# A STUDY OF A GRID ARCHITECTURE BEHAVIOR FOR LARGE DATA TRANSFERS

Carlos Jaime Barrios-Hernandez

**UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS**

**ECOLE DOCTORALE STIC**
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

# THESE

*Pour obtenir le titre de*

***Docteur en Sciences***

*de l'Université de Nice-Sophia Antipolis*

***Mention: Très Honorable***

*Présentée et soutenu par:*

**CARLOS JAIME BARRIOS HERNÁNDEZ**

# ETUDE DE TRANSFERT HAUT DÉBIT SUR DES GRILLES DE CALCUL

*Thèse Dirigiée par:*
**Michel RIVEILL**, *Université de Nice-Sophia Antipolis et* **Yves DENNEULIN**, *Université de Grenoble*

SOUTENUE LE 6 OCTOBRE 2009

*Jury:*
**Pascale Vicat-Blanc Primet**, *Professeur à l'Institut National de Recherche en Informatique et Automatique (INRIA), Rapportrice*
**Michel Daydé**, *Professeur à l'Université de Toulouse, Rapporteur*
**Pierre Sens**, *Professeur à l'Université de Paris, Examinateur*

**NICE-SOPHIA ANTIPOLIS UNIVERSITY, FRANCE**
DOCTORAL SCHOOL IN INFORMATION AND COMMUNICATION SCIENCES AND TECHNOLOGIES

# THESIS

*Presented to obtain the title of*

**DOCTOR IN SCIENCES**
**OF THE NICE-SOPHIA ANTIPOLIS UNIVERSITY**
COMPUTER SCIENCE SPECIALITY

*Presented by:*
**CARLOS JAIME BARRIOS HERNÁNDEZ**

# A STUDY OF A GRID ARCHITECTURE BEHAVIOR FOR LARGE DATA TRANSFERS

*Advised by:*
***Michel Riveill***, NICE-SOPHIA ANTIPOLIS UNIVERSITY
***Yves Denneulin***, UNIVERSITY OF GRENOBLE

*Reviewers:*
***Pascale Vicat-Blanc Primet***, NATIONAL INSTITUTE OF RESEARCH IN COMPUTER SCIENCE
***Michel Daydé***, UNIVERSITY OF TOULOUSE

*Examiner:*
***Pierre Sens***, UNIVERSITY OF PARIS

**LABORATORY OF INFORMATICS OF GRENOBLE AT MONTBONNOT (LIG), MESCAL TEAM AND LABORATORY OF INFORMATICS, SYSTEMS AND SIGNALS OF SOPHIA ANTIPOLIS (I3S), RAINBOW TEAM**
**FRANCE, TUESDAY, OCTOBER 6TH 2009**

*What do I do when my love is away*
*(Does it worry you to be alone)*
*How do I feel by the end of the day*
*(Are you sad because you're on your own)*
*No, I get by with a little help from my friends,*
*Mmm, get high with a little help from my friends,*
*Mmm, gonna to try with a little help from my friends ...*

# ACKNOWLEDGMENTS

*I'm would like to thank for the support among these years to many people: professors, colleagues and friends that helped me and encouraged me in different moments of my thesis life. I hope that a page is sufficient ;-) .*

*First, my advisers, Prof. Michel Riveill and Prof. Yves Denneulin, not only for their guide, also for their example and patience.*

*A special thanks to members of jury, Professors Pascale Vicat Blanc-Primet, Michel Daydé and Pierre Sens.*

*On the other hand, I take this opportunity to thank to Prof. Claudia Roncancio and Mme. Patricia Suarez.*

*I thank to the two host labs: LIG (specifically to the old ID-Lab and the Mescal Team), for more of five years and the I3S Lab (specially to Rainbow team), for more of three years of support and interaction. I express my gratitude to my colleagues and friends: Yiannis, Pedro, Olivier, Pierre, Adrien, Maxime, Thomas, Luiz Angelo, Jesus, Jonathan, Jean Dennis and the others than may be I forgot..., the administrative people: Annie Claude Vial-Dallais, Christian Seguy and Stellio Panagopoulos, and of course, the permanent researchers and professors Bruno Gaujal, Oliver Richard, Bruno Raffin, Gregory Moune, Jean-Marc Vincent, Derick Kondo, Dennis Tristan and the others of this excellent laboratory. A special thank to Prof. Briggite Plateau, director of the LIG Lab. For the side of Nice-Sophia Antipolis, I thank to Javier, Marcel and Tristan. Special thanks to Sabine Barrere, administrative assistant of I3S. As well, I take this line to thank to Prof. Johan Montagnat for their important comments in a critical moment of the research work.*

*Special thanks to Aladdin-Grid'5000 people for their special attention during the tests time, mainly to Pierre Neyron.*

*The thesis work was enriched with the work of two great research laboratories: the LAAS at Toulouse, France and the UFRGS-GPPD at Porto Alegre, Brazil. Special thanks to Professors Herve Aubert and Fabio Coccetti and to my colleague Fadi Khalil, in Toulouse. In Brazil, I thanks a lot to Professors Phillipe Navaux and Nicolas Maillard and the students Rodrigo Kassick and Marcia Cera.*

*But the thesis life is not only research and study, so, I thanks a lot to my friends Evelio, Gabriel, Carlos Hernan, Lina Judith, Lina Maria, John Jairo, Diana, Paola John Alexander, Santiago... "mis paisanos" from Colombia. Lupita, Luis, Noé,... "mis amigos" from Mexico. Bibiana, Lucas, Patricia, Kiev,... "meus parceiros" from Brazil. Christianne, Lisbeth, Franck,... "mes amis" from France. Vassilis, Eleni, Tatiana..., in general to all my friends from different nationalities that made my life very pleasant among these years of doctoral studies.*

*Thanks to my family, for their support in the midst of difficulties.*

*Finally, but not the less important, a very great appreciation to Amelita and Guillermo, my "adoptive parents" in France.*

# Contents

# List of Figures

# List of Tables

# 1. Introduction

Dramatic improvements in the capacity and capability of sensors, storage systems, computers, and networks are enabling the creation of data archives of enormous size and value [42]. Grid Computing environments support the execution of applications that produce and consume such data volumes. This production and consumption of data implies processing, transfer and storage. In general terms, the data treatment in each one of the activities and the relation between them corresponds to the data management.

Often, data volumes managed in Grid computing systems grows during the execution time. The capacity of the resources with the processing, transfer and storage is strongly related. Then, the behavior of the system changes during their use, due to the capacity limits of the resources.

Actually, data transfer is a critical process in Grid Computing systems. Because the various formats, sizes and contents of the data that are transferred in messages add complexity and unpredictability to the system behavior. In the same manner, the different technologies and topologies implemented in the Grid computing networks contribute to the complexity and variation of the performance during the data transfer. Frequently, the transfer is more expensive than the processing.

Although Grid Computing systems have architectural features, such as multiprocessing, high bandwidth capacity, large storage capacity and so on, that ensure high performance, users require to know how to exploit these features optimally. Modeling and performance evaluation provides mechanisms for knowing the state of a system and predict their behavior, in order to take advantage of the resources with a minimal cost, in other words, to exploit the system optimally. Obviously, modeling and performance evaluation are large subjects for study and a given problem can be treated on different ways.

Models are needed to predict and analyze data transfer behavior together with tools to implement them. Not only to make a *forecast* of the transfer time in function of the size of message and the number of resources involved, but in order to identify the characteristics which influence the behavior of the data transfer.

To handle this, this work propose a restriction of the problem, covering only the Grid computing applications that transfer large volumes of data during their execution time, in

real time.

## 1.1 Research Context

High Performance Computing Systems (and Grid Computing Systems) allow scientists to treat data *efficiently* in accordance with specific needs. In other words, Grid Computing technology allows to perform science with shared computer science facilities. These facilities can be summarized as technological capacities and possibilities such as great storage, efficiency, ”dynamicity”, heterogeneity, ”pervasivity”, concurrency, high bandwidth and so on. On the other hand, we can consider other aspects, such as a high quality of service, collaborative work, safety between others [14] [42].

Despite technological trends implemented in Grid Computing, the observation and analysis of the performance of the applications in their interaction with the infrastructure is necessary to provide efficiency. Grid computing is particularly complex to observe and analyze, the processes implies parallel and distributed-heterogeneous tasks and concurrency. Nevertheless, the Grid Computing community makes important efforts to propose models and methodologies to observe and describe Grid Computing architectures, Grid Computing platforms, Grid Computing applications and even Grid Computing users (Communities).

This thesis work was done in the general context the High Performance Computing research, specifically Grid Computing research. The work covers a given domain, performance evaluation in realtime.

The observation of the behavior of a system, during a real use, demands an easy, feasible and low cost in the monitoring or measurement of the process and tasks. In the Grid Computing community, this case corresponds to an observation on Production Grid Computing platforms.

The work done in this thesis, includes the performance evaluation of massive data transfers between nodes of Production Grid Computing system. The study treats the large volumes of data consumed and produced by applications that are transferred on high bandwidth networks. The studies associated to this thesis allow to propose analytical models and to implement performance evaluation techniques in monitoring tools.

This thesis proposition has been developed in two teams of two different laboratories: The *Rainbow*[1] Team of the Informatics, Signals and Systems Laboratory, I3S[2], at Sophia Antipolis and The *Mescal*[3] Team of the Laboratory of Informatics of Grenoble, LIG[4] at

[1] http://rainbow.essi.fr
[2] http://www.i3s.unice.fr
[3] http://mescal.imag.fr/
[4] http://www.liglab.fr

Montbonnot-Saint Martin, in France.

Research interests of these teams contribute to address the problem with a hybrid focus. In the case of the Rainbow team, one of the research theme is the services orchestration and enactment on a large scale distributed infrastructure. About the Mescal Team, it exist the interest about the performance evaluation and simulation of large deterministic and probabilistic systems, mainly large scale distributed architectures. This interaction allows to treat the problem under study from two levels, an application and a fabric levels, observing the reciprocal action between both.

## 1.2 Problem Description

Scientific and industrial applications that run in Production Grid Computing platforms produce and consume great sets of data. This production and consumption requires high bandwidth data transfer. Despite the use of high performance networks, the use and concurrency of the shared resources of the Grid computing platforms, saturates these resources differently during the data transfer.

When a specific application runs on a Grid computing infrastructure, the data transfer time varies according to the state of the shared resources (mainly the network resources) and the total quantity of bytes involved in the transfer. Then, there are two important points to handle: one, the requirements to describe the massive data transfer to predict performance, in order to optimize the use of the shared resources by the application; and two, the necessity to made this prediction during a real use.

In both cases, prediction implies systematic approaches to performance evaluation. These approaches permit to define metrics, methodologies and techniques to provide tools to specific observations[5].

Then, the problematic implies the definition of a predictive model. This model should be implemented in performance evaluation mechanisms during a real use. Of course, this predictive model must be simple, not expensive in terms of applicability and addressed to the analysis of high bandwidth data transfer. The high bandwidth data transfer analysis is important because it allows to identify the loss of performance during the data transfer, estimate the best case of use in accord with the available resources, to know the data transfer cost or allow efficient scheduling.

On the other words, the modeling will have to permit to compute the transfer time and observe the behavior of the massive data transfer to analyze the characteristics that affects their performance, such as size of message transferred, resources involved, network com-

[5]However, it is important to consider that these systematic approaches are based in theoretical, practical or theory-practical focus.

ponents, file systems, between others. These characteristics must be related with specific parameters that allows a pertinent description in the model. In consequence, the model proposal must be easily implementable in tools to be used during a real time.

## 1.3 Contributions

The contributions of this work follow the preoccupation expressed before, but we can describe the main results:

- *Propose a realistic performance evaluation technique for high bandwidth data transfer during the execution of Grid Computing applications.*

  The work made the analysis of different possibilities of implementation of the deterministic model to make a correct description of the parallel and massive data transfer process. In consequence, the model should be easy to implement in performance evaluation techniques and tools.

  On another hand, the performance evaluation techniques and derived tools should be exploited during real use, providing real time information.

- *Propose a model to predict parallel and massive data transfer among nodes during the execution of applications that run on Grid Computing platforms.*

  The work made for this objective allows to identify the main characteristics that affect the communication performance during massive and parallel data transfer. These characteristics are related to parameters that describe the interactions between the application and infrastructure resources, in terms of capacity, availability and use.

## 1.4 Document Structure

This manuscript organises the contributions presented above as follow:

Chapter two presents the state of art of the problem in study. This chapter contains a description of the technological context associated with the communication process in High Performance computing (HPC) and distributed systems, explicitly in Grid computing environments. This chapter presents also the current work to model the data transfer in the context of the research in Clusters and Grid Computing systems. A description of different modeling proposals using deterministic and non deterministic models appears to

position our work. The emphasis of this description is to present the possibilities of implementation of these models in performance evaluation tools and techniques addressed to ”on-live” performance evaluation needs.

Chapter three focuses in the $LogP$ model used to describe parallel data transfer. A discussion is presented here, about its utility, advantages and disadvantages to describe communications in parallel and distributed systems. The same chapter presents the pertinence of the use of the p$LogP$ propositions and, the early results of their implementation in the realistic performance evaluation tools, to observe parallel communications in high performance computing platforms, such as clusters.

Chapter four presents our contributions to the realistic model of massive and parallel data transfer, using deterministic modeling derived from the $LogP$ model.

Chapter five shows the results of the realistic modeling using our proposal, to validate their utility in the performance evaluation of parallel and massive data transfer in Production Grid Computing platforms. Measures presented in this chapter were taken during a real use of the platform.

The chapter six, divided in three sections, presents the discussion, conclusion and consequences of this PhD work. The first section of this chapter presents the discussion about the results of this thesis proposal. This chapter aims to determinate the precision and utility of the model in the realistic performance evaluation of massive and parallel data transfer. The second section presents the conclusion of this PhD thesis. The consequences of this work, are presented in a final section, to offer development prospects to the implementation of new features in our model. On another hand, we present guidelines to the development of performance evaluation tools and techniques using our focus.

# 2. State of Art

## 2.1 Technological Context

Technological trends in distributed high performance computing, such as Grid computing systems, is to offer high capacity in communication at the price of increased complexity in their use. The Grid computing community has identified several aspects related with multiple use cases to describe Grid computing communication, taking into account the technological evolution.

Basically, a communication process is simple: a message is transferred between a sender to a receiver from a source among a medium with an objective of use [149] [107]. The complexity appears when the elements involved in this communication grow and the message transfer process exceeds the capacities of those elements.

For example, in a Grid Computing system, as shown in the Figure 2.1 the source of the information delivered by the message may be the result of an operation in a processor or a data set in the disk. Sender and receive can be cluster nodes of a particular platform.

On the other hand, the communication occurs at different levels. In the same Figure 2.1, we can see a cluster level and a Grid level that relies on the two platforms. This multi-level transfer implies an environment exchange and a transmission by different network devices.

Obviously, the communication channel has a specific protocol involved. And, the utility of the information contained in the message is given by the user (For example, the user can select the data source and the interpretation of the results to use the information contained in the data).

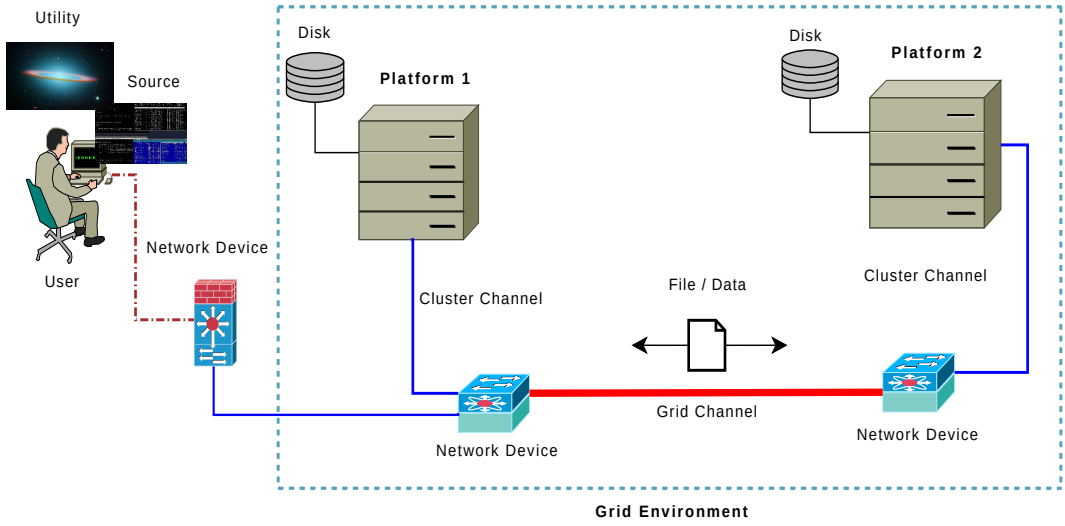


Figure 2.1: *Some Grid Computing Communication Elements*

Often, the communication process is not stable. The transfer time is not constant during communication, it changes in accordance with several situations, as interruptions, interferences, availability of resources, pre-processing or post-processing delay among other situations.

This situation should be interpreted as a communication process that occurs more or less efficiently with the characteristics of the environment used. The efficiency can be described in terms of speed, throughput, deliverance or coverage.

The communication technology implemented in the Grid computing infrastructure affects the performance of the communication process. Capacity, availability, security, speed are determined by the technical characteristics of channels, links, protocols, codes, etc.

In summary, in Grid computing data transfers exist characteristics that add complexity. For example, the transmission channels can be heterogeneous. Others characteristics should be that the data transfer can be simultaneous, parallel, and moreover, that the data are in various formats or sizes.

### 2.1.1 Technological Trends on Grid Computing Communication

Grid Computing technologies support the sharing and coordinated use of various resources in distributed geographic organizations[14] [42]. These organizations are dynamic and they are known as *Virtual Organizations (VOs)*.

The shared resources of the Virtual Organizations implies technology developments. Technology is integrated on systems and the work of technology systems, needs a description of their characteristics to maximize its use. Thus, the system technology is described to define the function and purpose of the system. Actually, in terms of services, specialized communities uses the Open Grid Services Architecture (OGSA) [45] [115] [44] to define and describe Grid Services and Grid Computing Architectures.

Generally, in Grid computing, the descriptions are made following the proposition for distributed systems[1] and web services. Then, well known descriptions, such as the layered Grid Architecture model presented in [42] present the various components grouped in layers interrelated.

These points are treated here for two reasons: first, because the data transfer implies a relation between the different layers of the Grid Computing systems (resources, network, applications), and second, because the data transfer characteristics are associated with network-architecture characteristics.

In network technology terms, since the introduction of the optical transmission, the relation cost-performance is very important[134]. The capacity of the backbone transmission systems are enormous and these technological capacities have been used totally, because a tendency of the applications that runs in distributed computer systems, is the use of all available resources. The same tendency exists in Grid Computing.

Figure 2.2 presents the evolution of the transmission network. In this figure is possible to see the trend of the backbone capacity and their growth among the years.

[1] Also, this description may be more complex if the components of the distributed system are parallel systems.

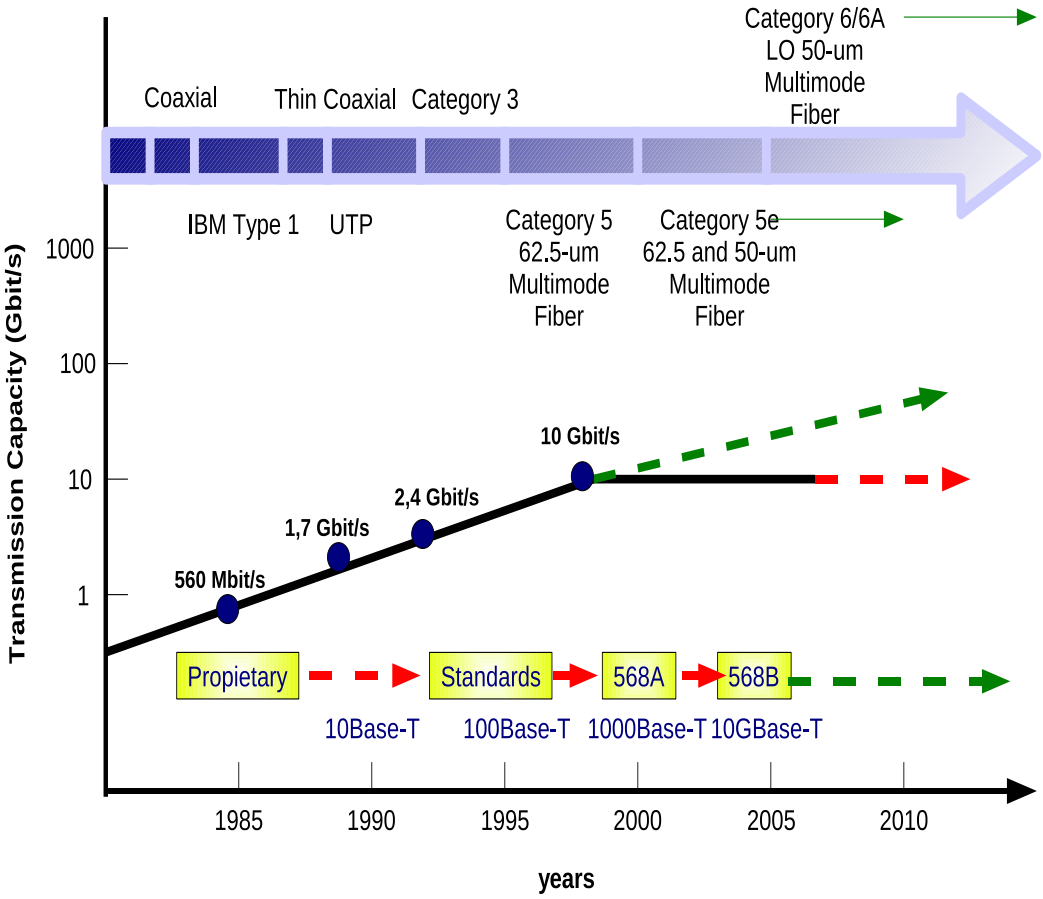


Figure 2.2: *Evolution of Transmission Networks*

In the Figure 2.2, the black line represents the trend of the backbone capacity growth. There are four main dots for 560Mbit/s, 1.7Gbit/s, 2.4 Gbit/s and 10Gbit/s transmission capacities implemented. Each capacity corresponds to a stage in cabling development, represented in the Figure 2.2 by the arrow.

The arrow shows connection wires since 80s, as coaxial type wire (Coaxial, IBM Type 1 and Thin Coaxial), coming to UTP (90's) and finally, actual type wires like Category 5 and Category 6 wires.

An important aspect observed in the Figure 2.2 is the evolution of the use of proprietary networks to standards definitions. For example the collection of IEEE 802.3 standards [72] [73] [74] defines specification for LAN and WAN interconnections, that are clearly exploited in Grid computing.

Figure 2.2 also presents the evolution of physical connection technologies. Of course, these structured cabling systems allows the physical communication between nodes and network devices (hubs, switches, routers), and links distributed sites, a principle of the Grid computing interaction. So, technologies that have evolved in the time such as 10Base-T 100Base-T, 1000Base-T and 10GBase-T, and implementations of structured cabling systems as 568A and 568B [138] are used in Grid computing infrastructures.

Nowadays, the transmission technology implemented in Grid computing infrastructures research to give an excellent performance in relation with high speed connections, high bandwidth capacity and cost. These characteristics are the same for High Performance Computing platforms. Observing the statistics of interconnection systems used in *HPC* infrastructures, from the Top 500 site [140], it is possible to identify the *most popular* interconnection systems.

Often, the Grid computing infrastructures are a merge of the interconnection of distributed HPC platforms. Projects such as EU-EGEE [34], Grid5000 [55], TeraGrid [139] between others, use different technology. This diversity in technology is also observable in the case a specific Grid Computing platform interacts with other platforms with a lower network performance (I. e. Relation between European EGEE infrastructures and EELA [35] networks or French Grid 5000 Infrastructure and Brazil-Porto Alegre cluster platforms).

Independently of the technology in use, the communication is possible with the use of a defined protocol. Naturally, protocols have common properties . And in the same way, the protocol is sensitive to the characteristics of the communication.

Other different networking issues have been proposed for Grid Computing Infrastructures. Network topology, for example, depends of the distribution of the local platforms, that in effect corresponds to community members of the VOs. A description about this topic is presented by the Grid High Performance Networking Group of the Open Grid Forum [126], focused in the relationships between networking infrastructure and Grid Computing applications.

Clearly, communication trends for Grid computing infrastructures follow the same directions proposed for communications in distributed and large scale systems. Implementations and protocols in this direction, are located in specific layers to particular requirements or specific architectures. The interactions of the applications with the network infrastructure of the Grid computing system, are supported by protocols. And their performance is affected by the effectiveness of the relation between the protocols and the applications.

### 2.1.2 Technological Trends on Grid Computing Storage

Computer applications use, produce and transfer information. Information is digital data. Data storage in computer systems refers to computer components, elements, devices, and recording media that retain digital data used for computing, for limited or unlimited intervals of time. Obviously, the information must be *found* and *recovered*.

If we observe only the case of transfer, when the data transfer is delimited in the time that an application runs, even though the large capacity of components, elements, devices and recording media, the storage capacity is restricted to the virtual space size during the execution. Then, when a process is executing, the information quantity that is stored in the virtual address space is limited by the virtual space size. Clearly, this implies that the information must be stored after the process's end.

Moreover, in our context, some process can access simultaneously data, in accordance with their utility. To address these situation, the information stored in the components, elements, devices and recording media are saved in file systems [136] [137].

On the other hand, computer systems use different storage types. These storage types are organized in a storage hierarchy around the CPU, as a trade-off between performance and cost (it is to say the cost per bit). And the selected organization type affects the time that the data are accessed.

In Grid Computing systems, the storage can be described as a common file tree, shared by all machines. The access must occur with a minimal delay, also storage must support heterogeneity, aggregation of the unused storage space and coherence [143].

Evidently, the placement of data onto storage systems has a significant impact on the performance of scientific computations and on the reliability and availability of data sets [25]. This affirmation concerns storage systems in Grid Computing platforms also. As a consequence, the performance of the data transfer is sensitive to the data placement too. Data access depends to the characteristics of the file systems implemented [103] [53] in the Grid computing infrastructure. Then, it is necessary to propose data management mechanisms, integrate these mechanisms with the communication protocols and clearly, know data management behavior in order to predict their efficiency.

Different parallel file systems are implemented in Grid computing infrastructures. For example, Lustre [101], PVFS [120], NFSp [111], NFSg [143], dNFSp [82] between many others. In the same manner, mechanisms for data management [86], such as the schedulers [91], are proposed for the specialized community.

In all cases, each one of the file systems or data management mechanisms implemented in the Grid computing infrastructure affects the performance of the data transfer process. Then, it is necessary to quantify the impact of the file system in the data transfer process.

### 2.1.3 Impact

Technological opportunities in computing are exploited to implement efficient mechanisms in data management. In the case of the our problem, data transfer in HPC and Grid Computing implies a knowledge of the architectural features, obviously, related with communication, networking and storage.

The evolution of the computer systems needs different levels of abstraction and, as is explained in this part, the conception and description of these systems correspond in the same sense to technological trends. For example, clearly it is possible to see the similar features between the TCP model and the Layered Model proposed by OGSA.

On the other hand, as mentioned before, consumption and production of data involve data transfer and data storage. In consequence, it is very important to guarantee efficient data access. Different developments are proposed for architectural researchers, data management specialists and communication researchers with the objective to provide an efficient data access. These technological trends are implemented in HPC and Grid computing and shows how the different propositions to distributed systems or computer network has been use to made proposals in HPC and Grid Computing and identify each one of the possible architectures. For example, differences between Clusters and Grid Computing infrastructures from the interconnection point of view.

Computer applications each time are more complex and the volumes of information increase dramatically. And, the same evolution of HPC systems and their integration into more complex infrastructures, such as Grid computing platforms implies technology opportunities and scientific and industrial advances. Nonetheless, each new advance suggests a new use and the complexity (at same time that opportunities) grows. For example, due to these technological opportunities and latest needs, Grid computing environments should interact with others. Nowadays, they are projected to guarantee an interaction with another entities contained in Cloud Computing environments [62].

## 2.2 Modeling Data Transfer in HPC and Grid Computing

Computing intensive applications running on HPC platforms or Grid Computing infrastructures, rely on parallel algorithms expressed in a formal manner using models of parallel architectures. Algorithms are translated in a program language associated with a runtime commonly based on OGSA or Web Service Oriented-Architecture [13].

For example, many algorithms are used to treat scheduling problems. Several scheduling algorithms contain implementations of distributed data transfer models. Then, to treat

the scheduling problem, it exists specifications of the elements that affect the Grid computing scheduling. These specifications can be grouped in three categories, observing the close relationship between scheduling and communication.

First, the specification of the application goal, normally is described as computational tasks and data sets, as a general model. Second, the specification of the resources and their interconnection in a platform model as a network model. And third, the specification of the behavior in terms of predictability or cost, as a performance model.

For example, an application performs a computation for a specific problem in science. It requires the identification of the inputs and outputs to determinate the type of data that it uses. Between the input and output, it exists some compute processes related with tasks and compute jobs. This general abstraction of inputs, processes and outputs should describe a general model. In the same way, applications run on a determined infrastructure and use resources. In the case of a parallel or distributed application, it runs among *elements* interconnected between them, so, the specification of resources and how this interconnection is due is a network model. Finally to estimate their behavior and the cost of the resources use, it is necessary to describe the processes as function of the use, capacity and availability of the resources in the platform, as a performance model.

An interesting work about network modeling issues for Grid Computing applications scheduling [23] presents a extended description about these assumptions.

On the other hand, exists the problem of the *representation*[2] of distributed data, data transfer and data access of the process. Also taking in account the effect in the implementation of the models in algorithms. This subject is treated by specialists in languages for parallel and distributed computation. Propositions in this domain help to build implementations to schedule communication in parallel and distributed applications [128].

In synthesis, the implementations of models to handle communication scheduling in HPC and Grid Computing, are proposed following the application goals, the characteristics of the available platform and searching a maximal performance.

Various models, originally proposed for clusters, are extended to Grid platforms. Deterministic and non deterministic models appear and treat the problems with different degrees of accuracy and complexity.

The discussion about the use of deterministic or non deterministic strategies to modeling communication is long but can be limited to performance evaluation needs. In fact, if we make the hypothesis that the experiments and measures are performed in a continuous time, all the events have to be ”serialised”, not using randomness values.

Our main interest is to reduce the complexity with the use of the *easily* implementable models to performance evaluation requirements. Evidently, deterministic models make no use of stochastic mechanisms and the number of variables are limited and sorted to

[2] Or modeling, as a formal statement of features.

real measurable conditions.

For example, there are models as $PRAM$ [41], $BSP$ [144], $SPM$ [78], $LogP$ [28] between others, that aim of implement many communication algorithms or performance evaluation of communication on associated architectures with a deterministic way.

Some models have evolved in derived models to add new parameters to specify cases of implementation or cases of observation. For example for $PRAM$ exist derivate models as $XRAM$ [27], and in the case of $LogP$ many extensions exist that are treated with special attention here.

### 2.2.1 $PRAM$ Models

The $PRAM$ Model or *Parallel Random Access Machine* eliminates the focus on miscellaneous issues such as synchronization and communication to exploit mainly the concurrency.

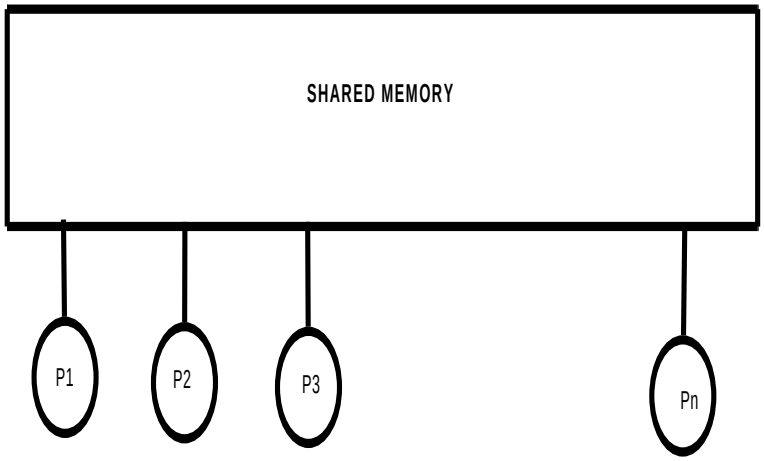


Figure 2.3: *PRAM Model*

The Figure 2.3 shows a picture of the general PRAM model. Each processor execute a step of the algorithm at the same time. There can be several restrictions on the processor access to memory and depending on the restrictions on memory access for data contention.

Due to this data contention, there are different PRAM Models:

- Exclusive Read, Exclusive Write (EREW): At each time step of the algorithm two processors cannot access the same memory location.

- Concurrent Read, Exclusive Write (CREW): At each time step of the algorithm two processors cannot write to the same memory location but several processors can be read from the same memory location.

- Concurrent Read, Concurrent Write (CRCW): At each time step of the algorithm, several processors can read from and write to the same memory location.

Measures of $PRAM$ model are associated with the technique use to implement the $PRAM$ algorithm. For example, if $T_p(n)$ denotes the time complexity of a $PRAM$ algorithm when it uses $p$ processors, and $W(n)$ the work complexity, then

$$T_p(n) \leq T(n) + \frac{W(n)}{p} \tag{2.1}$$

Where $T(n)$ is the time complexity of a $PRAM$ algorithm with as many processors as needed.

On the other hand, the work complexity is

$$W(n) = \sum_{i=1}^{T(n)} P_i \tag{2.2}$$

Where $P_i$ is the number of processors working in the $i - th$ iteration.

This basic information offered by $PRAM$ is not sufficient to analyze data transfer directly, same for homogeneous systems. In other side, $PRAM$ uses ignore important performance bottlenecks in modern parallel machines, because it assumes a single shared memory in which each processor can access any memory cell in unit time. Indeed, this point of view does not allows its implementation to Grid computing analysis.

The $PRAM$ Model is synchronous, and there are simultaneous access by multiple processors to the same location in shared memory. Obviously, Asynchronous $PRAM$ [6] differs of the $PRAM$ model, because the process runs asynchronously, then there is an explicit charge for synchronization and in consequence, there is a non-unit time cost to access the shared memory.

Other extensions of the $PRAM$ model has been proposed to treat characteristics such as memory contention and latency. In the case of memory contention many works propose a division of the memory into modules, each of which can process one access request at a time [83] [106]. In the latency case delay models, the delays between the time of production of the information by a processor and their use by another is measured as a communication latency [1] [118].

A tuned version of $XRAM$ model [46] allows to treat communication problems, as data-movement intensive problems [2]. Extensions of these $PRAM$ Models address communication problems, and they has been developed for specific cases. Some of these models are applied to scheduling programs with communication delays. An interesting work about it is possible to see in detail in [7].

Other different extensions are proposed to specific uses, specific architectures and specific implementations. But each extension is target to a specific architecture and does not allow a general approach. Studies about their efficiency, adaptability and implementations are made by different authors [50] [66] [92], they are not presented in this document, because is out of the scope of this thesis.

### 2.2.2 $BSP$ Models

The Bulk-Synchronous Parallel model $BSP$ [144] model suggests a parallel computer that consists into a set of processors with local memory, an interconnection mechanism that allows point-to-point communication, and a mechanism for barrier-style synchronizations.

The $BSP$ model appears like an unified model candidate for parallel computing. The $BSP$ properties allows an easy implementation and this is one of the most popular models.

The $BSP$ properties allow are easy design and write of parallel codes. Also, it should be close enough to physical reality that computer architecture can be designed in terms of efficient hardware. Finally, it should be mathematically tractable to aid in analysis of algorithms.

In mathematical terms, the $BSP$ model is described with three elements: processor/memory modules, an interconnection network and a synchronizer, as shown in Figure 2.4.

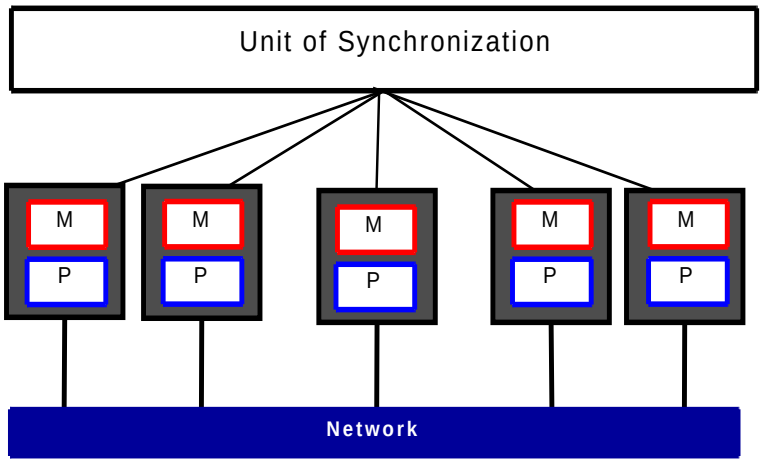


Figure 2.4: *BSP Model Architecture*

On the other hand, Figure 2.5 shows the model of execution for $BSP$. The green box represents the local computing on each processor, after a beginning of the *super-step i*. A $super - step$ is a sophisticated mathematical artifice to make a sequential composition of local computation and communication, limited by a barrier.

