Tuning Application in a Multi-cluster Environment

Eduardo Argollo[1], Adriana Gaudiani[2], Dolores Rexachs[1], Emilio Luque[1]

[1]Computer Architecture and Operating System Department, Universidad Autónoma de Barcelona. 08193 Barcelona, Spain.
eduardo.argollo@aomail.uab.es {Dolores.Rexachs,Emilio.Luque}@uab.es
[2] Instituto de Ciências, Informática, Universidad Nacional de General Sarmiento, Buenos Aires, Argentina.
agaudi@ungs.edu.ar

Abstract. The joining of geographically distributed heterogeneous clusters of workstations through the Internet can be a simple and effective approach to speed up a parallel application execution. This paper describes a methodology to migrate a parallel application from a single-cluster to a collection of clusters, guaranteeing a minimum level of efficiency. This methodology is applied to a parallel scientific application to use three geographically scattered clusters located in Argentina, Brazil and Spain. Experimental results prove that the speedup and efficiency estimations provided by this methodology are more than 90% precision. Without the tuning process of the application a 45% of the maximum speedup is obtained whereas a 94% of that maximum speedup is attained when a tuning process is applied. In both cases efficiency is over 90%.

1 Introduction

The usage of heterogeneous clusters of computers to solve complex scientific problems became ubiquitous in universities departments around the globe. These systems represent a cost-effective tool to achieve data intensive computation. The joint of these clusters for a parallel application execution could enhance the application's problem study by achieving faster results or by increasing its dimension.

The evolution of Internet made usual the interconnection and organization of distributed clusters in computational grids [1]. Although it is not a trivial matter to reach efficiency in heterogeneous clusters [2] and, despite the simplicity of physically interconnecting them, this complexity is increased in a multi-cluster environment [3]. It has been shown that parallel applications written for a single cluster do not run efficiently without modifications on multi-cluster systems [4].

The difficulties are even magnified when Internet represents the multi-cluster interconnection network because of Internet's unpredictable latency, throughput and performance limitations. The study of workload distribution policies is then crucial for obtaining applications speedup in such a heterogeneous and unstable environment.

This paper describes a methodology to migrate master-worker parallel applications from their original cluster to a multi-cluster environment. The proposed methodology targets to decrease the execution time in the multi-cluster environment guaranteeing a pre-established threshold level of efficiency. The methodology "inputs" are the desired

efficiency, some characteristics of the application (computation and communication volume) and multi-cluster system (computers performance and networks throughputs).

The methodology described in this paper is based in a developed hierarchical master-worker system architecture and an analytical system model [5]. Hierarchical approaches proved to be a viable alternative for handling the communication heterogeneity in a multi-cluster [6, 7].

The system architecture enables the collection of clusters to be seen as a single entity, allowing the transparent transition of a master-worker application to the multi-cluster environment, overcoming possible problems with Internet communication [8]. The analytical system model is based on the computation-communication analysis and evaluates which level of collaboration is possible (if possible) between clusters. Before the methodology is used, the application is adapted to the architecture.

The methodology can be divided in three basic phases: Local Cluster Analysis, Multi-Cluster Analysis, and Application Tuning. The Local Cluster Analysis evaluates the possibility for the application in the original cluster to get performance contribution from external clusters.

The Multi-Cluster Analysis evaluates the speedup that each available external cluster can add to the local one executing the specific application. Each cluster's resources are then selected to reach the evaluated speedup respecting the target efficiency.

The Application Tuning phase evaluates the possibilities of tuning the application for improving the reachable performance. For doing this the methodology provides some guide and parameters values to the application developer.

We present our methodology applying it to the execution of a complex scientific application migrated to a multi-cluster system composed of three clusters located in Argentina, Brazil and Spain. The selected application was described in [9] and it studies the problem of short-term memory storage in the central nervous system through simulations of a ring of bistable oscillators. The phenomenon is called Stochastic Resonant Memory Storage Device (SRMSD).

