A Grid–QoS Decision Support System using Service Level Agreements

Matheus Bandini[1,2], **Antonio Roberto Mury**[1], **Bruno Schulze**[1], **Ronaldo Salles**[2]

[1]Scientific Distributed Computing (ComCiDis)
National Laboratory for Scientific Computing (LNCC)
Av. Getúlio Vargas, 333 – 25651–075 – Petrópolis – RJ – Brazil

[2]Department of Computing Engineering
Military Institute Engineering (IME)
Rio de Janeiro – RJ – Brazil

{mbandini,aroberto,schulze}@lncc.br, salles@de9.ime.eb.br

Abstract. *The availability of better services and products has become a challenge in all segments, from economics to high technology. This paper proposes an approach to Quality of Service for Grid Computing that involves the analysis of the grid resources and the categorization of types of services into Service Level Agreements (SLAs). This work focuses on the definition of grid indicators to create a Decision Support System that helps to identify the Quality of Service offered by the resources available in the grid, as well as the improvement needed for these services. The system also includes benefits for grid users, such as the quality of the information provided about the grid resources, which allows users to manage and use the available resources in an easy way, improving the system's QoS from the user's point of view.*

1. Introduction

The technological development has considerably increased the computer hardware capability, allowing developers and researchers to imagine new ways to fully exploit this available computing capacity [Foster et al. 2001]. However, the high prices of the last generation computers and the constant hardware upgrades becomes unavailable for most users and institutions [Foster et al. 2002].

An alternative to solve this problem, cutting the cost of constant hardware improvement, is the use of Grid Computing and its technologies, as it can expand lifetime before the hardware becomes obsolete. The idea is to use a certain number of resources (computers and devices) interconnected via local networks or the Internet, allowing to increase the computational power of one or more processing resources in a cooperative way to accomplish a common task.

The computational grids allow the use of geographically distributed computing resources and distributed network, such as processing and memory through security schemes involving the use of certificates that identify the users, making them able to use the resources [Foster et al. 2001].

On its native form, a grid operates basically as follows: a user sets the number of resources he wants to run his application and send it along with the files and parameters needed for implementation. Then, the grid middleware used will take charge of the task to distribute them to the number of resources specified by the user, without worrying about special needs required by application or their users.

The use of Grids, in their natural form, answered the needs of the scientific community. However, due to the growth in the use of the Commercial Grids, the need to improve the Grid Computing QoS in order to optimize the environment performance and offer custom services has been considered important to better meet the users needs [Sahai et al. 2002].

This work aims to apply Quality of Service in an environment of Grid Computing, optimizing the use of resources and making the required QoS choice process transparent for the users. This way, it is possible for users who do not have technical knowledge about computers or Grid Computing itself to use the Grid environment.

The objective of this paper is to offer improvements to the services offered to users. This can be done by improving the way they interact with the system and charging different pricing for distinct services. The use of indicators and a multicriteria method evaluation has been considered to measure the usage of the system and specify Service Plans that will be chosen by the users.

The concept of Project used in this paper is the same as the one in [Schulze 2006], where several users are associated to a project. These users add resources associated to a project (A) and every time a resource is used by another project (B), credits will be added to the project (A). In a similar way, project (A) will consume its credits when using resources of the project (B). The main objective of this paper is to include Quality of Service in the Grid environment proposed in [Schulze 2006], using Grid performance indicators and Service Level Agreements.

The paper is organized as follows: Section 2 presents some existing works that address Grid-QoS problems and issues. Section 3 presents an overview for both Grid Computing and Network Quality of Service, discussing the analysis which defines the set of parameters. Section 4 presents the Grid indicators for QoS and the methods used to obtain them. It also presents a weight definition for Grid resources and services according to the pre–determined parameters. Section 5 presents a summary of the Analytic Hierarchy Process (AHP). Section 6 presents experiments used to validate the set of parameters and the charging proposed process in this paper and comments the results obtained from work. Section 7 presents some conclusions and future work.