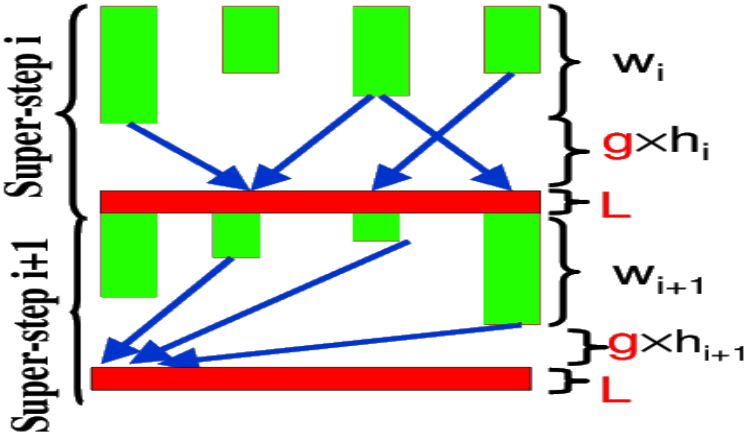


Figure 2.5: *BSP Model of Execution*

In the Figure 2.5, the blue arrows represents the global (collective) communication between processors, and the Global synchronization is the large red barrier. This global

synchronization is the exchanged of data available for the next super-step.

The nomenclature of $BSP$ as is shown in the Figures 2.4 and 2.5, presents $p$ as the number of processors, $L$ represents the global synchronization, $g$ is the phase of communication (1 word at most sent of received by each processor). Then, the cost of a super-step $i$ should be considered as

$$Cost(i) = (max_{0 \leq x=p} W_i^x) + h_i * g + L \tag{2.3}$$

Where p is the number of processors, $h_i$ represents the communication delays and $w$ the compute time. In general, this expression 2.3 says that the general cost of the program is the addition of the each super-step cost.

In terms of communication, $BSP$ considers that the communication is massive. This consideration makes possible to bound the time to deliver a whole set of data by considering all the communication actions of a super-step as a unit. However if the maximum number of incoming or outgoing messages per processors is $h$, then such a communication pattern is called an $h - relation$ [144].

About the $g$ parameter in $BSP$, this parameter measures the permeability of the network to continuous traffic addressed to uniformly random destinations. This situation is defined such that it takes time $hg$ to deliver an $h - relation$.

At this point, it is important to say that $BSP$ does not distinguish between sending 1 message of size $m$ or $m$ messages of size 1, then the general cost should be estimated with the simple relation $m * g * h$.

Using $PRAM$ model to describe $BSP$ model, this may be considered a general case of $PRAM$ [80], because if the $BSP$ architecture has a small value of the total number of local operations performed by all processors in one second over the total number of words delivered by the communications network in one second, then it can be regarded as $PRAM$.

Different implementations and studies have been developed by the $BSP$ community [19], studies as synchronization in parallel algorithms [52], communications in applications [51], specific architectures [69] between others.

The specific mathematical structures proposed for $BSP$ complicates sometimes the implementations. Another disadvantage is that in general for $BSP$, the data locality is good but the description of the network locality is bad, sometimes confusing.

### 2.2.3 $LogP$ Model

In the same way that $BSP$, $LogP$ [28] [29] is proposed to describe a machine-independent model for parallel computing, simple, for parallel machines interconnected by networks with limited bandwidth and significant latency.

The model is based on four parameters that represents the computing bandwidth, the communication bandwidth, the communication delay and the efficiency of coupling communication and computation. These parameters are *L* for latency, *o* for overhead time, *g* for gap between transfers and *P* for number of processors.

The general assumption of the $LogP$ [28] model proposes a transfer between two processors. When we send a consecutive set of messages between pairs of processors, it is possible to estimate the transfer time as a function of the time that takes the sender processor to send the message ($O_s$), the time of reception of the receiver processor ($O_r$), the time that the message takes in the channel ($L$) and the space of time between the consecutive messages (known as gap $g$).

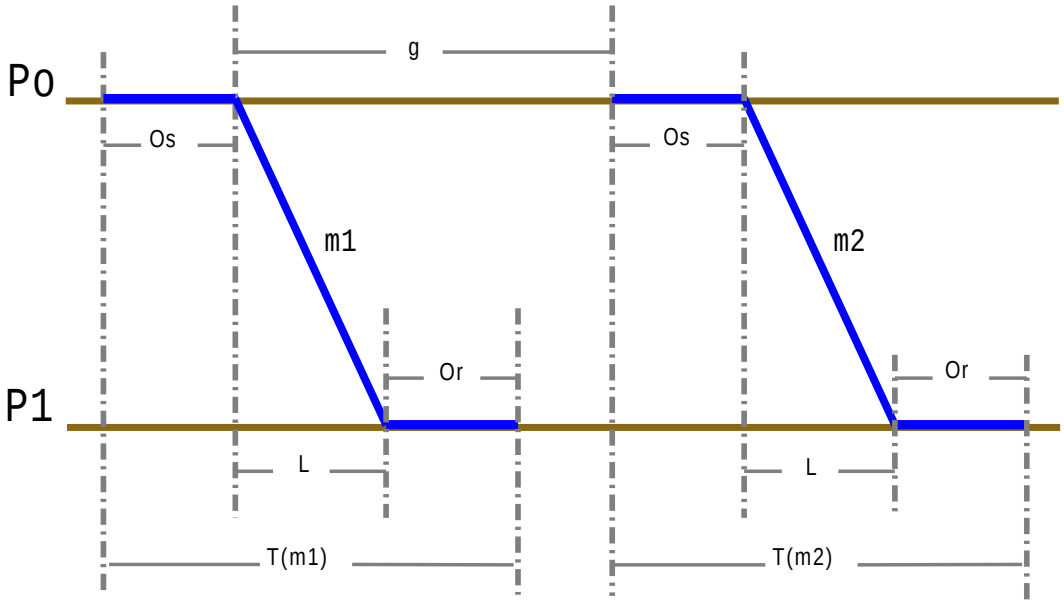


Figure 2.6: *LogP Basic Transfer*

Figure 2.6 shows the simple assumption of $LogP$ [28] for the transfer of two consecutive messages, $m_1$ and $m_2$ of small size, between two processors $P_0$ and $P_1$. The messages take a time to leave completely the sender processor ($P_0$). This time is known as *overhead send time* $O_s$.

In the same way, the messages take a time to be received completely by the receive processor ($P_1$). This time of reception is known as *overhead received time* $O_r$. In the same Figure 2.6 the communication latency is identifiable as $L$. Finally, the space of time between two messages consecutive is known as the *gap* by message $g$.

Then, in terms of $LogP$ model, an expression of time transfer is proposed, in the most simple case of transfer of a message $m$ between two processors $p$:

$$T(m) < o_s(m) + o_r(m) + L(m) + g(m) \tag{2.4}$$

Clearly, The first part of the expression is the general expression of transfer time $T(m) = o_s(m) + o_r(m) + L(m)$ for one message that can be reduced in $T(m) = 2O + L(m)$, where $O$ involves both sending and reception time. The addition of $g(m)$ corresponds to a transfer in continuous time proposed by $LogP$.

The objective to use these parameters proposed in $LogP$ model is the possibility to capture experimental values easily to describe the transfer behavior in terms of capacity, use and availability. This availability is clearly identified in the measures presented in this work.

However, the problem with $LogP$ is that it is proposed for describing transfers of small messages mainly[3]. To treat large message transfers, we consider the $LogGP$ model.

### 2.2.4 $LogGP$ Model

The possibilities of the $LogP$ framework are explored by several extensions. A first interesting extension is $LogGP$ [3] model that proposes the incorporation of large messages in the $LogP$ model. This incorporation allows to treat massive point to point communications and it should be addressed to different architectures.

Different algorithms for some uses has been developed using the $LogGP$ extension, mainly in message passing applications [70] [133]. The $LogGP$ model adds the parameter *G*, that is the gap by byte for a large message. This parameter allows to treat the bandwidth in the transfer of large messages.

Introducing a new hypothesis for the transfer of consecutive *large* messages in the same platform, the behavior of the communication process may be observed with the same methodology than for *short* messages. However, it is necessary to discuss two aspects: the underlying packet size of the platform and the definition of *large* message.

The underlying packet size, $w$, of the platform is the largest amount of data transferable in one segment of bytes by the platform. It exists different ways to estimate this capacity, for example, considering the communication protocol window scale of data (as the TCP window scale) or considering the maximum segment size of the buffer in a node (i. e. in a machine or network device).

Knowing the underlying packet size, $w$, it is possible to estimate if a message $m$ transferred is *large* or *short* in relation with the transfer capacity of the platform.

[3]In fact, the description initial of $LogP$ proposed by Culler et al [28] for the existing technology and type of transfers at that time, take into account only low bandwidth transfer.

A message $m$ is *short* if the quantity of bites contained is equal or lower than the underlying packet size. Hence, a $m$ message is *large* when the quantity of bytes contained is greater than the underlying packer size. In consequence, the message $m$ will be *cut* in $k$ packets to be transferred. Obviously, the number of $k$ packets may be estimated with $[\frac{m}{w}]$.

In our case, we suppose that if the message $m$ is much *larger* than the packet size $w$, it is to say $m >> w$[4], then, the use of $LogP$ model is not sufficient because it exists a time related with the transfer of each $k - packet$.

Taking the same transfer of two messages presented in the Figure 2.6 but this time, two *large* consecutive messages are transferred, Figure 2.7 appears showing the communication process between processors with the assumptions of the $LogGP$ model [3].

We can observe in the Figure 2.7 the transfer of two consecutive messages $m_1$ and $m_2$ between the processors $P_0$ to $P_1$. As the size $m$ is much larger than $w$, the message is divided into $k$ packets. Each one of the $k - packet$s is send to the receiving processor and between them exists a space of time known as *gap by byte for large messages* $G$.

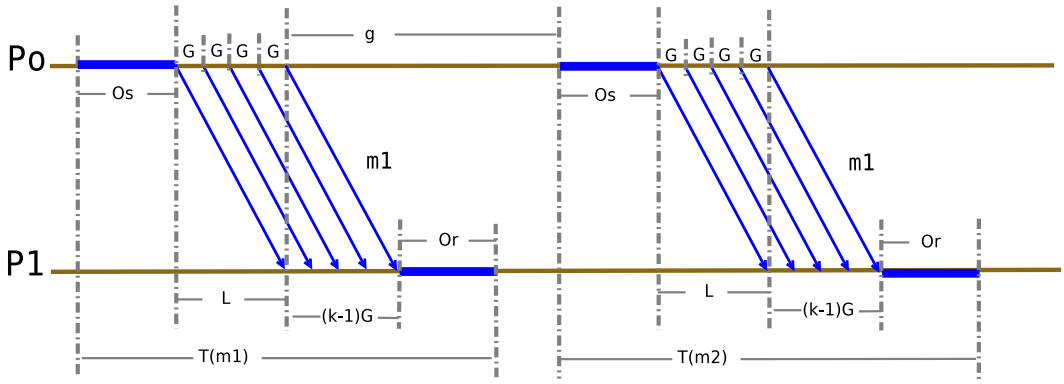


Figure 2.7: $LogGP$ *Basic Transfer*

Then, for a message transferred $m$, it is possible to present an expression of time transfer for a large message in terms of the $LogGP$ model:

$$T(m) = O_s(m) + L(m) + (k-1)G(m) + Or(m) \tag{2.5}$$

Where, $O_s(m)$ and $O_r(m)$ are the overhead send time and overhead received time. In opposition to the equation (2.4), the *"little"* $g$ does not appear in this equation (2.5) since

[4]This assumption is useful for our definition of high bandwidth transfer that it is treated later in this document.

we assume the transfer of a single message is divided in $k - packet$s.

The $LogGP$ model allows to solve some questions associated with our problematic. However, in this proposition some aspects are not treated, such as the parallel massive transfers among heterogeneous platforms (using heterogeneous networks), the capacity and availability predictions, between others.

### 2.2.5 Parametrized Modeling of $LogP$ from Measures

The *parameterized* $LogP$, $pLogP$ Model [87] appears for a practical use to cover measurement for messages of different sizes as slight extension of $LogP$ and $LogGP$. $pLogP$ uses the same parameters of $LogP$ as a function of the message size ($m$). $P$ is the number of processors, $L$ is the end-to-end latency (combining all contributing factors such as copying data to and from network interfaces and the transfer over the physical network), $O_s(m)$ is the send overhead, $O_r(m)$ is the receive overhead, and $g(m)$ is the gap.

The $pLogP$ model permits to propose a procedure to build measurement tools to know the behavior of a data transmission in terms of $LogP$ and $LogGP$ (eventually) parameters. To understand the model, we consider necessary to present a detailed description as follows.

The Figure 2.8 shows the message transmission as modeled by $pLogP$. The times for sending and received a message of size $m$ when both sender and receiver simultaneously start their operation are introduced like $s(m)$ and $r(m)$ respectively. The time at which the sender is ready to send the next message is denoted as $s(m) = g(m)$[5]. Every time the network itself is the transmission bottleneck, $O_s(m) < g(m)$, and the sender may continue computing after $O_s(m)$ time. Because the message uses the network, and obviously, the next message cannot be sent before $g(m)$. However, for sufficiently long messages, receiving may already start while the sender is still busy, so $O_s(m)$ and $O_r(m)$ may overlap [87].

[5]Because it is supposed that the space between consecutive messages is almost equal to the time to send the first message and to be ready to send a second message.

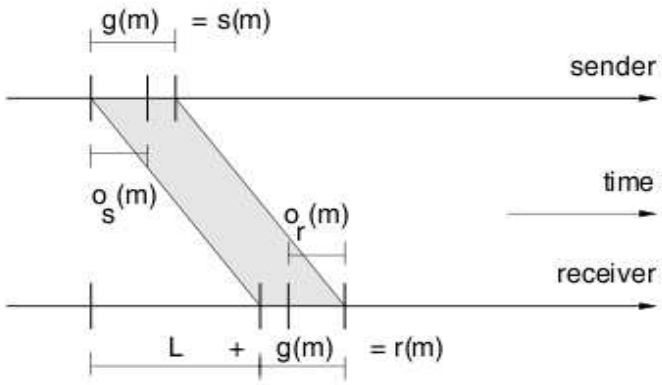


Figure 2.8: $pLogP$ *Message Transmission*

In the same Figure 2.8, $r(m) = L + g(m)$, is to say, the time at which the receiver has received the message. Latency $L$ is the time that it takes for the first bit of a message to travel from sender to receiver. The message $gap$, in consequence, adds the time after the first bit has been received until the last bit of the message has been received.

An important point of the use of $pLogP$ reported in Figure 2.8 is when the sender transmit several messages in a row, the latency will contribute only once to the receiver completion time but the gap values of all messages sum up, so $r(m_1, m_2, ....m_n) = L + g(m_1) + g(m_2) + ... + g(m_n)$.

The $pLogP$ model is a methodological proposition that provides mechanisms to capture the parameters of the $LogP$ model ($L$, $O$, $g$ and $P$) using a fast measurement procedure that is fundamentally a *Round Trip Transfer* ($RTT_n$) presented in the Figure 2.9.

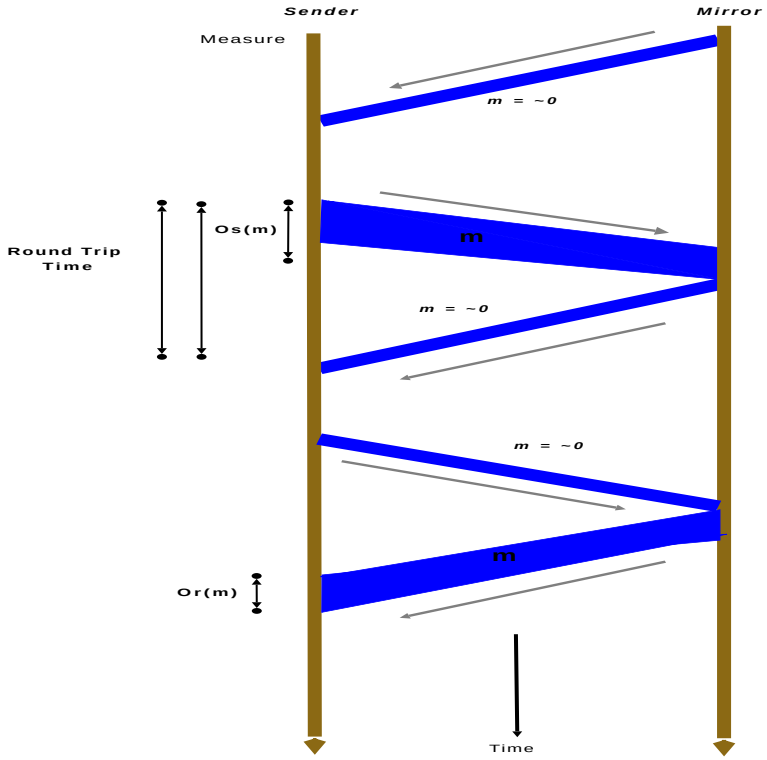


Figure 2.9: $pLogP$ *Fast Measurement Procedure*

As is described in [87], the $RTT_n$ consists in $n$ messages sent in a row by measure, and a single empty reply message that is sent-back by the mirror to sender. The procedure starts with $n = 10$. The procedure starts with a small number of messages in a row in order to speed up the measurement. The number of messages $n$ is doubled until the gap $g(m)$ per message changes only by = $\epsilon = 1\%$. Clearly, the saturation is assumed to be reached. The time measured for sending the so-far largest number of messages (without reply) is $n * g(0)$.

With this, it is possible to ensure that if the number of messages are sufficiently large then the round-trip time is dominated by bandwidth rather than latency.

The method takes advantage of the saturation to obtain $g(0)$. The $g(0)$ value is used for deriving other values, thus for this reason it is measured first. For each size $m$, two message *roundtrips* are necessary from *measure* to *mirror* and back (generally, $RTT(m) = RTT_1(m)$ is used). In the first roundtrip, measure sends an $m - bytes$ message and in turn receives a $zero - bytes$ message. The procedure measures the time for just sending and for the complete roundtrip. The send time directly yields $O_s(m)$.

In consequence, the latency $L$ is obtained with the Round-Trip Time (RTT) of an empty message ($sndcount = 0$). Thus, the latency can be measured as

$$Latency = \frac{RTT(0)}{2} \tag{2.6}$$

The same parameters of $LogP$ are assumed but with emphasis in the measures captured by the round-trip transfer (See Figure 2.6). For example, the latency $L$ that is the end-to-end latency from process to process, combining all contributing factors such as copying data to and from network interfaces and the transfer over the physical network in the platform. Then, following the procedure presented in [87], also the latency may be calculated with:

$$Latency = \frac{RTT(0) - 2g(0)}{2} \tag{2.7}$$

To calculate the values of the main parameters of the model, the latency $L$ and the gap $g(m)$ can be determined by solving the equations for $RTT(0)$ and $RTT(m)$ presented in [87], then, for example, to estimate $g(m)$ is necessary to resolve 2.8:

$$g(m) = RTT(m) - RTT(0) + g(0) \tag{2.8}$$

Where $RTT(m)$ and $RTT(0)$ are the first and second roundtrip time measure made to $zero - bytes$ and $m - bytes$ messages, in accordance with the procedure described in the Figure 2.9[6].

The second roundtrip, the *measure* processor (or *sender* processor) sends a $zero - bytes$ message, waits for $\Delta > RTT(m)$ time and then receives a $m - bytes$ message. Measuring the receive operation now yields $O_r(m)$, because after waiting $\Delta > RTT$ time, the message from *mirror* is available at measure immediately, without further waiting.

If we suppose that among the transmission, the reception may already starts while the sender is still busy, then $O_s$ and $O_r$ may overlap. In the same way that $L$, $g$ cover all contributing factors.

To extend these assumptions to a transfer of a message sufficiently large, $g(m)$ of $LogP$ is used to cover $O_s$ and $O_r$, then $g(m) \geq O_s$ and $g(m) \geq O_r$. More details about this process are discussed in [87].

Table 2.1 shows the possible parameters measured by $pLogP$ in terms of $LogP$ and also in terms of $LogGP$ [87]. This table shows how the $pLogP$ model includes the originals $LogP$ and $LogGP$ models. Obviously, the parameters presented in terms of

[6]Observing that $RTT(0) = 2 * (L + g(0))$. and $RTT(m) = L + g(m) + L + g(0)$.

$LogGP$ are applied to the transfer of *large* messages. The authors of $pLogP$ determine that with the use of $m = 1byte$, it is possible to represents any other short message, and in consequence the transfer in function of $LogP$.

| LogP/LogGP | pLogP Parameters |
|---|---|
| L | $L + g(m) - (O_s(m) + O_r(m))$ |
| O | $\frac{O_s(m)+O_r(m)}{2}$ |
| g | $g(1)$ |
| G | $\frac{g(m)}{m}$, but only if $m$ is sufficiently large |
| P | $P$ |

Table 2.1: $LogP$ and $LogGP$ parameters in terms of $pLogP$.

Then, a network $N$ involved in a transfer of a message of size $m$ may be characterized by $N = (L, O_s, O_r, g, G, P)$, where the use of $g$ or $G$ depends of the size of message $m$.

The method has limitations that are explained in detail in [87]. However, they are many interesting limitations. First, the procedure enforces that pipelines will always be drained between individual message pairs, assuming that message headers carry on the back of the another message. Then, the flow control information that reset senders to their initial state after each message roundtrip. This assumption does not works for all protocols but, is used by the authors because it works in TCP and Myrinet. In any case, we have retain the method because our experiences are in TCP and similar protocols also.

Another limitation is the extreme symmetrical assumption of the network. This situation is not always true, for example, on wide area networks as Grid computing networks, where the achievable bandwidth or the network latency may be different, due to possible asymmetric routing behavior or link speed. Furthermore if the machines used to measure and mirror processes are different (like in hierarchical infrastructures, such as fast and a slow local platforms on Grid computing infrastructures), then the overhead for sending and receiving may depend on the direction in which the message is sent.

However, the methodological process given by the $pLogP$ model is used by our approach. The abstraction of the data transfer proposed by the model is simple and allows to reduce the complexity of the process, same for the case of the parallel and massive data transfer among wide area networks or on Grid computing environments but with a reinterpretation of the parameters $o$, $g$ and $P$.

In synthesis, the $pLogP$ parameters associated with the network characteristics remains but a derivation of them is necessary to explain other characteristics and behaviors. For example, the utilisation or availability of the links that interconnect the processors

or the overhead time when the transfer process implies an exchange between different environments. These and other aspects are analyzed and treated by our proposition.

### 2.2.6 Other $LogP$ Extensions

Since 1993, when the original proposition was presented, $LogP$ evolved in different propositions that added specific parameters for specific needs. The framework supplies by $LogP$, allows a simplistic modelling and address its use to specific behaviors and architectures. The $LogP$ model in its original version has been interesting to analyze problems such as the broadcasting and summation [84], scheduling [153], optimization [5] [33] , implementations on specific architectures [32] [89], implementations on specific networks [47] [94], message passing applications [68] and so on.

As $LogGP$ or $pLogP$ shown above, another extension inspired by $LogP$ is the $LoPC$ model [40], it adds the *C* parameter to treat the contention cost for message in parallel algorithms on a multiprocessor or network of workstations. Network contention is also treated by an extension named $LoGPC$ [108], specifically in message-passing programs.

On other hand, to analyze synchronization costs, the $LogGPS$ [75] model is proposed more like an extension of $LogGP$ that $LogP$. In this model, a parameter *S* appears to define the threshold for message length, above which synchronous message are sent.

Heterogeneous $LogGP$ or $HLogGP$ [17] merges like a proposition to take in account the heterogeneity in parallel platforms. In this model, the $LogGP$ parameters are organized in arrays: a latency matrix that corresponds to the latency between heterogeneous pairs, a gap *g* and overhead *o* vectors that correspond to respective gap by bite and overhead cost in message transfers, a Gap *G* matrix that associate the transfer of large messages between heterogeneous nodes, and the *P* processors vector that represents the vector of computational power in the platform. The most interesting of this model is that it does not add a new parameter but changes the treatment of the $LogGP$ parameters.

Without adding other parameters, parametrized $LogP$ or $pLogP$ [87] introduces a methodology to fast measurement of $LogP$ and $LogGP$ parameters to minimize intrusiveness cost and completion time in measurement techniques. This methodology allows to build monitors and benchmarking tools for performance evaluation goals. The methodology is presented in the section 2.2.5.

In addition with performance evaluation uses, $log_nP$ Model extension [22] appears to provide analysis techniques to predict communication cost in distributed systems.

Other interesting extensions of $LogP$ model involves hierarchical memory models, as is the case of the $LogP-HMM$ and $LogP-UMH$ Models [95]. These models allows to capture network communication costs and the effects of multilevel memory, such as local cache and I/O.

Finally, models like $LogGP$ [3] and $LogGPC$ [108] considers others factors, associated with the saturation of the network and the contention. Also, models like $LogGPS$ [75] or $HLogGP$ [17] considers factors like the synchronization in message-passing programs (collective-communications) and the heterogeneity in clusters respectively.

Each derivation of the $LogP$ model adds a new parameter in behalf to explain a specific behavior. But, each *specific* model excludes the parameters of the other ones.

### 2.2.7 Other Models

Different approaches exist to model data transfer or communication process in parallel and distributed platforms. The degree of complexity and their accuracy are engaged in each implementation of the model, in behalf of the user needs.

In the last sections are presented the most interesting models in accordance with our needs, related with the contribution of this work. However, it is important to mention other models that have different approach.

One of the first model proposed to communication modeling is the $Hockney$ model [67]. This model uses only the latency and the bandwidth to calculate the communication time in function of the message size. It is a very simple model and normally used to describe simple communications.

The $Postal$ Model [8] [9] is other basic model that aims to get mechanisms to build broadcasting algorithms in networks with specific properties.

Other models are the specific models for $MPI$ collective communications. These models uses mechanisms proposed by models as $LogP$, $BSP$ or $PRAM$ and search tuning algorithms to broadcast collective communication algorithms [15] [104] [142].

Several sophisticated models appears to analyze other characteristics, as fault tolerance, bandwidth sharing and bandwidth loss, packet-delay, between others. These models uses techniques such as stochastic methods, game theory approaches, quantum computing and so on.

The degree of complexity grows with these models and their implementation or use in practical solutions is really difficult or only limited to simulation environments or specific-restricted behaviors. For this reason, these models are not considered into our proposal.

## 2.3 Discussion

The description of the data transfer process (like any phenomena described in scientific terms) requires the development of abstractions to model their behavior. The abstractions are formalized in mathematical models that allow the formal description not only of

the architectural characteristics, also the implementation of algorithms to communication protocols, transfer strategies in applications and the interaction application-infrastructure.

Each one of the presented models has specific characteristics and they are proposed to describe and predict behaviors in particular conditions and architectures. These architectures have technological characteristics and they have an impact in the applications that run on its and that transferring data. So, observing the technological trends in the platform and in accord with our objectives, the models have been compared to find the best suited to massive high bandwidth data transfers.

Comparing the general proposition of $LogP$ with $BSP$, $LogP$ is similar that $BSP$. The main difference is that $BSP$ does not use the overheads and adds barriers in the communication process[7].

A comparison between $LogP$, $PRAM$ and $BSP$ in terms of accuracy, shows that the accuracy of $BSP$ in comparison with $LogP$ and $PRAM$ is similar[8]. The discussion between them is about the implementation of the models, sometimes complex.

For example, $BSP$ model demands the use of sophisticated mathematical mechanisms, as is the case of the *supersteps*. On the other hand, practical aspects like locality, shows that this in $BSP$ is critical.

Table 2.2 presents a comparison between some of the model presents in this part of the thesis document. The models are $PRAM$, $BSP$, $LogP$, $LogGP$ and in a general manner *stochastic models*. The target characteristics are the accuracy, complexity, cost (this cost represents the add values of the model in their implementation) and the main feature.

| Model/Characteristics | Accuracy | Complexity | Cost | Feature |
|---|---|---|---|---|
| **PRAM** | Medium | Medium | Low | Homogeneous Architecture |
| **BSP** | High | Medium | High | Demands sophisticated maths |
| **LogP** | Medium | Low | Low | Only for short messages |
| **LogGP** | Medium | Low | Low | Suited for large messages |
| **Stochastic Models** | High | High | High | Implementation is complex |

Table 2.2: Comparison between Models

In the Table 2.2; a high accuracy represents a best approach in the description of the behavior treated by the model. The accuracy of several models are determined by the characteristics to observe, in this case, related with our problematic.

[7] It is possible to see different formal comparison proposed by some authors that have published results about [16] [122] [121] [147].

[8] Sometimes, in $BSP$ the degree of accuracy can be lower when it does not do low level processing [16]

A low complexity for us, represents a better possibility to implement the model into a technique (in addition with a measure tool). Normally, the low complexity is related with a low cost.

## 2.4 Conclusion

Observing the comparison between models presented in the Table 2.2, in accordance with our objectives, we find that $LogP$ and $LogGP$ are more interesting than the others. In the case of the $LogP$ model, their approach is interesting for us, because the parameters offers a specific description of network and communication characteristics simple. Theses possible descriptions should be treated to analyze behavior and performance easily, in consequence, the use of the model allows to build performance evaluation tools. Indeed, we have selected the specific extension for our study that considers large message transfer that subsumes the $LogP$ and $LogGP$ models: the $pLogP$ model.

The parameters of $LogP$ and $LogGP$ models included in $pLogP$ model capture the relevant aspects of massive and parallel data transfer that can be used in distributed architectures. The latency $L$, gaps $g$ and $G$, overheads $O_s$ and $O_r$ provide information about the behavior of the system during the data transfer process, in terms of availability, capacity and use, in function of the number of processors $P$ involved. Using these parameters associated with network entities we can estimated the bandwidth in the shared resources and analyze different factors, such as the data transfer cost for example.

Of course, other models provide parameters associated to network characteristics also, but in some cases, the number of parameters is expensive, then, the implementation of the model begins very complex, and for performance evaluation needs, adds several cost such as intrusiveness cost. A comparison about this is presented in the Table 2.2.

Nevertheless, modeling process remains difficult, despite the use of parameters to build ”parameterized models”. They allow simplify the modeling. Parameters can be assigned to specific characteristics and provides an easy identification of their values. In the case of the transfer of great volumes of data in parallel, parameterized models provide mechanisms to measure performance for the set of the resources implied in the transfer. Moreover, for distributed systems parameters should be associated to one isolate resource.

# 3. Modeling and Measuring Parallel and Massive Data Transfer with $pLogP$

Modeling is important to implement performance evaluation techniques. Actually, these techniques can be gathered in systematic approaches that defines many techniques and methodologies to specific observations and standards [77] [90] [96] [130]. In any case, these techniques starts from theoretical bases.

We have presented the theoretical assumptions of the $LogP$ model and their extensions to describe data transfer. The utility of the $LogP$ model family is the possibility to identify and capture the relevant aspects of message transfer in parallel and distributed architectures. For each new aspect to observe, authors use more or less $LogP$ parameters.

As is shown before, the $LogGP$ model adds a parameter $G$ for modeling the gap per byte for long messages. This parameter, as the others $L$, $g$ and $o$ depend on the message size $m$. In the same sense, for practical uses, we have described the parameterized $LogP$ or $pLogP$ model in the section 2.2.5, that provides mechanisms to fast measurement of the $LogP$ parameters.

In this chapter, we presents the practical possibilities of $pLogP$ Model. It allows in principle, the measurement of data transfers inside clusters or between multiple clusters connected via wide-area networks, such as the case of a Grid computing infrastructure. The early results presented here shows the methodological utility of the $pLogP$ proposal but also the limitations in the case of the massive and parallel data transfer on Grid computing infrastructures.

## 3.1 Measurement Methodology

Experimentation implies a measurement methodology that has a close relation with the platform testbed. Measurement methodology is proposed in agreement with the propositions of $pLogP$ methodology, and the used techniques aims to measure the specific characteristics involved in both $LogP$ and $LogGP$ models.

Characteristics of Cluster and Grid Computing infrastructures are different. Each experience or observation needs to be attributed with timing information and an historic description of the environment, indicating when and in which conditions a certain observation has been made.

The measurement efficiency provided by the $pLogP$ model, reduces the cost of measurement due to intrusiveness and the technique should be implemented easily in runtime tests. On the other hand, $pLogP$ methodology can be implemented on real systems, because it takes in account the changes in the network behavior during the application runtime.

### 3.1.1 Tests Description

The fundamental $pLogP$ test consists in simultaneous transfer of fixed messages of $m$ bytes between a given number of processors $p$. The fixed messages are transferred with the use of a benchmark tool and a simple application to send/receive messages. The objective of this basic test is to capture the parameters of $pLogP$ during a parallel data transfer among a limited number of processors.

The test is made using a main benchmark tool. The used benchmark tool is a modification of the *LogP MPI Benchmark Multitest Tool* [10] [11] [87], that is part of the MagPIE suite [88] [87] proposed for the Albatross Project [6] in Netherlands. The original modification has been proposed by Luiz Angelo Barchet-Steffanel in the context of LaPIe project [10].

The modification made to *LogP MPI Benchmark Multitest Tool*, aims to permit the parallel and massive send and reception of the messages between different platforms. The working principle is the same than the original benchmark tool. We send simultaneous messages of size $m$ among the links between two different nodes. Each node represents an end component only[1]. Each one of the messages uses a link to be transferred. Evidently, the test measurements corresponds to the $LogP$ parameters in the same sense that the original tool: overhead times (reception and send), latency and gap in the runtime.

Each link is filled with a large size message $m$ and, since it is a parallel transfer of data, the total amount of transferred bytes $M$ in the test is expressed:

$$M = m * \frac{p}{2} \tag{3.1}$$

Where $p$ is the number of processors involved in the transfer, always between pairs. In tests, the size of the messages $m$ transferred between $p$ processors is increased, from

[1]For our beings, this node is mainly a set of processor-memory-network card. Actually, the nodes has multiple cores (multiple processors) but for our tests, only one core by node is selected.

$1MB$ to $250MB$. Also, the number of pairs of processors involved is increased. In consequence, the total quantity of transferred bytes growths and the network is quickly saturated.

For example, following the equation 3.1, if a message transfer of $m = 50MB$ among $p = 10$ processors occurs, the total quantity of bytes transferred will be $M = 250MB$.

Of course, the relation between pairs of processors and links between them suggest an other analysis. This aspect will discussed later.

The tests are repeated several times to capture sufficient values of gap ($g(m)$), overhead times ($O_s(m), O_r(m)$) and latency ($L(m)$).

For example, a simple example of a line script to show the execution of the *LogP MPI Benchmark Multitest Tool* with the last characteristics ($m = 50MB$ and $p = 10$) is:

```
mpirun -np 10 -machinefile maqsgrid logp_multitest  -min-size 5242880 -max-size 5242880 -o OutputFile
```

The first part of the script calls the execution of the code `logp_multitest` invoking the `mpirun` environment for $10$ processors. Each processor corresponds to a machine/node and are recorded in the `maqsgrid` file. The second part defines de size of the message and the output file.

An output file is produced for each pair of processors, for example, for the last script, a typical output is:

```
# LogP network performance data: logp_test.Send.Recv.8.9
# Latency =   64.50
# time      bytes os        os_min    os_cnfint or        or_min    or_cnfint g
1158932612       0 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0002319
1158932613 52428800 16.1575521 16.1328125 0.0858335 14.6132812 14.5976562 0.0513423 16.2431790
```

In the output example, we can see that there are three comment lines. The first commented line presents the performance type of measure, in this case for the execution of the code $logp\_multitest$, between the nodes 8 and 9. Second commented line presents the latency measure at moment at moment of tests. The last commented line describe the measures taken. In the case of the column the first column shows the time in the clock system during the test. The second column shows the quantity of bytes transferred by row (here $50MB$ are $52428800$. The next columns present the values of $O_s$, $O_r$ and $g$. Observing the results, the first measures results are the measures for the first transfer ($m = 0$) and the second measures line provides the measures for the second transfer ($m = 50MB$ in this case). The values represented as $os\_min$ and $or\_min$ are the minimal values for $o_s$ and $o_r$ respectively. The values with the extension $\_cnfint$ indicate the confidence interval difference (by default the confidence interval is $\approx 90\%$)

The test protocol can be summarized as follows:

- Objective: Capture of the transfer parameters (Latency, gap, overhead times) associated with network characteristics (Bandwidth, Transfer Cost), to measure the performance and observe the behavior of a massive data transfer, to describe and predict performance of this type of transfer..