Experiments to validate the methodology by checking the predicted and the real obtained efficiency and speedup are also shown aside the methodology explanation. The experiments proved that for this application it is possible, without the application tuning changes, to reach 45% of the maximum speedup with a level of efficiency over 90%. After the tuning process, based on the Application Tuning recommendations, 94% of the maximum speedup was achieved, maintaining the efficiency over 90%.

The following sections present our study in further detail. Section 2 describes the methodology phases. The SRMSD problem, the system architecture and the basic adaptation of the application to the proposed architecture are explained in section 3. In section 4 the three phases methodology is applied to the selected application, to show the application tuning process for the above mentioned multi-cluster system. Finally, conclusions and further work are presented in section 5.

2 Migrating form a single cluster to a multi-cluster system

The methodology to adapt a parallel application to a heterogeneous multi-cluster environment has as basic goals the *decrease of the execution time*, maintaining the *efficiency* over *a selected threshold*.

The methodology is divided in three basic phases providing an estimated efficiency and speedup, and the selected resources to be used in each cluster. The methodology flow diagram is shown in Fig.1.

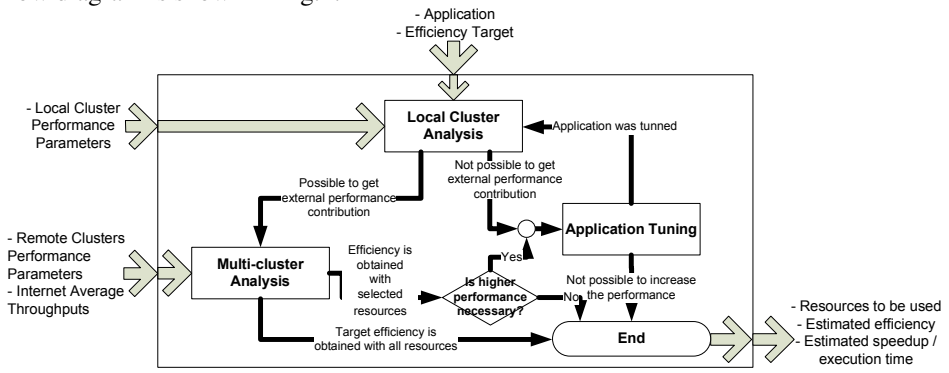

Fig. 1. A flow diagram of the multi-cluster tuning methodology.

The speedup is the metric for measuring the performance improvement. It is defined as the ratio between the application original execution time in the local base cluster and the execution time in the multi-cluster system.

For parallel applications in homogeneous clusters the efficiency is defined as the ratio between the execution speedup and the number of computers. This ratio is not directly appropriate when heterogeneous computers are used [10]. For our work efficiency is redefined as the ratio between the obtained and available performances.

The first phase is the Local Cluster Analysis that requires the algorithms communication volume and the local cluster *performance parameters* (the performance of each computer running the application and the local area network throughput) to evaluate if the application is apt to receive external clusters' performance collaboration.

When the remote clusters can collaborate raising up the speedup then the Multi-Cluster Analysis is the following phase. On the contrary, the application should be tuned to be able to use the multi-cluster resources. The Multi-Cluster Analysis has as "inputs" the *performance parameters* of all the remote clusters and the average value of Internet throughput between the clusters.

The Multi-Cluster Analysis identifies for each remote cluster the resources that can collaborate with the local cluster, keeping the efficiency threshold. This analysis also provides the attainable speedup for the application execution using these resources.

There are two possible outputs from the Multi-Cluster Analysis. The first output, when the obtainable performance is satisfactory, leads to the end of the process, the other output, for improving the performance, is the Application Tuning phase.

The Application Tuning evaluates the possibilities of adapting the application data distribution strategy for improving the collaborative performance. Some *performance recommendations* are presented to guide the possible application changes. If the application is changed the application needs to be re-evaluated.

Detailed explanations of the methodology phases are provided in section 4.

3 The testbed description: the application and the architecture

3.1 The SRMSD problem and simulation application

The Stochastic Resonant Memory Storage Device SRMSD application [9] represents a numerical simulation to study the response of a system of bistable oscillators coupled unidirectional driven by a source of external noise and a periodic and temporal stimulus.