2. Related Works

Through these last years, many systems and architectures have been proposed trying to offer solutions that are capable of supporting Quality of Service for Grid Computing Applications. However, Grid–QoS represents not only the technologies involved on the system development, it also requires other requisites, such as the availability and reliability of the resources and services, agreements between users and the service providers and a friendly interface. Some proposals of existing systems and architectures will be presented next, as well as its benefits and restrictions.

The Framework GARA (General-purpose Architecture for Reservation and Allocation) [Foster et al. 1999], [Roy 2001] is a system capable of providing support to Quality of Service for Grid Computing applications, allowing developers to specify end–to–

end Grid–QoS parameters. It reserves computer resources with an uniform treatment. These reservations guarantee that the resource manager will give the specified Quality of Service to the application. Alhough GARA is a very popular Grid–QoS framework amongst the Grid community since it is a Globus Toolkit based application, it does not employ some important Grid technologies, such as Web Service and the OGSA (Open Grid Service Architecture). These incompatibilities cause GARA applications not to work properly with Web Service and OGSA compliant applications. GARA also does not supply functionalities such as dynamic resources monitoring and Service–Level Agreements used for service negotiations [Al-ali et al. 2004].

Trying to soften some of the limitations of GARA, the Grid–QoSm (Grid QoS Management) [Al-ali et al. 2004] was developed with the objective to provide a Quality of Service system for Grid Computing applications that is compatible with the technologies GARA is not. This assures that Grid–QoSm can receive applications that use both Web Services and OGSA. Using QGS (QoS Grid Service), a component of OGSA, Grid–QoSm can provide services such as negotiation and resource reservation and allocation with certain Quality of Service [Al-ali et al. 2004].

Grid–QoSm, comparing with GARA, brings some technologies advantages. However, it has also some restrictions about how users interact with the system. It does not present a friendly interface to use the system. It does not charge for different services and resources, since there is no payment process or Economy model. These two are key items, together with the scheduling and reservations systems, to provide Quality of Service for Grid Computing applications [Bandini et al. 2008].

Other systems give different approaches to the Grid–QoS problem, as an example there is AppLeS (Application–level Scheduling System) [EPCC 2003], which, according to [Li and Baker 2005] is an adaptive application-level scheduling system that can be applied to the Grid. Any application submitted to the grid has its own AppLeS instance. The architecture of AppLeS has been designed to allow that the system performance and utilization may be experienced from the perspective of an application using the system [Li and Baker 2005]. AppLeS does not use Service Level Agreements (SLA) to guarantee or provide Quality of Service, otherwise, it measures the performance of the application on a specific site resource and utilizes this information to make resource selection and scheduling decisions [Li and Baker 2005].

A distributed specialized parametric modeling system called Nimrod/G [Nimrod/G 2005] has been created to help the development of complex parametric experiments. Nimrod/G takes advantage of many features provided by the Globus Toolkit [Foster 2005] such as the discovery of resources and services and its digital certificate based security. Nimrod/G also introduces the concept of a computational Economy to measure the resources usage. Nimrod/G however, does not have any kind of agreement to assure Quality of Service for its users. So, it can not charge for different levels of service.

An architecture that provides high level services, combined with the management of this services through SLAs is presented in the VRM (Virtual Resource Manager) [Burchard et al. 2005]. It uses SLA to negotiate the service level between users and the system and guarantee the required QoS (CPUs, network bandwidth and storage) is achieved. It also offers features such as control of long–running jobs and fault tolerance.

However, there is no kind of Economy model, which means there is no way to charge the different classes of service held in the SLAs.

The processes of classification and choice of resources in a grid computing based on the guarantee of a certain provision of level of service can be seen as a complex decision-making process. To deal with this type of complexity, Multi-criteria Decision Making Method presents itself as high recommended support tool. [Wu and Chang 2007a] and [Wu and Chang 2007b] classify it as a process that involves multiple criteria on which the AHP (Analytic Hierarchy Process) can be used. Its characteristics make it possible to work with both qualitative and quantitative elements, as well as to assist on the structuring of the problem through a hierarchy, besides the correspondence of this approach as the one adopted for the evaluation of QoS. [Sun et al. 2007] presents the AHP method as an effective tool to obtain QoS indices that are used to determine which service will be the best one for a specific user.