- Used Method: Execution of a benchmark tool to make a massive data transfer between selected pairs of a specific platform testbed. The platform is not isolated and it is in a normal activity. Various sizes of messages are transferred among various nodes on the platform. In fact, the tests are classified in accordance with the size of message transferred among a link.

- Analysis: The different measures are analyzed to observe the capacity and use of the resources and make a relation with the network characteristics and the data transfer behavior.

This test protocol is used to obtain the early results presented above and the other results presented later in this document.

### 3.1.2 Platform Description

Almost all tests were done on the Grid5000 [55] infrastructure, the French experimental grid platform for research in large-scale architectures, high performance computing and grid computing.

The platform involves 9 sites geographically distributed in France and some interaction with international projects, as the Das-3 infrastructure in Netherlands [30], Naregi Project in Japan [110] and the UFRGS supercomputing infrastructure in Porto Alegre, Brazil [26].

The Grid'5000 platform is highly reconfigurable, *controlable* and *monitorable*. Which permits to design, develop and follow different experiences in different conditions.

The experiments made on Grid'5000 involve software layers between the network protocols up to the applications.

Figure 3.1 shows the infrastructure and network characteristics of Grid 5000 including two of the international interconnections: the connection with the NAREGI project and the DAS-3 infrastructure.

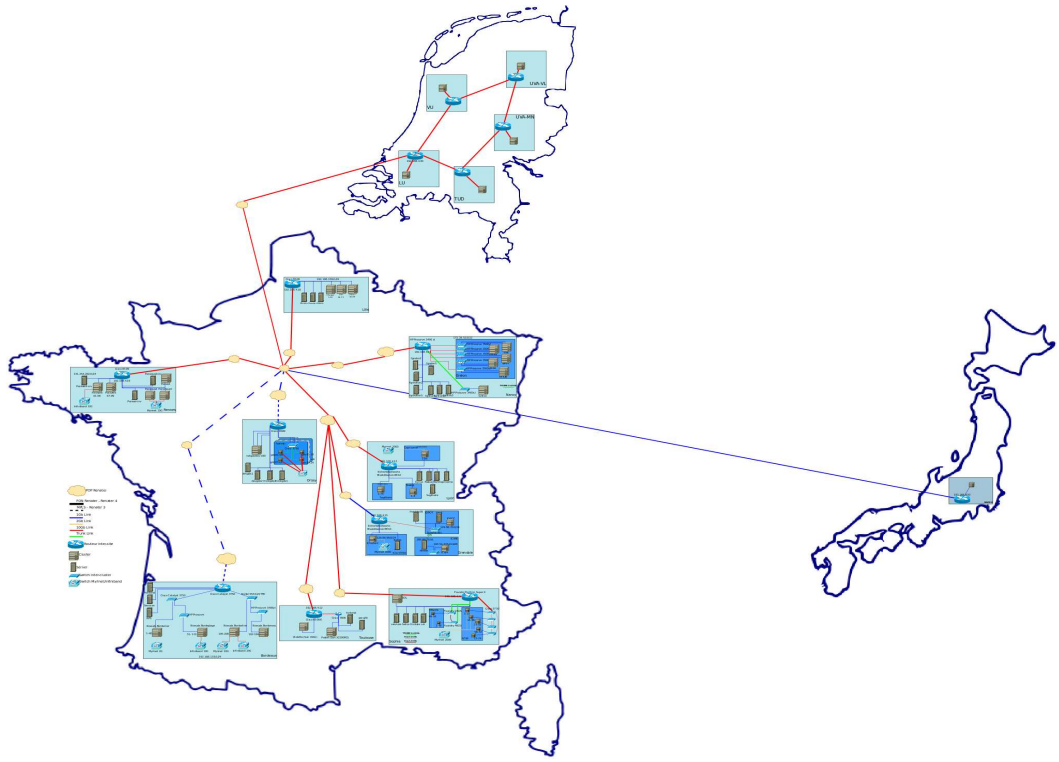

Figure 3.1: *Infrastructure and Network Characteristics of Grid 5000 with External Sites*

In the Figure 3.1, the clouds represents the Point of Presence (POPs). The huge black lines represents the dark fiber connection and the non continuous lines are the Multi protocol Label Switching (MPLS) connections. Blue lines are the 1 Gb/s links between some POPs and local platforms as is the case of Grenoble and the external link to Japan. Yellow lines are 2 Gb/s links and red lines are the 10 Gb/s links. Trunk links, that usually interconnect switches are represented with green lines.

Also, Figure 3.1 shows internal characteristics of the local platforms. The blue pies represents the routers inter site. Each site has at least one cluster with a high performance configuration. For example, Figure 3.1 shows the different clusters that can be in one place.

Clusters are represented in the Figure 3.1 with gray array boxes (or racks) and servers by long gray boxes. Switches are represented by simple blue boxes in the case of inter-cluster switches and composed blue boxes in the cases of Myrinet or Infiniband switches.

Grid'5000 backbone network infrastructure is provided by *RENATER* [124], the French National Telecommunication Network for Technology, Education and Research.

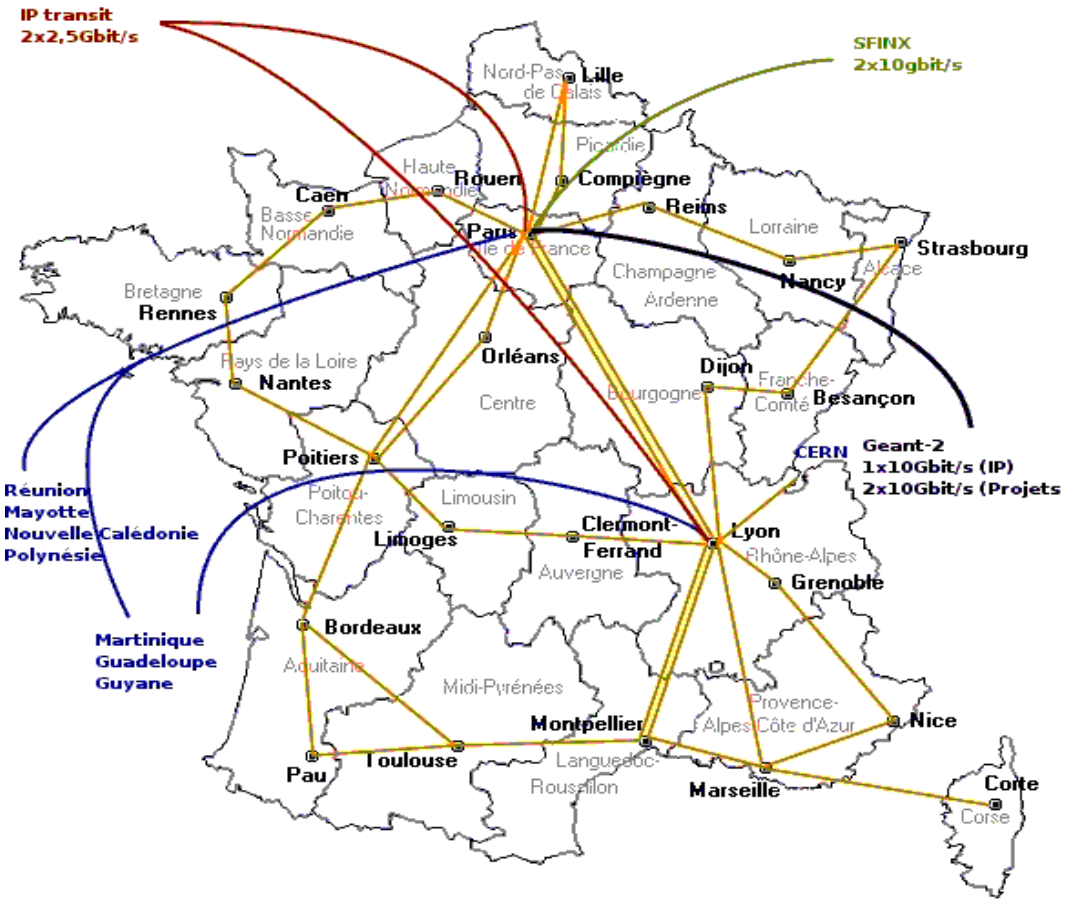


Figure 3.2: *Network Backbone of Grid 5000*

The RENATER network offers about 30 Points of Presence (POPs) in France, at least one POP for each region on, which metropolitan and regional networks are connected on. More than 600 sites (Universities, Research Centers, government entities, etc.) are interconnected through this network. Between these more 600 sites are the sites that are involved in Grid 5000 infrastructure. Figure 3.2 shows the Backbone of Grid 5000 provides for RENATER.

The *standard* architecture is based on 2,5 Gbit/s leased lines and provides IP transit connectivity, interconnection with GEANT-2[2], overseas territories and the SFINX (Global Internet exchange), as shown in the Figure 3.2.

[2]http://www.geant2.net/

Nowadays, Renater network is in a phase named RENATER-4. RENATER-4 introduces a dark fibre infrastructure allowing to allocate dedicated 10 Gbit/s ”lambdas” for specific research projects. It also provides interconnection with GEANT-2 and it greatly increases the capacity compared with the standard network. The Grid’5000 sites will see each others inside this VLAN (10Gbit/s).

Inside every Grid’5000 site has a specific network hardware and configuration for each local platform. Normally, they are different and may be contain high speed network technologies, such as Myrinet and Infiniband network infrastructures.

Grid 5000 is a heterogeneous platform. However it exist identical configurations in the local platforms. For example, the GDX cluster nodes are 186 IBM eServer 326m machine nodes with 2x AMD Opteron 246 (2GHz, x86-64) processors and a memory ram of 2GB DDR RAM for node is used. Each GDX node has a storage capacity of 80GB SATA Hard Disk and a Gigabit Ethernet network interface.

In the case of Grillon cluster nodes are 186 IBM eServer 326m machine nodes with 2x AMD Opteron 246 (2GHz, x86-64) processors and a memory ram of 2GB DDR RAM for node is used. Each node has a storage capacity of 80GB SATA Hard Disk and a Gigabit Ethernet network interface. All infrastructures characteristics can be consulted in the Grid 5000 project site [55].

As is says before, the set of nodes of each cluster during the tests are booked but not isolated at the moment of the experiments.

## 3.2 Early Results

First observations with the methodology described, show the utility of the $pLogP$ proposed as a technique to fast measurement in data transfer. These observations were done by specific tests explained before and their outcomes are of the tests realised by us.

The results presented here, correspond to different experiments transferring messages of $m$ size between $1KB$ to $50MB$. In other words, the quantity of bytes transferred corresponds to transfers of message of *small* sizes to *large* sizes.

On another hand, the number of processors is increase in pairs, from $2$ to $100$ processors. Processors are in two clusters of the Grid’5000 platform. These two clusters are in the Grenoble site: IDPOT and I-Cluster 2.

IDPOT [3] is an experimental Beowulf cluster, with 48 nodes Bi-Xeon Dell 1600SC 2.5 GHz processors, 1.5 Gb. RAM ECC and a Gigabit network (Fast Ethernet).

I-Cluster 2[4] platform contains Itanium-2 technology. This cluster is made of 104

[3] At moment of this writing process, IDPOT not is available to tests.

[4] At moment of this writing process, I-Cluster 2 is dismantle to build a more powerful cluster: Genepi.

nodes Itanium-2 with 64 bits, 900 MHz and 3.0 Gb of RAM, a Myrinet and a Fast Ethernet network. The system provides a total of 208 processors and 312 Gb of RAM memory and a disk capacity of 7.5 Tb. During the tests only has been used the Fast Ethernet network.

### 3.2.1 Cluster Transfer Measurements

Observing the gap in the transfer cluster measurements, the increase of the values each time that the number of nodes involved in the test grows is evident. The observed behavior is easy to describe in terms of the $LogP$.

Figure 3.3 shows the measures of the $pLogP$ parameters $g$ (blue line), $O_s$ (cyan line) and $O_r$ (red line) for a transfer of messages of $1KB$ between determined quantity of nodes.

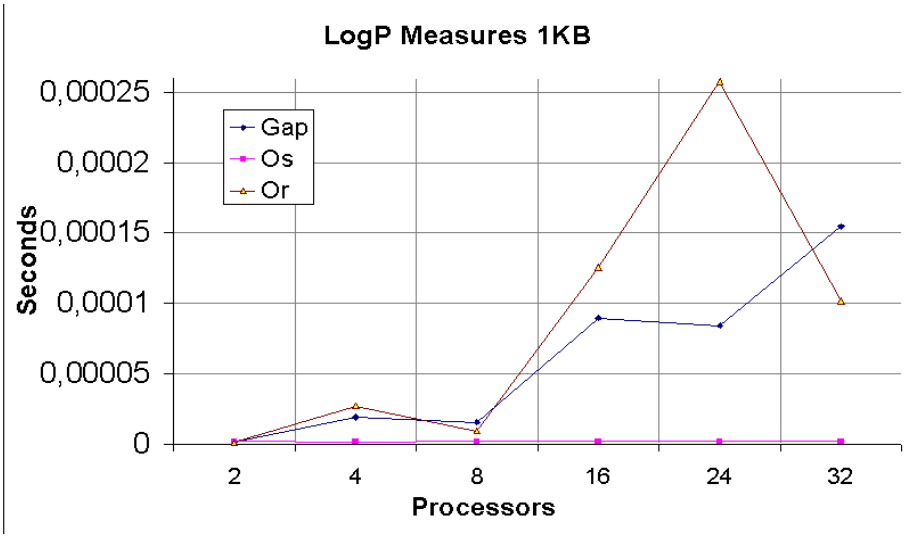


Figure 3.3: $pLogP$ *measurements in IDPOT Cluster - 1KB Transfer*

The low values observed in the different measures are obvious. The growth in the values is due to the increase of the number of nodes that suggest a saturation in the network. Theoretically, the values of the gap and overheads must present a slight increase.

For this transfer of $m = 1KB$, the values of the latency are not shown, because the values are really small ($\approx 0$) and remain stable.

Advancing to the observations for long messages, the Figure 3.4 shows the same measures, but for a transfer of $1MB$. The values of the gap presents in this case corresponds to the gap $G$ for long messages. Evidently, the values of the different parameters $g$ (blue

line), $O_s$ (cyan line) and $O_r$ (red line) of the $LogP$ model are highest for this transfer than for the transfer of $1KB$ (In fact, a message of $1MB$ not is a small message really). The growth in all measures is due to the increase of the number of nodes too.

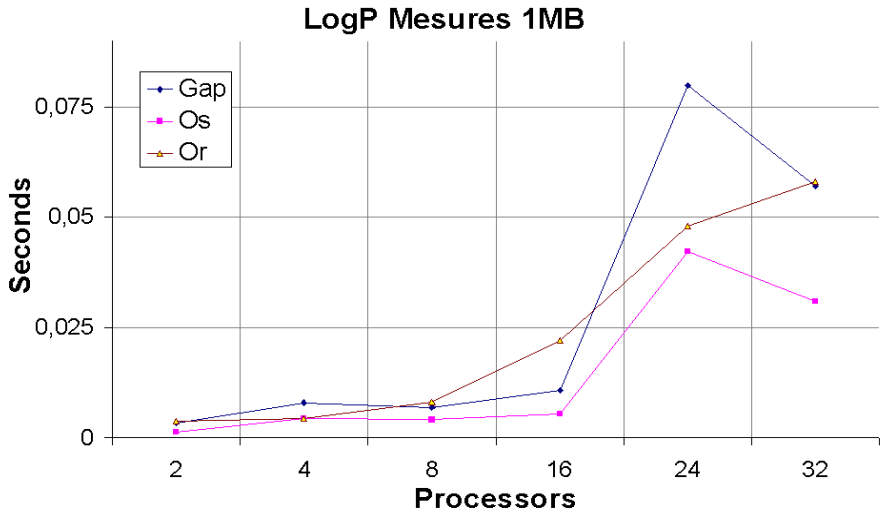


Figure 3.4: $pLogP$ *measurements in IDPOT Cluster - 1MB Transfer*

Contrary to the observations when $m = 1KB$, these observations shows the equivalence in the curves of the $O_s$ and $O_r$ measures. However the values of the $O_s$ remains lower than $O_r$ values. The overhead associated with the messages shipment reaches similar values to the overhead reception because the increase of the quantity of bytes transferred affects the deliverance time to the transmission channel. In other words, the time that exists between the first byte and the last byte that leave the source is important, and adds a significant cost. Same for the reception time due to the increase of the quantity of bytes in the message.

Following with the test to observe what happens with other transfers of more large messages using this technique, the size of the messages is increased to $10MB$, $50MB$ and $100MB$. For these cases also, the values of the gap corresponds to the gap for long messages.

At this point, a specific analysis is necessary to understand the behavior and explain the observations. The assumptions of $pLogP$ or the other extensions can be used but in any case they add a complexity degree important in the analysis.

The behavior begins to be unstable when the quantity of bytes transferred reaches a saturation of the resources, in a first approach saturating of the links between the proces-

sors.

Taking the gap like reference measure to see the time between two consecutive messages, it is possible to observe how the increase in the values of the $g$ parameter is important.

A first test presented in the Figure 3.5 shows a comparison between the gap observed for two transfers: $1MB$ (red line) and $10MB$ (blue line).


Figure 3.5: *Observed gap in ID Cluster for Long Messages*

The curve of $10MB$ has important quantity and quality differences with the curve of $1MB$, due mainly to the increase of the bytes transferred and their use of the shared resources. On another hand, the measure of this gap is taken between two consecutive messages of $1MB$ and $10MB$, then the delay of time that finish to go the last byte of the first message and the first byte of the second message is important, and affects this measure.

Tests in other platforms confirm these assumptions. Figure 3.6 shows a comparison of the gap measured during the transfer of messages of size $10MB$ (red line), $50MB$ (green line) and $100MB$ (blue line) in Icluster-2 platform.

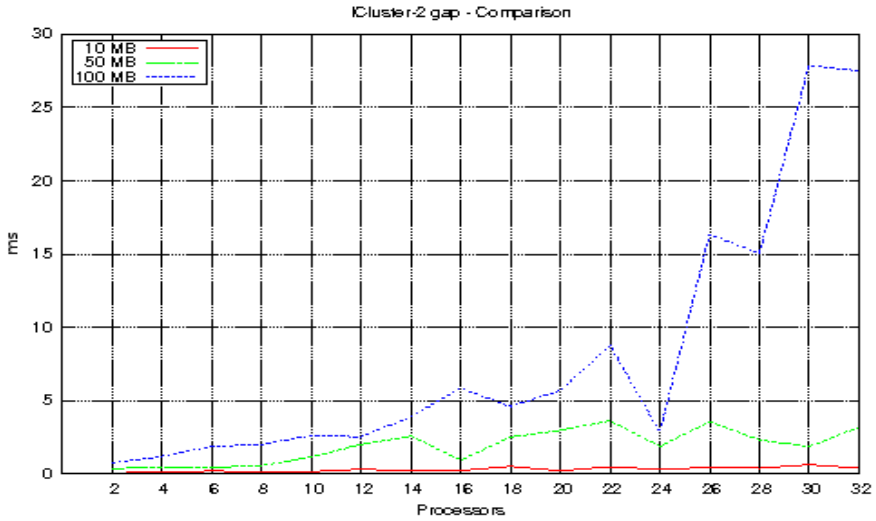


Figure 3.6: *Observed gap in ICluster-2 Cluster for Great Messages*

In this Figure 3.6, is observable the increase of the gap time in function of the size of message. And also, is notable the increase of gap values when the quantity of nodes that are involved in the transfer grows.

The latency normally is considered in one-way, and provides information about the time from the start of message transmission to the start of message reception. In other words, the latency measures allows to know the delay between the initiation of a network transmission by a sender and the receipt of that transmission by a receiver. At this point, it is important not to confuse the latency with the *throughput*. The throughput is the number of messages successfully delivered per unit time.

Figure 3.7 shows the latency comparison between three different transfers in ICluster-2 platform. Transfer of messages of sizes $m = 1MB$ (red line), $10MB$ (green line) and $50MB$ (blue line). The measures of the latency are in microseconds and the transfers are between 2 to 32 processors respectively.

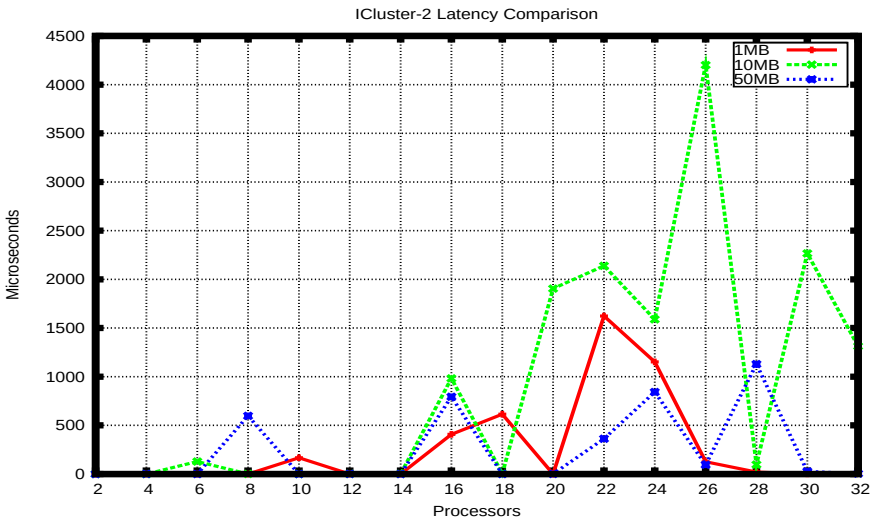


Figure 3.7: *ICluster-2 Latency Comparison*

As is presented before, the latency values represented in the Figure 3.7, the delay of time that takes a message among the effective link between a pairs. In accord with the $LogP$ hypothesis, the latency is measured since the moment when the first byte of the message is leaving the sender and start its ”trip” among the link to the moment until the last byte arrives to the receiver and completes the transmission. Of course, at moment that more processors are implied in the transmission, the network may be saturated. Then, the latency is increased.

On the other hand, due to the growths of the number of bytes, the message is cut in $k-packets$, then it exists an additional time delay that corresponds to gap by $k-package$. Then, the latency is affected.

The important differences observed in the measures, suggest an influence not only in the total quantity of bytes transferred between nodes, but also in the use of the links between these nodes. In consequence, for the transfers of messages of large size it is necessary to propose an analysis that take into account different factors related with the quick saturation of resources due to the high quantity of bytes involved.

The set of measures made for the cluster transfer shows the pertinence of the $pLogP$ model descriptors and their methodology for fast measurement of the parallel and massive data transfer. However, the information provided in terms of processors and the increase of the gap values for the really large message suggest that is important to analyze the behavior taking into account network characteristics such as the number of links used between the processors, the bandwidth generated by each message transfer and the per-

formance of the switch (and other eventual resources involved during the data transfer).

### 3.2.2 Grid Transfer Measurements

When the transfer of a *small* message occurs between distributed platforms, this process implies a *small* occupation of the resources and in addition, the behavior remains stable with little values of gap time.

The management of data is optimal because the shared resources are not saturated. Observations for these *little* transfers shows, in the same way that for the cluster observations, the adequate use of the $LogP$ assumptions.

Following the same technique proposed in $pLogP$, Figure 3.8 presents the different values of gap measured for different transfers between two clusters in Grid'5000 platform: GDX and Grillon cluster.

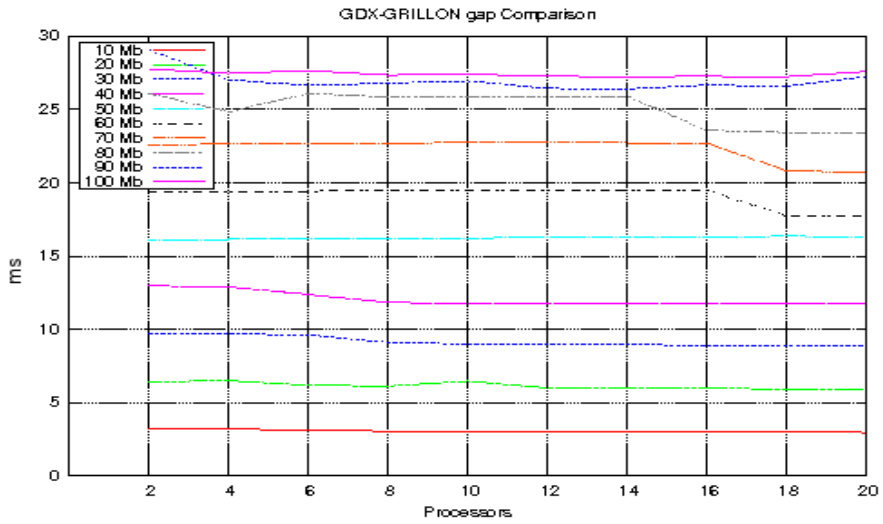


Figure 3.8: *Observed gap in GDX-Grillon Clusters Transfer for Many Messages*

In the Figure 3.8, we observe on increase in the gap time related with the amount of bytes that are transferred in the Grid communication process between the two clusters. The small sizes of message (under $10MB$) are omitted in this figure.

The different measures remains stable in some values, with precise disturbances, due mainly to the *real* use or *on-live* state of the Grid platform.

The observation of the gap, as a reference measure is interesting because the gap is the related reciprocal inverse value to the bandwidth. Then, the gap provides information

about the *delay* in the transfer among the link due to the division of the message in *packets* of a determined size. Clearly, the values of the gap remains *stable* for each one of the measures, change in its value are correlated to the size of bytes transferred by message.

Although this behavior is expected, it is necessary to observe the phenomenon on the links and on the switch. The relation between pairs and their description is not sufficient specially when the amount of data transferred is considerable.

In all cases, $LogP$ characteristics are adapted to describe the behavior, but do not resolve questions that are normally treated with the model extensions of $LogP$, increasing the degree of complexity.

On another hand, if the behavior looks stable for each one of the size of messages transferred, some questions like the influence of the devices, the intra-cluster communication in the general transfer or the real use influence in the behavior, need specific analysis and other assumptions.

Figure 3.9 shows the irregular behavior of the latencies in the tests, for transfer of $m = 10MB$ (red line), $30MB$ (blue line) and $100MB$(cyan line).

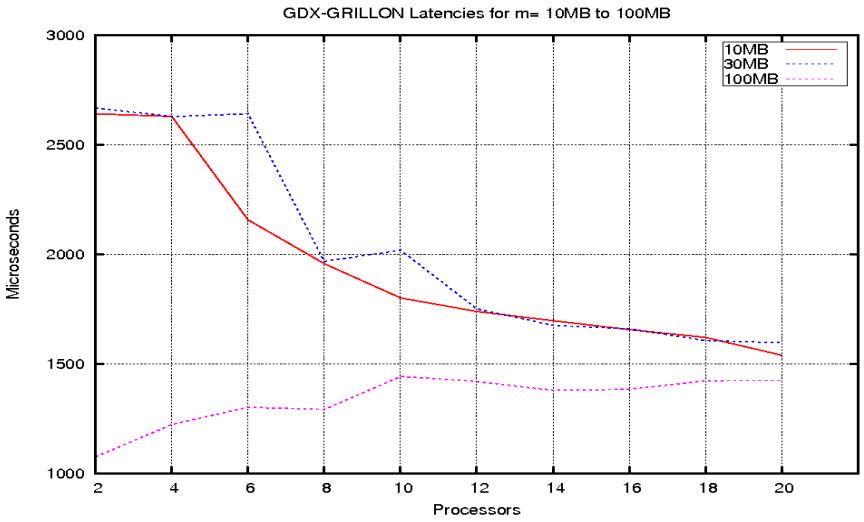


Figure 3.9: *Observed Latencies in GDX-Grillon Clusters Transfer for Many Messages*

The interested point of this figure, is how the values of the latency fall to reach a stable value that remains, for the lines of $10MB$ and $30MB$. However, is not the case for $100MB$, that start in low values and after growths to reach the same latency value of the other transfers. This situation suggest that it exists an influence of the network activity of

the link to cause a delay in the latency. Then, it is necessary to interpret the behaviors in terms of links.

As is presented by Kielmamn et al in [87], the method has some limitations. A first one, the difficult to measure the gap $g$ values in the case of large messages. To handle this, is necessary to calculate the relation between $G$ and $g$ proposed by $LogGP$. This relation adds an error in the estimation of the delay by byte for the long messages.

On the other hand, the measures sometimes show that the $g(m) < O_r(m)$, which is contradictory to the parameterized $LogP$ assumptions. This situation is caused by the variations in the behavior of the message reception that depends on whether the incoming message is expected to arrive [87].

It is important to say that the measurement procedure described above assumes that the network links are symmetrical, and this situation is not present in some wide area networks, such a Grid Computing network. Then, the achievable bandwidth and their associated gap may be different in both directions, the network latency too. So, a reinterpretation of the parameters and is necessary a limitation of the test procedure to handle this.

## 3.3 Discussion

The early results presented in this chapter, shows how the procedure proposes for $pLogP$, captures the relevant parameters associated with characteristics of message transfer between processors in distributed architectures, in terms of $LogP$ model and the slight extension of $LogGP$ model to handle large messages transfer.

These parameters are the latency $L$ that provides information about the use of the network, the $g(m)$ gap that should be associated with the bandwidth and provides information about the capacity of the network, the $O$ overhead send and receive times and the number of processors $P$ involved in the process.

The latency $L$ depends more on the distance between nodes and the number of links used than on the quantity of bytes transferred. For this reason the values of the latency remain stable for the tests set. It is not the case for the gap , it evolves as a function of the quantity of bytes transferred as it is presented in the last early results. Of course, due to the relation between the gap and the bandwidth by link, the bandwidth by link is affected too. However, the values of the bandwidth in the switch, evolves as a function of the number of links during the transfer.

Early tests show the advantages of $pLogP$ uses. And as it limitations in the description of the phenomenon observed when the number of bytes transferred by messages grows considerably. For this, the mechanisms of capture have been modified in the main tool

used for the tests, but the limitation in the description of the behavior, justify to add or reinterpret the parameters of $LogP$ and $LogGP$ use in $pLogP$ technique.

On the other hand, a perspective of evaluation on the platform testbed is necessary. In our problematic, the main interest is to observe and predict the data transfer behavior in Grid computing environments. Observations in clusters, contribute to understand one of the levels of the data transfer on the Grid platform. In other words, for us the cluster platform is a local component of a large infrastructure, the Grid infrastructure.

The set of equations presented in the section 2.2.5 allows to analyze and predict the behavior of the data transfer. These equations are implemented in the benchmark tool and used for the respective analysis. However in terms of utility, the early results presented offer more a procedure to fast measurement of $LogP$ parameters than a predictive approach.

Obviously, a systematic approach is necessary to define the levels of the data transfer, and identify the resources used during the data transfer. Actually, in the case of the $LogP$ proposition, the extensions add parameters associated with new characteristics. But, each new aggregation implies a new degree of complexity. And in consequence, a new degree of complexity does not allow an easy implementation of a model in a technique for performance evaluation during a real use.

Inside clusters or a homogeneous HPC platforms, some network performance characteristics remain stable, as is the case of the overhead times. Other variations of the network performance characteristics depend on the number of links or quantity of bytes transferred by message[5]. For example, we saw in the early tests presented before, the important variations of the bandwidth by link are due to the increase of the quantity of bytes by message transferred, and the latency variations due to the increase of the number of links. However, when the data transfer involves heterogeneous systems, wide area networks, and also massive and parallel data transfer, the network performance characteristics may be affected by several factors and other network performance characteristics, such as the overhead times and requires a reinterpretation.

## 3.4 Conclusion

In behalf of the pertinence of the test procedure provided by $pLogP$, as is said before, it is necessary to propose a reinterpretation of the parameters of $LogP$ to give specific information for our beings and to permit an analysis of the transfer characteristics in terms of network characteristics. This information is related with the growing of the quantity

[5]Indeed they are variations due to the performance of the network protocol (in this case TCP), as is observed for the non linear behavior of the gap and overhead measures in relation with the size of message.

of bytes during a massive data transfer, the increase of the links between processors used, the behavior exchange between the possible network levels on a Grid Computing infrastructure, the sensibility of the file system and their influence during the data transfer and so on. These information that are not given for $pLogP$ clearly.

In practical terms, the time of intrusiveness and completion due to the use of the monitoring and test tools derived of the model, are an important factor, because the objective is to implement performance evaluation mechanisms during a real use of production Grid computing platforms.

# 4. Contributions to Modeling of Parallel and Massive Data Transfer on Grid Computing

## 4.1 Introduction

Grid Computing platforms are scalable distributed systems. The applications that run in Grid computing platforms, often consume and produce large amount of data, this implies massive and intensive data transfer among the Grid computing network.

Even though that technological challenges can give more capacity and more speed of transfer, the efficiency of the interaction between the deployed applications and the physical resources in the platform is crucial for an appropriate use of Grid Computing technology (and related e-science emergent technologies). The transfer occurs to carry the requirements of use of the application and the availability and/or capacity of the network resources. The communication process in a Grid computing platform involves sharing network resources between users, applications and services.

Globally, the resources of a Grid computing platform are heterogeneous (this is a main feature of a Grid Computing infrastructure that was mentioned repeatedly throughout this document). The distributed remote local platforms are different, with heterogeneous nodes and the network can dynamically change in topology and also in technology (i.e. for suppression or addition of devices due to availability or due to technological evolution). All changes among the Grid Computing platform affect the behavior of the system, and in consequence, the performance of the Grid Computing applications.

The characterization of the communication behavior is crucial to anticipate the applications performance. Different scenarios are possible because a change of a resource or a untimely and dynamic action during application runtime could affect the transfer and make difficult the prediction of the behavior.

To handle these, various strategies are investigated, from the analysis of technological

characteristics of the physical resources, to the use of formal expressions of communication process or modelling algorithms to implement protocols. Previously, some of them were presented and discussed (Mainly in the Section 2.2).

As said before, the description of transfer characteristics is proposed in terms of parallel architectures. This description enables to build the algorithms, to implement them in applications or protocols associated with a runtime, commonly based on $OGSA$ [43]. Moreover, several models to describe the behavior or to predict the performance are proposed from different points of view. Depending of the abstraction levels of the models, they are used in the implementation of monitoring tools and benchmarking applications or they are used to modify and to implement algorithms to build transfer protocols or data storage/management services in scalable architectures.

The model proposal of this thesis work is addressed to the parallel data transfer of great volumes of data, using $pLogP$ assumptions, that in the same time, uses the hypothesis of $LogP$ and $LogGP$ models. We profit the simplicity of the abstractions and the fast measurement methodology given by $pLogP$.

Our approach aims to treat this parallel and massive data transfer between heterogeneous nodes, on a distributed systems context, specifically in Grid computing environments. Consequently, we have heterogeneous resources, each local platform is a $HPC$ infrastructure (specifically they are clusters) and is necessary a definition of hierarchical levels of observation (Cluster level, Grid Computing level).

The heterogeneity of the resources requires that we consider different factors that affects the parallel and massive data transfer: the file system influence, the transfer costs associated with use of the platform, the capacity of the shared resources , etc. Furthermore, we have the principal assumptions that the applications run in concurrence and that is necessary to make the performance evaluation in real time.

At this point, we aim to make a minimal and implementable model to analyze and evaluate the behavior of the communication process in high bandwidth networks on Grid Computing infrastructures during real time. Consequently, it will possible to predict or tuning applications running on these platforms. To satisfy, a reinterpretation of the original parameters of $LogP$ and $pLogP$ is made to create a parameters reduction.

The communication process between two nodes of a Grid Computing platform is described taking into account the network parameters strongly associated with the infrastructure characteristics. This description is observed in the studied models. Then, observed the network characteristics, we propose different hypothesis.

A first assumption of our contribution analysis is that the data transfer occurs in parallel (in a synchronized manner or not) between nodes in HPC or Grid Computing platforms. Then, is necessary to consider the busy resources during the transfer and the consequences in the general capacity of the platform and application performance.

In our context, we use the time to measure the *transfer time*[1]. This is the *estimated time for the completion of a data transmission*. The transfer time is sensitive to specific characteristics related with the *environment* where occurs the data transmission (channels, links, technical features, distance), state of this environment (availability, capacity, use), entities implied in the transmission (senders, receivers, nodes), policies of the transmission (synchronization, non-synchronization, flooding) and obviously, the data (type, size).

The different possibilities of the relation between these characteristics, elements, entities, states, politics and data allow the description of the behavior in terms of latency, bandwidth, traffic, cost and so on.