The physical phenomena underlying such short-time storage is that of stochastic resonant (SR) [11], that is, the presence of external noise is essential in sustaining the stored information for an appreciable time once the external stimulus has disappeared. Under this condition the set up acts as a SRMSD.

The application is used to perform several numerical simulations involving rings with different numbers of links and delay times in the coupling. The result is the evaluation of the power-spectral density of the first oscillator during the travelling signal loops, averaging it over a suitable ensemble of N initial conditions in order to hinder fluctuations. The windowed Fourier transform is used to provide a picture of the decaying process.

The SRMSD model and its simulation in a single-cluster environment were previously developed by the Argentinean research group using the master-worker paradigm [9]. The master distributes to all workers the command to execute one simulation. The worker then generates a random set of initial conditions and simulates the traveling signal, sending back to the master a tri-dimensional matrix representing the simulation result. When the master receives one result back, it sends to this worker another simulation command, reaching a dynamic load balancing in the heterogeneous cluster.

When all the N simulations matrices results are received and added by the master, the master calculates the average value of these tri-dimensional matrices, writes it in a file and terminates the program execution.

3.2 Multi-cluster architecture

To interconnect clusters in such way that it is possible to reach efficient collaboration in a multi-cluster environment a run-time architecture was created. This architecture is a hierarchical master-worker on which the remote clusters are called sub-clusters, with their sub-masters and sub-workers.

The architecture supports the overlapping of computation and communication in the workers and a dynamic on-demand tasks distribution policy. To isolate the local network and Internet, overcoming the problems on the Internet communication between clusters, the architecture includes a module called Communication Manager (CM) [8].

The required modifications in the single-cluster SRMSD application to be executed in the multi-cluster system were to add the CM module and to allow the sub-master to execute receiving its amount of work from the main cluster. The integration of the CM simply implies the creation of a special worker process that communicates through Internet. For the master, this “special worker” acts like any worker with the difference that it has a higher performance (correspondent to the whole remote cluster) and a higher latency (because of the use of Internet). Fig. 2 shows the three testbed clusters.

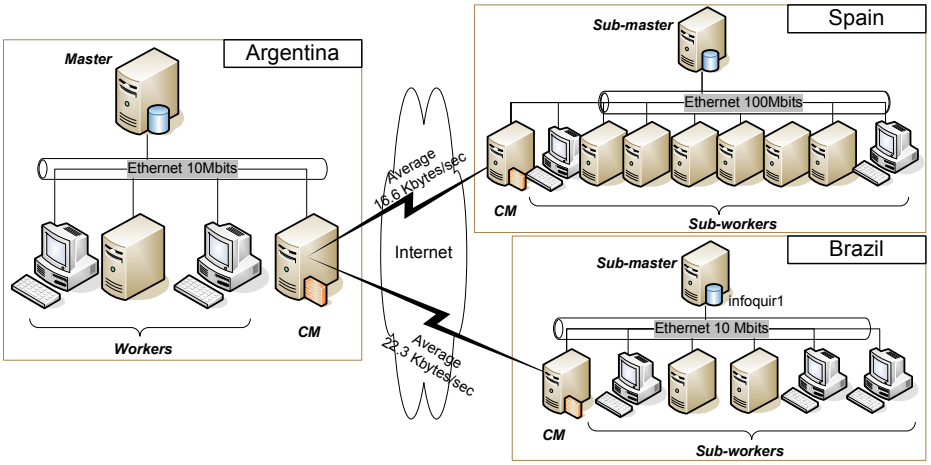

Fig. 2. Multi-cluster testbed system with the local cluster in Argentina.

4 Applying the methodology to the SRMSD

Throughout this section the methodological phases are applied to the SRMSD application migrating it from its original single-cluster version (Argentina) to the multi-cluster (Argentina, Brazil and Spain) scenario. The target threshold efficiency is fixed in 85% and the SRMSD application is executed for a set of N=500 initial conditions.

