Could be improved the efficiency of SPMD applications on heterogeneous environments? *

Ronal Muresano, Dolores Rexachs, and Emilio Luque

Universitat Autònoma de Barcelona
Computer Architecture and Operating System Department (CAOS)
Barcelona, SPAIN
{rmuresano}@caos.uab.es
{dolores.rexachs, emilio.luque}@uab.es

Abstract. The goal of this work is to execute SPMD applications efficiently on heterogeneous environments. Applications used to test our work are designed with message-passing interface to communicate and are developed to be executed in a single core cluster. However, we create a methodology to execute efficiently these SPMD applications over heterogeneous architectures. The SPMD applications are selected because they present high level of synchronism and communications; both elements could generate challenges when we want to obtain our objective, which is defined as to obtain an improvement in the execution time while maintaining the efficiency level over a threshold defined by programmer, taking into consideration the communications heterogeneities present in a multicore cluster. This objective is achieved using a mapping and scheduling strategies included in our methodology. Finally, the results obtained show an improvement around 40% in the best case of efficiency in SPMD applications tested, when our methodology is applied.

1 Introduction

Actually parallel applications are designed to execute complex computational problems and this execution could be finished in a long time. However the objective of high performance computing (HPC) is to execute application faster and efficiently, for this reason has been included new technologies to HPC which generate challenges, one of them has designed integrating set of cluster in an architecture called multicluster in this case the heterogeneity is presented in computations and communications. Other technology included is multicore nodes, in these nodes the heterogeneity is presented through communications paths for these reason both architecture are divided in level and will be explain bellow.

Our work is oriented on the inclusion of multicore technology in HPC which have allowed that applications could be executed in environment with more computational power, in order to obtain a fast execution. However, multicore clusters add high heterogeneities in communication paths and these heterogeneities generate communication troubles if are used message-passing applications which

* Supported by the MEC-Spain under contract TIN2007-64974

are developed to be executed within a single core cluster or homogeneous architecture, and these applications are executed on heterogeneous architecture as multicore cluster.

The work is focused on managing the communication heterogeneities to improve the performance metrics of parallel applications. We are mainly focused in two performance measures, efficiency and execution time. Both are affected by the number of PEs included in the parallel environment, and we have to administrate the workload in order to determine the adequate number of PEs and number of tasks needed for executing applications efficiently, taking into consideration the characteristics of the environment. We work with SPMD applications due these applications have a repetitive behavior.

Then, the objective is to do an improvement in parallel execution time when is used a heterogeneous environment while maintaining the efficiency over a threshold defined by programmer. The multicore cluster presents a set of heterogeneities due its different communications paths. Then our goal is to manage the speeds and bandwidths of the communications paths and to execute SPMD applications efficiently. This is realized through methodology divided by three phases characterization, mapping and scheduling.

In the mapping, the workload is distributed between PEs according the communications latency and the communications numbers presented by PEs and the communication paths where communications are made. Moreover, The scheduling allows us to develop an overlapping strategies with the aim to eliminate the communication inefficiencies present in multicore cluster.

The work is structured as follows: section 2 defines the problem formulation, followed by section 3 which describes related works. A description of the methodology is illustrated in section 4, likewise, section 5 describes the implementation. Section 6 describes the performance evaluation, and finally conclusions are in section 7.

2 Problem Formulation

Evolution in the parallel programming field has allowed that scientific applications can be programmed with more complexity and accuracy. These precisions require high computational power and clusters are generally limited by the number of nodes. Such limitations originate application scalability and performance issues; causing programmers to find suitable solutions that will improve application metrics. On the other hand, there are many computer centers within organizations and universities, which have computational power to execute parallel applications. However, these centers are usually underutilized by having potential resources in an idle state.

In order to benefit from such computational cluster capacities and to execute applications faster, some of these computational centers could be combined for creating a cluster architecture called multicluster (Fig.1). However, to use this kind of architecture, the programmer must consider computational environment heterogeneity. A multicluster environment has different computational and in-

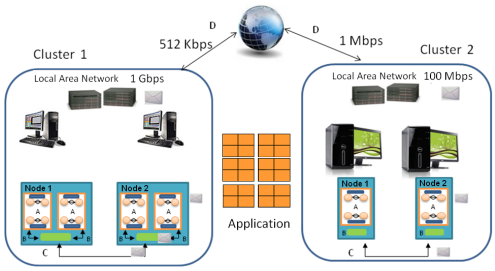


Fig. 1. Multicluster with multicore Node

terconnection network architectures, and both elements have to be managed if performance wants to be improved.