In this chapter, we present our contribution to modeling of parallel and massive data transfer on Grid computing in the following sections: section 4.2 presents the definition of the massive and intensive data transfer, section 4.3 treats the data transfer cost, the overhead times analysis is presented in the section 4.4, the file systems influence is treated in the section 4.5, the section 4.6 handle a specific case of transfer due to low capacity performance of the network and finally the section 4.7 presents a general conclusion of our contributions.

## 4.2 Massive and Intensive Data Transfer

The $LogGP$ [3] model says that a large message $m$ transferred between a pairs of nodes or processors is divided in $k - packets$, where each *gap by* $k - packet$ may be understood as $(k-1) * G(m)$. Then, taking into account this assumption of the $LogGP$ model, we propose a parameter of the *gap* captured between the $k - packets$ transferred, as:

$$\hat{g} = (k-1) * G(m) \tag{4.1}$$

As the bandwidth is a rate of data transfer, measured in the time, is to say $B = \frac{m}{T}$, using the expression (2.5), the bandwidth estimated is:

$$B = \frac{m}{o_s(m) + (k-1) * G(m) + L + o_r(m)} \tag{4.2}$$

Now, if we suppose that a message of size $m$ is very large ($m \approx \infty$), the overheads $O_s$ and $O_r$ are negligible in relation with the latency $L$, because the link between the

[1]The first measure to characterize a transfer is the time. The time is a fundamental quantity used in sequential events, to compare the durations of events and the intervals between them and allows to define other quantities, such as delay, velocity, etc.

processors is saturated, the message is cut in accordance with the underlying capacity transfer of the link, then the message transfer occurs by packets, and it exists a delay time of the transmission, so, the latency with the $(k-1)*G(m)$ values the expression 4.2 becomes:

$$B = \frac{m}{(k-1)*G(m)+L} = \frac{m}{\hat{g}(m)+L} \tag{4.3}$$

Presently, as the delay time of the transmission is dominated by the gap observed. Then, the high values of the $\hat{g}$ will be more important that the latency existent during the transmission, so, the equation 4.3 becomes:

$$B = \frac{m}{\hat{g}(m)} \tag{4.4}$$

Essentially, the expressions (4.2), (4.3) and (4.4) represent the *gap* as the reciprocal value of the end to end bandwidth from processor to processor. This relation between the gap and the bandwidth is a principle of the $LogP$ model.

On the other hand, a preoccupation of the massive and intensive data transfer analysis are the consequences of the parallel data transfer between nodes. Parallel transfers saturate the network quickly, which is common in HPC and Grid Computing applications. Also parallel transfer may be synchronous or asynchronous.

Then, to make an analysis of the interaction between the nodes, is proposed the observation of the data transfer in terms of links between them. For the side of the requirement to implement parallelism and synchronism (or asynchronism) in the data transfer, is proposed the use of barriers.

A barrier defines a start process to send the messages and also a count of the reception time of the parallel and simultaneous transfer between the nodes. A barrier can be used to synchronize all processes in a communicator an it is easy to implement. Barrier implementations exist into $MPI$, $PVM$, $OpenMP$ and others well known libraries.

With the hypothesis of the transfer process between pairs of $p$ processors using one link by pairs, named $W_p$, each $W_p$ transmits the same quantity $m$ of bytes contained in the $m$ message [2]. Obviously, the quantity of bytes contained in the message $m$ is really larger than the underlying transfer capacity of the link $w$, ($m >> w$), then it exits a certain number of $k-bytes$ for each $k-packet$ one generated. On the other hand, the total quantity of $M$ bytes transferred among $W_p$ links in a platform, may be calculated with

$$M = W_p * m \tag{4.5}$$

Obviously, $W_p$ corresponds to the number of links used during the parallel transfer.

[2] The number $W_p$ is estimated by $\frac{p}{2}$.

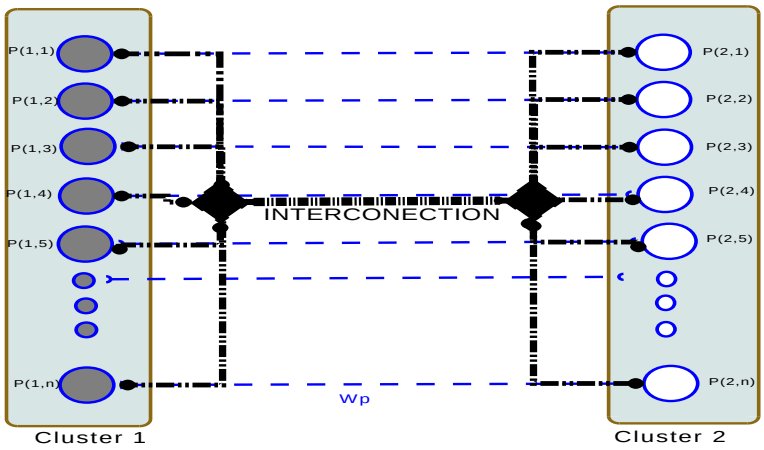


Figure 4.1: *Transmission among $W_p$ links*

Figure 4.1 represents a parallel and simultaneous transmission among $W_p$ links between nodes. The set of nodes are in two local platforms, each one is a cluster. The connections are concentrated a connection point, represented in the Figure 4.1 by the black diamonds (in the real world by a router or switch). For example, the most simple case is an abstraction with direct connections, shown in the figure as blue non consecutive lines.

Remember the important assumption presented before related with the size of the message $m$, then, if the message $m$ is much larger than the capacity $w$, the message is divided in $k - packets$ that are transferred consecutively; two possible assumptions come: first, the large size $m$ of the message by links $W_p$ and the addition of each one of the transfers when a general $M$ quantity of bytes transferred represents a *massive* data transfer. Second, the cut or division of the $m$ message in $k - packages$ following the consecutive transmission represents an *intensive* data transfer.

Practically, the *intensive data transmission* does not depends of the size of the packages transferred because this type of transfer corresponds to successive requests between the sender and receiver nodes.

Then, the simultaneous transfer of message of large size produce a utilisation of the network resources with the attributes of intensive and massive transmissions (increasing the number of transfers). Likewise, the capacity of the network resources is easily saturated (connection points, interconnection links, etc.). This situation suggests a high bandwidth.

The term *high bandwidth* is frequently used to distinguish faster broadband connections from traditional or lower connections. However, this term corresponds to an idea or

concept more than a formal definition due to the technological trends.

Taking the expressions (4.2) to (4.4), the bandwidth is expressed as a function of the size of message $m$ and parameters of transfer related with the time. Then, a *high bandwidth* connection should be with a great capacity of transmission of data sets.

Observing the equation (4.2), we consider parameters associated with the availability of the resources, for example as is the case of the latency $L$. Also, we observe other parameters associated with the data transfer management, as it is the case, for example, of the $O_s$ and $O_r$. So, with the hypothesis that $m$ is sufficiently larger to gather the $O_s(m)$ and $O_r(m)$ times into $g(m)$, the *high bandwidth* transfer is guaranteed by a low latency, is to say a great availability of transference.

Other interesting situation is the possibility to estimate a *worst bandwidth* using only the space of time between the $k - packages$, it is to say the $\hat{g}$.

Then, from a measurement procedure to capture the parameters, we can represent a worst bandwidth as:

$$B_{worst} = \frac{m}{max(\hat{g})} \tag{4.6}$$

Where $max(\hat{g})$ is the maximal gap measured causing jams of the transmission (or faults due to the saturation among the transfer).

Other capacity measure, due to the possible connections in the network is the sum of each bandwidth by link. This sum of the bandwidth by link, observing all possible connections on the network, is known as the *bisection bandwidth*. For example, if we have a linear topology with a maximum of the shortest path between a given pair of nodes or diameter $d = n - 1$ the bisection bandwidth is $B = 1$. In the case of a Torus ring, the diameter is $d = \frac{n}{2}$ then the bisection bandwidth is $B = 2$. Other possible connections are explained in [63].

Then, at this point, we have analyzed the massive and parallel transfer in terms of capacity of the links, mainly used the estimation of the bandwidth. However is necessary to know how much is the cost of the data transfer to know the efficiency of the transmission in accordance with the quantity of bytes transferred. This transfer cost analysis is treated in the next section.

## 4.3 Transfer Cost

Transfer cost analysis is necessary to know the efficiency of a communication depending of the amount of bytes transferred. These measures of the efficiency should predict the performance of a implemented algorithm on a determined application. Of course, the influence of the network Quality of Service (QoS) [21] is important.

The type of transfer that is described in the past sections, implies the use of collective communications[3]. Collective communications transmit data among all processes in a group specified by an *intracommunicator object*. The barrier, serves to synchronize processes without passing data. Several libraries that implement message passing, such as $MPI$, provides functions to collective communications [129].

Collective communications are important in the development of parallel programs that have running on parallel machines or scalable platforms. These collective communications provides operators to conduct all to all nodes communication and on mesh networks. Also, collective communication operators provides mechanisms to guarantee synchronization and minimize the steps and conflicts among the transfer between network interfaces.

A way to analyze the transfer cost of these type of communication is observing the behavior of the completion time of the transmission [15]. Clearly, the completion time of a transmission is the time between the start of the transfer process from the first processor and the end of the transfer process in the last processor involved. Thus, if all processors start the collective operation approximately at the same time, the completion time of the transmission is affected mainly by the runtime structure of the communication algorithm. This assumption allows, in principle, to eliminate external factors for the analysis. However, this implies to observe the latency.

Latency not depends on the size of message transferred but we needs to observe the impact of the size of the message transferred to analyze the transfer cost. As we assume that a message of large size is transferred, the transmission time is affected by the overhead time and in consequence, is measured to consider the size of message in this estimation.

Actually, latency is considered almost larger than the gap. Thus, to reinterpret the overhead time it is necessary to see the latency like a maximum delay of time to transfer the message and to observe the possible bottlenecks presented in the I/O bus. This situation increases the amount of time where a node is busy during the transmission.

Then at this point, we have two situations: one, when the overhead is negligible in comparison with the latency (as is presented before) and two, when the overhead is considered in relation with the latency (for example due to the environment exchange[4]).

To simplify the analysis of transfer cost, we take the parameter $\lambda$, that is a relation between the latency $L$ and the our gap $\hat{g}$. Then, it is possible to distinguish three cases:

- In a first case, as it says above, the values of the latency are larger than the values of the gap, then for a message $m$ transferred, $\lambda$ is

[3]Collective communications are implemented to optimize communication within different clusters into a Grid platform.

[4]The estimation of the overhead due to the environment exchange is treated in the next section 4.4.

$$\lambda_{(m)} = \frac{L}{\hat{g}} \approx 1 \tag{4.7}$$

Equation (4.7) suggests that the *busy* time during the transmission is not important. It is to say, the time to send and the time to receive the message by a node, are small in relation with the gap between packets and the latency.

In other words, as the latency and gap values are similar, the cost of transmission is low, because the transmission delay, described by the latency is similar than the couped cost, described by the gap.

- Another case is when values $\lambda_{(m)} \ll 1$. We can suppose that the transfer cost is high because the gap values are significants in relation with the effective transmission delay determined by the latency $L$.

- Finally, in the case of the values of $\lambda \gg 1$, the capacity of the network is exceeded and this has a direct of the bandwidth. Then, these relative *high values* of the latency implies a high cost of the transfer.

To rule out a general expression, let us consider the overhead time in the transfer, merged into the equation (4.8). This expression is proposed in [15], but we present this in terms of $\hat{g}$ at the time that each communication takes in the $LogP$ model.

$$\lambda_{(m)} = \frac{2 * O + L}{\hat{g}} \tag{4.8}$$

Equation (4.8) presents $O$ as the overhead time, $L$ the latency observed during the transfer and $\hat{g}$ the measured gap.

Analyzing the possible values of $\lambda$, if $\lambda_{(m)} \ll 1$ the assumptions presented before may be extended for this expression. However, in the case of $\lambda \gg 1$, the high values of $\lambda$ may be caused by the busy time in the network devices, mainly in the sending and receiving time. For example, due to a saturation in the I/O bus.

On the other hand if we suppose an effective delivery during the sending or receiving process, the cost associated may be moved to the network interfaces that changes the level from the cluster network to the grid network. Then, the equation (4.8) will be:

$$\lambda_{(m)} = \frac{(O_c) + L + (O_G)}{\hat{g}} \tag{4.9}$$

Where $(O_G)$ is the overhead time corresponding to the change of cluster-grid network and $(O_c)$ is the overhead time corresponding to he cluster-network only.

Then, high values of $(O_G)$ involve a great busy time of delivery in the network interfaces, due to the saturation of the resource. A discussion about the influence of the change of cluster-grid network is proposed in the next section.

## 4.4 Cluster and Grid Overhead Time

Nodes in Grid computing are distributed over remote interconnected platforms. Likewise, nodes are often heterogeneous and the network can dynamically change in topology and also in bandwidth. When a transfer of data occurs in a grid computing platform, the data *travel* upon two types of environment: the intra-network that interconnects the nodes in the cluster and the external network associated to the grid that relies the remote cluster platforms.

The $LogP$ model [28] considers the transfer process in a homogeneous network. The nodes are linked by a connection that may be affected by the distance between nodes and the possible presence of the latency variations.

Now, if we observe a *single* communication between two clusters of a Grid computing platform, the message $m$ transferred leaves the sender node, named here $P_O$, and it *travels* through a link to the received node, named $P_1$, in the other cluster. Thus, a part of the message travel occurs in the cluster network, and other in the Grid network.

Figure 4.2 shows a minimalistic approach of the Cluster to Cluster transmission inside a Grid computing platform. On the other side, Figure 4.3 represents a general case, where they are different *undetermined* devices and networks. It is for this reason that it is represented by a cloud.

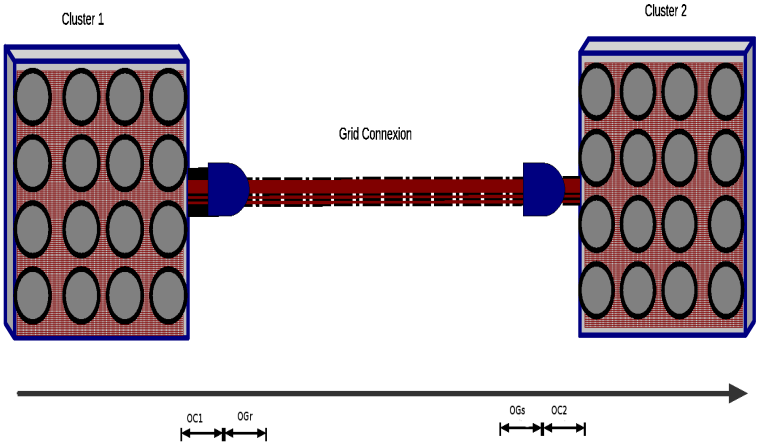


Figure 4.2: *Cluster-Grid Transmission*

Then, Figure 4.3 proposes an abstraction of the Grid data transfer where the Grid network may be understood as a *cloud* and we treat only the overhead (and gap variations) to determinate the transmission behavior. This point will be discussed later with all the elements.

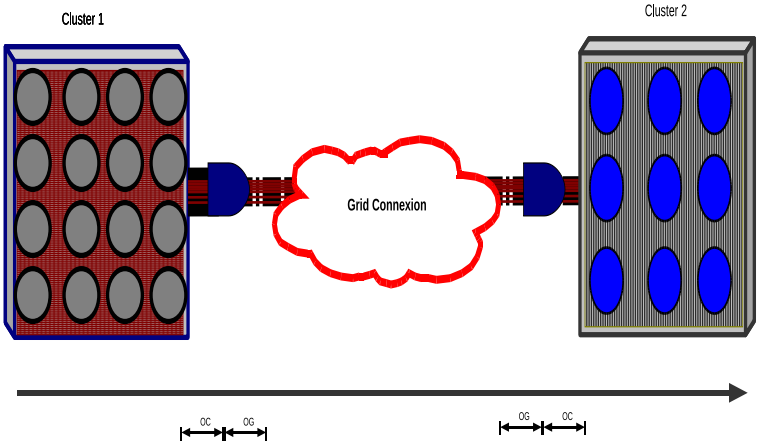


Figure 4.3: *Cluster-Grid Transmission General Approach.*

In the Figures 4.2 and 4.3 each overhead time is identified by the origin and destination. For example, the overhead time for the reception of the message $m$ from the cluster

1 to start the Grid environment travel is named $O_{C_1}$. Immediately this $O_{C_1}$ is followed by a overhead time of the start transference named $O_{G_r}$. At the moment the message $m$ leaves the Grid network, the overhead time $O_{G_s}$ and the overhead time $O_{C_2}$ follows the $O_{G_s}$ at once.

Then for this case, the overhead time of the change cluster-grid-cluster can be presented as:

$$O_{cgc} = \frac{O_{C_1} + O_{G_r}}{2} + \frac{O_{G_s} + O_{C_2}}{2} \tag{4.10}$$

The Figure 4.4 explains this hypothesis in function of the transfer processing of the message $m$. The message is transferred from a processor/node $P_O$ to a processor/node $P_1$, but in the transfer process crosses the cluster connections and the grid connections. This Figure allows to observe more clearly our analysis.

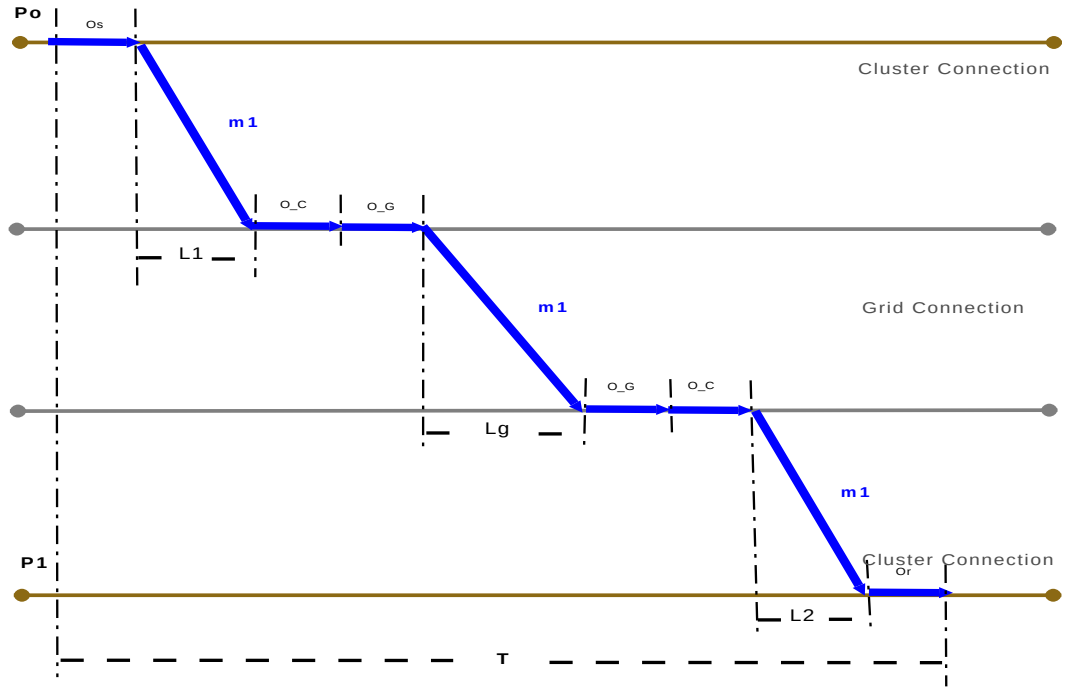


Figure 4.4: *Cluster-Grid Transmission Abstraction*

Returning to the right side of the Figure 4.2, we observe that it is more general. In real life, the clusters are different, the network intra cluster are heterogeneous, but we

suppose the devices have similar behavior in almost one sense. $O_C = O_{C_1} + O_{C_2}$ and $O_G = O_{G_r} + O_{G_S}$. Then the expression (4.10) may be presented as:

$$O_{cgc} = \frac{O_C + O_G}{2} \tag{4.11}$$

Where $O_{cgc}$ represents the overhead time of the change cluster-grid-cluster architecture, $O_C$ the general overhead time in the clusters due to intra-cluster network devices and $O_G$, the general overhead time in the Grid network, due to inter-platforms devices.

As the Figure 4.4 represents a transfer among two clusters, we presents latencies $L_1$ and $L_2$ associated with each one of the cluster transmission. The latency $L_g$ is the delay transmission associated with the Grid network.

Then, when the message $m$ arrives to the Grid connexion, the times $O_c$ and $O_g$ and vice versa can be measured as time difference at the moment that $m$ is received and sent by the network devices.

These devices making the relation between the cluster network and the inter-cluster network. As a consequence, during the transfer upon the inter-cluster network, the latency $L_g$ exists. In consequence, our latency $L$ in the Grid transfer becomes $L = L1 + Lg + L2$.

This analysis modifies the equation (2.4) and obviously the assumptions of the equation (2.5). Then, for our transfer of a message $m$ of great size in a link $Wp$, the transfer time $T(m)$ becomes:

$$T(m) = L + O_{cgc} + \hat{g}(m) \tag{4.12}$$

Where $L$ is the total latency that contains the latency during the intra-cluster network and the inter-cluster network (Grid connection) and $O_{cgc}$ is the overhead cluster-grid-cluster time, measured as a differential time in the linked network devices. The parameter $\hat{g}(m)$ is the measured gap (not included in the Figure 4.4).

On the other hand, is necessary to observe how the data placement affects the data access and in consequence the data transfer. Then, an analysis of the file systems influence is proposed and treated in the next section.

## 4.5 File Systems Influence

Observing the influence of the file system in the transfer process between nodes that correspond clients, servers or storages devices, it is agreed that the activity of the file system is important and affects the general performance during the communication. This

activity may be multiple and independent and it demands a service operation in the network that requires a time to satisfy the service requests[5].

Nowadays, the efficient use of the network resources on parallel and distributed architectures may be beneficial for storage requirements or the data management.

To handle this, they are proposed many distributed file systems or management strategies to guarantee an efficient data management and consequently, an efficient data transfer. Of course, there are an influence of the file system or management strategy on the data transfer.

Many works like [36], [103] have made studies about the sensitivity, modeling and performance prediction of high performance networks associated with the file systems and they address this problematic in an interesting way.

The use of parallel and distributed file systems implies a very large amounts of data and a continuous data transfer. Algorithms of data placement, data structures and query methods are implemented to obtain a relative efficiency for a specific physical machine architecture. In fact, the goal of a parallel file system is to allow a standard performance with a maximal adaptation, in terms of architectural characteristics. Different strategies of data management are implemented not only for file systems but also for many applications at different levels as it is possible to see, for example, in [86]. In terms of the applications, the efficiency is guaranteed by the balancing and distribution of the data without increase of the costs of communication.

A hybrid point of view allows to observe these characteristics into a set: applications send/receive data in concurrents process in accordance to sizes of data, concurrents access, allocation and scalability [64]. This point of view may be applicable also to general file systems.

Actually, a way to understand the effect of the file system in the transfer process is to see the response time in the node, observing its relation with the throughput time. In fact, the gap $g$ is dominated by the message size and the transfer within the network. Thus, the gap changes as a function of the packet size managed by the transfer protocol. The latency $L$ can vary, for example, where the network switches are affected by a congestion. Conversely, it exist an additional delay due to the file system [103].

[5]This subject is part of the work developed by the CAPES/COFEBUC action between the LIG-Montbonnot Laboratory at Montbonnot-St Martin, France and the GPPD-UFRGS Laboratory at Porto Alegre, Brazil.

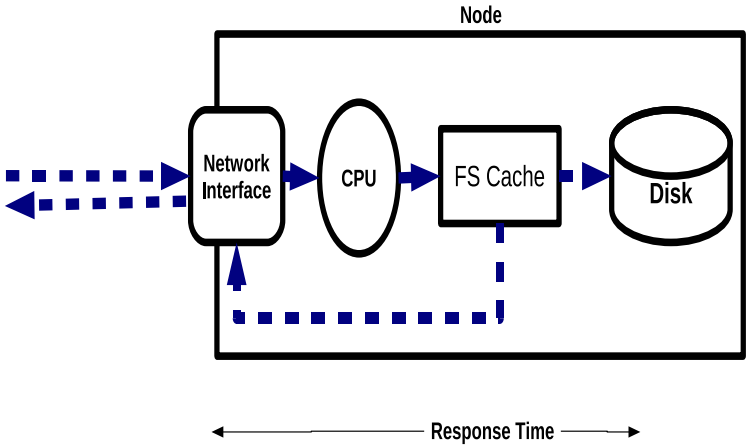


Figure 4.5: *Model to Response Time in a Node*

When the message arrives to the node, the interface network deliver the message to the interface processor that is placed into the receive queue. After, in accordance with the demanded task, the message is written or not in the disk, as shown in the Figure 4.5. Consequently, the total latency is increased and thus the transfer time.

It would seem that the change of write or read the block data in a determined file system is unknown, but there exists mechanism to know this, and technical implementations (i.e. the command $iostat$). Thus, to infer a delay time in the transfer process as function of the file system $T_{FS(m)}$, the equation (4.13) shows:

$$T_{FS(m)} = \frac{m}{m_t} * \frac{1}{tps} \tag{4.13}$$

Where $m$ is the size of message, $m_t$ is the relation between the size of a block transferred and the block size of the file system that vary in accordance with the specific file system and $tps$ is the number of transfers by second, with the assumption that each transfer on the disk correspond to a block.

## 4.6 Analysis of Anomalies during the Data Transfer

A significant case of study, is when exists a worst transfer due to an unreliable network or a quickly saturation of the devices (as router or switches). Actually, this assumption is observable by the important variations of the latency or loss of bandwidth [12].

Consequently, variations represent abrupt increases of the transfer cost. To handle this, two sub-cases are used: first, for a cluster data transfer and second, for a Grid data transfer. The discrimination of the two sub cases search to identify the factors that affect the latency and the bandwidth.

Firstly, we can use the cluster network in different behaviors and to omit the influence of the overhead send/receive time. For the next subcase, we use all factors. However, in each one of the sub-cases exists a variation of the size of messages and the number of nodes and links between them involved during the transfer.

In the cluster subcase, we take the equation (4.7) for our type of transfer high bandwidth explained before. Obviously the values of $\lambda \gg 1$ are for a worst network that presents a high latency $L$.

Normally, when the number of nodes is increased, the latency grows due to the use of the resources, then it's possible to observe different of values of $\lambda$[6].

In the Grid subcase, we use the expression (4.9). It is possible to omit the values of the overhead send/receive time and consider only the values of the change of cluster-grid network:

$$\lambda_{(m)} = \frac{L + (2 * O_{gc})}{\hat{g}} \tag{4.14}$$

Where $\lambda_{(m)}$ is the cost relation, $L$ the general latency, $(2 * O_{gc})$ is the overhead exchange and $\hat{g}$ the measured gap.

At this point, it is necessary to observe the variation of the bandwidth due to the influence of the gap during the transfer.

Taking the equation (4.6) presented in the section (4.2), we consider the values of $max(\hat{g})$ measured without saturation causing jams of the transmission (or faults due to the saturation among the transfer) as we says before.

Thus, experimental results in the case of really worst case of networks, suggest the possible utilization of a *anomaly parameter* to correct the bandwidth curve.

Now, knowing the influence of the *great* gap, using the expression:

$$B = \frac{m}{max(\hat{g})} = \frac{m}{\hat{g}} * \bar{A_p} \tag{4.15}$$

Where $B = \frac{m}{k * \hat{g}}$ is the bandwidth at time of the saturation due to the worst network and $\bar{A_p}$ is the anomaly parameter.

The $max(\hat{g})$ parameter is used because the delay time between each one of the $k - packages$ is very long, then the other parameters that affects the bandwidth have negligible values.

[6]These descriptions are based in experimental results exposed in the early results.

$\bar{A}_p$ is experimental and can be calculated as a function of the differences in the bandwidth curve.

## 4.7 Discussion: Interpreting $pLogP$ as $LO\hat{g}W_p$

In this chapter, we have made a reduction and a reinterpretation of parameters from $LogP$ and $LogGP$ models included in $pLogP$ model to describe the data transfer behavior in applications using files running on Grid computing platforms.

Taking the advantages of the $pLogP$ model, we can use the main parameters $L$, $O_s$, $O_r$, $g$ (or eventually $G$) and $P$ to describe and predict the parallel and massive data transfer behavior, but with a reinterpretation in the case of the overhead, gap and number of processors involved in the data transfer, to handle it.

This reinterpretation have consequences in derived analysis, such as the case of the transfer cost. On the other hand, we can analyze the influence of the file systems in these types of data exchanges and possible particularities due to differences in the performance of the network.

With the assumption that the transfer of the message of size $m$ occurs simultaneously and synchronized upon $W_p$ links, the $M$ total bytes transfer over the network can be characterized as:

$$N_G(M) = (L, \hat{g}, \hat{G}, O_s, O_r, W_p, m, O_{cgc}, T_{FS(m)}) \tag{4.16}$$

Where $L$ is the latency, $\hat{g}$ is the gap associated with the division of the message, $O_s$ and $O_r$ are the respective overhead send/receive time, $W_p$ the number of links used during the transfer that relies the nodes, $m$ the size of message transferred by link, $O_{cgc}$ is the overhead time associated with the change of cluster-grid network and $T_{FS(m)}$ is the delay associated with the file system.

Observing the Figure 4.4 we can identify that the *Total Time Transfer* is equal to: *Time to Send + Time to Reception + Time of Transference + Additional Delays*. (Also this relation is observable in the Figure 4.6 presented later.)

The *Time to Send* and *Time to Reception* are considered by the overhead send and receive respectively. The *Time of Transference* or time of trip among the network is considered by the latency. However, due to the influence of the variations in the size of message, the gap is fundamental and affects the total time transfer.

It exists other factors that add delays to the transfer, such as the data access in the disk, considered by the influence of the file system and the overhead due to de environment exchange (Cluster-Grid exchange) taken by the overhead exchange.

Thus, it is possible to propose a general expression for the time transfer in relation with the equations (2.4) and (2.5):

$$T(m) = O_s + O_r + L + O_{cgc} + \hat{g} + T_{FS} \tag{4.17}$$

Where the first part of the equation 4.17 corresponds to the *transfer time* without the node strictly speaking and the second part to the influence of the file system and the transfer within the node. In the large of this chapter, each one of the equation 4.17 elements are treated.

Now, assuming the supposition of a low influence of the overhead send/receive time in the nodes in relation with the other parameters, we can presents this expression 4.17 in another manner,

$$T(m) = L + O_{cgc} + \hat{g} + T_{FS} \tag{4.18}$$

Of course, there are limitations to this approach, for example, we can not observe stochastic facts. However our objective, explained in the section 4.1 is to achieve a *minimalist model* implementable to the performance evaluation in ”live” tests. Sometimes stochastic models involve several parameters that do not allows to build practical approaches due to their complexity.

In fact, there are consequences, discussed in the past sections: the presumption of the transfer cost in a first time with the relation between the latency and the gap, and after, between the general latency and the delay associated with the file system.

Other consequences can be inferred associated with our hypothesis, as the possible addition of many parameters to new transfer characteristics, like is observable in the evolution of $LogP$ model.

However, in general, the methodological advantages of our approach facilitate the measurement to performance evaluation of running applications in Grid computing platform because they do not add many parameters. We emphasize that our approach proposes a reduction and reinterpretation of the parameters to explain the behavior of the different characteristics involved in the massive data transfer in Grid Computing platform.

## 4.8 Conclusion

This proposal reach a level of abstraction where we can analyze large transfers on heterogeneous architectures, knowing only a general latency, considering the Grid network as a cloud, taking into account the overheads measured by the exchange of the environment ($O_{cgc}$ begins $O$ simply), and obviously the gap that provides information about the time delay due to the size of the message.

The complexity of the real process, observed graphically, for example, in the Figures 4.2 and 4.4 should be abstracted with simplicity by the Figures 4.3 and 4.6

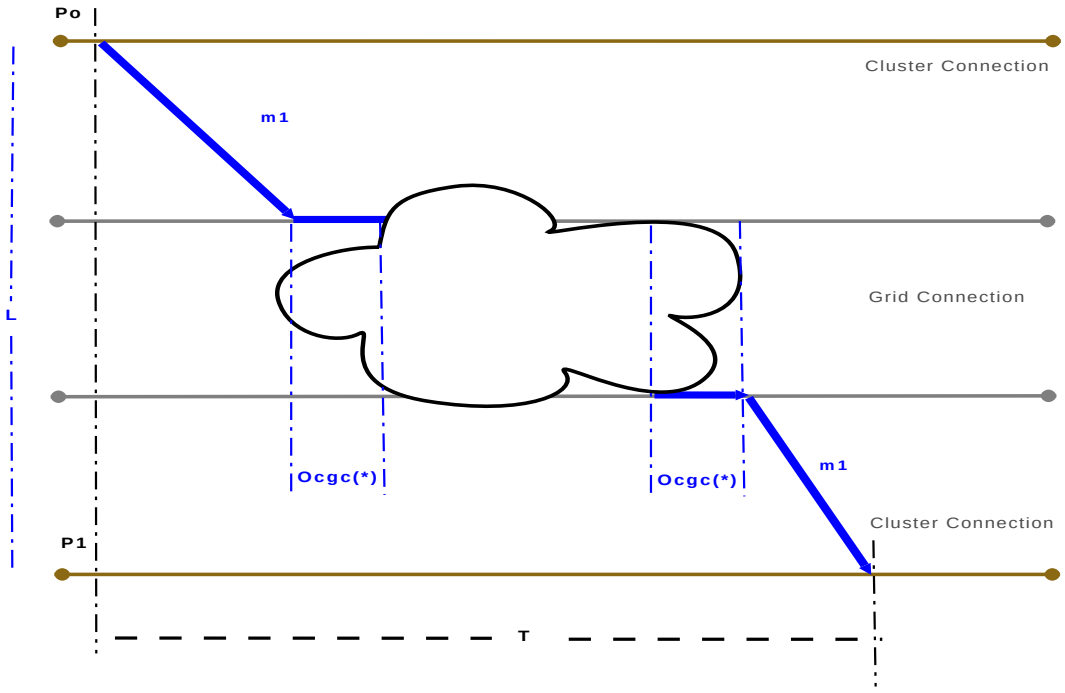


Figure 4.6: *Cluster-Grid Transmission Reduction Abstraction*

Figure 4.6 shows this simple abstraction for the case of a transfer of a message of large size $m_1$ between two nodes ($P0$ and $P1$) among a Grid computing network (without specify the number of links). The general latency is represented in the left (vertical line in blue) and the overhead time exchange for each one of the cluster-grid environment exchange by $O_{cgc}(*)$. Of course the only value of the $O_{cgc}$ will be calculated with the expression 4.11, presented before in the section 4.4.

In the same way, the Figure 4.6 shows the two types of connections: Grid and Cluster connections are determined by *gray zones* and the Grid network is represented by a cloud. The Total Transmission Time or Total Transfer Time will be the horizontal end line in black in the down of the Figure 4.6 and named with $T$. It involves the addition of the parameters treated before.

The abstraction of the File Systems influence to treat data access patterns is presented in the section 4.5 by the Figure 4.5. This analysis of the file systems influence permits to observe the variations in the data transfers between different file systems used in the

system under study, even more taking into account that in Production Grid computing platforms we can find different file systems to select in accordance with our necessities.

In general, practically if the transfer does not imply a high bandwidth data transfer, is to say, transfer of short messages, our model may becomes $LogP$. If the transfer implies a high bandwidth data transfer in homogeneous systems the model may becomes $LogP$. In the case of a high bandwidth data transfer among heterogeneous architectures (as Grid Computing platforms), each one of the parameters described before contributes to explain this massive and parallel data transfer.

At this point, it is necessary to validate the utility of the model to implement mechanisms of performance evaluation to observe the data transfer behavior. To handle this validation of our proposal, several cases will be presented in the next chapter.

# 5. Realistic Performance Evaluation using $LO\hat{g}W_p$

The main interest of this work is to propose an implementable and a simple model to observe the high bandwidth data transfer behavior on grid computing platforms. In the past chapter, the contribution to modeling is presented as a reinterpretation of the $LogP$ parameters into a new $LO\hat{g}W_p$ model.

The validation of the model utility is a difficult process. An analytical model requires a confrontation with measurements or with simulation. This work is aimed at performance evaluation, then, two techniques are presented: first, testing with a benchmarking tool and second, monitoring the behavior in real processes into a Grid Computing platform. The first technique is based on the use of the methodology presented in the section 3.1. We use the same modification of the *MPI LogP Multitest Benchmark Tool* (presented before in the same section 3.1) and we measure the main parameters of our $LO\hat{g}W_p$ model (presented in the chapter 4, *Contributions to Modeling of Parallel and Massive Data Transfer on Grid Computing*) and their behavior is analyzed.