The Fig. 3 shows the methodology flow diagram with the internal structure of its phases – Local Cluster Analysis (LCA), Multi-Cluster Analysis (MCA) and Application Tuning (AT).

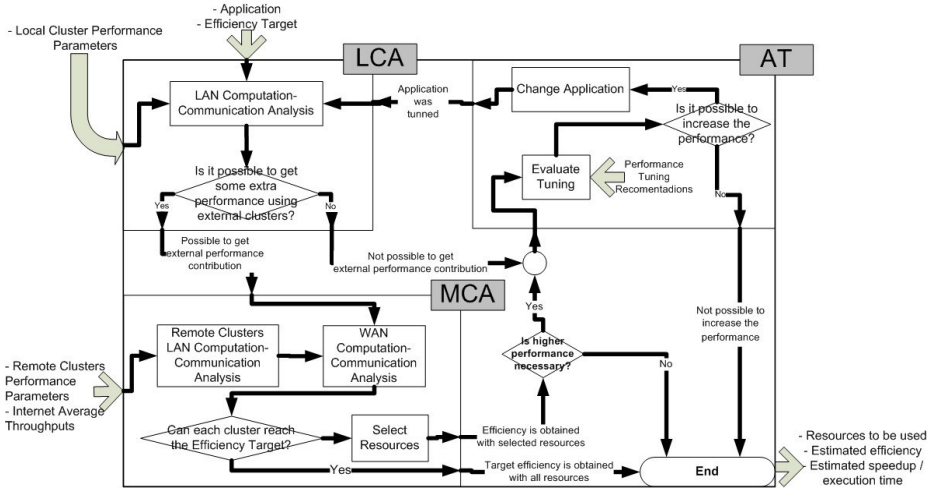

Fig. 3. Methodology flow diagram and phases internal structure.

4.1 Local Cluster Analysis

The main process inside the LCA is to apply the computation-communication analysis [5] to the local cluster. This analysis is oriented to the estimation of the speedup and efficiency that can be obtained in the parallel application execution.

If the analysis' concludes that the application is not able to obtain the whole *Available Performance* in the local cluster then the application should be tuned (AT phase). Otherwise the next phase is the MCA.

a) General computation-communication analysis

Considering the master does not represent the bottleneck, a parallel application execution time is either limited by the computers performance (computation-bounded) or by the network throughput (communication-bounded). The *Maximum Performance* (*MaxPerf*) for a specific execution is achieved when the execution is computation-bounded. This happens when the application's computation time is greater than or equal to its communication time.

For a worker task running on a processor, the *computation time* (T_{Cpt}) is defined as the ratio between the task number of *operations* (*Oper*) and the processor *performance* (*Perf*): T_{Cpt}=*Oper/Perf*. The *communication time* (T_{Comm}) is the ratio between the volume of *data communication* (*Comm*) (worker task data from and to the master) and the *network throughput* (*TPut*): T_{Comm}=*Comm/TPut*. The *MaxPerf* is the performance that can be obtained when $T_{Cpt} \geq T_{Comm}$ (Eq. 1).

In a multi-cluster environment there are two communication levels: intra-cluster and inter-clusters. To calculate the *MaxPerf* for the intra-cluster ($MaxPerf_{intra}$) it is necessary to consider the local-area throughput and for the *MaxPerf* for the inter-cluster ($MaxPerf_{inter}$) the Internet average throughput between the clusters.

The *Available Performance* (*AvPerf*) is the addition of each worker performance executing standalone the application tasks. The *Estimated Performance* (*EstPerf*) that a cluster can provide to the multi-cluster system is then minimum value between its AvPerf, $MaxPerf_{LAN}$ and $MaxPerf_{Inet}$ (Eq.2). The *Estimated Efficiency* for each cluster is the ratio between the cluster *EstPerf* and *AvPerf* while the *Estimated Speedup* is the ratio between the cluster *EstPerf* and the local cluster *EstPerf* ($EstPerf_{LC}$) (Eq. 3).

