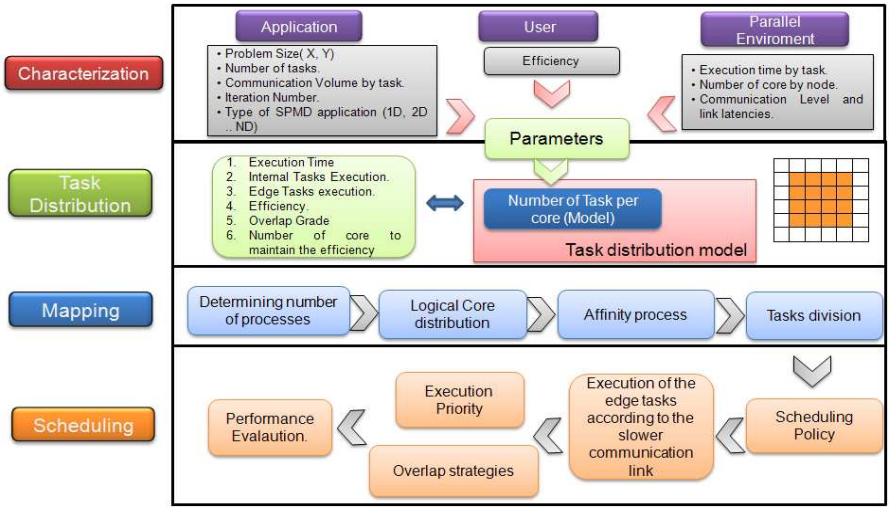

Fig 2. Methodology for Efficient Execution of SPMD applications on Multicore Cluster

The mapping phase target is to allocate the set of tiles calculated among the number of core necessary to maintain our goal. In this phase, we consider that MPI messages may be communicated through different communications paths when a Multicore cluster is used, and these communications generate issues due to the link latencies.

Finally, the scheduling phase objective, this phase coordinate the execution orders of each tile within the core. This process is realized in two parts, the first is to generate the priority assignment in which the highest priorities are established for tasks which have communications through slowest links and then is to apply the overlapping strategy in which the internal computation and the edge communication are overlapped.

This method has been tested with different benchmarks and applications over different multicore cluster composed between 8 to 4096 cores and the results shows an improvement around 40% in efficiency in application tested. An example of this improvement can be detailed in figure 3, where experimental evaluation makes clear that to achieve a better performance in SPMD applications; we have to manage the communication heterogeneities. Finally our methodology evaluates the environment through the characterization phase and we apply our model with real values of the multicore architecture.

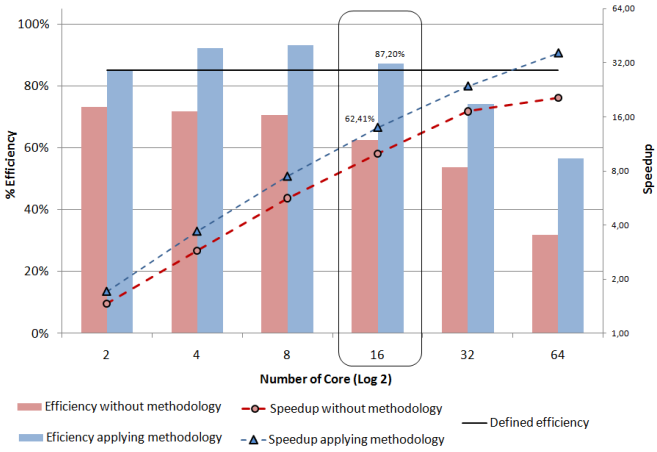

Fig 3. Efficiency and Speedup improvement in a heat transfer application

References:

[1] M. McCool. Scalable programming models for massively multicore processors. Proc of the IEEE, pages 816-831, (2008).
[2] G. Mercier and J. Clet-Ortega. Towards an efficient process placement policy for mpi applications in multicore environments. Euro PVM/MPI, Lecture Notes in Computer Science Springer, 5759/2009:104{115, (2009).
[3] L. M. Liebrock and S. P. Goudy, "Methodology for modeling spmd hybrid parallel computation," Concurr. Comput. : Pract. Exper., vol. 20, no. 8, pp. 903–940, 2008.
[4] R. Muresano, D. Rexachs, E. luque., How SPMD applications could be efficiently executed on multicore environment, IEEE International Conference on Cluster Computing (Cluster 2009).
[5] R. Muresano, D. Rexachs, and E. Luque. Methodology for efficient execution of spmd applications on multicore clusters. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID), IEEE Computer Society, pages 185-195, (2010).
[6] R. Muresano, D. Rexachs, and E. Luque. A tool for efficient execution of spmd applications on multicore clusters. International Conference on Computational Science, ICCS 2010, Procedia Computer Science, 1:2593{2602, (2010)
[7] R. Muresano, D. Rexachs, and E. Luque. Combining Scalability and Efficiency for SPMD Applications on Multicore Clusters, The 2011 International Conference on Parallel and Distributed Processing Techniques and Application PDPTA 2011, pages XX-XXX (2011)