3. Quality of Service Overview

A service or product that is assumed to have quality is the one that satisfies its users, no matter the area of interest. To achieve the desired quality, this service or product must follow certain rules and patterns in order to match with its quality categories. That is exactly what happens with both Quality of Service for Network Communications or for Grid Computing Applications. In fact, the very concept of Quality of Service and its parameters work together in these two cases, as some parameters of Grid–QoS are the parameters that define Network Quality of Service [Bandini et al. 2008]. A common and important factor in every system that offers different services with distinct quality is to charge for the level of service that is being used [Sahai et al. 2002].

The Network Quality of Service concept emerged with the main objective of providing better Internet services than the "best–effort" services that were being offered since the 1970s. In this context, several patterns and protocols where proposed and developed since then to promote better services. All of them converging to the same four key parameters that, if measured and controlled, can assure that a network can support Quality of Service. These parameters are the ***delay*** of a network, the variation this delay occurs, called ***jitter***, the transmission rate of a network (***throughput***) and the ***packet–loss rate*** of a network [Katchabaw et al. 1998].

To provide a better service, together with a friendly interface capable to attract users from research, academic and commercial communities, the Grid–QoS idea adopted some of the Network QoS concepts. Based on these concepts, a qualification of services was specified long ago in [ISO 1995]. These qualifications have been taken into consideration to define Service–Level Agreements:

- *Best effort*: The service provider system does not guarantee that the QoS patterns will be maintained as previously accorded. This means that the service can eventually be degraded or improved, compromising the charging agreement;
- *Compulsory*: The service provider system monitors the resources and can attend the application with the required QoS. However, if there is no resource that supports the requirements, the service can be interrupted or cancelled;
- *Guaranteed*: The service provider system accepts the user application and guarantees the Quality of Service, as soon as there are resources that attend to the

pre–established agreements.

After defining the kind of service that will be provided, the QoS parameters that specify the Service–Level Agreements can be set. In the Grid–QoS case, these parameters are: processing capacity, the percentage of available memory, the disk space required for storage and the priority level of a job. All these parameters work together with the network QoS parameters to specify the SLAs and make different guarantees and charges. Table 1 shows an example of a SLA specification based on these parameters.

Table 1. Example of SLAs definition for Grid–QoS

SLA	CPU (% available)	Memory (% available)	Storage (% available)	Priority	Network delay	SLA wage (%)
Standard	less than 50	less than 40	less than 10	Low	< 500 ms	+ 0 %
Master	50	40	15	Medium	< 200 ms	+ 25 %
Premium	80	70	25	High	< 100 ms	+ 50 %

As it can be seen, the hardware features must be considered on the Grid–QoS evaluation and SLAs specifications, there are other important indicators that can be used in order to improve the reliability of the Grid–QoS Provider System. The methods used to obtain these indicators are shown in section 4.

Table 2. Hardware weight definitions according to its features

Resource Type	CPU cores	CPU Total Frequency	Memory Total	weight (*f*)
A	2	Up to 1.0GHz	1024 MB	0,2
B	2	Up to 1.0GHz	1536 MB	0,4
C	2	Up to 1.0GHz	2048 MB	0,6
D	2	From 1.0GHz up to 1.5 GHz	1024 MB	0,8
E	2	From 1.0GHz up to 1.5 GHz	1536 MB	1,0
F	2	From 1.0GHz up to 1.5 GHz	2048 MB	1,2
G	1	From 1.5GHz up to 2.0 GHz	1024 MB	1,4
H	1	From 1.5GHz up to 2.0 GHz	1536 MB	1,6
I	1	From 1.5GHz up to 2.0 GHz	2048 MB	1,8
J	2	From 1.5GHz up to 2.0 GHz	1024 MB	2,0
K	2	Over 2.0 GHz	1024 MB	2,2
L	2	From 1.5GHz up to 2.0 GHz	1536 MB	2,4
M	2	Over 2.0 GHz	1536 MB	2,6
N	2	From 1.5GHz up to 2.0 GHz	2048 MB	2,8
O	2	Over 2.0 GHz	2048 MB	3,0
P	4	Over 2.0 GHz	4096 MB	3,4
Q	8	Over 2.0 GHz	4096 MB	3,8