$$T_{Cpt} \geq T_{Comm} \Rightarrow MaxPerf = Perf \leq \frac{Oper * TPut}{Comm} \quad (1)$$

$$EstPerf = \min(AvPerf, MaxPerf_{LAN}, MaxPerf_{Inet}) \quad (2)$$

$$EstimatedEfficiency = \frac{EstPerf}{AvPerf}; EstimatedSpeedup = \frac{EstPerf}{EstPerf_{LC}} \quad (3)$$

b) SRMSD simulation computation-communication analysis

For the SRMSD simulation application we consider each single simulation as the task to apply our computation-communication local and remote analysis. Each worker performance is then measured in simulation tasks per second. The worker *available performance* value should be obtained by the execution of simulations tasks in each cluster computer.

The number of operations (*Oper*) is the total number N of simulations. The communication for each simulation task consists of one integer value (representing a command for a simulation execution) from the master to the worker, and a 601x31x31 float-point elements matrix with the simulation results from worker to master. The total communication volume for one task is then CV=2,310,248 bytes. The total communication is N times CV. For the SRMSD application Eq.1 become in Eq.4.

$$MaxPerf = \frac{N * Tput}{N * CV} = \frac{Tput}{CV} \quad (4)$$

In this Local Cluster Analysis step the efficiency and speedup are evaluated for the local cluster (Argentina). Table 1 shows the testbed parameters, and the estimated and experimentally obtained values for the Speedup and the Efficiency in the Argentinean Cluster. The testbed parameters are the number of computers, the LAN throughput and the *Available Performance* in tasks/second. The Eq. 3 were used to estimate the speedup and efficiency. The comparison from the experimental speedup and the estimated one is our estimation precision.

Table 1. Testbed parameters, estimated and experimental Speedup and Efficiency, and estimation precision for the local cluster stand alone execution of the application.

Cluster	#Com	LAN TPut (Mbytes /sec)	*AvPerf* (10^{-3} Tasks /sec)	Estimated Speedup	Estimated Efficiency	Experimental Speedup	Experimental Efficiency	Estimation Precision
Argentina	4	1	1.754	1.000	100%	0.918	92%	**92%**

The experimental data shows that a high-level of efficiency is possible although the theoretical peak could not be achieved. The differences between the estimated and experimental efficiency values are mainly caused by light load-imbalance. Since the application is able to execute efficiently in the local cluster, the next phase is the MCA.

4.2 Multi-Cluster Analysis

The first step of the MCA is to apply the computation-communication analysis to each remote cluster. The MCA parameters, the estimated and experimental Speedup and Efficiency values for the clusters are displayed in Table 2.

Table 2. Testbed parameters, estimated and experimental Speedup and Efficiency, and estimation precision for the multi-cluster execution of the application with all resources.

Cluster	#Workers	Inet Tput (Kbytes /sec)	*AvPerf* (10^{-3} Tasks /sec)	Estimated Speedup	Estimated Efficiency	Experimental Speedup	Experimental Efficiency	Estimation Precision
Argentina	3		1.754	1.000	100%	0.886	89%	**89%**
Brazil	5	25.3	3.106	1.771	100%	1.621	92%	**92%**
Spain	8	21.0	23.570	5.307	39%	5.795	43%	**92%**
Total	**16**		**28.430**	**8.077**	**50%**	**8.302**	**51%**	**97%**

From the Table 2 we can infer that it is possible to get a speedup of 8 for the application execution using the external clusters. The experimental speedup of the Spanish cluster is higher than estimated because during the experiments the real Internet average throughput was 23.0 Kbytes/sec, greater than the estimated average used in the analysis.

The estimation data in Table 2 shows that the target threshold efficiency for the cluster in Spain is not reachable. To attain the efficiency threshold it is necessary to select the resources to be used in this cluster. This adjustment is done by selecting the workers for which the addition of *Available Performance* is approximate to the *Estimated Performance* (Eq. 3). Table 3 shows the data using just the selected computers of the Spanish cluster.

Table 3. Testbed parameters, estimated and experimental Speedup and Efficiency, and estimation precision for the multi-cluster execution of the application with selected resources.