Executing parallel applications in this environment is a challenge due to the workload allocation for each Processing Element (PE), and the amount of tasks that will be assigned to each core, node, or cluster set. Another level of heterogeneity which can be included in a multicluster environment is the multicore node (Dual or Quad Core). A multicore node adds one more level of complexity to the multicluster, due to its different internal communication levels; which must be considered suitable strategies for workload mapping.

A Multicluster has different types of communications, some of them through network links such as: local area network(LAN) or wide area network (WAN); and others by internal processor buses like core-to-core communication through cache memory or communication between chip processors via main memory. All these communications have different speeds and bandwidths and they represent a challenge when the programmer wants to manage them for efficient application execution. The heterogeneity present in a multicluster with multicore nodes can generate that performance metrics such as efficiency, speedup and execution time worsen. An inadequate mapping strategy could decrease the effectiveness of the parallel application in Multicluster environments.

One more element to consider in this environment is that applications are designed under parallel programming paradigms such as, Master/Worker, SPMD, etc., each of them having a different communication patterns and execution models. The programmer has to evaluate if the execution of these parallel paradigms can improve application performance in a multicluster environment.

A methodology to migrate a master-worker parallel application from its original cluster to a multicluster environment was developed by Argollo [1] using a Master/Worker (M/W) paradigm. The proposed methodology targets are to decrease the execution time in the multicluster environment guaranteeing a preestablished threshold level of efficiency. Unlike the M/W paradigm, the behavior of SPMD is to execute the same program in all PE, but with a different set of data task. These applications are synchronized and they have a high communication volume in each iteration, making the execution of SPMD applications on a heterogeneous communication environment a challenge.

From the above problem, we plan to develop a methodology for SPMD applications in heterogeneous communication systems, considering an efficient execution. The objective is to execute in the shortest execution time possible,

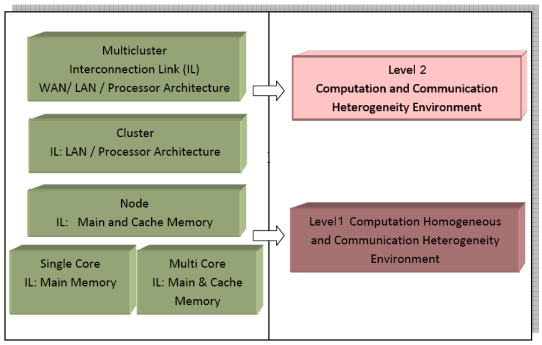


Fig. 2. Levels and Complexity Architecture

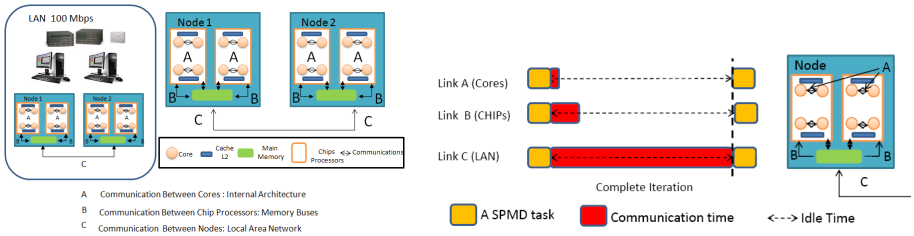
maintaining the efficiency level over a threshold value defined by the programmer. Moreover, our work considers a SPMD application which permits the set up of a number of tasks greater than the PE present in the environment. For this reason, tasks must be distributed between PEs, considering important key elements such as number of communications between PEs, communication volume and links involved in the communication process.

To solve the problem, our research has been divided in layers (Fig 2), which allow us to define a multicluster architecture, showing the heterogeneity between the different network links and buses, and also show the computational hierarchy between PEs. Additionally, the complexity levels in charge of identifying the computational and communication parameters present in the environment.

Dividing the problem through the different complexity levels presented before (Fig 2); allows to give solutions in levels, which are able to resolve the inherent complexity of heterogeneous environments present in a multicluster with multicore node. To analyze the influence of communications, the first step is to make a characterization of the environment including different bandwidths for each level and size of computation. Then, obtaining the computation and communication time by task in the PE, allow us to develop the mapping and scheduling strategies applied to obtain the best execution time, maintaining the efficiency.

This work presents a proposal for level 1 (Fig 2). Level 2 is currently under development. At level 1 the heterogeneities is presented in communications paths, and we try to manage them with the aim to maintain the efficiency parameters.