The size $m$ of the messages transferred is of $10MB$ to $250MB$. The quantity of processors on two local platforms (clusters) start with 2 processors that corresponds to one (1) link $W_p$ and finish with $100$ processors that corresponds to 50 $W_p$ links. In fact, we consider these sizes of message as *large* messages, because their transfer involve a sufficient charge to saturate the underlying capacity of the network resources.

Hence, the total bytes quantity $M$ transferred during the tests follows the expression 4.5. For example, if in one test that involves $12$ $W_p$ links, with a transfer of a message $m$ of $100MB$ by link, the total quantity $M$ will be $1200MB$ ($12 * 100MB = 1200MB$).

Test were made in two layers to observe the two network levels: cluster environment tests and grid computing environment tests. Clusters tests aim to acquire measures that involve intra-cluster transfer. On the other hand, Grid Computing tests aim to acquire values of the Grid transfers between nodes of two different clusters.

In this chapter the validation of our proposal is presented taking in account each one of the considerations presented in the content of the chapter 4: the general massive and data

transfer behavior in the cluster context (section 5.1) and in the Grid context (section 5.2), the overheads due to cluster-grid exchange (section 5.2.1), the transfer cost analysis (section 5.3), the file systems sensibility (section 5.4) and the analysis of the anomalies in the worst case of a performance network (section 5.5). Moreover, two cases of performance evaluation in real systems using our proposal are presented (section 5.6).

## 5.1 Cluster Tests Results

Various local platforms of the Grid 5000 platform are used in the tests, but we present in this document significant results of specific sites. In this type of tests, all tests are made in one cluster of the local platforms. This tests allows to observe the behavior of the intra-cluster transfer, observing the different parameters related with network characteristics.

The first measure of reference observed is the gap $\hat{g}$, presented before in the section 4.2. Initially we observe this measure of gap $\hat{g}$ by link and later, on the switch. The gap provides information about the throughput and as reference measure can be used to observe the efficiency in the delivery of a message.

The gap, as all parameters, is measured for several transfers (between 10-50 times) and after a statistical treatment is possible to observe maximal gap values, minimal gap values and calculate an average gap. In the same way, the measured data has a little dispersion, that is increased when the size of message is increased, but remains between $0.4ms$ and $0.8ms$ approximately.

Figure 5.1 shows the gap measured for the Grillon Cluster of the Nancy site, for the transfer of $250MB$ by link. This figure presents the average values of the gap (blue), the gap minimal (green) and the gap maximal (red). During this tests the latency remains in $60$ microseconds.

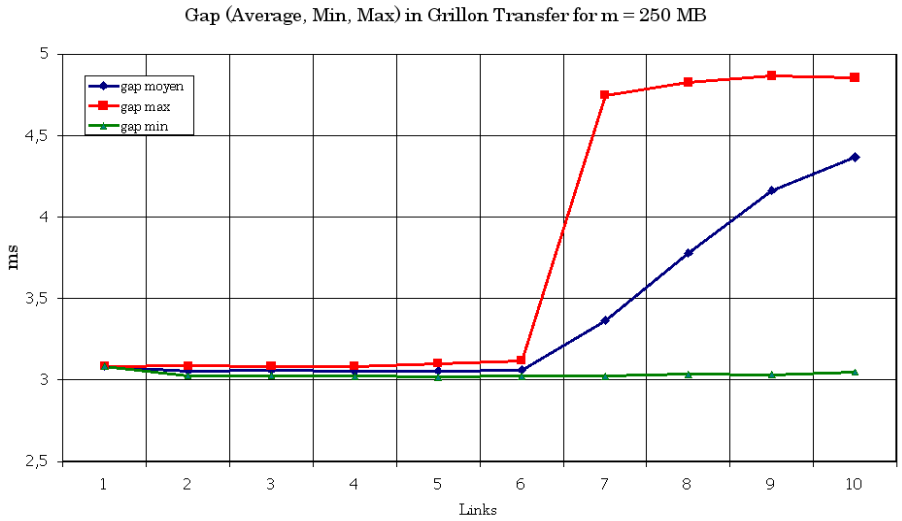


Figure 5.1: *Measured gap for Grillon Cluster*

Observing the gap, it remains stable until the use of $6$ links and after grows to remain stable for the rest of $W_p$ links with a slight increase. In the three curves, the increase is observed of the gap values. As expected, the maximal gap presents the more slope in the curve and the gap minimal presents a little increase.

The saturation noticeable in the tests with the increase of the gap, is reached quickly because a large quantity of bytes is transferred. The increase in the delay between the transfer of $k - packets$ that is composed the message is due to this saturation. This gap increases follows until a value related with the capacity of the network resources, such as the case of the links. However, is possible to find higher values, e.g. when exists a switch contention.

Before analyzing the switch behavior, we present the Figure 5.2. It shows the measures of gap $\hat{g}$ (blue line), overhead send $O_s$ (cyan line) and overhead reception $O_r$ (red line) for a transfer of $250MB$.

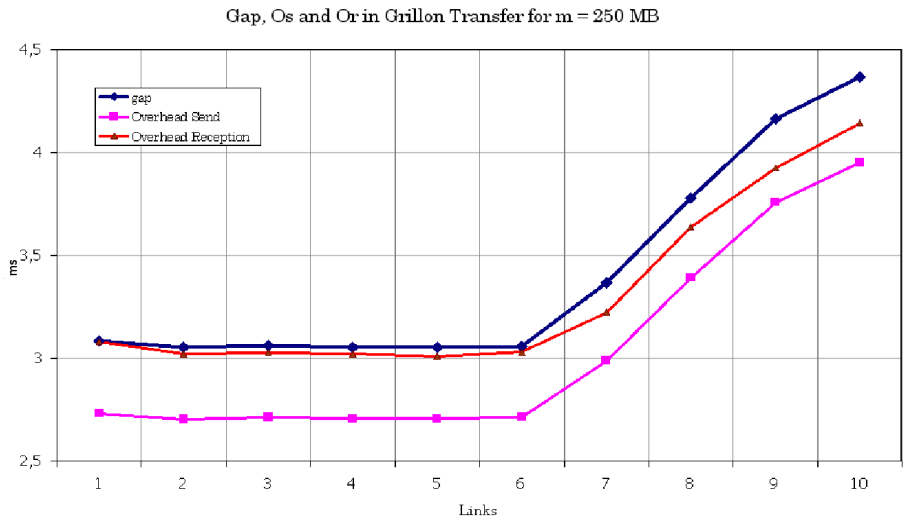


Figure 5.2: *Gap and Overhead measures in Grillon Cluster*

The Figure 5.2 shows a similarity of the three measured values. In fact, as the data transfer is synchronized by a *barrier*, the reception of a message does not starts before that the message is completely sent. The space between the reception time and the send time is affected by the gap.

A situation interesting to observe is when the transfer involves $6$ links. In this critical point, the total quantity of bytes transferred is $1500MB$ among the $6$ links, the delay in all measures grows before this quantity of links used, and it implies an important saturation, due to the important activity added in the network, that implies an increase of the data transfer cost that is discuss later.

Analyzing the behavior of the switch, the Figure 5.3 shows the influence of the gap in the bandwidth. Obviously, the gap is influenced by the switch behavior, and this influence is measurable in their bandwidth. The gap influence can be interpreted in terms of effectiveness. In the case of the gap minimal, this gap corresponds to the more effective links, unlike the case of the maximal gap measures.

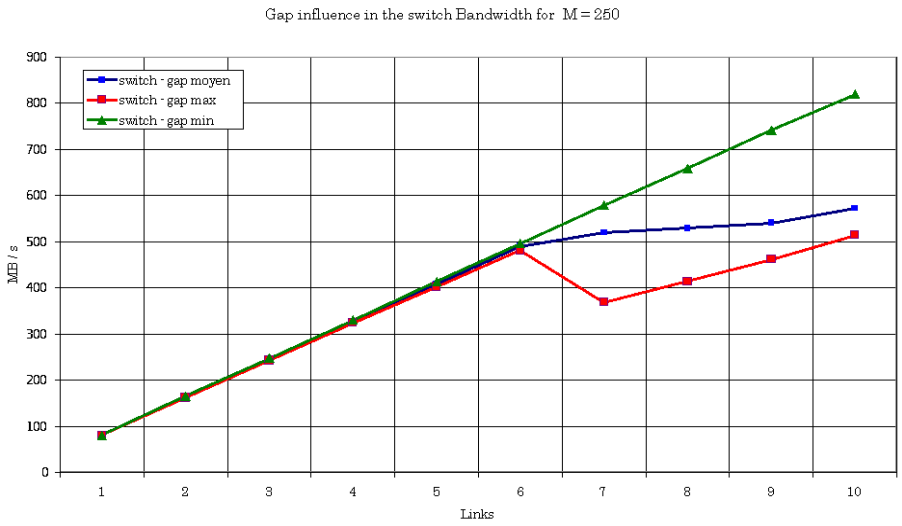


Figure 5.3: *Influence of the gap in the Bandwidth of Grillon Switch Cluster*

Figure 5.3 shows, in the same way as Figure 5.1, values of the gap influence: minimal (green line), maximal (red line) and average (blue). Due to the reciprocal relation between the gap and the bandwidth, a maximal gap represents a low bandwidth (comparing with the others values in the Figure 5.3).

When the bandwidth observed in the measures that uses the minimal gap grows, the data transfer cost decrease. However, it is important to remember that the bandwidth is affected only by the gap, due to the size of message transferred and our hypothesis discussed about the saturation, as is presented in the section 4.2, *Massive and Intensive Data Transfer*.

The next Figure 5.4 presents the bandwidth by links for the transfers of $5$ different sizes of messages $m$. The messages are transferred in parallel among determined quantity of links ($W_p$). The different values of bandwidth in the transfer of the $m = 50MB$, $m = 100MB$, $m = 150MB$ and $m = 200MB$ present a similar behavior. Conversely for the case of $m = 250MB$ the decrease of the bandwidth occurs when the number of nodes grows. Initially, the bandwidth grows in the same way than the others, until the use of $6$ links.

Immediately, occurs a fall in the bandwidth and decrease to a minimal value[1]. Observing the gap analyzed before, the *falls* in the bandwidth occurs when the gap increases. These *falls* are a bandwidth loss due to saturation, mainly due to the quantity of bytes

[1]This last minimal value remains when we increase in our tests the number of links to $50$ for this transfer.

transferred. This behavior for the transfer of $m = 250MB$ suggests an analysis presented before in the Figure 5.3 specifically for the case of the gap influence in the switch.

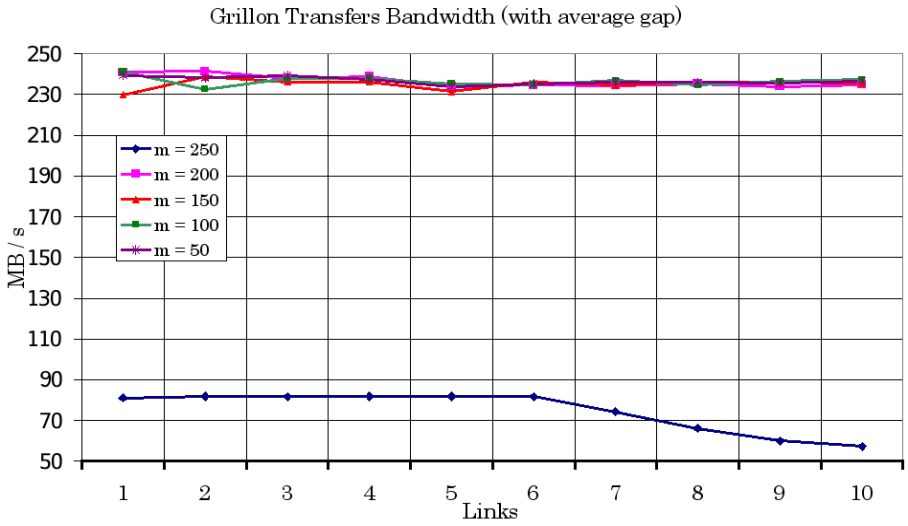


Figure 5.4: *Bandwidth by links in Grillon Cluster*

On the other hand, in the Figure 5.4 exists differences in the bandwidth values between each one of the curves, due mainly to the difference of the quantity of bytes transferred[2]. However, this difference is more important between the measures of $250MB$ and the others. This low bandwidth suggests a saturation that produces high values of the gap by $k - packet$.

In fact, it is necessary to remember that with the network saturation the gap is increased, and also with the increase of the nodes the bandwidth is shared between the connections. In consequence, the bandwidth reached for each link is limited. The values of the gap $\hat{g}$ are most important in the $m = 250MB$ transfers than for the others, due to the saturation and for some connections, there are highly gap values that may correspond to packet loss, a characteristic of saturated networks.

The relation gap-bandwidth advise to make a further examination of the switch behavior for all data transfers. The bandwidth values presented before, indicate that the shared bandwidth in the network depends of the number of simultaneous connections, however, these results are not sufficient to insure that the links between the two clusters are reached. The next Figure 5.5 presents the analysis of the switch behavior, for transfer

[2]Treated in the model by the expressions 4.2, 4.3 and 4.4.

of $m = 50MB$, $m = 100MB$, $m = 150MB$, $m = 200MB$ and $m = 250MB$.

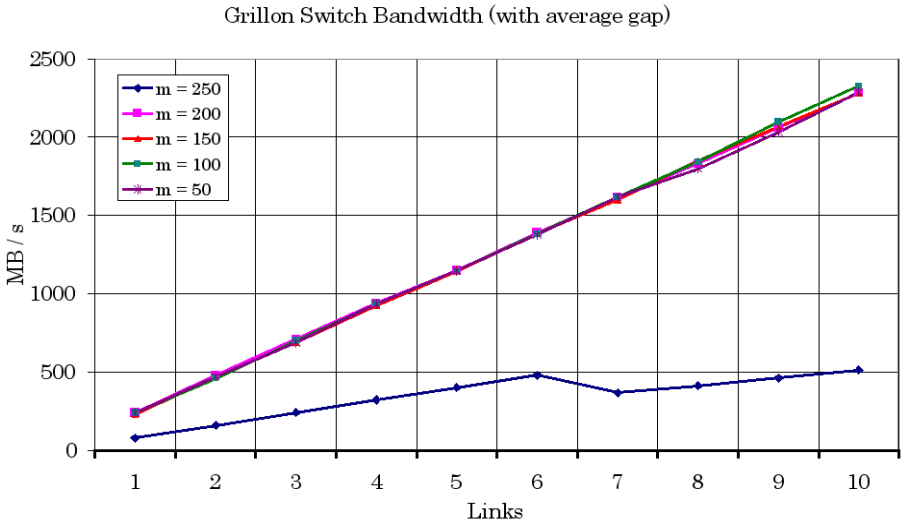


Figure 5.5: *Switch Bandwidth of Grillon Cluster*

In the Figure 5.5 is shown the increase of the bandwidth in the switch in all transfers of messages of $m = 50MB$, $m = 100MB$, $m = 150MB$ and $m = 200MB$. Also, it exists an important difference between the bandwidth switch of $m = 250MB$ and the others, in the same way that the Figure 5.4. As is presented before, the bandwidth switch for the transfer of $m = 250MB$ presents a regular increase until $6$ link. After there are a bandwidth loss, due to saturation of the switch (bottleneck). After this fall, the increase follows a small slope and reach to another bottleneck.

The switch behavior presents a sensibility to the increase of the connections. Each connection transfers a determined quantity of bytes between a pair. But the influence is not clear in the number of connections related with the size of the message. Therefore, an analysis is necessary that takes the number of links to observe this influence during the transfer.

Figure 5.6 shows the bandwidth by link, between $W_p = 1$ until $W_p = 10$ links that links pairs of processors. Each one of the link quantities is identified with a different color.

The Figure 5.6 contains the size of the messages transferred and the ordinates axe has the bandwidth values in Megabytes per seconds.

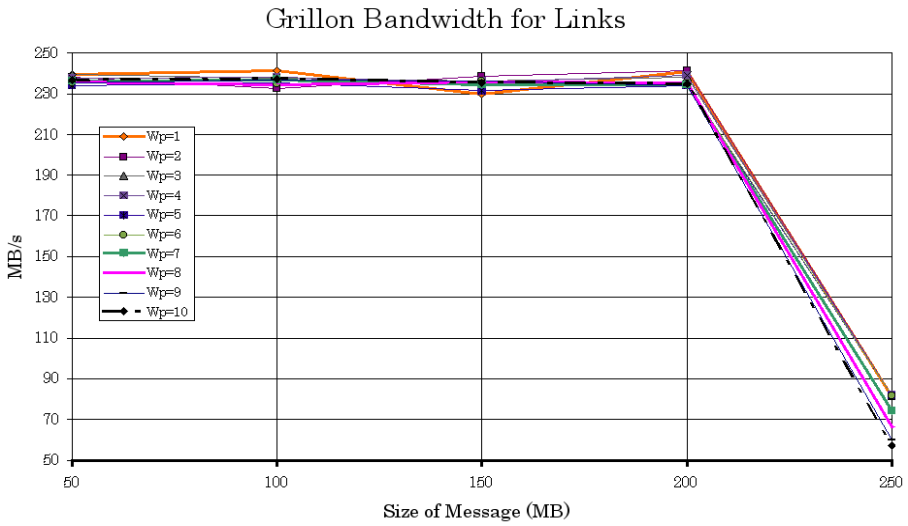


Figure 5.6: *Bandwidth by Links involved in a Transfer on Grillon Cluster*

It is interesting to observe in the Figure 5.6, how the increase of the links adds more bytes in the network, but the general behavior remains stable. Of course, the growth of the messages induces an increase of the gap, but the network manages the transfer guaranteeing a constant and stable high bandwidth for each quantity of links. However, the management of the transfer is efficient for the messages of sizes between $m = 50MB$ and $m = 200MB$, but not for $m = 250MB$ , because the capacity is outgrown. In consequence, the bandwidth decays for this $250MB$ transfer.

In other words, the connections are saturated when the quantity of $m$ bytes transferred exceeds the capacity of the network. This saturation is pronounced in the measures like bandwidth loss. In the practice, saturations causes perturbations in the transfer, such as bottlenecks. Precisely, the dramatical fall in the bandwidth values with $m = 250MB$ is due to the growth in the quantity of bytes transferred by link that affects the network capacity. The space of time by $k - packet$ grows and the bandwidth falls.

We have done the test in different local platforms of Grid'5000 infrastructure. The behavior observed in several clusters varies for the same tests. These differences are due to the influence of the architecture and the technological features of the platform in the transfer. For example, the Figure 5.7 shows the measured gap for the GDX Cluster during the transfer of $m = 250MB$. The gap values are presented in the same way that Figure 5.1: gap average (blue), gap minimal (green) and gap maximal (red).

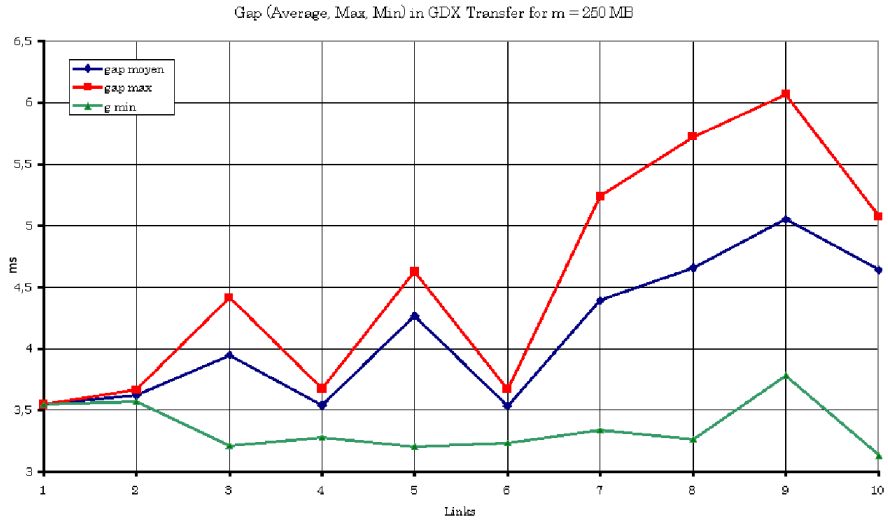


Figure 5.7: *Measured gap for GDX Cluster*

The gap time measured in GDX cluster is lower than the gap time observed in Grillon cluster. Also, Figure 5.7 shows that the *gap curves* are different qualitatively in relation with the Grillon cluster gap curves.

Without making a comparison with other behaviors, the different gaps present important irregularities in specific quantity of links involved. However, is observable an increase of the gap values to reach a specific value in them, almost in the gap maximal and gap average.

Observing the curves of the gap maximal and gap average, we find that the behavior is very similar and their values are close. The Figure 5.7 shows the same points of growth of the gap for these results.

Clearly, the original behavior of the minimal gap is opposed to the behavior of the maximal gap and average gap (only is the same with the use of $1$ and $2$ links). Nevertheless, when $9$ links are used in the transfer, a similar increase in the gap value is remarkable in the three curves.

In the same way than Figure 5.2, Figure 5.8 shows the measures of gap (blue line), overhead send time (cyan line) and overhead reception time (red line) for GDX cluster tests.

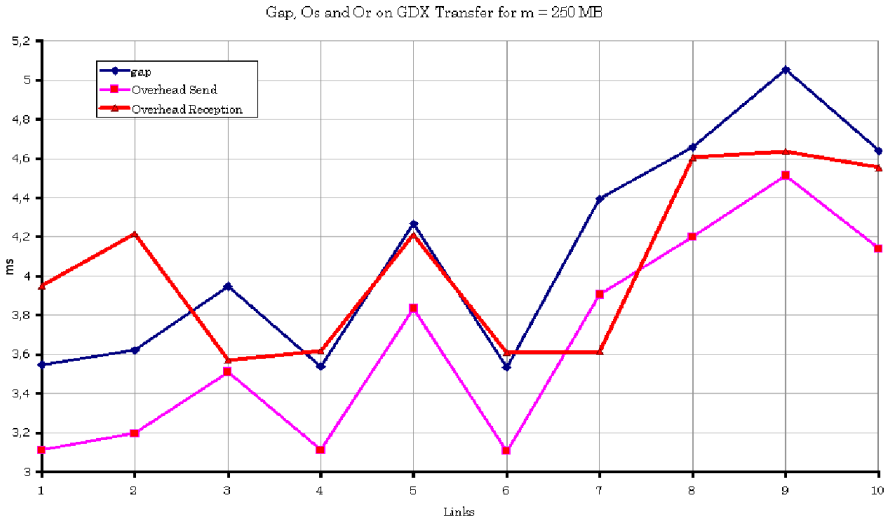


Figure 5.8: *Gap and Overhead measures in GDX Cluster*

Notwithstanding that these measures are less stable than the activity in the Grillon cluster tests, the results remains in the interval expected. The three measures are similar, the synchronization method used in the tests with the barrier guarantees that the message receive does not start before the message has been completely sent, and the differences between the overhead send time and the overhead receive time are explained by the gap.

Figure 5.9 presents the bandwidth observed during the messages transfer in the GDX cluster. The sizes of messages are the same than for other tests: $m = 50MB$, $m = 100MB$, $m = 150MB$, $m = 200MB$ and $m = 250MB$.

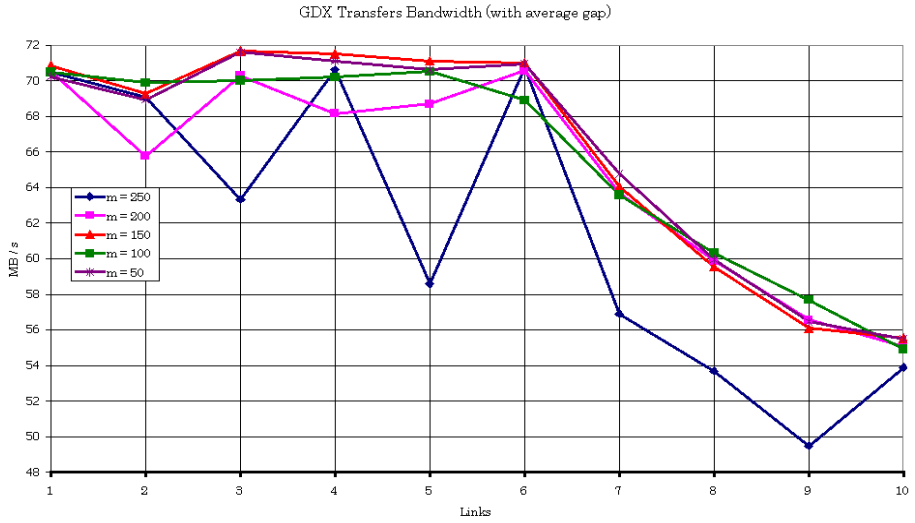


Figure 5.9: *Bandwidth in GDX Cluster*

The bandwidth values presented in the Figure 5.9 are stable for the first transfer of messages, it is to say for the sizes $m = 50MB$, $m = 100MB$, $m = 150MB$ and $m = 200MB$. The bandwidth observed during the transfer of $m = 250MB$ is different. This bandwidth presents dramatic falls. And these falls occurs for different number of links used.

In spite of these observations, the decrease observed in the bandwidth is interesting because for all curves, it follows a similar tendency. The analysis of the gap permit to explain this behavior. The growth of the gap produces an important delay in the transfer that causes the falls in the bandwidth values. The saturation due to the quantity of bytes transferred is noticeable in this cluster, and more when the parallel transfer increases.

The analysis of the switch behavior is proposed with the Figure 5.10. The Figure 5.10 presents the test results of the bandwidth in the switch during the transfer on GDX cluster.

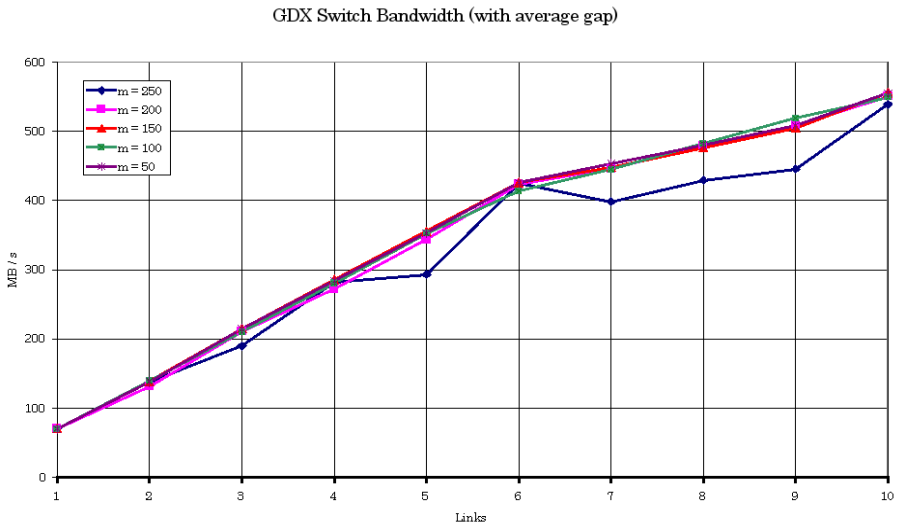


Figure 5.10: *Bandwidth in GDX Switch Cluster*

In the Figure 5.10, the switch bandwidth grows for all transfers similarly. They are falls of the bandwidth for the transfer of $m = 250MB$, but generally their behavior remains following the same increase that the other transfers.

In general, the management of the connections made by the switch in GDX Cluster search to maintain a high bandwidth during all transfers. Contrary of the observations in Grillon cluster, where exists an important loss of the bandwidth due to the saturation when $250MB$ are transferred.

In synthesis, the particular behaviors observed in each cluster suggests that it is important to take in account the gap influence in the intra-cluster transfer. Obviously, the gap is affected by the quantity of bytes transferred and each local platform has implemented specific mechanisms of transfer managed associated with their architecture.

At this point, is important to remember that the gap represents the reciprocal value of the end to end bandwidth from node to node, then it establishes a relation between the gap and the bandwidth to analyze.

Observing the figures 5.4, 5.5, 5.6, 5.9 and 5.10, the bandwidth observed follows a tendency. In the case of the switch analysis, bandwidth grows because the transfer manage made for the switch search to guarantee the maximum capacity of transfer. In the case of the links analysis, the bandwidth decreases because the capacity of each link is reached, and the general network state is affected by the total quantity of bytes transferred.

On another hand, there are two aspects that should be analyzed to explain the disturbances in the measures: (1) the saturation of the network caused by the test at the transfer

time, and (2) the contention in the benchmark program.

Making the test with a latency of $60$ microseconds on Grillon transfer, and using the general expression 4.17 presented in the last section[3], is possible to estimate the total transfer time for all sizes of messages, as is presented in the Figure 5.11.

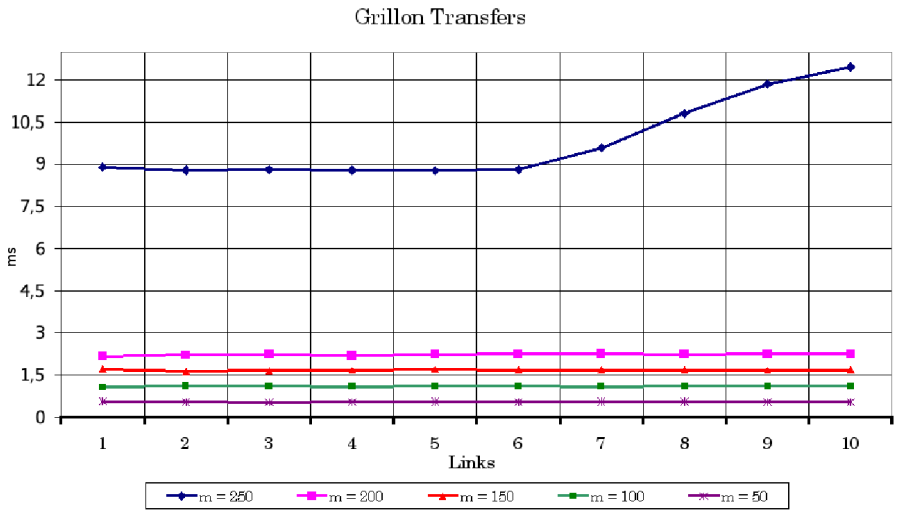


Figure 5.11: *Transfer Time in Grillon Cluster*

The high values of the gap presented for the transfer of $250MB$ affects the general transfer time, as is possible to see in the Figure 5.11. However, it is clearly that the increase of the links in this transfer adds a delay, and this situation is observed also in the other measures for other platforms.

In the case of the test on GDX transfer, the measured latency is the $80$ microseconds at moment of the tests. The transfer time estimated for each one of the sizes of messages are presented in the Figure 5.12.

[3]Of course, without the use of the $O_{cgc}$ overhead, because the transfer occurs in one cluster. It the same sense, we suppose a minimal influence of the file system ($\approx 0$) for this time estimation and for the next estimation on GDX Cluster.

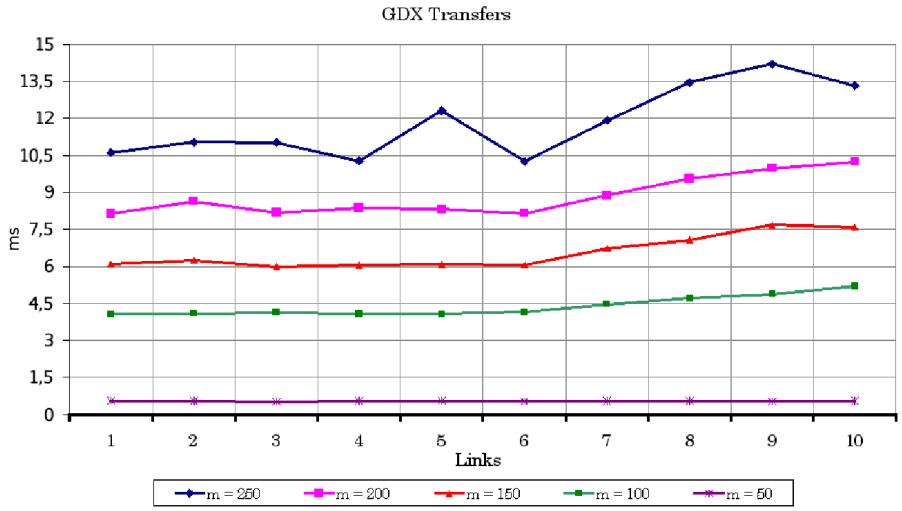


Figure 5.12: *Transfer Time in GDX Cluster*

In these tests, the space between transfer times is more stable (without little disturbances, due to the cluster activity). Clearly exists a most difference between the transfer of $50MB$ and the others, but remains stable. The other transfers, presents a clear growth in their transfer time due in the same sense by the growths in the gap.

Comparing the times observed in these two platforms, presented in the figures 5.11 and 5.12, is possible to observe that the transfer time remains between similar intervals. Of course, there are differences due mainly to the specific architectural characteristics and the use of each one of them.

The cluster computing allow to analyze and describe the behavior of the massive and parallel transfer using our proposal. In fact, we have used the size of the messages transferred $m$ and the number of links involved in the transfer $W_p$, as main characteristics. Knowing the latency $L$, and supposing that the overhead is negligible in accordance with our hypothesis presented in the chapter 4, we arrives to estimate the gap $\hat{g}$, and using this gap, we analyze the bandwidth by links and the bandwidth in the switch.

The opportunities provided by our model simplify the capture of values to analyze the behavior, moreover if we search to describe the data transfer in terms of network characteristics only. Evidently, this simplicity is provided also by $pLogP$ for this type of transfer in clusters. However, it is not the case for grid data transfer, that is treated in the next section.

## 5.2 Grid Computing Tests Results

The Grid computing tests makes the same type of transfer of the Cluster tests. In the section 3.1, *Measurement Methodology*, we have presented the description of these types of tests. The main goal of these tests is to observe the behavior capturing the parameters associated with our proposition: values of gap $\hat{g}$, overheads (mainly the overhead time due to the environment exchange), latency at moment of the transfer occurs to estimate the transfer cost, bandwidth, file system influence between others.

Grid computing transfer has specific characteristics. Remember the characteristics of the transfer "out" of the cluster, when the transfer process of a message of size $m$ occurs the message transferred leaves the cluster environment in a moment to trip in a wide network and finally arrives to another cluster environment. Undoubtedly, the transfer time is affected by the changes of the environment, represented by overheads presents in the inter-cluster switches, PoPs and routers.

Another important characteristic is the direction of the transfer. Even though the results selected shown here have a similar behavior, the direction of transfer is defined in each one of the figures.

Figure 5.13 shows the results of the gap measures for a transfer of messages of size $m = 250MB$. The transfer occurs between the GDX and Grillon clusters in Grid'5000 platform. Measures of gap corresponds to gap average (blue), gap maximal (red) and gap minimal (green).

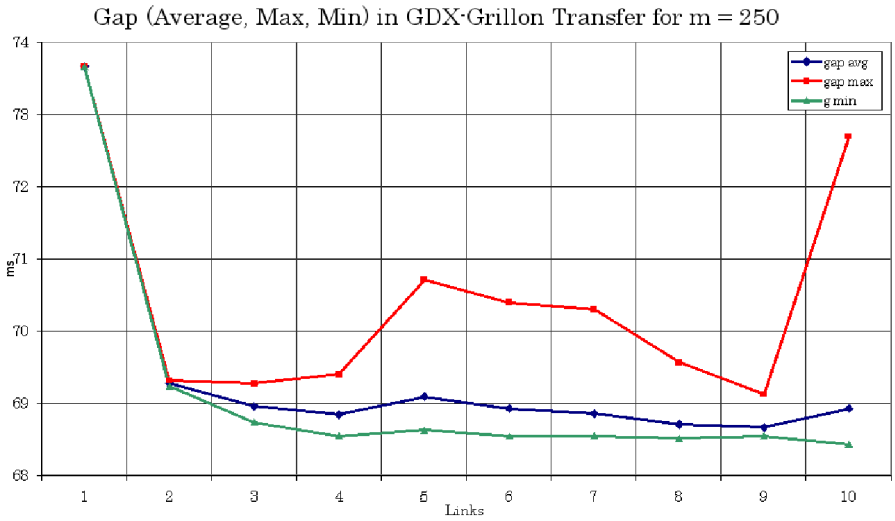


Figure 5.13: *Measures of gap for GDX-Grillon Transfer*

In the Figure 5.13, the initial transfers that involve few links present falls in the gap values. After, the gap remains stable. However, observing the gap maximal curve, when the number of links grows considerably, the gap values increase to reach one more time a relative high value, near to the original value of the gap.