Cluster	#Workers	Inet Tput (Kbytes /sec)	*AvPerf* (10^{-3} Tasks /sec)	Estimated Speedup	Estimated Efficiency	Experimental Speedup	Experimental Efficiency	Estimation Precision
Argentina	3		1.754	1.000	100%	0.858	86%	**86%**
Brazil	5	25.3	3.106	1.771	100%	1.621	92%	**92%**
Spain	3	21.0	8.885	5.065	100%	4.863	96%	**96%**
Total	**11**		**13.745**	**7.836**	**100%**	**7.341**	**94%**	**94%**

This experiment shows that the speedup decreased from 8.3 to 7.3 while the efficiency of the Spanish cluster increased from 39% to 96%. This means a better utilization of the resources or a lower cost, in case of paying for remote resources utilization.

Even though the execution speedup can be 8, comparing the total *AvPerf* and the Argentinean cluster *AvPerf* in Table 2, we can infer that potential speedup of the multicluster is 16.2. To try to get the maximum speedup it is necessary to apply the Application Tuning phase.

4.3 Application Tuning

At the MCA we concluded that, without modifications, the SRMSD application can not obtain the whole of the Spanish cluster performance. Based on the data in Table 2, the value for the Spanish cluster $MaxPerf_{inter}$ (Eq.4) is of 9.30×10^{-3} tasks/sec. Comparing this value with the Spanish cluster *Available Performance* (23.57×10^{-3} tasks/sec), we can conclude that Internet communication between Spain and Argentina is saturated.

In this case, the Application Tuning (AT) phase intends to improve the application computation-communication ratio to enhance the obtained performance. It is necessary then to evaluate the increase in the inter-cluster granularity, executing more tasks with the necessity of less communication.

The final operation at the SRMSD application is to average the different simulations results. The granularity could be changed if sub-masters perform the addition of workers partial results and just communicates to the local master the result of a certain number S of additions. This granularity change means that each result simulation communicated (CV bytes) represents S simulation tasks. The number S of simulations that need to be aggregated before sending the results depends on the Internet throughput. S should be small enough to avoid load imbalance and big enough to reach the desired performance.

Once the tuning strategy is set, the application needs to be changed and re-evaluated. Nothing was changed in the local execution and the LCA can be skipped. On the remote execution, for N simulation tasks (*oper*), (N/S)*CV bytes are communicated (*comm*). Applying it in the Eq. 2, the new $MaxPerf_{Inet}$ is Eq. 5.

$$MaxPerf_{Inet} = \frac{S * Tput}{CV} \quad (5)$$

The Spanish cluster has an *Available Performance* of 23.57 $x10^{-3}$ tasks/sec. Using this as the $MaxPerf_{Inet}$ in the Eq. 5 we can conclude the number of aggregated simulations S should be 2.53. The amount of simulations needs to be an integer and for choosing its value it is necessary to evaluate the impact of this number in the balance of the load.

For S=2 we would not achieve the totality of the Spanish performance. For S=3 the load-imbalance is increased. Since the value 3 is still not significant when compared to the total number of tasks (N=500) we choose this value for the application tuning.

Table 4 shows the testbed parameters, the estimated and experimental Speedup and Efficiency for the tuned application. This table shows that with the AT the speedup was increased from 7.34 to 15.33. The efficiency in all clusters was 95% and, since all the clusters' resources were used, this means 95% of the total *AvPerf*.

Table 5 presents a summary of the predicted values and the experimental results obtained as results of the different phases of the proposed methodology.

Table 4. Testbed parameters, estimated and experimental Speedup and Efficiency, and estimation precision for the multi-cluster execution of the tuned application.

	Testbed parameters		**Theoretical peak performance**		**Real performance**		
Cluster	**Inet Tput (Kbytes /sec)**	***AvPerf* (10^{-3} Tasks /sec)**	**Estimated Speedup**	**Estimated Efficiency**	**Experimental Speedup**	**Experimental Efficiency**	**Estimation Precision**
Argentina		1.754	1.000	100%	0.889	89%	**89%**
Brazil	25.3	3.106	1.771	100%	1.594	90%	**90%**
Spain	21.0	23.570	13.437	100%	12.848	96%	**96%**
Total		**28.430**	**16.208**	**100%**	**15.332**	**95%**	**95%**

Table 5. Estimation and experiments speedup, efficiency and execution time comparison for different steps along the methodology.