For this reason, we propose a methodology to evaluate the computational and communication parameters of SPMD applications. To develop a mapping strategy, the environment heterogeneity is evaluated, allowing us to assign a task set to each PE, maintaining the efficiency level. This Mapping strategy intends to manage the workload unbalance caused by different communication link latencies. Otherwise, workload unbalance would certainly decrease the application performance. Once the mapping is finished, scheduling strategies are considered. The scheduling is based on overlapping internal computation and edge communication. The overlapping process is made considering the architecture hierarchy and the communication latency.



(a) A multicore cluster environment. (b) Communications of SPMD tasks.

Fig. 3. multicore cluster and SPMD tasks assigment.

An example of the problem is shown in figure 3(b), in this case, tasks are assigned on a multicore node figure 3(a), which are executed in an iteration of SPMD application. Each task has a similar computational execution time; however the communication process could be different. For example, if a task has a communication with another task but in different node, the communication is made through network link (LAN) and this link has more latency than communications which are made through internal processor architecture. The slower communication limits the time defined for each iteration, then, tasks that finishes before of this time must wait until the iteration ends (Fig. 3(b)). This idle time could generate an inefficient time to the execution and could decrease the performance metrics in parallel application. Our methodology allow to give a solution, which permit to maintain the system efficiency.

3 Related Work

To achieve the objective of this work, we divided the conceptual study in three main aspects, mapping, scheduling and multicore in SPMD application. In mapping topic, Virkram [2] has studied a suitable strategy which permit to improve some performance metrics in SPMD applications. However, this work is mainly focused on seeking the best speedup, obtaining the lower execution time. This mapping tries to search the maximum number of nodes which application need to execute without evaluate the efficiency level. Additionally, different kind of mapping are studied some are statics [3] and others dynamics [4]. The statics mapping are focused on homogeneous architecture of single core node, obtaining different manner to distribute the workload between nodes. These distributions are by rows, columns, blocks or through acyclic blocks, etc. On the other hand, dynamics mapping presents their distribution based on the computational power of the PE inside environment.

Furthermore, exits works oriented to study the effects of communications links on multicore architecture [5], and multicluster in [6], where the communications delays and bandwidths are evaluated with the aim to obtain an improvement of the efficiency within the environment. Also, these works present different communications levels and the way to administrate them in order to manage the troubles generated by communication parameters and even more when communications are different. is important in a heterogeneous environment to manage the workload properly, due an incorrect distribution could generate inefficiency in the system when a SPMD application is executed [7].

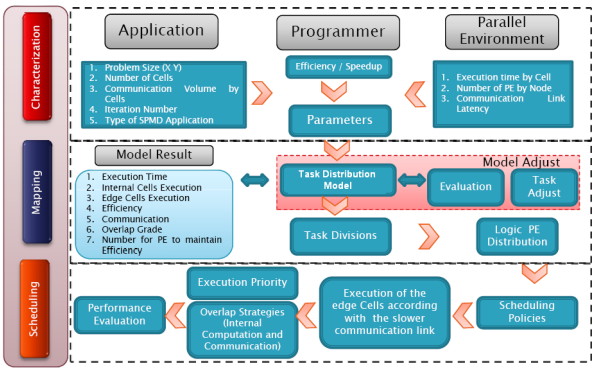


Fig. 4. Methodology for efficient execution of SPMD applications.

Additionally, scheduling strategies have been studied [8], to be developed for large-scale architectures which use heterogeneous distributed systems for SPMD tasks, the objective of these scheduling are to minimize the execution time of SPMD tasks, but they do not use an overlapping strategies to minimize the inefficiency generated. Additionally, in order to obtain a better performance metrics for SPMD application, the evaluation have been made in a multicore cluster [9], and we can appreciate the system degradation when are added more PE to system.

4 Methodology

The methodology developed is composed by three phases: characterization, mapping and scheduling of SPMD tasks and are detailed below(Fig. 2).

4.1 Characterization Phase

The main function of this phase is to determine the parameter which will be included in mapping distribution model. The characterization is made through a testing of the environment where communication and computation values are determined. This phase is divided in three types of inputs.

One of them is the application parameters. The application parameters offer to our methodology information related with some application characteristics such as problem size, number of cells, iteration number etc. Additionally, this phase determine the application behavior within the application.

Another element included is the parallel environment characteristics, in this section is evaluated the computational and communication time of a task inside different communication links. The latencies and bandwidths are evaluated in order to obtain the heterogeneous characteristics within environment. Finally, this phase is concluded with the programmer parameters, the objective is to determine the efficiency level require to execute the SPMD application on a heterogeneous environments.