Using the service–level and hardware parameters, it is possible to establish a charging formula, as defined in [Bandini et al. 2008]. This formula can also be modified to supply Grid–QoS on demand and to benefit, for example, large scale applications.

$$C = S + \alpha(S.t) + \beta(S.f) \tag{1}$$

The value *C* represents the total submission cost for one resource; the value *S* is the total execution time of a submission; the parameter *t* is the wage that varies according to the type of service (Table 1); the parameter *f* is the hardware weight used for the submission (Table 2); the parameters α and β serve as equilibrium factors for the equation and can be used to increase or decrease the total submission cost, for example, for applications that are using the resource for a long period of time.

4. Grid Indicators

The information generally available in a Grid infrastructure related to its services are: CPU type, number of processors, amount of memory, storage space, load (% CPU used and available memory), architecture, operating system and the network performance parameters described in section 3. Some of these items are fixed while others may vary over time.

The fixed items are: CPU type, number of processors, total amount of memory, architecture and operating system; and the variables items are: load, storage space, competing process, free memory, throughput and latency, where these last two elements are the key ones for network performance.

These elements are important because they contribute to the overall capacity view, but the performance analysis is extremely complex, because in addition of its heterogeneity, changes may occur in the individual resources capabilities. These changes imply that the performance analysis of a grid should take into account other indicators.

The QoS as saw before refers to a broad set of variables and technologies. Its goal is to provide guarantees that a service has the ability to deliver predictable and favorable results. The performance indicators, here proposed, within the QoS scope are: the index of resources availability, the rate of utilization of resources and the task average execution time.

a) The index of resources availability is obtained by calculating the ratio between the total time resources were available and the total Grid operation time. This indicator measures the degree of availability of the resources when the Grid remains active. It is possible to include in this indicator a qualitative assessment through the application of weights based not only on the amount of time but also in the slot of time during the available period.

Table 3 presents an example of the weights application depending on the time slot that a resource is available, considering the peak hours of Grid operation time.

Table 3. Weight definition cost for time period

Time	*weight*
08:00:01 - 12:00:00	3
12:00:01 - 13:00:00	2
13:00:01 - 18:00:00	3
18:00:01 - 22:00:00	2
22:00:01 - 08:00:00	1

Table 3 may also be adopted in a Credit Economy Model, where the resources added to a Grid will be rewarded with credits, and the users will be charged when using these resources. This table, when coupled with the possibility of an earlier resource reservation, encourage users to use their time out of Grid attended hours, freeing this time for those most critical applications.

b) The resources utilization rate is obtained by the ratio between the number of tasks submitted to a resource and the total number of tasks successfully executed in this resource.

This indicator will only be correctly used if it is possible to detect if the error was due to server restrictions or to incorrectness in the submitted application (task).

The two indicators previously presented are respectively related to the availability (a) and reliability (b) of the resource.

c) Task average execution time is obtained by the ratio between the total time spent performing tasks and the number submitted tasks. It is understood as the total spent time. This indicator is counted from the task submission until the end of their execution, this is done to include network delays.

Although this indicator is highly influenced by the variety of tasks submitted to the grid and by their complexity, it solves the complexity of the resource usage variation in a Grid. To get more reliable results and to be able to compare the use of Grid resources, it is important to consider the use of standard test tasks. The use of this kind of test has the following advantages:

1. It allows resources comparison, considering the whole set of features (CPU type, number of processors, amount of memory, storage space, load – % CPU used and memory available, architecture, operating System, network parameters, the influence of other programs installed that competes with the submitted task).
2. it allows the distinct assessment of the resources according to their execution performance for a particular class of problem, and
3. it allows the instantaneous assessment of a resource or a group of resources.

Restrictions on its use:

1. the period of use of the standard test task should be well established for non competition with the legitimate tasks,
2. the criterion in choosing the task pattern must be compatible with the type of the actual use of the Grid.
3. It will be always an approximation of reality.

The use of standard tasks test must be linked to the Grid control in terms of average time of the real tasks submitted. This will serve as a comparison parameter for those times calculated. The assessment is made considering the two results, performance of a standard task test and achieved average time for the real tasks. In the first case, a comparison, in terms of absolute class problem of the standard task test, and in the second case, considering the diversity of tasks submitted to the Grid.