This behavior is expected, because each one of the environments (cluster and grids) and the devices that are in the middle of the Grid platform (and take part of the Grid transfer) adds delays between the $k - packet$ transmissions. Thus, the data management in the Grid wide connection guarantee a high capacity in comparison with the data management in the cluster platforms, on the other hand the transmission capacity is more high. Thus, the curves of the gap are more stable in the grid transfers than in the cluster transfers.

In fact, when the capacity term is used here, this term not only is related with the hardware features. It is important to observe that the topology and network management inside of the cluster are different than the Grid. Moreover, if the shared resources during the cluster transfer are highly committed by the users (same if the different process that runs in concurrence does not implies an external interaction), giving a high concurrency.

The Figure 5.14 shows the gap and overhead measures for a transfer between GDX and Grillon clusters. Indeed, the values of overhead, corresponds to the overhead exchange $O_{cgc}$, during a transfer of $m = 250MB$ and with a latency of $L = 1415,13$ microseconds at moment of the tests.

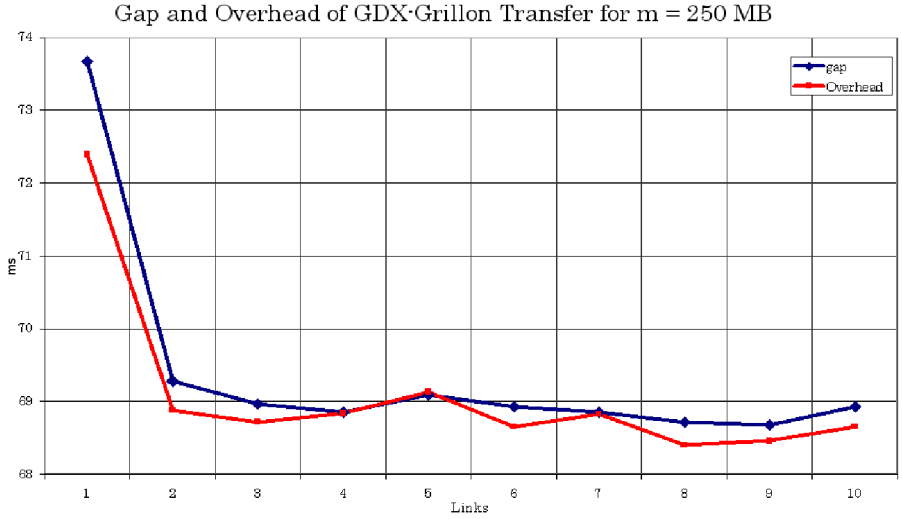


Figure 5.14: *Gap and Overhead Measures for GDX-Grillon Transfer*

The values of the overhead exchange $O_{cgc}$ are close to gap $\hat{g}$ values. It is not a surprise

because the delay due to the gap in the transmission affects the time of reception and sent of the message in the network device. On the other hand, the measures of $O_s$ and $O_r$ that corresponds to overhead in the nodes are negligible with regards to their values with the other parameters in this type of tests.

Figure 5.15 shows the bandwidth for the Grid transfer between GDX and Grillon clusters. The size of messages transferred are $m = 50MB$, $m = 100MB$, $m = 150MB$, $m = 200MB$ and $m = 250MB$.


Figure 5.15: *Bandwidth for GDX-Grillon Transfer*

Observing the Figure 5.15, the bandwidth values for the transfer of $m = 50MB$ remains constant in all quantity of links used. The relative quantity of bytes transferred does not saturate dramatically the network. At this point is important to say that at moment of the tests, the latency was $2652,70$ microseconds. For the other transfers of messages, $m = 100MB$, $m = 150MB$ and $m = 200MB$, the bandwidth growths to reach a relative high value quickly and after remains stable and the latency was of $1375,27$ microseconds.

The difference of bandwidth values is not much, only of $0,5$ tenths between $m = 50MB$ transfer and the others. But, for an application that runs in a Grid environment in concurrency with other applications, this difference should add an important cost in the total performance.

Taking in account the scale of the measures, different explanations explain this behavior. First, the relative *low* bandwidth values for the transfer of $m = 50MB$ remain stable because there are not a real saturation in the devices or links, due to the high capacity of

the channels and Grid devices.

Second, for all size messages transfer, a quick saturation is reached, caused by the mechanisms of management of data in the high speed networks used in the Grid interconnection. The goal is to guarantee a small growths in the delays for saturation or deliverance. In other words, provides a high bandwidth connection.

### 5.2.1 Contention and Overhead

Actually, contention in the short messages is ignored, but in the transfer of large messages, as it is the case of this work, analysis of the contention is important.

Different works propose extensions of the $LogP$ model, add parameters to estimate the contention [40] [108]. The parameter $C_n$ is added to study the impact of the data and process mapping, in other words, locality or the message size on network contention.

In our approach, the overhead values measured permit to know in an experimental way, the delay due to contention network and the delay due to contention in the interfaces. Both cases, with a hard influence of the size of message, make possible to build a relation between the number of nodes/processors that are implied in the transfer and the quantity of links that interconnect them.

The Figure 5.16 shows the overhead time measured in the grid transfer between GDX and Grillon clusters, for different sizes of messages, each one identified with a specific color.

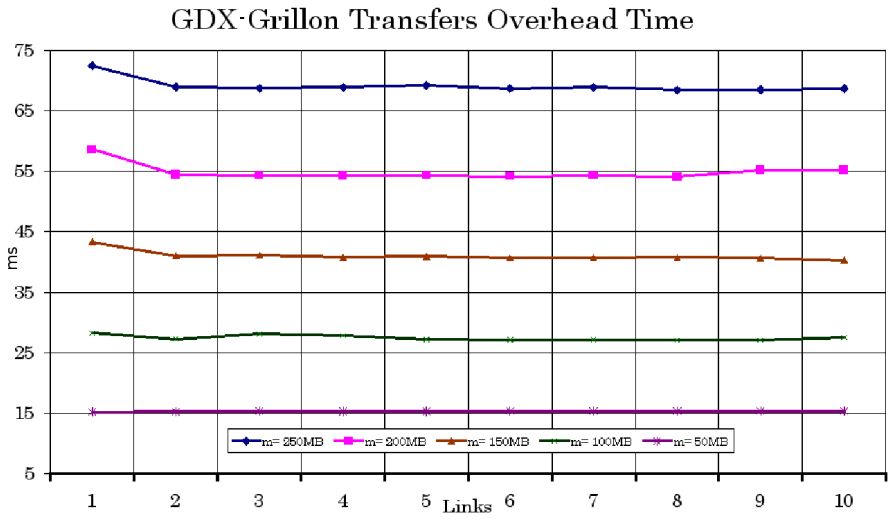


Figure 5.16: *Overhead in GDX-Grillon Transfer*

The overhead time represents the delay due to the environment exchange (Cluster-Grid), in accord with the hypothesis presented before, in the section 4.4, *Cluster and Grid Overhead Time*.

In the Figure 5.16, we can observe how the increase of the bytes transferred affects the overhead and obviously, adds a delay that impacts the network contention. The lowest values of overhead time corresponds to the smallest size of messages, in this case for a transfer of $m = 50MB$, and the highest overhead time corresponds to the largest size of messages, it is to say, $m = 250MB$.

The overhead time remains constant when is increased the number of links, because the time that take the message to change the environment is the same. This time is only limited by the capacity of the device responsible of the connection cluster to grid, and normally, this capacity is not affected by the underlying capacity of the communication protocol or by the link capacity.

In the same Figure 5.16, it is possible to see a little variation between the use of one link and the rest. The variation is more notable when the size of messages grows. These little variations in the first values of the overhead time are due to the change of the network state at moment to start the tests.

To analyze the different overhead time increase during the transfers, it is necessary to present the results as a function of link quantity involved in the parallel transfer. In this sense, the Figure 5.17 shows the overhead time values of the different transfers by link.

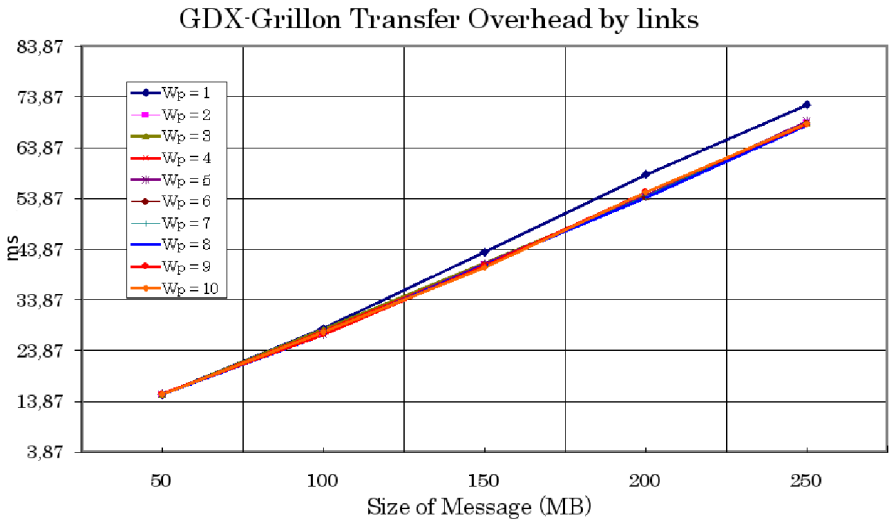


Figure 5.17: *Overhead in GDX-Grillon Transfer by links*

Clearly, Figure 5.17 permit to see the influence of the size of message in the transfer time. The increase of the overhead time is linear, when the size of message is increased independently of the number of links used at time of the transfer.

Thus, the size of message is the main factor that affects the overhead time in a Grid transfer. The little variation observed by the transfer using a link not affects really this affirmation, because the growth tendency of the overhead time observed in the Figure 5.17 presents the same linear behavior.

Other characteristics as the influence of the synchronization or the heterogeneity could be studied with these results. For example, in the case of the synchronization, it is possible to modify the tests sending asynchronous messages, but our interest is to see a parallel and massive data transfer among the two types of networks (Cluster network and Grid network). In the case of asynchronous messages, a delay should be added in the general transfer time.

On the other hand, to analyze the heterogeneity, it is necessary to take into account the topology of the network with techniques associated to discover the specific topology and architectural features of the platform. For example, the delay adding by the processor interruption or specific memory management, or the delay adding by the different capacities of the network devices presents in each one of the involved platforms between other possibilities.

These cases are analyzed with methodologies proposed by several authors using test results similar to our tests [17] [40] [75] [108].

At this point, we have observed different measures for high bandwidth data transfer. The measures are related with network characteristics, using our reinterpretation of the $pLogP$ parameters with $LO\hat{g}W_P$ parameters. We take measures of $\hat{g}$ to analyze the delay of time due to partitioning of the message in $k - packages$ and drawing the relation between the gap and the bandwidth, we succeeded to analyze and measure the bandwidth. In the same sense, we can take the latency at the moment of tests and we can consider the occupation of the network and their availability.

Grid data transfer implies environment exchange. The environment exchange creates a backlog in the data transfer, that is possible to measure when the data leaves the local platform where the message is sent and when arrives to the remote local platform involved in the reception. This overhead $O$ allows to analyze factors as the contention in the exchange network devices.

On the other hand, our proposition reaches to analyze the data transfer behavior considering the number of links between nodes. In fact, in terms of communication, the number of links used to interconnect the nodes give more information that the number of processors. In our approach, we use, the most simple presumption: the nodes are linked between peers for a link among them. Using the number of links, we can observe what

happening during a parallel transfer. In these simultaneous transfers the quantity of bytes that travels each link, add a charge to the network. And when we say before, permits to describe the communication process associated with network characteristics.

However, an estimation of the total transfer time and each one of the last values of $LO\hat{g}W_P$ is sufficient to understand the behavior of a data transfer. It is necessary to analyze the transfer cost, that allows to qualify the efficiency or not of the transfer.

## 5.3 Transfer Cost Analysis

Knowing the efficiency of a communication as a function of the amount of bytes transferred, it is possible to analyze the transfer cost. The methodology used in this work implies the observation of the completion time of the transmission.

Basically, as is described in the section 4.3, we can use the relation between the latency, the gap and the overheads to observe efficiency. This relation is known as $\lambda$ and it allows to estimate the transfer cost, depending the values of $\lambda$.

Figure 5.18 shows the values of $\lambda$ that represents the relation between the $L$ latency and the $\hat{g}$ gap by number of links that relies the nodes. The latency remains low in these tests ($60$ microseconds)

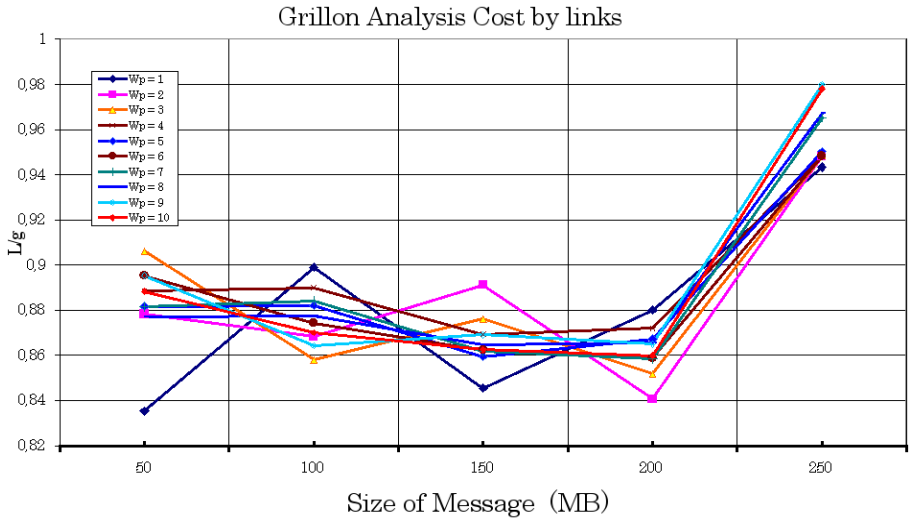


Figure 5.18: *Cost Analysis of the Transfers in Grillon cluster (Low latency)*

In this Figure 5.18, is possible to see how the increase of the size of message affects

the transfer cost. Obviously, the total quantity of bytes transferred is increasing each time that a new links is added. In consequence, each new link between processors adds a transfer cost. The values of the relation $L/\hat{g}$ are less than 1. This situation supposes that the cost generated by the transfer is due to the delay in the gap and this figure shows that this transfer cost decreases when the difference between the Latency and the gap is lower.

Growing the latency ($\approx$ 1 second), the Figure 5.19 shows the transfer cost behavior represented by the relation $L/\hat{g}$. These $\lambda$ values changes however the similarity in the curves shown in the Figure 5.19.

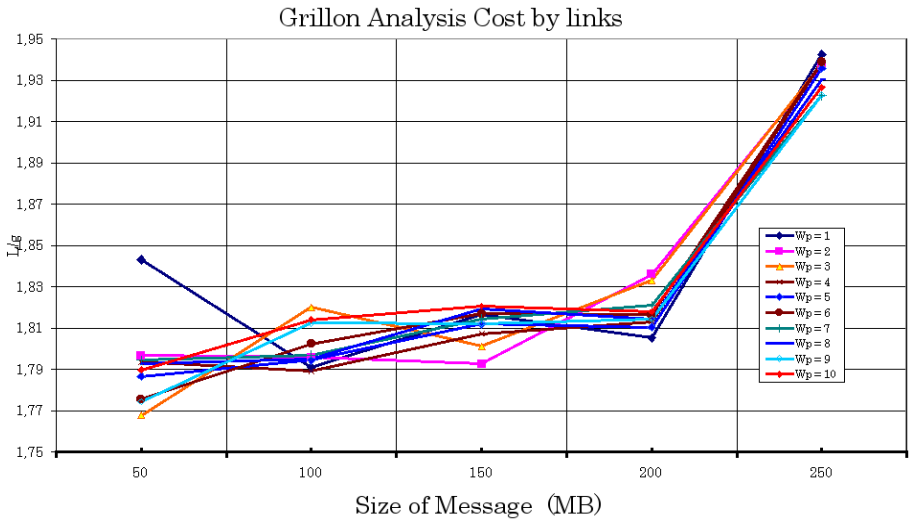


Figure 5.19: *Cost Analysis of the Transfers in Grillon cluster (High latency)*

In this figure, the values of the relation $L/\hat{g}$ are high, and more of 1. Obviously, this behavior is due to the cost related with the high values of the latency. The gap $\hat{g}$ values are the same values of the previous test.

The idea to changes the latency values and to observe these two different behaviors, reach to observe the different cases of cost presented before in the section 4.3 about the $\lambda$ values. In the first case, where $\lambda < 1$, the data transfer cost is dominated by the gap $\hat{g}$. The second case, shows $\lambda > 1$ values, that implies that the data transfer cost is dominated by the high latency $L$.

Now to analyze the data transfer cost between two local platforms that implies a grid computing exchange, we use the parameters $\hat{g}$, $L$ and $O$ of our model, using the expression 4.9 explained in the section 4.3, *Transfer Cost*. These grid computing transfers are conducted in the same way of the other past experiences (Massive and parallel data transfer

among the links that interconnect nodes in two different local platforms).

Figure 5.20 contains the analysis cost in relation with the size of message transferred in a two types of Grid computing transfers.First, using a network with a capacity of $1GB/s$ and second, with a network with a capacity of $2.5/GB/s$. The latency is $1413,49$ microseconds. (Actually, the capacities of the Grid entities are most important than the cluster as is also their use[4].).

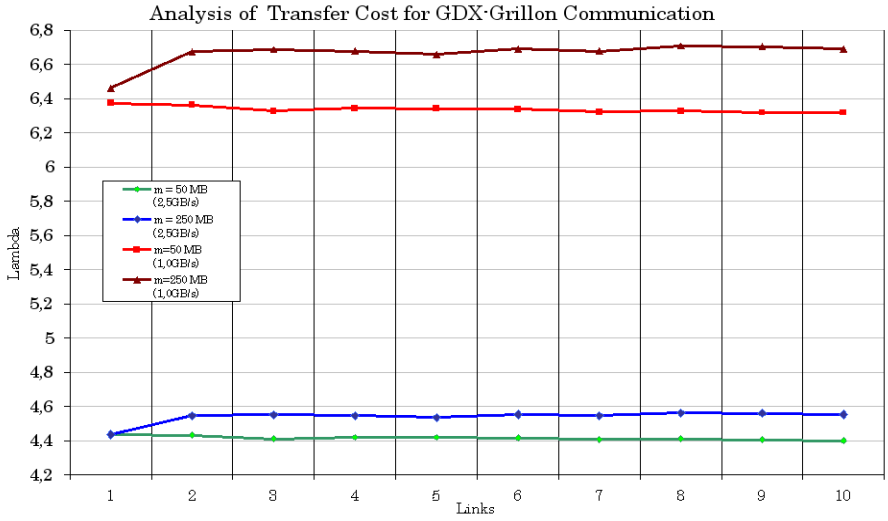


Figure 5.20: *Transfer Cost Analysis for GDX-Grillon clusters*

Although the capacities of the Grid entities, in any case is affected by the transmission capacity of the communication channel. As is observed in the Figure 5.20, the high values of $\lambda$ represents a large difference between the gap and the latency, due to the high use of the Grid network, independently of the data management shown by the stable values of the gap.

Figure 5.21 shows the transfer cost but in relation with the number of links implied in the transfer with a high latency. The cost is related with the latency and the possible high values of $O_{cgc}$ that involves a possible busy time of the deliverance in the network interfaces, and in consequence, contention or saturation in the devices.

[4]On another hand, the management of the data transfer in a Grid network is more efficient than a cluster network. However, the increase of the transfer cost between the different sizes of messages is noticeable.

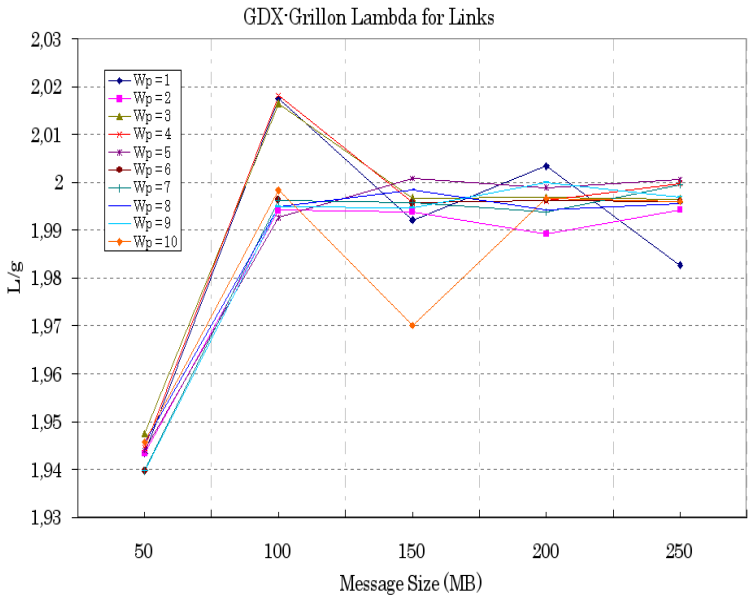


Figure 5.21: *Transfer Cost Analysis for GDX-Grillon clusters by links*

The specific $\lambda$ analysis provides information about the transfer cost due to gap values and effective transference delay, determined by the latency. For other latency values, the behavior is the same, with the similar curves with an increase of the values if we use more high values of the latency. In fact, the transfer cost is high on the Grid computing network, because the exchanges of the environment, distance and use of the network adds *time delays* to the transfer.

Consequently, the different parameters proposed have a role description. The gap $\hat{g}$ provide important information about the delays due to the data transfer itself, but the transfer cost is only known with the $\lambda$ relation. In the same sense, the analysis of the overheads time give information about the time involved during the environment exchange, and allows to analyze saturation or contention that generate busy time in the network devices. But the influence with the transfer cost is quantified implying the $\lambda$ analysis, as is presented before.

Indeed, there exist other factors important to observe, that are the case of the data placement and the data access. To handle these, we propose an analysis of the file system influence in the data transfer process, presented in the next section.

## 5.4 File Systems Sensibility

Section 4.5 presents the theoretical assumptions of the File System influence in the data transfer[5]. To handle these assumptions, is measured the parameters proposed by our methodology to observe the behavior of data transfer in various file systems on each one local platform.

The results presented here, correspond to experiences made in two file systems in GDX Cluster: NFS [127] and dNFSp [82]. Basically, they consist in sending and receiving data upon a cluster deployed with NFS and, later, with dNFSp.

The goal of these tests is to measure the transfer time of the cluster platform upon the two systems. The delay between consecutive transfers of messages leads to define a relationship between the number of nodes used in the transfer and the size of the messages transferred. The volume of data transfered between nodes by link is the same so the total amount increases with the number of nodes.

Figures 5.22 and 5.23 show the influence of the file system in time for transfers of $50MB$ upon NFS and upon dNFSp respectively. For NFS tests (Figure 5.22) , $1$ NFS server is used with $59$ clients. The Figure 5.22 shows the time average (red) and the maximal (green) and minimal (blue).

[5]This subject is part of the work developed by the CAPES/COFEBUC action between the LIG-Montbonnot Laboratory at Montbonnot-St Martin, France and the GPPD-UFRGS Laboratory at Porto Alegre, Brazil.

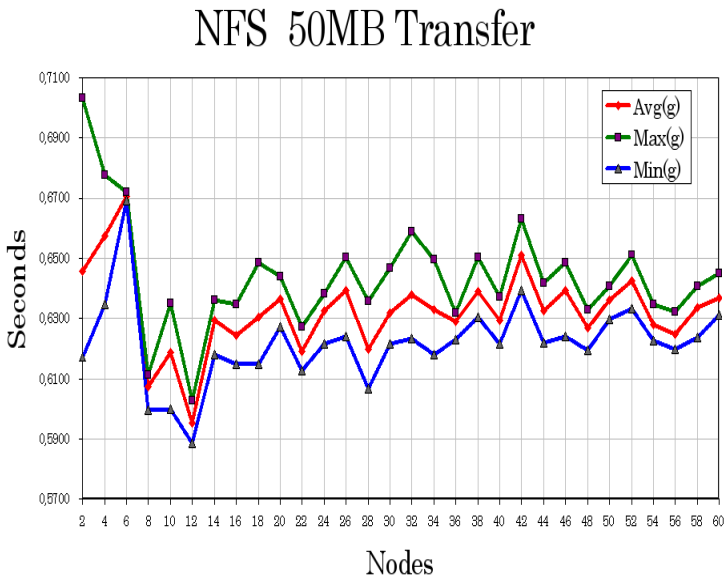


Figure 5.22: *File System Sensibility for 50MB transfer in NFS*

The measure of time was taken during consecutive transfers between nodes in the platform. The number of processors is shown in the $x$ axis and the measure of the time in the $y$ axis, in seconds. A total of $60$ nodes were used in these tests. The tests were repeated 3 times for each case.

Figure 5.23 presents the dNFSp tests. In these tests, $6$ nodes are used as metaservers and IOD's; the remaining $54$ nodes are clients. The Figure 5.23 shows the time average (red) and the maximal (purple) and minimal (blue).

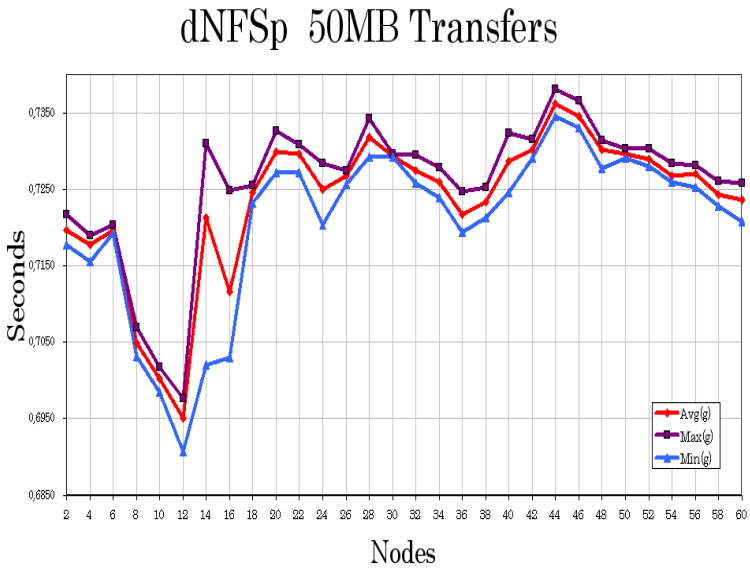


Figure 5.23: *File System Sensibility for 50MB transfer in dNFSp*

Comparing the Figures 5.22 and 5.23, we can exhibit the behavior of the two file systems and quantify an influence on the transfer. Observing the measures in the NFS tests, it is possible to observe the initial saturation represented by the high values. When the number of nodes increases, these values decrease to remain in intervals more stable. For dNFSp tests, the time falls to a minimal value and then it rises and remains stable.

To continue with the analysis of the file system on the transfer, we propose an analysis of the bandwidth for the data transfer in each one of the file systems used. In this way, the figures 5.24 and 5.25 show the analysis of the bandwidth taking into account the file system influence for both files system in study, done on real tests. The bandwidth is presented for transfers of $50MB$ for both NFS (Figure 5.24) and dNFSp (Figure 5.25) deployed in the platform.

The Figure 5.24 shows the maximal bandwidth (blue), minimal (green) and average (red). For each one of the measures is take the maximal influence of time presented before, and also take in account the maximal gap in the transfer for all measures.

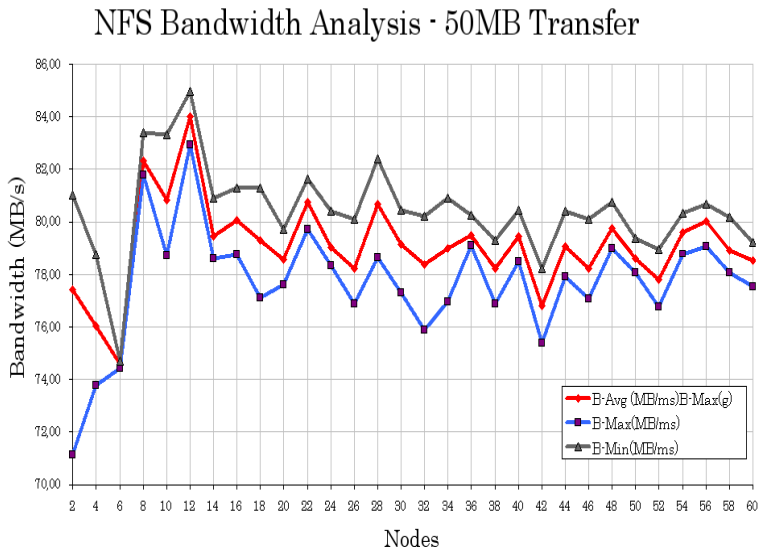


Figure 5.24: *File System Sensibility - Bandwidth for 50MB transfer in NFS*

The two figures show, for both cases, the quick saturation of the network and, after that, a stable behavior. For the $50MB$ transfers in NFS, the values seem to increase. For dNFSp, on the contrary, after the saturation, we can observe a decrease of the bandwidth. However, the decrease and increase of values are not steep. The management done by the dNFSp servers guarantees a stability in the transfer. Those variations are the consequence of the increase of processors (clients). They are minimal as can be seen in Figure 5.25.

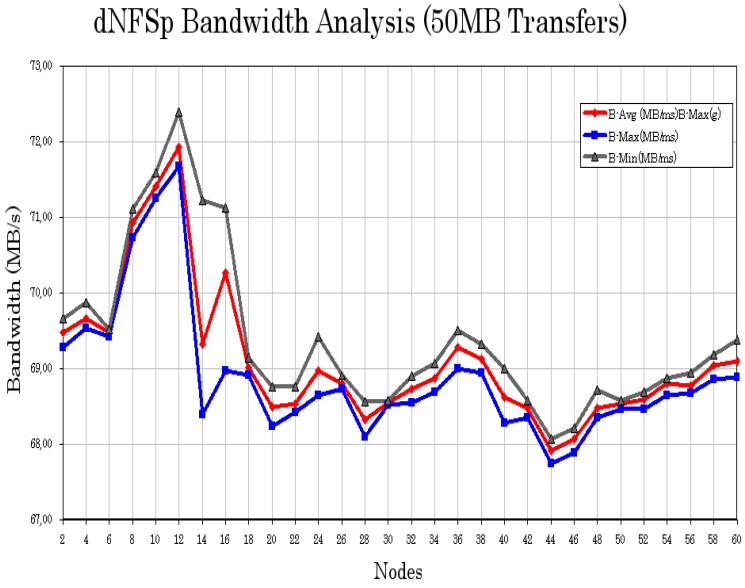


Figure 5.25: *File System Sensibility - Bandwidth for 50MB transfer in dNFSp*

The Figure 5.25 shows the maximal bandwidth (blue), minimal (green) and average (red) as the last Figure 5.24 . In the same way, for each one of the measures is take the maximal influence of time presented before, and also take in account the maximal gap in the transfer for all measures.

Comparing the behavior of the two file systems, it is possible to observe the similar bandwidth performance, the difference lies in the saturation point. Since the bandwidth is also influenced by the number of synchronized connections, it's necessary to observe the total bandwidth as is shown in the Figures 5.26 and 5.27 .

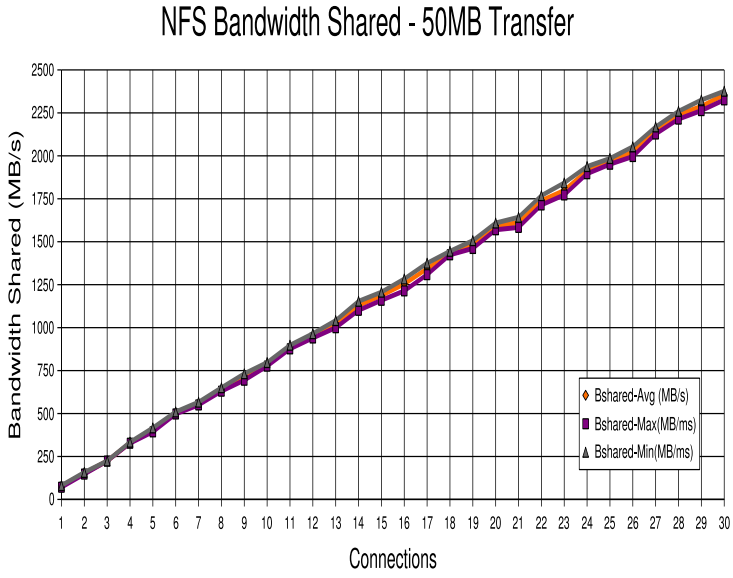


Figure 5.26: *File System Sensibility - Bandwidth Shared for 50MB transfer in NFS*

In the Figure 5.26, a quick look may find that the behavior is similar, however, the space between the two bandwidths grows when the number of connections is increased. Same behavior for the case presented in dNFSp (Figure 5.27).

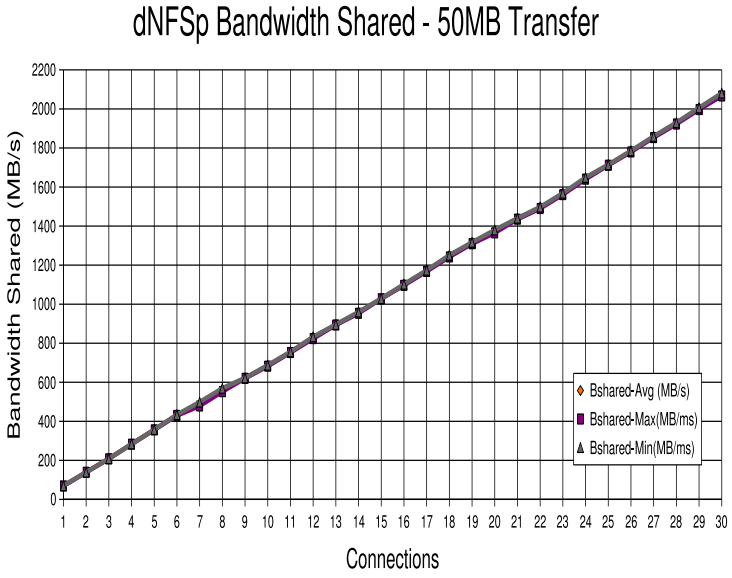


Figure 5.27: *File System Sensibility - Bandwidth Shared for 50MB transfer in dNFSp*

In the Figure 5.26, a quick look may find that the behavior is similar, however, the space between the two bandwidths grows when the number of connections is increased.

The comparison between dNFSp and NFS shows that there are a similar behavior in the transfer for both systems. A first approach suggests that NFS is *slightly better* for this case of transfer. However there is a difference in the configuration of the test, the use of six servers for dNFSp and one for NFS affects the network behavior. Sensibility in network transfer explains this difference, but, the high rate of transfer guaranteed by the increase of the bandwidth if the number of clients grows and the total bytes transferred shows the high efficiency in the data transfer process for dNFSp.

It is clear that several aspects affects the data transfer behavior during the execution of application that implies a data-massive production, transfer and consumption. We exposed in this section a performance measurement of the file systems sensitivity to handle data access and data placement. In the section 4.5, *File System Influence*, we have integrated the effect of the file system in the transfer process to our model, as a delay time in function of a specific file system.

The results presented here shows a quantification of the file system influence in accordance with the characteristics of the transfer (size of message, number of links used between the clients and servers, latency), However to considers the values of $T_{FS}$ is necessary the use of the modeling proposed before in the section 4.5, and made an integration

with the network characteristics in accordance with the expression 4.13.

Now, considering the relation $m_t = 48768$ for a maximal size of block possible in NFS $= 32k$ and a quantity total of blocks transferred for $m = 50MB$, of $= 1524$. The values of $tps$ are between $7,0$ and $7,8$. Then, the $T_{FS}$ estimated should be $0,14$ seconds approximately in these tests.

In the case of the dNFSp, the relation $mt = 49280$ for a maximal size of block possible of $= 64k$ and a quantity total of blocks transferred for $m = 50MB$, of 770. The values of $tps$ are between $5,0$ and $5,7$ Then, the $T_{FS}$ estimated should be $0,19$ approximately for the tests presented before.

In fact, there are other aspects, such as the case with is used a network with a worst performance in relation with the actual capacities of a Grid infrastructure. This *worst* case may be rare, due to the technological possibilities. However, when this *worst performance* occurs, it produces falls in the bandwidth, high delays in several measures, as is presented in the next section.

## 5.5 Anomalies in the Worst Case

Following with the realistic performance using our model, we present the case of the parallel and massive data transfer occurs in low performance networks. Section 4.6 presents a special case for worst cases of transfer. A strategy to characterize the worst behavior in anomalies that has seen in tests has falls of bandwidth or great values of gap, overhead and another parameters due to saturation is to estimate a *anomaly value or parameter*.

Two different scenarios of tests have been used to identify the falls of bandwidth a time of the saturation due to the low capacity of the network or their intensive use, and in consequence, to calculate the anomaly parameter $\bar{A}_p$.