4.2 Mapping Phase

The objective of mapping strategy is to determine the number of tasks to manage the computational idle time generated by communication paths. Then we have

to evaluate the communication path slower in order to assign tasks to all PEs present until the idle space is covered as is shown in figure 5.

Once the model is calculated, we could determine some model result such as: execution time, overlap grade, number of PEs necessaries according with the efficiency defined, internal execution time, and communication time. etc. This values are estimated with the analytical model defined by the author in [10].

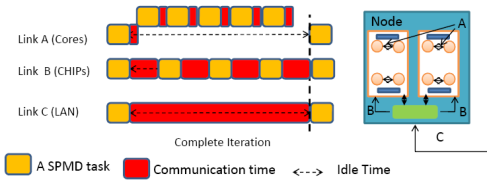


Fig. 5. Communications managing through a mapping strategy.

4.3 Scheduling Phase

The objective of the scheduling phase is to establish the tasks executions. The tasks are divided in two type, one defined as internal tasks and the other are defined as edge tasks. The scheduling works assigning priorities to tasks, where the highest priorities are established for tasks which have communications through slower links. The objective to assign priorities is to overlap the internal computation and edge communication.

The priorities are assigned in the follow way, firstly, tasks with two external communications are selected and the priority 1 is assigned, followed by tasks with one external communication with priority 2 and then are execute all internal tasks which will be overlapped with the edge communications 5, this tasks have execution priority number 3 Figure 6.

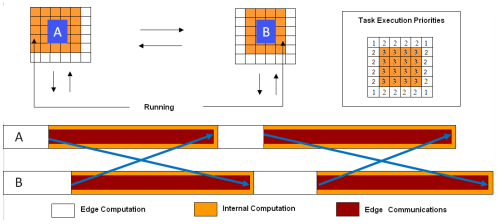


Fig. 6. Scheduling process priorities

The priorities organize the way to execute the SPMD applications, and also permit to manage the delays generated by communications paths. The priority to execute edge tasks allows us to have all the internal tasks time to communicate the edge information for next iteration. This strategy gives possibilities to overlaps and administrates the inefficiencies generated by communications.

5 Implementation

To implement our methodology, we develop a framework in C where the characterization, mapping and scheduling are included. This allow us to execute the application including our program module, where these modules determine the characteristics within environment and we could develop the mapping with a tools to solve linear inequalities in order to obtain values for X_i and Y_i which are the amount of SPMD task that will be assigned to each PEs. The source code 1.1 shows how to assign the module within the SPMD applications.

```
Start
characterization();
Core_Affinity();
mapping_distribution_workload();
    for iteration=0 to N
                          //scheduling process
      Edge_computation();
      Communication_process();
        Internal_computation();
    End For
End
```

Source Code 1.1. SPMD Methodology Algorithm

Another element including in the framework is the core affinity process, the objective is to assign to each core the amount of tasks correctly in order to minimize the number of communication through slower communication paths.

6 Performance Evaluation

Our experiments were conducted on a multicore cluster DELL with 4 nodes, each node has 2 Quadcore Intel Xeon E5430 of 2.66 Ghz processors, and 6 MB of cache L2 shared by each two core and RAM memory of 12 GB by blade and we use a heat transfer, wave equation and Laplace application.

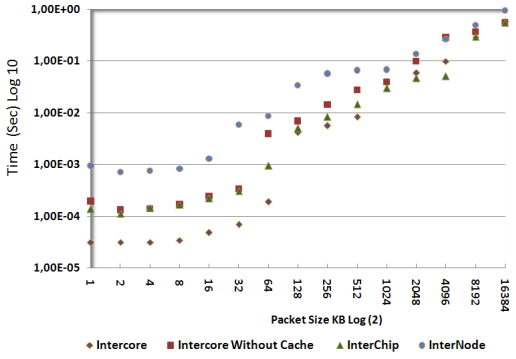
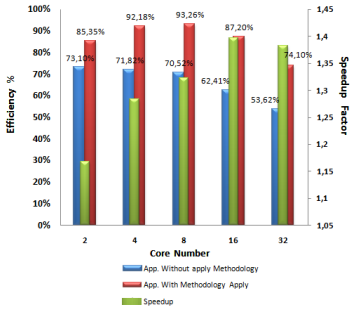
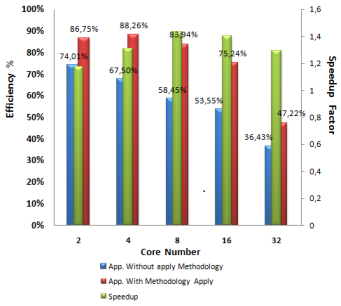


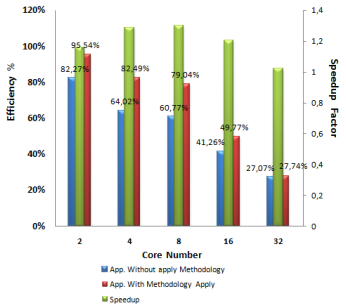
Fig. 7. Communication characterization on multicore cluster



(a) Heat Transfer Application



(b) Laplace Equation



(c) Wave Equation

Fig. 8. Efficiency and Speedup in different SPMD applications.