	Estimated Indexes Values		**Experimental Indexes Values**		**Prediction Precision**
Description	**Speedup**	**Efficiency**	**Speedup**	**Efficiency**	
Original application single cluster execution.	1	100%	0.918	92%	92%
Intermediate evaluation in the MCA phase: All clusters with all resources.	8.077	50%	8.302	51%	97%
After the MCA phase: All the clusters with selected resources.	7.836	100%	7.341	94%	94%
After the AT phase: All the clusters with all resources **with tuned application**	16.208	100%	15.332	95%	95%

5 Conclusions and Future Work

A multi-cluster environment using Internet as inter-communication network represents a cost-effective way to speedup the execution of scientific applications. This paper presented a methodology for adapting a parallel application from a single cluster to a multi-cluster environment, keeping a threshold level of efficiency. The methodology phases were presented and an example application was used to follow the methodological steps.

To validate the methodology, experiments were done in different stages. The results proved that the estimation is over 90% precision.

The application execution time was reduced to 12% its original single cluster time just with the adaptation to the proposed architecture. When the application was tuned the execution time was reduced to 6% its original value. The efficiency of the system was kept over 90%.

Our methodology should be considered as a low level, direct resources utilization of computational grid environments. Therefore it provides an upper limit of the environment utilization, providing a good prediction of the maximum available speedup guarantying a defined resources utilization efficiency.

Future lines are to extend the multi-cluster computation-communication model to other parallel programming paradigms and to include in the model other system parameters (memory size, cache size, ...).

References

[1] I. Foster. “The grid: A new infrastructure for 21st century science”. Physics Today, 55(2), pp 42–47, 2002.

[2] Olivier Beaumont, Arnaud Legrand, and Yves Robert. “The master-slave paradigm with heterogeneous processors”. IEEE Trans. Parallel Distributed Systems, 14(9):897-908, 2003.

[3] B. Javadi, M. Akbari and J. Abawajy. “Performance analysis of heterogeneous multi-cluster systems”. ICPP 2005, pp 493-500, 2005.

[4] Bal, H.E.; Plaat, A.; Bakker, M.G.; Dozy, P.; Hofman, R.F.H., "Optimizing parallel applications for wide-area clusters". Proceedings of IPPS/SPDP 98, pp.784-790, 1998.

[5] E. Argollo, D. Rexachs, F. Tinetti and E. Luque. “Efficient Execution of Scientific Computation on Geographically Distributed Clusters”. In LNCS vol. 3732, 2004.

[6] Aida, K.; Natsume, W. & Futakata, Y. “Distributed computing with hierarchical master-worker paradigm for parallel branch and bound algorithm” CCGrid 2003, pp. 156-163, 2003.

[7] Nieuwpoort, R.V.; Kielmann, T. & Bal, H.E. “Efficient load balancing for wide-area divide-and-conquer applications”. PPoPP '01, ACM Press, 34-43, 2001

[8] E. Argollo, J. R. de Souza, D. Rexachs, and E. Luque. “Efficient Execution on Long-Distance Geographically Distributed Dedicated Clusters”. In D. Kranzlmüller et al. editors, Proceedings of the 11th Euro PVM/MPI 2004, LNCS vol. 3241, pages 311--319, 2004.

[9] M.F. Carusela, R.P.J.Perazzo and L. Romanelli. “Stochastic resonant memory storage device”. Physical Review, 64(3 pt 1):031101, 2001.

[10] L. Colombet, and L. Desbat. “Speedup and efficiency of large-size applications on heterogeneous networks”. Theoretical Computer Science, 196 ,pp 31-44, 1998.

[11] Bruce McNamara and Kurt Wiesenfeld. “Theory of stochastic resonance”. Physical Review, A 39, pp 4854 – 4869, 1989.