This indicator is also an important parameter for choosing new resources, as it allows to simulate their inclusion to the Grid and evaluate the impact they have on its performance, guiding the choice and acquisition of new resources.

The establishment of SLAs can be guided by a set of tests getting the application profile. This initial classification, when thought in terms of QoS, not only helps the choice of most appropriate resources to the projects, but also assists in the transparency of services to the users. For users and projects with more complex applications, it can identify the resources that are best suited to their purposes in advance.

The use of standard tasks must be seen as a tool to assist the Grid Manager to verify the resources performance.

The increase in processing time of a test task may indicate the occurrence of the following problems: communication problems, processes competition in the resource, hardware problems, among others. This information, associated with the type of task, indicates whether the problem lies, if in the system configuration, running a certain kind of code, or hardware problems.

The Project Prioritization Process is a typical decision–making problem, where the characteristics of these resources and indicators serve as evaluation elements of the grid environment for managers, for projects and users that require services and for those who provide Grid services. The approach proposed here will result in values, as weights, to be applied both to those who hire and those who provide the services. The decision–making method used, given the characteristics of the problem, will be a multicriteria decision–making method and among the different schools and methods, the Analytic Hierarchy Process (AHP) will be used.

The AHP method was chosen not only because it is a long used and tested method, but mainly because it is capable of capturing both the quantitative and the qualitative analysis of a particular decision-making problem.

5. Summary of the AHP Method

Among the various analytical methods to support the decision making complex problems, Multicriteria Decision Making Methods (MCDM) has the ability to aggregate all features considered important to offer better results for the problem.

The Analytic Hierarchy Process method (AHP) [Whitaker 2007] is a Multicriteria Decision Method in which the problem is decomposed into hierarchical levels, thereby facilitating its structure and subsequent evaluation. It determines, through the synthesis of the decision agents evaluations, an overall measure for each one of the evaluated elements, classifying them.

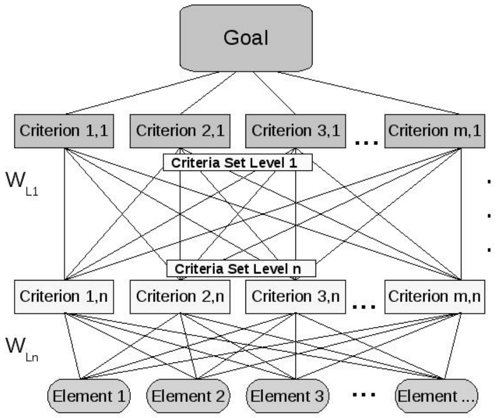


Figure 1. AHP Basic Hierarchy - Symmetrical Hierarchy

The hierarchy is a structure that represents the dependence of various levels in a sequential manner. Figure 1 shows an example of a hierarchy. Basically it has a set criteria divided into N levels and a set of elements that are evaluated in relation to the last level (N) of criteria. The weights of the assessment from higher levels of criteria

(W$_{L1}$ to W$_{Ln}$) are then aggregated with the weights obtained through the assessment of elements/alternatives of the final level, providing the final valuation.

After the construction of the hierarchy, each decision maker will make a pairwise comparison of the elements, within each level of the hierarchy, creating a square matrix decision (Figure 2), where he will state, from a pre-set scale (Figure 3), their opinion/preference among the elements based on the element of the next higher level of the hierarchy.

$$\left\{ \begin{matrix} E_{(1,1)} & E_{(1,2)} & E_{(1,3)} & \dots & E_{(1,n)} \\ E_{(2,1)} & E_{(2,2)} & E_{(2,3)} & \dots & E_{(2,n)} \\ \dots & & & & \\ E_{(n,1)} & E_{(n,2)} & E_{(n,3)} & \dots & E_{(n,n)} \end{matrix} \right\}$$

Figure 2. Pairwise Comparison Matrix

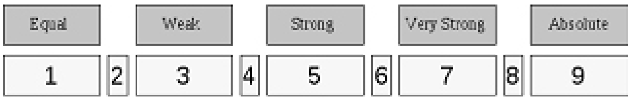
This will be done for all levels of the hierarchy. The result matrix is called dominance matrix that expresses the number of times that an alternative dominates the other, or is dominated by it. The main diagonal of the dominance matrix will be filled by a set of values representing a non-dominance of an alternative on the other.