The first step is to evaluate the communications paths, in this cluster there are four communication types which are defined as, intercore, intercore without cache, interchip and internode, each of them has a different communication behavior as is shown in figure 7.

The figure presents the differences between each link that could be approximately one and half order of magnitude in some cases. Additionally, communications have been tested with different packet sizes because we noticed that communications do not have a linear relationship with the packet size.

Once finished the characterization, we could evaluate the mapping and scheduling. Figures 8(a),8(b) and 8(c) show how could be the improvement between the original application and when we apply our methodology. The result shows an improvement around 40% in best case, and allows us to evaluate the effectiveness of mapping and scheduling over the SPMD applications tested. The numbers of core used to test are fixed and we can observe how the efficiency is improved and maintained while the core number is below by PEs number calculated by our methodology. The PEs could be observed until our methodology find the highest speedup between the application with an without our methodology.

Finally, the experiment reported in this section made possible the analysis of our methodology with different SPMD applications. We have achieved through our methodology maintain the efficiency in a heterogeneous communication environment. Additionally, results show how the communication could be managed in order to improve the performance metrics within application and how the efficiency of a SPMD application could be improved.

7 Conclusion and Future Work

This work allows us to demonstrate how a SPMD application could be executed efficiently in a heterogeneous environment. The efficiency is maintained due the mapping and scheduling strategies, where in both we try to manage the communication latency. Our methodologies through mapping enable to improve the execution time while the efficiency is managed. We could set the amount of tasks necessaries each PE according with the value of the slower communication path.

Finally, the execution order permits to develop an overlapping strategy between internal computation and edge communications, allowing us to control the inefficiencies generated by communications.

Some important future lines consist of generalizing the methodology to include other scientific computation applications, and the selection of the optimal number of PEs necessary to execute efficiently. Also we want to include the characteristics to our model in order to execute a SPMD application in a heterogeneous environment as a multicluster environment.

References

1. R. D. Argollo E. and L. E., "Tuning application in a multi-cluster environment," *Lecture Notes in Computer Science*, vol. 4128, p. 78, 2006.
2. K. Vikram and V. Vasudevan, "Mapping data-parallel tasks onto partially reconfigurable hybrid processor architectures," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 14, no. 9, p. 1010, 2006.
3. F. Guirado, A. Ripoll, C. Roig, X. Yuan, and E. Luque, "Predicting the best mapping for efficient exploitation of task and data parallelism," *Lecture notes in computer science*, pp. 218–223, 2003.
4. S. Sanyal and S. Das, "Match: Mapping data-parallel tasks on a heterogeneous computing platform using the cross-entropy heuristic," *19th IEEE International Parallel and Distributed Processing Symposium*, pp. 64b–64b, 2005.
5. F. Trahay, E. Brunet, A. Denis, and R. Namyst, "A multithreaded communication engine for multicore architectures," *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pp. 1–7, 2008. [Online]. Available: http://dx.doi.org/10.1109/IPDPS.2008.4536139
6. A. Plaat, H. E. Bal, and R. F. H. Hofman, "Sensitivity of parallel applications to large differences in bandwidth and latency in two-layer interconnects," *HPCA '99: Proceedings of the 5th International Symposium on High Performance Computer Architecture*, p. 244, 1999.
7. L. Pastor and J. L. Bosque, "An efficiency and scalability model for heterogeneous clusters." *CLUSTER '01: Proceedings of the 3rd IEEE International Conference on Cluster Computing*, p. 427, 2001.
8. M. Panshenskov and A. Vakhitov, "Adaptive scheduling of parallel computations for spmd tasks," *ICCSA 2007*, vol. 4706/2007, pp. 38–50, 2007.
9. L. C. Pinto, L. H. B. Tomazella, and M. A. R. Dantas, "An experimental study on how to build efficient multi-core clusters for high performance computing," *11th IEEE International Conference on Computational Science and Engineering*, pp. 33–40, 2008.
10. R. Muresano, "Aplicaciones single program multiple data (spmd) en ambientes distribuidos," Master's thesis, Universitutonoma de Barcelona, 2008.