In the first scenario, two old clusters of the Grid'5000 platform interconnected with a normal network (not the high speed network) are used. The results of these tests are presented in past works [10] [11], but not with the analysis of the falls of bandwidth, represented with a parameter of anomalies. The clusters has been the IDPOT Cluster and ICluster-2, in the Grenoble site.

*IDPOT* was an experimental Beowulf cluster at ID-IMAG Laboratory. It is made with 48 nodes Bi-Xeon Dell 1600SC 2.5 GHz processors, 1.5 Gb. RAM ECC and a gigabit network. Since 2004, IDPOT works with Linux 2.4.26. ID-POT does not available for works after 2006.

*I-Cluster2* was the first cluster with Itanium-2 technology in France. This cluster is made of 104 nodes Itanium-2 with 64 bits, 900 MHz and 3.0 Gb. of RAM, a Myrinet and

a Fast Ethernet network. The system provides a total of 208 processors and 312 Gb. of RAM memory and a disc capacity of 7.5 Tb. For more information visit the I-Cluster2 web site.

The second scenario is a high capacity network with full use. This behavior is not evident, but in the experiences show here, it is possible to identify the falls in the bandwidth and estimate different $\bar{A}_p$ values.

Figure 5.28 shows a comparison between IDPOT and I-Cluster bandwidth for a transfer of $100MB$ between pairs of processors. This bandwidth corresponds to a measure using the assumption that exists a link between a pair of processors. Notorious falls in the bandwidth are observed due to rapid saturation that causes bottlenecks in the transfer. The latency is irregular and it presents different values as is shown later.

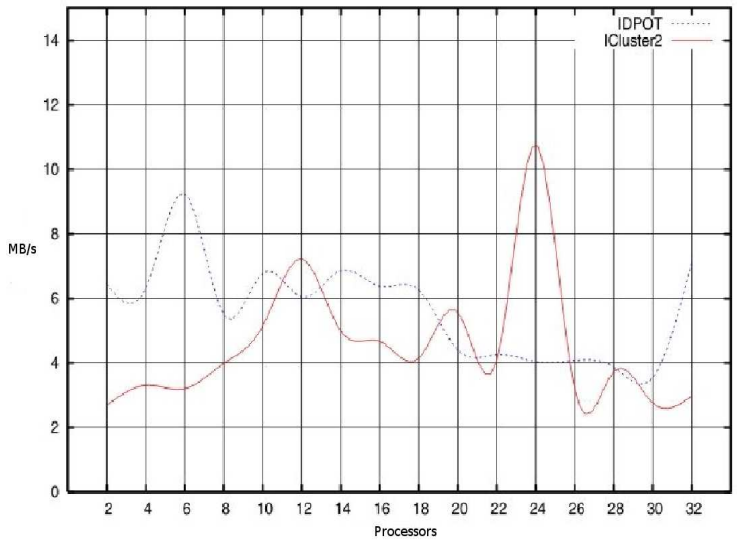


Figure 5.28: *I-Cluster2 and IDPot Clusters Bandwidth Comparison*

Observing the latency behavior in one of the cluster, in this case in I-Cluster2, presented in the Figure 5.29, disturbances increase the latency in determined numbers of pairs. Indeed, the latency is measured by each transfer by link, and as is observable in the Figure 5.29, the behavior is irregular.

In the same way that the Figure 5.28, the delay time due to the quick saturation of the network causes high and irregular values of the latency.

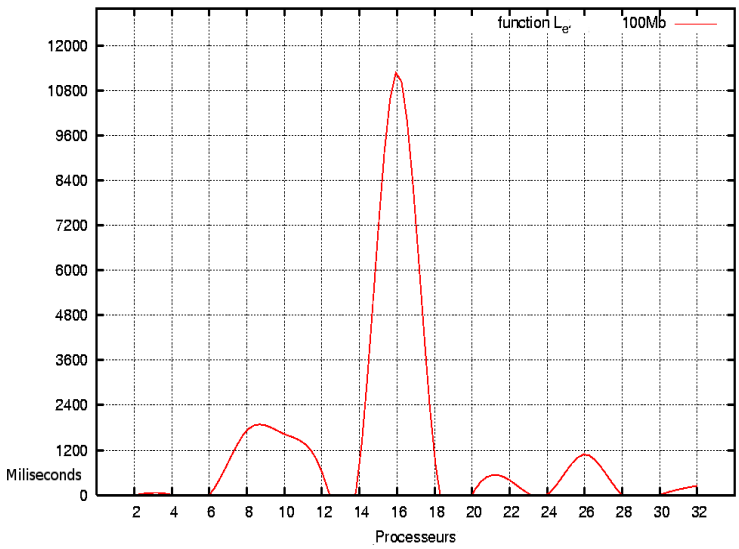


Figure 5.29: *I-Cluster Cluster Latency Disturbances*

Observing Figures 5.28 and 5.29, two intervals of performance are identified, in relation with the number of processors. Then, it is possible to propose two values of $\bar{A}_p$ that correspond to each interval.

Clearly, the values measures for the latency are very high for a normal use. Then, clearly the values possibles for $\bar{A}_p$ will be high ( In this case is $\bar{A}_p << 1$, however a case should be with values of $\bar{A}_p >> 1$).

In the second scenario, using the information of the Grid transfer between GDX and Grillon clusters, is possible to estimate transfer cost in worst cases of network availability, for example, by excessive use that causes congestion in the resources.

Figure 5.30 shows the bandwidth measured for a speed congestion of the resources, due to a high real use in relation with the availability of the network. The values of the bandwidth are low in comparison with the experiences presented before, but the behavior remains stable.

With other observations, for example, with a excessive use in two different networks, also using the information of the GDX-Grillon transfer, the Figure 5.31 appears to show very high values of lambda and in consequence, an excessive cost of transfer due in this case for the great values of the latency. The values of $\lambda$ are predicted using the relation between the latency, the gap $\hat{g}$ and the overheads, presented in the section 4.3.

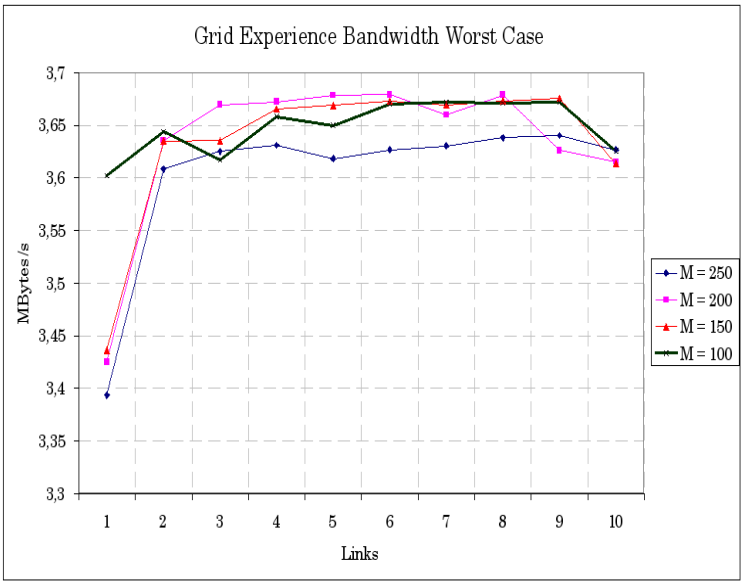


Figure 5.30: *GDX-Grillon Transfer: Bandwidth Worst Case Measurement*

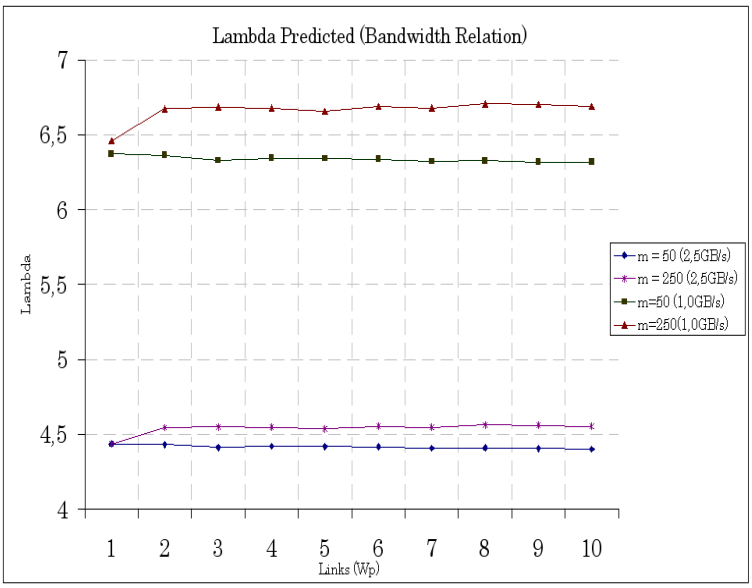


Figure 5.31: *GDX-Grillon Transfer: Lambda predicted for Worst Case*

Different issues for the anomaly parameter may be proposed. For example, $\bar{A}_p$ can be describe as limitations of a determined network or architecture. Also, the anomaly parameter may be useful to predict the behavior of a data transfer for an application that currently runs between two very different networks.

In this sense, the table 5.1 is proposed to show the $A_p$ values comparing the two different measures presented here[6] : a *worst* case for a network of $2.0GB/s$ and a *normal* case for a network of $10GB/s$.

| $W_p$ | $B_{worst}$ | $B$ | $A_p$ |
|---|---|---|---|
| 1 | 3,39 | 3,39 | 0,00 |
| 2 | 3,60 | 3,62 | 0,02 |
| 3 | 3,61 | 3,63 | 0,02 |
| 4 | 3,60 | 3,61 | 0,01 |
| 5 | 3,53 | 3,62 | 0,09 |
| 6 | 3,55 | 3,63 | 0,08 |
| 7 | 3,56 | 3,63 | 0,07 |
| 8 | 3,59 | 3,64 | 0,05 |
| 9 | 3,61 | 3,63 | 0,02 |
| 10 | 3,44 | 3,63 | 0,19 |

Table 5.1: $\bar{A}_p$ for Worst Case of $250MB$ Transfer $2.0GB/s$ .

Nevertheless the values of the $A_p$ should be considered $low$ to be representatives in this two cases, it shows a characteristics value that is observable when we analyze the transfer cost, and more, considering the disturbances in the latency of the network, a feature in low performance networks.

Now, following with our proposition to realistic performance evaluation, the observation of data transfer behavior is necessary to validate the applicability of our proposition. These observations in real systems is presented in the next section.

## 5.6 Observations in Real Systems

In the context of this research, it is very important to validate the use of our model to evaluate the performance in real systems. The reduction of parameters and the methodology techniques proposed in this work, aims to provide an implementable and simplistic model to predict and performance evaluation.

[6]Where B is the bandwidth.

In the last sections, different tests are presented using a specific benchmark tool. However the observation on real systems implies the use of tools and interpretation of the results with respect to the model.

In this chapter, the validation has been made mainly on a specific system: Gedeon. However, other cases are studied and the results of our analysis are described in a special section.

### 5.6.1 Gedeon

The Gedeon Project [49] is a project that search made an hybrid system between a File System Management and a Data Base Management System integrate in a Grid computing middleware to make data management. The project is addressing the bioinformatics community needs.

The work associated with Gedeon, implies data access, synchronization, semantic process, community support and so on. Figure 5.32 shows the position of the Gedeon middleware in a general system of data management for two possible uses. The data come from instruments and are stored in specialized data bases. The treatment of data begins after a request and depending of this, the data access implies only a meta-data access or a completion data transfer.

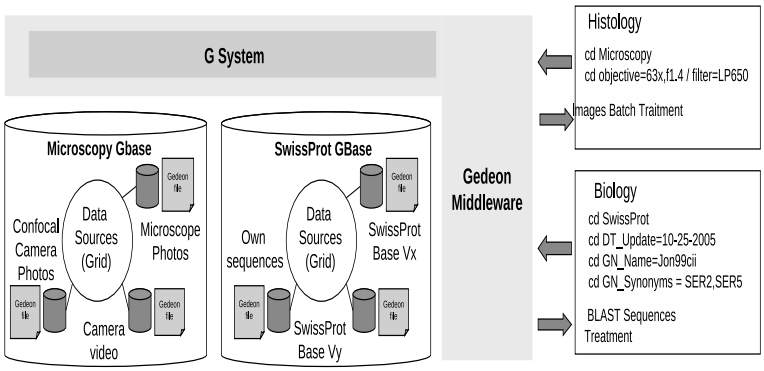


Figure 5.32: *Gedeon System*

In any case the data has been shared between peers and the data management is proposed in different levels, mainly in meta-data level to improve the performance in data access. Figure 5.33 shows the data distribution for two clusters.

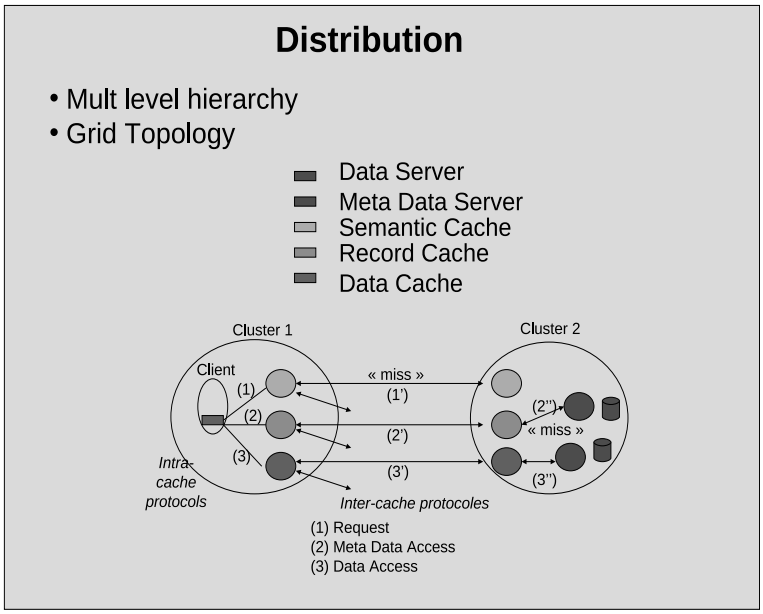


Figure 5.33: *Gedeon Data Distribution*

In the Figure 5.33, there are three phases for the data distribution. First, the request by the client, second, a meta-data access and third, the data-access. In each one of the phases, different entities take part. Obviously, in the first phase, the semantic cache and in the subsequent phases the meta-data and data servers, and the record cache.

The measurement process for this real case of performance evaluation of data transfer implies the use of monitoring tools such as *codes* synthetics. These codes are placed to capture the different parameters of our proposition. Then, we can measure the total transfer time and estimate the parameter $\hat{g}$ using our hypothesis presented before.

The observations include the measured of gap for 4 different requests, that corresponds to 4 different sizes of responses: $54,74MB$, $75,3MB$, $132,83MB$ and $150,39MB$ and 4 different semantic work charges. The tests has been made between 1 to 30 servers, using the GDX and Grillon clusters of the Grid'5000 platform. The references $R90$, $R60$ $R30$ and $R0$ are the semantic work charge, where $R0$ represents an uniform work charge and $R90$ represents $90\%$ of non-uniformity in the requests[7].

Taking the gap as reference measured, the Figure 5.34 shows the measured gap by the observations in Gedeon system. It is interesting to see the fall in the values of the gap with

[7]In fact, this type of requests are part of the general requirements to observe in Gedeon Project to characterize the availability and the response time of the system to different type of requests.

the increase of the number of server used for all type of the requests. The blackout values between 5 to 7 servers are very small values of gap. The latency was 60 microseconds at moment of tests.

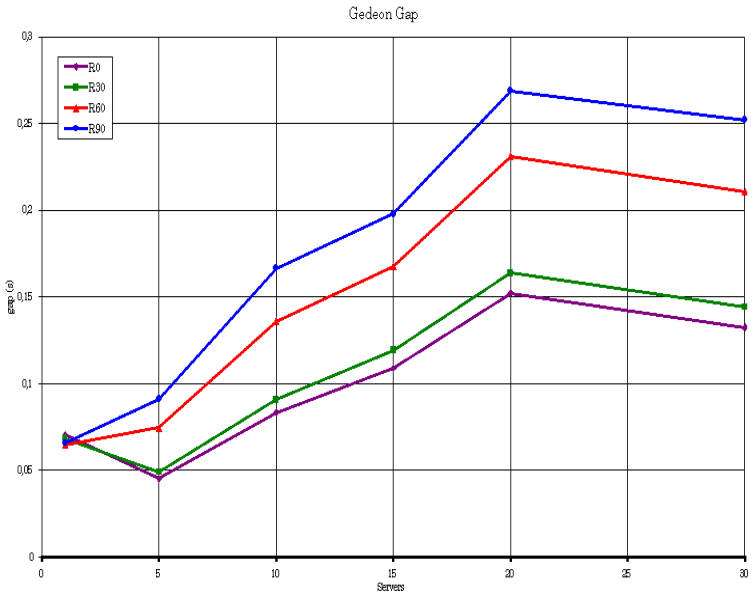


Figure 5.34: *Gedeon Gap measured*

In the Figure 5.34 is observable a relative decreased in the values of the gap. As the size of message transferred by links are not sufficiently great to saturate the network, the values of the gap remains on low values.

Drawing on the information given by the gap, the Figure 5.35 shows the bandwidth presented in the experiences in Gedeon System. The bandwidth estimation was made for 4 different measures, in accordance with the last measures of the gap presented in the Figure 5.34

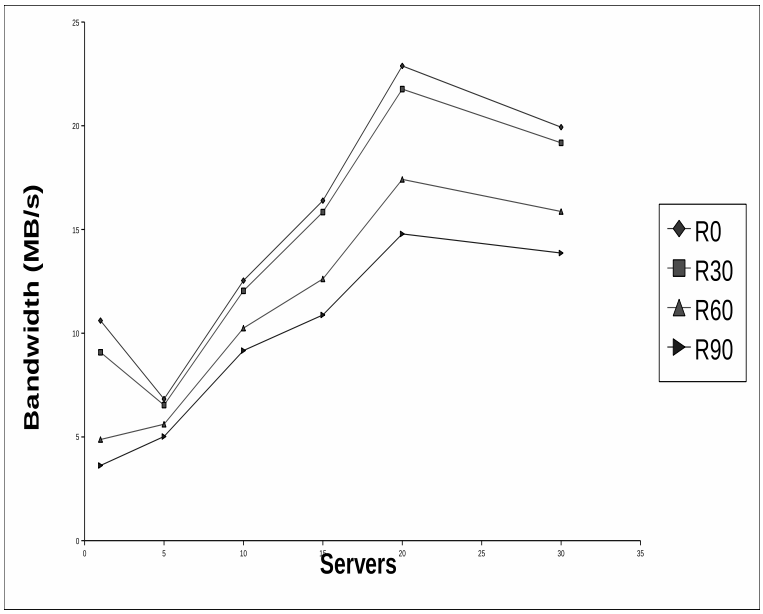


Figure 5.35: *Gedeon Bandwidth Measured*

In the Figure 5.35 the falls of the bandwidth for the $R0$ and $R30$ request, are due by the size of the answers. They are by: $150,39MB$ in the case of the $R0$ and $132,83MB$ for $R30$

Finally, the Figure 5.36 shows the transfer estimated time for the last tests, using our proposition. Indeed, the high values of time are for the transfer which implies more quantity of bytes transferred.

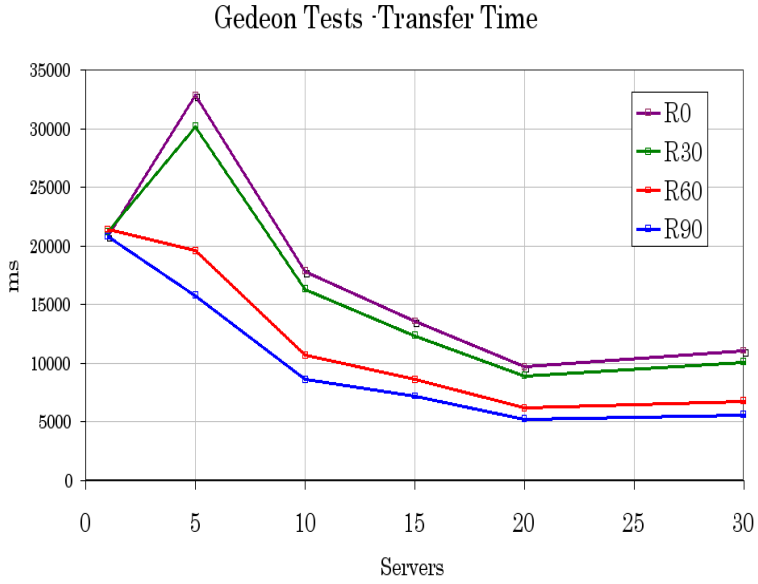


Figure 5.36: *Gedeon Transfer Time*

In all cases, how the transfer time is affected by the $gap$, it exist an increase in the values of the transfer time. However, due to the increase of the capacity of transfer guarantee by the grows in the bandwidth, the transfer time decays to stable values that remains when the number of server increase to 30.

Analyzing the results of these tests in Gedeon System, we can conclude that the bandwidth observed is sufficient to guarantee a stable performance, same when the number of servers and clients is increased. In the same way, for all sizes of transfers and all type of the requests, the data transfer can be reduced, obviously, increasing the number of nodes.

In practical terms, this experience to estimate the performance in terms of data transfer using $LO\hat{g}W_p$ is interesting, because our approach allows to profit the facility to capture the values of the gap and make a relation with the bandwidth, that provides information allows the capacity of transfer of the system.

Due to the type of tests, we have not presented the values of the overhead, because they are very close to the gap values, and in any case for our needs, the estimation of the bandwidth in the use of Gedeon demands consider the gap as reference measure. However, the overhead delays are considered in the estimation of the transfer time, presented in the Figure 5.36, and also the other parameters such the latency and the number of links.

## 5.7 Image Deployment

The methodology has been used in the observation of another cases, mainly in Grid Computing applications that deliver great data set volumes. One of the cases is the deployment of an image in the nodes of Grid'5000 platform, in the context of the ANR-MEG Project [105]. The project aims to develop applications to multi scale modelling for Electromagnetism studies in Grid platforms.

To runs the compute environments is necessary to deploy the image that contains the MEG compute tools in several nodes. Each image contains 600 MB approximately. This deployment in Grid '5000 platform is made with the use of Kadeploy of Ka-Tools [81].

The deployment process in one node requires 250 seconds. Where a important percentage of the time corresponds to transmission of the server in the node. Obviously, when the number of nodes implied in deployment is increased in the same cluster, the deployment time grows reach use the maximal capacity of the network, after this point, the behavior remains.

In the same way, when the nodes implied are of different clusters, the observations shows the same behavior observed in the use of the benchmark tool presented in the chapter 3.1. Figure 5.37 shows the gap observed for two different transfers between two clusters of Grid'5000, in this case Pastel cluster in Toulouse site and Azur cluster in Sophia-Antipolis site . The average of the transfer time is 130 seconds, and they are used two MEG Images, one with $300MB$ and another with $550MB$. The latency was $\approx 60$ microseconds.

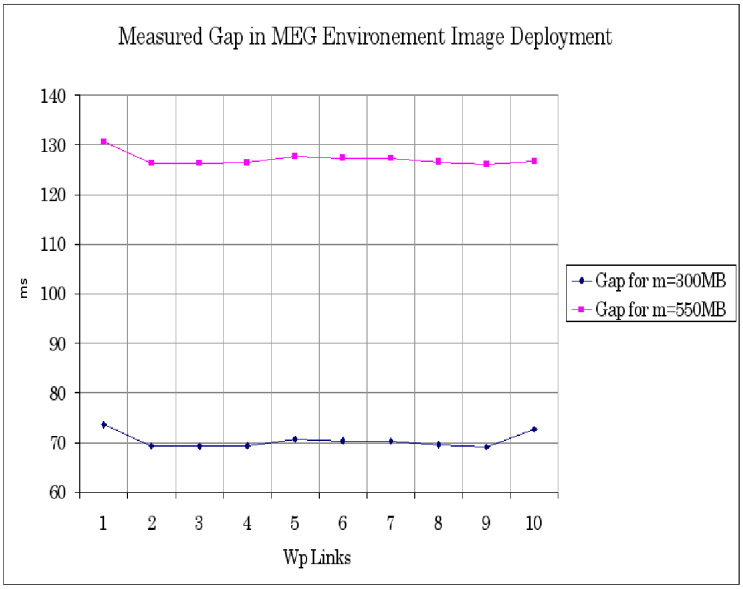


Figure 5.37: *Gap measured in MEG Environment Deployment*

On another hand, the bandwidth in the same transfer is presented in the Figure 5.38. Same that the experiences made with the benchmark tool, the bandwidth grows to a point for 2 links and after remains stable in interval for the two measures between $4,2MB/s$ and $4,5MB/s$.

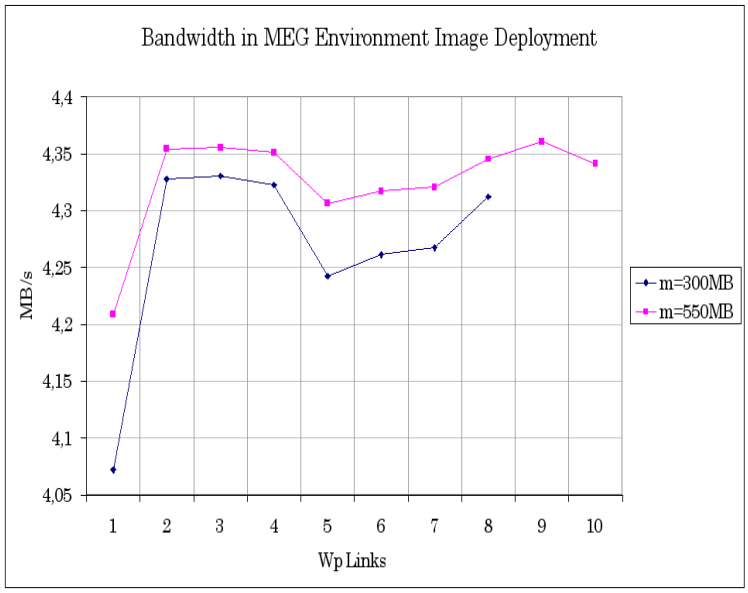


Figure 5.38: *Bandwidth in MEG Environment Image Deployment*

The bandwidth measures for the two transfers show an initial increase, due to de quickly saturation of the link, scaling a most high possible value of bandwidth for the transfer. In the Figure 5.38, we can see that although the difference in bytes transferred, the bandwidth has similar values. Obviously, these values are affected by the increase of the gap, but as is possible to see in the Figure 5.37, the gap remains stable for this type of transfer, but with high values.

This analysis of deployment process using our approach is interesting, because allows to observe the behavior in terms of capacity and use of the network during a massive data transfer, such as the transfer of an environment image among different nodes in real time.

In the same way that the other tests, the capture of the gap values allows to estimate the bandwidth during the data transfer, with a minimal cost for intrusiveness. Also we can estimate the transfer cost observing the $\lambda$ relation and also using the other parameters we can exhibit the different delays associated with the network exchange, as is the case of the overhead for example.

## 5.8 Discussion

Modeling to performance evaluation of distributed environments, as Grid computing, is complex. The implementation and use of the models, such as experiments or monitoring of real systems, is difficult. The utility of a model in performance evaluation is measured not only by its capacity of prediction but also by portability and pertinence in the description provided. On the other hand, the testing and validation of this type of proposals require an understanding of the utility of the mathematical formalisms of the model and a knowledge about the platform testbed and observer/user needs (What is important or interesting to see, and Where?).

In this document we have said that communication is a critical process to measure and monitoring, moreover when the communication process implies distributed systems and a necessity to observe the communications in real time, during a current use. Unfortunately, often *critical* is associated with *complex*. However, in order to propose a realistic approach, useful for the performance evaluation in real time, we propose a parameters reduction and a reinterpretation of the parameters used in $LogP$ for our needs. This approach allows to simplify the observation of the behavior, identify the key parameters and propose a methodology to implement this approach with performance evaluations techniques.

The measurement techniques used give information about the data transfer behavior in terms of network characteristics. Precisely, this information allows to estimate the capacity, availability, loss and delay of the data transmission on the Grid computing platform during the use of a specific application. Results of the tests presented here, shows the utility of our approach, as a systematic and hybrid perspective[8] that permits to develop an analytical model applicable to data transfer on Clusters and Grids environments. Of course, taking into account important architectural differences that affect the behavior.

Clearly, the analysis of the results permits to observe different information to set our case of transfer. For example, the relation between the transfer cost and the size of message transferred by link. Or, the increase of the network activity due to the increase of the number of connection between nodes and also, the influence of the file system activity.

Even thought a model tries to cover a large range of possible cases, is not possible to build *universal* models and the limitations of use of a model must be identified. Our proposed model does not manage fault tolerance, it always supposes that the delivery message arrives to their destination and the prediction is made for the case of parallel, massive and simultaneous transfer in Grid computing platforms. Evidently is possible to treat special situations, such as the case when the data transfer occurs in a non-high bandwidth network of for an use of the resources that cause saturation or congestion. A first

[8] In terms of theoretic and practical point of view.

approach is to estimate the degree of the possible transfer perturbations as disturbances or anomalies.

The anomalies are identifiable as falls in the bandwidth or dramatical and sudden increase of the transfer time, latency, gap or overhead values. In opposition with the general argument of our thesis, that is not adding new parameters, we propose the use of the anomaly parameter to quantify the loss bandwidth. However, is possible to describe the data transfer behavior without use new parameters, such as the case of the network contention, that we can explain in terms of overheads due to the environment exchanges (Cluster-Environment to Grid Environment).

Actually, the advantages of our model proposal and their integration with the measurements permit an easy description of the data transfer behavior and an efficient measurement. Apparently, the descriptions proposed by $pLogP$ or the other models are sufficient to handle the problematic exposed, but as is presented in the development of the past discussions and how the tests shown, the representation of the data transfer process demands the use of network entities, this type of description is not provided by any $LogP$ extensions enough.

## 5.9 Conclusion

The tests presented in this chapter, show the advantages of the $LO\hat{g}W_P$ implementation for performance evaluation techniques and the important information provided by the modeling for various analysis. For example, we can analyze the influence of exchange of data using different file systems or the overhead time due to the change of environment cluster-grid, using real data in real time. In other words, $LO\hat{g}W_P$ model offers good possibilities to realistic performance evaluation.

However it exists limitations clearly observed in the presented tests that suggest additional experiments to identify the precise causes of these limitations. These additional experiments are conducted currently in the context of derived works.

Furthermore, a current additional focus of research is the validation of the model on Grid computing simulation frameworks. This focus allows to explore another possibilities of the model, addressed to design of algorithms to communication management from theoretical assumptions.

# 6. Thesis Conclusion

## 6.1 General Discussion

The production and consumption of large volumes of information in Grid computing systems generate parallel and massive data transfers. So, this type of data transfer is a critical factor in Grid computing application performances[1]. It implies the intensive use of shared network resources between users, applications and services. Often, this high use changes dynamically, involves heterogeneity and concurrency.

Then, the opportunities and capacity offered by Grid Computing become non familiar in terms of efficient use. To handle this, several strategies are proposed to design algorithms to schedule or management communication and data access, based on formal expressions of communication process, models and so on, to implemented protocols. All with the purpose to take advantage of the resources for a minimal cost. However, the problem of the prediction remains. Then, the performance evaluation appears to handle the prediction problem.

Performance evaluation aims to describe systems to predict their behavior. Commonly, the Grid computing and HPC community, propose models to describe transfer characteristics in terms of parallel architectures, based in OGSA. With the different levels identified clearly (infrastructure, resources and applications), it is important to observe the interactions between them, to guarantee an adequate description of the phenomenon (in this case, data transfer) and an good prediction.

On the other hand, we have the need to observe and predict the behavior during the real use of the Grid computing systems. In fact, applications that runs in Production Grid computing platforms can not be interrupted for performance evaluation[2]. A Production Grid computing platform guarantees a pervasive and continuous service, it can not be isolated in one only application execution. A Grid computing application, that runs in

[1] Even though advances in technology provide high capacity in network resources and communication protocols.

[2] In practical terms, because in any case an application should be stopped in any time but is not correct.

these type of platforms, is in concurrence with others and their observation is affected by this situation. So, the techniques of monitoring implemented to observe and analyze behaviors during a *real time* should not be complex and they cannot add overheads due to intrusiveness in the system in observation.

The research involved in this thesis work, propose an analytical model that considers infrastructure characteristics and application specifications to the specific problem of the parallel and massive data transfer on Grid computing platforms. Of course, the utility of the analytical model is in terms of prediction and implementation in easy performance evaluation techniques. Our proposal uses few parameters to analyze the behavior and the measurement of these parameters, as is presented before, is easily and it can be made using synthetics codes (implemented in benchmark tools or monitor systems). Indeed, the problem proposed is at the same time a challenge because it eliminates the complexity of the problem to guarantee implementation of the model in *cheapest*[3] performance evaluation techniques.

Performance evaluation community classified this type of approach as *hybrid point of view* of performance evaluation [77] [90] [96] [130]. This hybrid point of view, demands a detailed analysis of all network characteristics and transfer scenarios to select key parameters to provide information about the data transfer behavior. Also, the model must be simple and easily implementable. Several models consider interesting scenarios, but for example, implies a large quantity of parameters or sophisticated mathematical mechanisms that hardens their implementation and limits their application to very particular cases.

Our proposition has been tested and the obtained results shown their easiness in their implementation[4]. The collected information by the techniques implemented allow to made descriptions of behavior in terms of capacity, availability and use, using measures of bandwidth, latency, transfer times, transfer delays between others, presented in the content of this manuscript. Obviously, this information allows to characterize the performance of an application in a moment and take decision during their execution (increase or decrease of resources in use, network or local platform selection, prediction of conclusion time, bottlenecks, bandwidth loss).

[3] Cheap in terms of cost and intrusiveness.

[4] This *easiness* has an additional consequence that is the low delay added in the measures by their low intrusiveness cost.

## 6.2 Contributions

The general contribution of this thesis is to offer an approach to understand the influence of the network and communication characteristics in applications running in Grid computing platforms that make massive and parallel data transfer on high bandwidth networks associated with Grid computing systems.

### 6.2.1 Performance Evaluation

The modelling proposed in this work provides performance evaluation contributions that permit identifying the behavior in terms of capacity, availability and use. The measurement techniques derived of the analytical model are easily implemented without adding important or non controlled workloads.

In another words, our proposition provides an easy and portable methodology to test high bandwidth data transfer[5].

The model allows a systematic approach, that in the same sense, provides mechanisms to evaluate the high bandwidth data transfer in different environments, taking architectural characteristics and defined a correct performance metric to describe each one of the measures. For example, the descriptions made about the bandwidth by links or the bandwidth in the switch, use common network measures. The various performance criteria considered permit individual and global measurements, as is presented in the long of the chapter 5, *Realistic Performance Evaluation using* $LO\hat{g}W_p$.

Consequently, this analytical modelling permits the design of new tests and experiments. The measurement can be validated by another techniques and drop the experimental non familiar related with the complexity of the systems in observation.

Furthermore, analysis of data provided by the measurement methodology, allows to identify derived factors or consequences of the measures, such as contention, transfer cost, transfer time between others.

For example, in the case of the measurement made in cluster platforms, we capture easily the values associated with the parameters of $LO\hat{g}W_p$ (in the same way that $pLogP$), making the description of the behavior with network characteristics. Of course, the main advantages are for the case of the Grid computing transfer, where we can use the $LO\hat{g}W_p$ model to treat transfer between heterogeneous platforms, treat the environment exchange and estimate transfer costs, availability of the resources, capacity in real time, between others characteristics.

[5]The term *high bandwidth* used in this work is associated with faster connections and high capacity of transference.

Finally, the measurement techniques provide here can be implemented in monitors, sensors of systematic programs as benchmark tools. As is the case of the codes implemented in the benchmark tool used, the *LogP MPI Benchmark Multitest Tool* and in our further modification of this benchmark tool.

In the case of the monitors, they can be considered as invasive mechanisms that add a known workload. For the sensors, they are non-invasive mechanisms, then it is possible to capture measures without add important workloads, and benchmark tools offer possibilities of comparison. The different possibilities has been explored in this work and the results presented before, confirm this contribution, as is possible to see in the chapter 5, *Realistic Performance Evaluation using* $LO\hat{g}W_p$.

In synthesis, we arrive to contribute with a realistic performance evaluation technique to treat massive and parallel data transfer on high bandwidth networks into Grid computing environments. This realistic performance evaluation technique is addressed to observe in real time the data transfer behaviors during the execution of Grid computing applications .