5.1. Fundaments of AHP

a) Attributes and Properties - a finite set of alternatives/elements compared according to a finite set of properties.

b) Binary Correlation - Comparison between two elements based on a given property, it is a binary comparison in which we can have the follow solutions: to be preferable to the other, measured by a basic scale, or to be indifferent to the other.

c) Fundamental Scale (Basic Scale) - It is a set of qualitative values that express the priority of one element in relation to the other, which will be correlated to a numerical scale of positive real numbers. In general there are five distinct differences: equal, weak, strong, very strong and absolute. This leads to a total of 9 assessment (figure 3)[Whitaker 2007].



Equal		Weak		Strong		Very Strong		Absolute
1	2	3	4	5	6	7	8	9

Figure 3. Fundamental Scale (Basic Scale) - Values 2,4,6,8 are the intermediates values of the basic verbal scale

In this model a special attention should be taken with uniformity and redundancy because the criteria at each level should be homogeneous and not redundant. A practical way to test a hierarchy, is to see if the elements of a higher level could be used as an argument for a lower level.

The existence of a decision hierarchy is a convenient way to break into steps, a complex problem in finding the explanation of cause and effect in a linear chain. After completion of the modeling phase, it follows the implementation phase of the method itself.

When using the pairwise comparison in order to summarize the priorities, rating this preference verbally and then transformed into a scale of reasons attributes it is possible to fully represent the preference in a given criterion, even when working with various levels of complexity.

It is important to note that the consistency in a decision matrix should serve as a warning to the “quality of the judgment”, so the “decision maker” had to be very careful with the use of mathematical procedures to obtain the consistency, as these can change significantly the result of the problem. The “decision maker” should be alerted during the process analysis and he, and only he, can change the judgment held. There are two types of uncertainties in AHP method: the first is number of alternatives and criteria established and the second is the assessment carried out.

The AHP method, has the advantage of eliminating the black boxes of algebraic calculations. Their correct use enables a simple way to understand how to turn concepts into values that meets the evaluation process. Using the decision matrix, the AHP method calculates partial results of each element of the hierarchy, called “impact value”, which represent numerical values of verbal assessment given by the “decision maker” for each comparison.

After the evaluations have been considered, the decision matrix has its columns normalized by the equations (2) and (3).

$$\sum_{j=1}^{n} a_{ij}, i = 1, 2, \ldots, n \tag{2}$$

$$W_{ij} = a_{ij} / \sum_{j=1}^{n} a_{ij}, i = 1, 2, \ldots, n \tag{3}$$

And the normalized main Eigen vector can be obtained by averaging across the rows (4).

$$W_i = \frac{\sum_{j=1}^{n} W_{ij}}{n}, i = 1, 2, \ldots, n \tag{4}$$

The normalized main Eigen vector is also called *priority vector.*

5.2. Services Analysis and Classification Methodology

The fist step to use the AHP method is the establishment of the hierarchy. The following items will be used for the analysis of the resources available in the Grid:

Quantitative elements - number of processors*velocity of processor, free memory average, CPU load average, storage space average, network parameters and indicators (Index of resource availability, resource utilization rate and average execution time).

Qualitative elements - relative importance for the project leaders of the performance criterion, the availability criterion, the reliability criterion and the security criterion.

The hierarchy may be a symmetrical hierarchy as shown in Figure 1 and may be a non symmetrical hierarchy as the hierarchy that will be used in this work, shown in Figure 4.

The performance criterion is based on the task average execution time. This average time is obtaining by the use of standard tasks and the historical use of the resource by the users. This is done to minimize the effects of resources variation through time as addressed in [Nemeth et al. 2004] and [Wu et al. 2007]. It also uses parameters, such as: number of processors*velocity of processor, free memory average, CPU load average, storage space average, network parameters. The availability criterion is based on the calculation of the index of resource availability. The reliability criterion is based on the resource utilization rate.