### 6.2.2 Modeling

The modelling contribution allows to propose a description of the high bandwidth data transfer process in terms of network characteristics. The analytical modelling begins from the *LogP* family assumptions, and our model is an extension of the model. In opposition with other extensions, in this description one parameter is added [6] and it proposes a re-interpretation of the key parameters: Latency $L$, overhead $O$, in this case takes into account the overhead due to the environment exchange mainly ($O_{cgc}$), $G$ gap by byte for long messages, but for us is the gap by $k - packet$ ($\hat{g}$) of a message of size $m$ and the number of processors $p$ involved, for us is more important the links that interconnect the nodes of the Grid computing infrastructure ($W_p$). In consequence, characteristics as the bandwidth $B$ (for example using the equations 4.4 or 4.6 ) to handle data transfer capacity, $T_{FS}$ to measure the influence of the File System (equation 4.13) in the data transfer process, the overhead time (equation 4.11) due to environment exchange between others should be estimated with several formal expressions for different cases, as is presented before in this manuscript.

On another hand, interesting consequences of the modelling contribution is the definition terms as massive/intensive data transfer and high bandwidth. This definition presents the high bandwidth as a bandwidth that implies the massive data transfer.

Indeed, massive data transfer is defined in terms of size of message $m$ transferred by link. Then, if the message $m$ contains a quantity of bytes greater than the underlying

[6] Only one, the $\bar{A}p$, in the specific worst case of network to explain disturbances.

packet size capacity $w$, that cut the message in $k - packets$, the message is great. For example, the underlying packet capacity for a TCP transfer can be the window transmission size.

As the message is divided in $k - packets$ and transferred consecutively, this situation involves a intensive data transfer. However, in practical terms, the intensive data transmission not depends of the size of the packages transferred. Thus, accordingly with these two hypothesis, the transfers analyzed in this work are massive/intensive data transfers.

To discuss high bandwidth, in this work is proposed the hypothesis of a high bandwidth connection have a great capacity to transmission data sets. In consequence, a high bandwidth data transfer supports massive data transfer and is defined in terms of this type of transfers.

Summarizing, we propose a model based in experimental observations to describe and predict parallel and massive data transfer among nodes during the execution of applications running on Grid Computing Platforms.

## 6.3 Future Works

Communication process in HPC and Grid computing is an active domain of study. The contributions offer by the results of this thesis leave more open questions that solutions. Our performance evaluation and prediction approach has allowed to limit the problematic, however, different study proposals can be addressed.

- For Analytical Modelling, an interesting perspective is the addition of more complexity, without adds new parameters. The objective is treat other possible cases, conserving the principle of easy implementation of the model in performance evaluation methodologies. Between the possible cases to study are non completion transfers that implies analysis of fault tolerance, asynchronous communications, non-collective communicants and active messages [146].

- Mathematical formalisms proposed to describe the different transfer characteristics can be used to define new entities or existing concepts. Is the case for example of our hypothesis to define High Bandwidth, massive and intensive transfer, underlying capacity, etc. Several concepts exist in Grid computing and it is necessary to build definitions that allows to made standard descriptions. And in consequence, these definitions may be used to identify specific network characteristics for existing network entities and new entities proposed[7].

[7]Due for example, to technology evolution.

- Another interesting perspective, derived of the last one is the integration in the model of different network entities, as non HPC nodes (such as non permanent devices or compute nodes, instruments, specific networks (as wireless networks) in order to propose a more general model to explain the behavior for any distributed systems, such as High Throughput Computing systems, Cloud Computing Systems, Voluntary Desktop Computing Systems.

- Exploiting the possibilities of the analytical model, another perspective is the placement of the model in a simulator framework. This approach permits to know mathematic limitations of the model and a comparison with others, in this case not only to performance evaluation beings, also to design and description of algorithms to schedule and data management.

- On another hand, an interesting possibility is use the analytical modelling to propose models to design data-transfer algorithms that can be implemented in schedulers, data managers, data transfer protocols between others. Always conserving the simplicity and portability of the modelling results.

- For Performance Evaluation perspective, a point important is the characterization of different protocols, architectures, network topologies and algorithm implementations to build description tables for specific platforms . This information is useful to planning infrastructure utilization or infrastructure implementations, to treat for example, scalability and heterogeneity.

- Finally, design and build monitors, sensors, tracers and benchmarking tools based in the analytical modelling proposed in this work, provides not only tools to predict with a high degree of accuracy and portability. Also, it is possible to present new performance measurement techniques that can be implemented in new or existents performance evaluation tools.

# Bibliography

[1] Aggarwal, A., Chandra, A., and Snir, M. *On Communication Latency in PRAM Computations.* Proceedings of the First Annual ACM Symposium on Parallel Algorithms and Architectures, Santa Fe, New Mexico, United States of America 1988.

[2] Akl, S., Cosnard, M. and Ferreira G. *Data-Movement-Intensive Problems : Two Folk Theorems in Parallel Computation Revisited.* Theoretical Computer Science Journal, Volume 95 , Issue 2, Pages: 323 - 337. United States of America, 1992.

[3] Alexandrov, A., Ionescu, M., Schauser, K. and Scheiman, C. *LogGP: Incorporating Long Messages into the LogP Model - One Step Closer Towards a Realistic Model for Parallel Computation*. Proceedings in 7th Annual ACM Syposium on Parallel Algorithms and Architectures, Santa Barbara, California, U.S.A. 1995.

[4] Allcock, B., Bester, J., Bresnahan, J., Chervenak, A., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnal, D and Tuecke, S. *Data Management and Transfer in High Performance Computational Grid Environments.* Parallel Computing Journal, Vol. 28, No. 5. Elsevier Science Publishers, Netherlands, 2002.

[5] Amme, W., Braun, P., Lowe, W. and Zehendner, E. *LogP Modelling of List Algorithms.* Proceedings 11th Symposium on Computer Architecture and High Performance Computing, Natal, Brasil 1999.

[6] Albatross Project Team. *Albatross Project Site.* http://www.cs.vu.nl/albatross/ Internet Site.

[6] Baldwin-Gibbons, P. *The Asynchronous PRAM : A Semi-Synchronous Model for Shared Memory MIMD Machines*. PhD. Thesis. University of California at Berkeley, United States of America 1989.

[7] Bampis, E., Guinand, F. and Trystram, D. *Some Models for Scheduling Parallel Programs with Communication Delays.* ACM Discrete Applied Mathematics Journal, Vol-

ume 72 , Issue 1-2, Special issue on models and algorithms for planning and scheduling problems, United States of America 1997.

[8] Bar-Noy, A. and Kipnis, S. *Designing Broadcasting Algorithms in the Postal Model for Message-Passing Systems.* Mathematical Systems Theory Journal. Volume 27, Issue 5, Special issue: ACM Symposium on Parallel Algorithms and Architectures, Pages: 431 - 452, Springer-Verlag New York, Inc. Secaucus, NJ, United States of America 1994.

[9] Bar-Noy, A. and Nir, U. *The Generalized Postal Model-Broadcasting in A System Withnon-Homogeneous Delays.* Proceedings in 9th Mediterranean Electrotechnical Conference, MELECON'98, Tel-Aviv, Israel 1998.

[10] Barchet-Estefanel, L., *LaPIe: Communications Collectives Adaptées aux Grilles de Calcul*. PhD. Thesis (In French), Institut National Polytechnique de Grenoble, Grenoble, France, 2005.

[11] Barrios-Hernández, C. and Denneulin, Y. *High Bandwidth Data Transfer Analysis and Modelling in Grids.* Proceedings in EXPGRID - Experimental Grid testbeds for the assessment of large-scale distributed applications and tools in The 15th IEEE International Symposium on High Performance Distributed Computing (HPDC-15 EXPGRID'06), Paris, France 2006.

[12] Barrios-Hernández, C., Denneulin, Y. and Riveill, M. *High Bandwidth Anomalies in Cluster and Grid Data Transfer*. Proceedings of the International Workshop on Advanced Topics in Network Computing, Technology, and Applications in conjuction with IFIP International Conference on Network and Parallel Computing (NPC), IEEE Computer Society. Dalian, China, 2007.

[13] Barry, D. (Editor) *Web Services and Service-Oriented Architecture.* Barry and Associates Inc. http://www.service-architecture.com Internet Site.

[14] Berman, F., Fox, G. and Hey, T. (Editors) *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley and Sons Ltd., West Susesex, England 2003.

[15] Bernaschi, M. and Iannello, G. *Collective Communications Operations: Experimental Results Vs. Theory* Concurrency: Practice and Experience, Vol 10(5), pag 359-386. John Willey and Sons ltd. 1998.

[16] Bilardi, G., Herley, K., Petracaprina, A., Pucci, G. and Spirakis, P. *BSP vs LogP.* Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures, Padua, Italy, 1996.

[17] Bosque, J.L. and Perez, L.P. *HLogGP: A New Parallel Computational Model for Heterogeneous Clusters.* Proceedings of IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004, Chicago, Illinois, U. S. A April 19-22, 2004

[18] Braden, R. (Editor) *RFC1122 - Requirements for Internet Hosts - Communication Layers* Network Working Group - Internet Engineering Task Force (IETF). http://www.faqs.org/rfcs/rfc1122.html Internet Document 1989.

[19] BSP Community *BSP Community Site.* http://www.bsp-worldwide.org/ Internet Site.

[20] Buyya, R. and Murshed, M. Journal of Concurrency and Computation: Practice and Experience, Volume 14, Issue 13-15, Pages 1175 - 1220, John Wiley and Sons, Ltd. United States of America 2002.

[21] Cysco Systems Group. *Internetworking Technology Handbook.* Third Edition. Cysco Systems Press Inc. United States of America, 2000.

[22] Cameron, K. and Ge, R. *Predicting and Evaluating Distributed Communication Performance.* Proceedings of the 2004 ACM/IEEE Conference on Supercomputing, Conference on High Performance Networking and Computing, Pittsburgh, PA, United States of America 2004.

[23] Casanova, H., *Network Modelling Issues for Grid Application Scheduling.* International Journal of Foundations of Computer Science, Vol. 16, No. 2. Pags 145-162. World Scientific Publishers Co. Mountain View, CA. United States of America 2005.

[24] Casanova, H, Legrand, A. and Quinson, M. *SimGrid: a Generic Framework for Large-Scale Distributed Experimentations.* Proceedings of the 10th IEEE International Conference on Computer Modelling and Simulation, Cambridge, England 2008.

[25] Chervenak, A., Deelman, E., Livny, M., Su, M., Schuler, R., Bharathi, S., Mehta, G., and Vahi, K *Data Placement for Scientific Applications in Distributed Environments*. Proceedings of the 8th IEEE/ACM Internation Conference on Grid Computing 2007. Austin, United States of America 2007.

[26] CLUMSSY Project Team. *CLUMSSY: Cluster Management and Scheduling System Site.* http://www.inf.ufrgs.br/clumssy/ Internet Site.

[27] Cosnard, M. and Ferreira, A. *Designing Parallel Non Numerical Algorithms.* Proceedings in International Conference in Parallel Computing '91. North Holland, Netherlands 1991.

[28] Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K., Santos, E., Subramonian, R. and Von Eicken, T. *LogP: Towards a Realistic Model of Parallel Computation.* Proceedings in Four ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, San Diego, California, U.S.A. 1993.

[29] Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K., Santos, E., Subramonian, R. and Von Eicken, T. *LogP: A Practical Model of Parallel Computation.* Communications of the ACM, Volume 39 , Issue 11, Pages: 78 - 85, United States of America 1996.

[30] DAS-3 Project Team. *The Distributed ASCI Supercomputer (DAS-3) Project Site.* http://www.starplane.org/das3/ Internet Site.

[31] Den Burger, M., Kielmann, T. and Bal, H. *TOPOMON: A Monitoring Tool for Grid Network Topology.* Lecture Notes In Computer Science; Vol. 2330, Special Issue to Proceedings of the International Conference on Computational Science-Part II, Springer-Verlag London, United Kingdom 2002.

[32] Dusseau, A., Culler, D., Schauser, K. and Martin, R. *Fast Parallel Sorting Under LogP: Experience with the CM-5.* IEEE Transactions on Parallel and Distributed Systems, Volume 7 , Issue 8, Pages: 791 - 805, IEEE Press, Piscataway, NJ, United States of America 1996.

[33] Eisenbiegler, J., Lowe, W. and Wehrenpfennig, A. *On the Optimization by Redundancy Using an Extended LogP Model.* Proceedings in Advances in Parallel and Distributed Computing, Shanghai, China 1997.

[34] EU - EGEE Consortium. *European Union - Enabled Grid for E-Science*. http://www.eu-egee.org . Internet Site.

[35] EU-EELA Consortium. *European Union - E-Science Grid Facility for Europe and LatinAmerica* http://www.eu-eela.eu/ . Internet Site.

[36] Faerman, M.; Su, A.; Wolski, R. and Berman, F. *Adaptive Performance Prediction for Distributed Data-Intensive Applications*. Proceedings of the 1999 ACM/IEEE Conference on Supercomputing. Portland, Oregon, U.S.A., 1999.

[37] Feiltelson, D. *Experimental Computer Science.* Introduction to Special Issue about Experimental Computer Science in Communications of the ACM, Volume 50, Issue 11 (November 2007), ACM New York, United States of America 2007.

[38] Feitelson, D. *Experimental Computer Science Site.* http://www.cs.huji.ac.il/ feit/exp/ . Internet Site.

[39] Ferrari, T. and Giacomini, F. *Network Monitoring for GRID Performance Optimization.* Computer Communications, Volume 27, Issue 14, Pages 1357-1363, Elsevier Science Publishers, Netherlands 2004.

[40] Frank, M., Agarwal, A. and Vernon, M. *LoPC: Modelling Contention in Parallel Algorithms.* ACM SIGPLAN Notices, Volume 32, Issue 7, Pages: 276 - 287, New York, NY, United States of America 1997.

[41] Fortune, S. and Wyllie., J. *Parallelism in Random Access Machines.* Proceedings of the 10th Annual ACM Symposium on Theory of Computing, San Diego, CA, United States of America 1978.

[42] Foster, I. and Kesselman, C. (Editors) *The Grid 2: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers - Elsevier Inc. New York, United States of America. 2004.

[43] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[44] Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramanian, R., Treadwell, J. and Von Reich, J. *The Open Grid Services Architecture, Version 1.0.*. Open Grid Forum Document GFD-I.030 Open Grid Services Architecture, http://forge.gridforum.org/projects/ogsa-wg Internet Document 2005.

[45] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. *Grid Services for Distributed Systems Integration.* IEEE Computer Volume 35, Issue 6, United States of America, 2002.

[46] Fraigniaud, P. *On XRAM and PRAM Models, and On Data-Movement-Intensive Problems.* Note, Theoretical Computer Science, Volume 194, Issues 1-2, Pages 225-237 Elsevier Science Publishers, Netherlands 1998.

[47] Frederiksen, J., Larsen, K., Noga, J. and Uthaisombut, P. *Journal of Algorithms.* Volume 48, Issue 2, Pages: 407 - 428, Academic Press, Inc. Duluth, MN, United States of America 2003.

[48] Ganglia Project Team. *Ganglia Monitoring System Site.* http://ganglia.info/. Project Internet Site.

[49] Gedeon Project Team. *Gedeon Project Site.* http://www-lsr.imag.fr/Gedeon/ . Project Internet Site

[50] Gerbessiotis, A., Lecomber, D., Siniolakis, C.J and Sujithan, K. *PRAM Programming: Theory vs. Practice.* Proceedings of the Sixth Euromicro Workshop on Parallel and Distributed Processing, Madrid, Spain 1998.

[51] Gerbessiotis, A. and Siniolakis, C. *Communication Efficient Data Structures on the BSP Model with Applications in Computational Geometry.* Lecture Notes In Computer Science; Vol. 1124, Special Issue to Proceedings of the Second International Euro-Par Conference on Parallel Processing-Volume II, Pages: 348 - 351, Springer Editors, Berlin, Deutchland 1996.

[52] Gerbessiotis, A. and Valiant, L. *Direct Bulk-Synchronous Parallel Algorithms.* Journal of Parallel and Distributed Computing, Volume 22 , Issue 2, Pages: 251 - 267, Academic Press, Inc. Orlando, FL, United States of America1994.

[53] Goglin, B. and Prylli, L. *Performance Analysis of Remote File Systems Access over High-Speed Local Network*. Proceedings of the 18th International Parallel and Distributed Processing Symposium, IPDPS 2004. Santa Fe, New Mexico, United States of America, 2004.

[54] Grid Cafe Team. *Grid Café* http://www.gridcafe.org Internet Site.

[55] Grid 5000 Team. *Grid 5000 Project Site* http://wwww.grid5000.fr Internet Site.

[56] Grid Talk. *Grid Talk* http://www.gridtalk-project.eu Internet Site.

[57] Gu, Y. and Grossman, R. *SABUL: A Transport Protocol for Grid Computing*. Journal of Grid Computing. Volume I. No. 4. Kluwer Academic Publishers, Springer Publishers, Netherlands 2003.

[58] Guillier, R., Hablot, L. and Vicat-Blanc Primet, P. *RR-6244 - Towards a User-Oriented Benchmark for Transport Protocols Comparison in very High Speed Networks.* Internal Report (In French), Inria Rhône-Alpes, ENS Lyon, France 2007.

[59] Guillier, R., Hablot, L., Vicat-Blanc Primet, P. and Soudan, S. *RR-6047 - Evaluation des Liens 10 GbE de Grid'5000.* Internal Report (In French), Inria Rhône-Alpes, ENS Lyon, France 2006.

[60] Guillier, R., Hablot, L., Kodama, Y., Kudoh, T., Okazaki, F., Vicat-Blanc Primet, P., Soudan, S. and Takano, R. *RR-6034 -A Study of Large Flow Interactions in High-Speed Shared Networks with Grid5000 and GtrcNET-10 Instruments.* Inria Rhône-Alpes, ENS Lyon, France 2006.

[61] Harakaly, R., Primet, P. and Bonnassieux, F. *Grid coordination by using the Grid coordination protocol*. Proceedings on IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004. Chicago, Illinois, United States of America, 2004.

[62] Hayes, B. *Cloud Computing.* Communications of the ACM, Volume 51, Issue 7, Pages 9-11, ACM New York, United States of America 2008.

[63] Hennessy, J. and Patterson, D. *Computer Architecture: A Quantitative Approach, Third Edition*. Morgan Kaufmann Publishers, San Francisco CA, United States of America, 2003.

[64] Hermman, E., Kassick, R., Avila, R., Barrios-Hernandez, C., Riveill, M., Denneulin, Y. and Navaux, P. *Performance Evaluation of Meta Data Transfer and Storage in Clusters*. In Proceedings of the Latinamerican Conference of High Performance Computing, pages 127-134. Santa Marta, Colombia., 2007.

[65] Hertel, C. *Implementing CIFS: The Common Internet File System* http://www.ubiqx.org/cifs/ Internet Open Book 2003.

[66] Hochstein, L., Basili, V., Vishkin, U. and Gilbert, J. *A Pilot Study to Compare Programming Effort for Two Parallel Programming Models.* Journal of Systems and Software, Volume 81, Issue 11, Pages 1920-1930, United States of America 2008.

[67] Hockney, W. *The Communication Challenge for MPP: Intel Paragon and Meiko CS-2*. Volume 20, Issue 3, Pages: 389 - 398, Elsevier Science Publishers B. V. Amsterdam, Netherlands 1994.

[68] Hoefler, T., Cerquetti, L., Mehlan, T., Mietke, F. and Rehm, W. *A Practical Approach to the Rating of Barrier Algorithms Using the LogP Model and Open MPI.* Proceedings in International Conference Workshops on Parallel Processing, ICPP 2005 Workshops, Oslo, Norway 2005.

[69] Hou, Q., Zhou, K. and Guo, B. *BSGP: Bulk-Synchronous GPU Programming.* ACM Transactions on Graphics (TOG), Volume 27, Issue 3, Special Issue to SIGGRAPH 2008, Article No. 19, United States of America 2008.

[70] Iannello, G. *Efficient Algorithms for the Reduce-Scatter Operation in LogGP.* IEEE Transactions on Parallel and Distributed Systems, Volume 8, Issue 9, Pages: 970 - 982, IEEE Press Piscataway, NJ, United States of America 1997.

[71] IEEE Standards Association. *IEEE Standards Association Site.* http://standards.ieee.org/ Internet Site.

[72] IEEE Standards Association. *IEEE 802 Standards for LAN/MAN CSMA/CD Access Method*. IEEE Standards Publications. United Stated of America 1983 - 2008.

[73] IEEE Standards Association. *IEEE 802.3 Standards for LAN/MAN CSMA/CD Access Method.* IEEE Standards Publications. United Stated of America 1983 - 2008.

[74] IEEE 802.3 ETHERNET WORKING GROUP. *IEEE P802.3ba 40Gb/s and 100Gb/s Ethernet Task Force.* IEEE Task Force Publications. United States of America 2008.

[75] Ino. F., Fujimoto., N. and Hagihara., K. *LogGPS: A Parallel Computational Model for Synchronization Analysis*. Proceedings of the 2001 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPOPP'01), Snowbird, Utah, USA, June 18-20, 2001 .

[76] Intel Inc. *ntel® Trace Analyzer and Collector 7.2 for Linux* or Windows*.* http://www.intel.com/cd/software/products/asmo-na/eng/306321.htm Internet Site.

[77] Jain, R. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons Inc. New York, United States of America 1991.

[78] Jin, D. and Ziavras, S.G *A Super-Programming Approach for Mining Association Rules in Parallel on PC Clusters.* IEEE Transactions on Parallel and Distributed Systems, Volume 15, Issue 9, Pages 783 - 794. United Stedes of America 2004.

[79] Joseph, J., Ernest, M. and Felleinstein, C. *Evolution of Grid Computing Architecture and Grid Adoption Models*. IBM Systems Journal. Volume 43., No. 4, United States of America, 2004.

[80] Juurlink, B. and Wijshoff, H. *A Quantitative Comparison of Parallel Computation Models.* ACM Transactions on Computer Systems (TOCS), ACM, New York, United States of America 1998.

[81] Ka-Tools Team. *Ka Clustering Tools Site.* http://ka-tools.imag.fr/ . Internet Site.

[82] Kassick, R., Machado, C., Hermann, E., Boher-Avila R., Navaux, P. and Denneulin, Y. *Evaluating the Performance of the dNFSp File System.* Proceedings of Cluster and Grid Computing Conference 2005. Cardiff, United Kingdom 2005.

[83] Karp, R., Luby, M. and Meyer auf der Heide, F. *Efficient PRAM Simulation on A Distributed Memory Machine.* Algorithmica Journal, Volume 16, Numbers 4-5, Pages 517-542, Springer Editors, New York, United States of America 1996.

[84] Karp, R., Sahay, A., Santos, E. and Schauser, K. *Optimal Broadcast and Summation in the LogP Model.* Proceedings of the Fifth Annual ACM Symposium on Parallel Algorithms and Architectures, Velen, Germany 1993.

[85] Kavoussanakis, K., Phipps, A., Palansuriya, C., Trew, A., Simpson, A. and Baxter, R. *Federated Network Performance Monitoring for the Grid.* Proceedings in 3nd International Conference in Broadband Communications, Networks and Systems, BROADNETS 2006, San José, CA, United States of America 2006.

[86] Khan, M.; Paul, R.; Ahmed, I. and Ghafoor, A. *Intensive Data Managment in Parallel Systems: A Survey*. Distributed and Parallel Databases. Vol. 7, Pages: 383-414. Kluwer Academic Publishers. Netherlands, 1999.

[87] Kielmann, T., Bal, H., and Verstoep, K. *Fast Measurement of LogP Parameters for Message Passing Platforms*. Lecture Notes In Computer Science; Vol. 1800, Proceedings of the 15th IPDPS 2000 Workshops on Parallel and Distributed Processing, Springer - Verlag, U.K., 2000.

[88] Kielmann, T., Hofman, R., Bal, H., Plaat, A. and Bhoedjang, R. *MAGPIE: MPI's Collective Communication Operations for Clustered Wide Area Systems.* Proceedings in Seventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'99), Pages 131-140, Atlanta, GA, United Stats of America 1999.

[89] Kort, I. and Trystram, D. *Assessing LogP Model Parameters for the IBM-SP.* Lecture Notes In Computer Science; Vol. 1470, Special Issue to Proceedings of the 4th International Euro-Par Conference on Parallel Processing, Pages: 255 - 262, Springer-Verlag London, UK 1998.

[90] Kurian-John, L. and Eeckhout, L. *Performance Evaluation and Benchmarking.* CRC Press, Taylor and Francis Group, Boca Raton, FL, United States of America 2006.

[91] Lebre, A. *aIOLi: Contrôle, Ordonnancement et Régularisation des Accès aux Données Persistantes dans les Environnements Multi-applicatifs Haute Performance*. PhD Thesis (In French), Institut National Polytechnique de Grenoble, Grenoble, France 2006.

[92] Lecomber, D., Sujithan, K. and Hill, J. *Architecture-Independent Locality Analysis and Efficient PRAM Simulations*. Lecture Notes in Computer Science, Volume 1225, Springer Berlin, Germany 1997.

[93] Legrand, A., Marchal, L. and Casanova, H. *Scheduling Distributed Applications: the SimGrid Simulation Framework.* Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid CCGRID'03, Tokyo, Japan 2003.

[94] Leon, E., Maccabe, A. and Brightwell, R. *Instrumenting LogP Parameters in GM: Implementation and Validation.* Proceedings of LCN 2002 27th Annual IEEE Conference on Local Computer Networks, Tampa, Florida, United States of America 2002.

[95] Li, Z., Mills, P. and Reif, J. *Models and Resource Metrics for Parallel and Distributed Computation.* Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS'95), Hawaii, United States of America 1995

[96] Lilja, D. *Measuring Computer Performance: A Practicioner's Guide.* Cambridge University Press, Cambridge, United Kingdom 2004.

[97] Lombard, P. *NFSP: Une Solution de Stockage Distribué pour Architectures Grande Echelle*. PhD. Thesis (In French), Institut National Polytechnique de Grenoble, Grenoble, France 2003.

[98] Long, G., Yuan, N. and Fan, D. *Location Consistency Model Revisited: Problem, Solution and Prospects.* Proceedings of the 2008 Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies PDCAT'08, Dunedin, New Zealand 2008.

[99] Lowekamp, B., Tierney, B., Hugues-Jones, R, Kielmann, T. and Swany, M. *A Hierarchy of Network Performance Characteristics for Grid Applications and Services.* GWD-C Network Measurement Working Group, Global Grid Forum Document. http://www.ogf.org/documents/GFD.23.pdf Internet Document 2003.

[100] Leese, M., Tyer, R. and Tasker, R. *Network Performance Monitoring for the Grid.* Proceedings in UK e-Science, 2005 All Hands Meeting, Nottingham, United Kingdom 2005.

[101] LUSTRE Team *LUSTRE: SUN Microsystems Cluster File System Project Site*. http://www.lustre.org Internet Site.

[102] Marchal, L. and Casanova, H. *RR-4596 - A Network Model for Simulation of Grid Application.* Internal Report, INRIA Rhône Alpes, ENS-Lyon, France 2002.

[103] Martin, P. and Culler, D. *NFS Sensitivity to High Performance Networks*. Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modelling of Computer Systems. Pages: 71-82. Atlanta, Georgia, U.S.A., 1999.

[104] Martinasso, M. *Analyse et Modélisation des Communications Concurrentes dans les Réseaux Haut Performance.* PhD. Thesis (In French), Université Joseph Fourier, Grenoble, France 2007.

[105] MEG Team. *MEG Project Site.* http://www-id.imag.fr/ denneuli/MEG/ Internet Site.

[106] Mehlhorn, K. and Vishkin, U. *Randomized and Deterministic Simulations of PRAMs by Parallel Machines with Restricted Granularity of Parallel Memories.* Acta Informatica Journal, Volume 21, Number 4, Pages 339-374. Springer Editors, Berlin, Deutchland 1984.

[107] Miller, K. *Communication Theories: Perspectives, Processes, and Contexts*. 2nd edition. McGraw-Hill Ed. New York, United States of America, 2005.

[108] Moritz, C. et Frank, M. *LogGPC: Modelling Network Contention in Message-Passing Programs*. IEEE Transactions on Parallel and Distributed Systems. Vol. 12. No. 4, United States of America 2001.

[109] MPI MCS-ANL Team. *Message Passing Interface Standard Site*. http://www-unix.mcs.anl.gov/mpi/ Internet Site.

[110] Naregi Project Team. *NAREGI: National Research Grid Initiative Site.* http://www.naregi.org Internet Site.

[111] NFSp Team. *NFSp - A Non-Intrusive Parallel NFS Server Project Site.* http://nfsp.imag.fr Internet Site.

[112] Novell Inc. *Novell Netware Product Information* http://www.novell.com/products/netware/ Internet Site.

[113] NWS Team. *NWS: Network Weather Service Site.* http://nws.cs.ucsb.edu/ewiki/ Internet Site.

[114] OAR Team. *OAR - Resource Management System for High Performance Computing Project Site.* http://oar.imag.fr Internet Site.

[115] Open Grid Forum. *Open Grid Forum Site*. http://www.ogf.org/ Internet Site.

[116] Ould-Khaoua, M., Sarbazi-Azad, H. and Obaidat, M. *Performance Modelling and Evaluation of High-Performance Parallel and Distributed Systems.* International Journal in Performance Evaluation, Volume 60, Issue 1-4. Elsevier, North Holland, Netherlands 2005.

[117] Pajé Team. *Pajé Site.* http://www-id.imag.fr/Logiciels/paje/ Internet Site.

[118] Papadimitriou C. and Yannakakis, M. *Towards an Architecture-Independent Analysis of Parallel Algorithms.* Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. Chicago, Illinois, United States of America 1988.

[119] Paxson, V., Almes, G., Mahdavi, J. and Mathis, M. *RFC 2330 - Framework for IP Performance Metrics.* Network Working Group. http://www.faqs.org/rfcs/rfc2330.html Interned Document 1998.

[120] PVFS Team *PVFS: Parallel Virtual File System Project Site* http://www.pvfs.org Internet Site.

[121] Rajasekaran, S. and Reif, J. *Handbook of Parallel Computing: Models, Algorithms and Applications.* Chapman and Hall/CRC Computer and Information Science Series, Taylor and Francis Group, Boca Raton, FL, United States of America 2007.

[122] Ramachandran, V., Grayson, B. and Dahlin, M. *Emulations between QSM, BSP and LogP: A Framework for General-Purpose Parallel Algorithm Design.* Journal of Parallel and Distributed Computing, Volume 63, Issue 12, Pages: 1175 - 1192, Academic Press, Inc. Orlando, FL, United States of America 2003.

[123] Renard, H., Robert, Y. and Vivien, F. *Data Redistribution Algorithms for Heterogeneous Processor Rings*. Proceeding of International Conference on High Performance Computing HiPC'2004, Bangalore, India 2004.

[124] RENATER GIP. *RENATER, Réseau National de Télécommunications pour la Technologie l¿Enseignement et la Recherche Site.* http://www.renater.fr Internet Site.

[125] Rice, D. *An Analytical Model for Computer System Performance Evaluation.* ACM SIGMETRICS Performance Evaluation Review, Volume 2, Issue 2, Pages: 14 - 30, ACM New York, United States of America 1973.

[126] Sander, V. (Editor) *Networking Issues for Grid Infrastructure.* Open Grid Forum Document GFD-I.037 Grid High Performance Networking Research Group, http://www.ggf.org/documents/GFD.37.pdf Internet Document 2004.

[127] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M. and Noveck, D. *RFC3530 - Network File System (NFS) version 4 Protocol.* Network Working Group, http://www.faqs.org/rfcs/rfc3530.html Internet Document 2003.

[128] Skillicorn, D. and Talia, D. *Models and Languages for Parallel Computation*. ACM Computing Surveys (CSUR), Volume 30 , Issue 2. United States of America 1998.

[129] Snir, M., Otto, S., Huss-Lederman, S., Walker, D. and Dongarra, J. *MPI: The Complete Reference.* MIT Press, Boston, United States of America, 1996.

[130] Standard Performance Evaluation Corporation *SPEC - Standard Performance Evaluation Corportion Site.* Internet Site.

[131] Sulistio, A., Poduval, G., Buyya, R. and Tham C-K. *On Incorporating Differentiated Levels of Network Service into GridSim.* Future Generation Computer Systems (FGCS), Volume 23, Issue 4, Pages: 606-615, Elsevier Science, Amsterdam, Netherlands 2007.

[132] Sun Microsystems Inc. *RFC 1094 - NFS: Network File System Protocol specification* Network Working Group. http://www.faqs.org/rfcs/rfc1094.html Internet Document 1989.

[133] Sundaram-Stukel, D. and Vernon, M. *Predictive Analysis of a Wavefront Application Using LogGP.* Proceedings of the Seventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Atlanta, Georgia, United States of America 1999.

[134] Tanaka, T., Inayoshi, M. and Mizuhara, N. *Overview of Communication Network Evolution*. Hitachi Review, Volume 47, No. 2, Japan, 1998.

[135] Tanembaum, A. *Computer Networks* Fourth Edition. Prentice Hall Publishers. United States of America, 2002.

[136] Tanembaum, A. *Modern Operating Systems* Third Edition. Prentice Hall Publishers. United States of America, 2008.

[137] Tanembaum, A. and Woodhull, A. *Operating Systems: Design and Implementation* Third Edition. Prentice Hall Publishers. United States of America, 2006.

[138] Telecommunications Industry Association *Federal Telecommunications Recommendation 1090-1997*. Federal Telecommunications Recommendations, United States of America, 1997.

[139] Tera Grid Consortium *TeraGrid Project* http://www.teragrid.org/ Intenet Site.

[140] Top 500 Team *Top 500 Supercomputer Sites* http://top5000.org Internet Site.

[141] Treadwell, J. (Editor) *OGSA, Open Grid Service Architecture: Glossary of Terms. Version 1.5*. Open Grid Forum Document GFD-I.081 Open Grid Services Architecture, http://forge.gridforum.org/projects/ogsa-wg Internet Document 2006.

[142] Vadhiyar, S., Fagg, G. and Dongarra, J. *Towards an Accurate Model for Collective Communications.* International Journal of High Performance Computing Applications, Volume 18, Issue 1, Sage Publications, Inc. Thousand Oaks, CA, United States of America 2004.

[143] Valentin, O., Lombard, P., Lebre, A;, Guinet, C. and Denneulin, Y. *Distributed File Systems for Clusters and Grids*. Proceedingson Prallel Processing and Applied Mathematics PPAM 2003, Czestochowa, Poland. Lectue Notes in Computer Science, Springer, Berlin, 2003.

[144] Valiant, L. *A Bridging Model for Parallel Computation.* Communications of the ACM, Volume 33 , Issue 8, Pages: 103 - 111, United States of America 1990.

[145] Vampir Team. *Vampir: Performance Optimization Tool Site.* http://www.vampir.eu/ Internet Site

[146] von Eicken, T., Culler, D., Goldstein, S. and Schauser, K. *Active messages: a mechanism for integrated communication and computation.* Proceedings of the 19th Annual International Symposium on Computer Architecture, Queensland, Australia 1992.

[147] Verstoep, K., Bhoedjang R., Rühl, T., Bal, H. and Hofman, R. *Cluster Communication Protocols for Parallel-Programming Systems.* ACM Transactions on Computer Systems (TOCS), Volume 22, Issue 3, ACM New York, NY, United States of America 2004.

[148] Waldbusser, S. (Editor) *RFC 1243 - AppleTalk Management Information Base.* Network Working Group (AppleTalk-IP Working Group of the Internet Engineering Task Force (IETF)). http://www.faqs.org/rfcs/rfc1243.html Internet Document 1991.

[149] Wikimedia Foundation Inc. *Wikipedia: The Free Encyclopedia.* . http://www.wikipedia.org/ Internet Site.

[150] Wolski, R. *Experiences with Predicting Resource Performance On-line in Computational Grid Settings*. ACM SIGMETRICS Performance Evaluation Review, Volume 30, Number 4, Pages 41-49, United States of America 2003.

[151] Wolski, R., Spring, N. and Hayes, J. *The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing.* Journal of Future Generation Computing Systems, Volume 15, Numbers 5-6, Pages 757-768, Elsevier, North Holland, Netherlands 1999.

[152] Zimmermann, H. *OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection*. IEEE Transactions on Communications, Vol. 28, No. 4. United States of America, 1980.

[153] Zimmermann, W., Löwe, W. and Trystram, D. *On Scheduling Send-Graphs and Receive-Graphs Under the LogP-Model.* Information Processing Letters, Volume 82 , Issue 2, Pages: 83 - 92, Elsevier North-Holland, Inc. Amsterdam, Netherlands 2002.