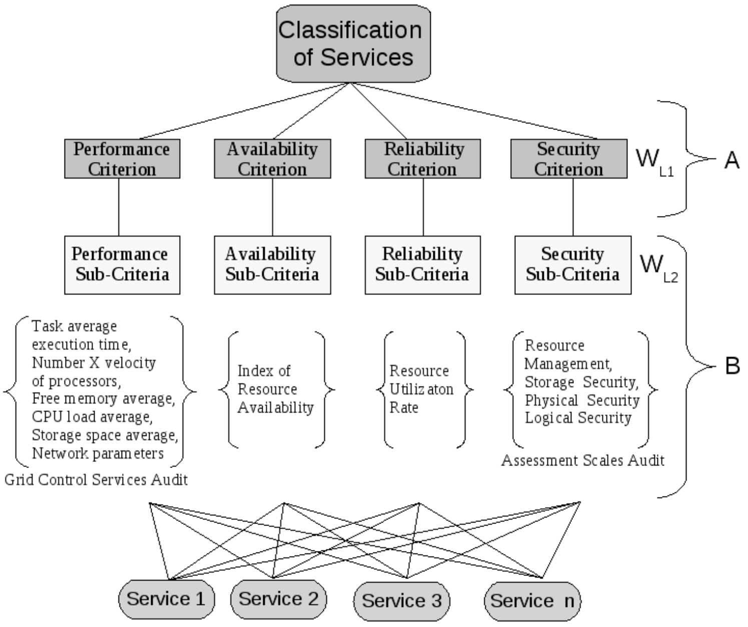

Classification of Services
Performance Criterion
Availability Criterion
Reliability Criterion
Security Criterion
W_{L1}
A
Performance Sub-Criteria
Availability Sub-Criteria
Reliability Sub-Criteria
Security Sub-Criteria
W_{L2}
Task average execution time, Number X velocity of processors, Free memory average, CPU load average, Storage space average, Network parameters
Index of Resource Availability
Resource Utilizaton Rate
Resource Management, Storage Security, Physical Security Logical Security
B
Grid Control Services Audit
Assessment Scales Audit
Service 1
Service 2
Service 3
Service n

Figure 4. Asymmetrical Hierarchy Classification of Services Model

The Security criterion is composed of four sub-criteria: Resource Management, Storage Security, Physical Security and Logical Security. The resource management means the level of control by the grid administrator for: preventing, granting, limiting and revoking access to the grid resources and his audit tools. The storage security means the level of security implemented for the storage data as: access control, data integrity, cryptography and data reliability [Office 2002].

Physical Security means procedures of the level of personnel control to access resources area, equipment maintenance, environmental safety and all other aspects of the physical and electrical infrastructure. The logical security means the procedures for the control level to access resources such as: backup policy, network and data security and public remote access. A detailed description of these procedures can be found in [West-Brown et al. 2003].

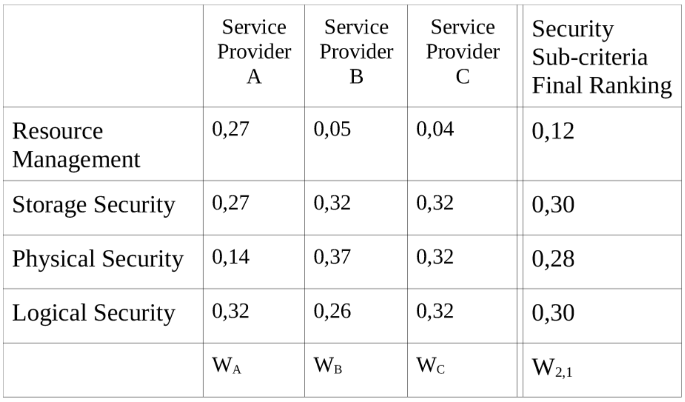
The evaluation of the level one criteria (performance, availability, reliability and security) and among the security sub–criteria (Resource Management, Storage Security, Physical Security and Logical Security) and among performance sub–criteria (average execution time, number of processors*velocity, free memory average, CPU load average, storage space average and network parameters) are made by a qualitative pairwise comparison. The other values of the hierarchy are quantitative values and are obtained by the Grid control services (resources features) and assessment scales audit (security procedures).

6. Experiments and Results

The experiments were conducted comparing three projects that use a Community Grid [Schulze 2006]. This case study examine them through the point of view of the resource providers and, from now on, they will be called Service Providers.

The results generated from the evaluation of the security criterion respecting their sub-criteria (Resource Management, Storage Security, Physical Security and Logical Security) are shown in Table 4.

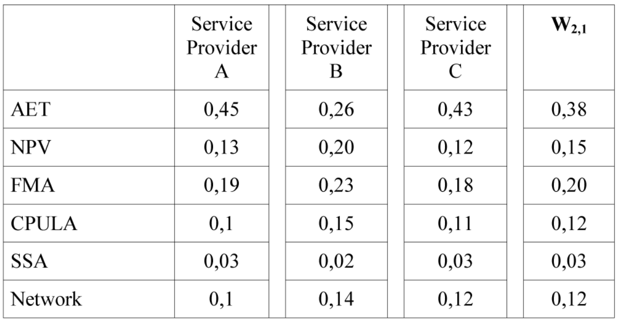
Table 4. Service Providers A, B and C normalized Security sub-criteria weigth evaluation and total Security sub-criteria weight



	Service Provider A	Service Provider B	Service Provider C	Security Sub-criteria Final Ranking
Resource Management	0,27	0,05	0,04	0,12
Storage Security	0,27	0,32	0,32	0,30
Physical Security	0,14	0,37	0,32	0,28
Logical Security	0,32	0,26	0,32	0,30
	W_A	W_B	W_C	$W_{2,1}$

In respect to the performance sub-criteria (Average execution time – AET, Number of Processors*Velocity – NPV, Free memory average – FMA, CPU Load average – CPULA, Storage Space Average – SSA, Network). Table 5 shows the results obtained.

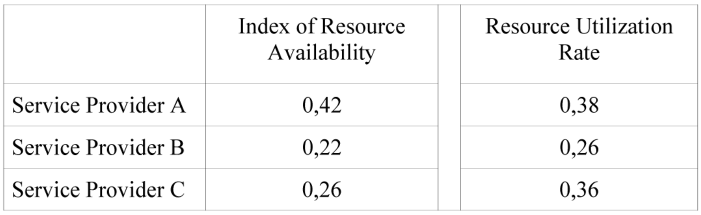
Table 5. Performance sub-criteria normalized weight evaluation (Service Provider A, B and C) and total Performance sub–criteria evaluation



	Service Provider A	Service Provider B	Service Provider C	$\mathbf{W_{2,1}}$
AET	0,45	0,26	0,43	0,38
NPV	0,13	0,20	0,12	0,15
FMA	0,19	0,23	0,18	0,20
CPULA	0,1	0,15	0,11	0,12
SSA	0,03	0,02	0,03	0,03
Network	0,1	0,14	0,12	0,12

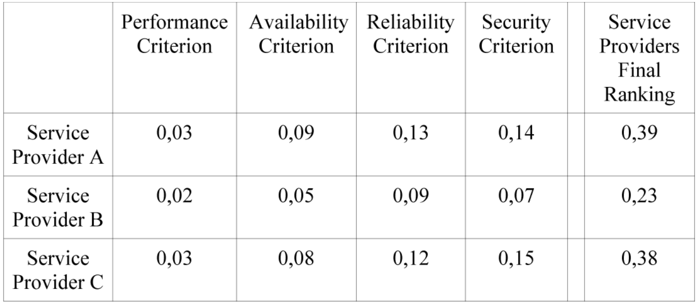
Table 6 represents the outcome of the other sub-criteria: Resource Availability Index and Resource Utilization Rate.

Table 6. Performance sub-criteria normalized weight evaluation Resource Availability Index and Resource Utilization Index



	Index of Resource Availability	Resource Utilization Rate
Service Provider A	0,42	0,38
Service Provider B	0,22	0,26
Service Provider C	0,26	0,36

Table 7. Consolidate weight results of first level criteria and Servers Providers Final Ranking



	Performance Criterion	Availability Criterion	Reliability Criterion	Security Criterion	Service Providers Final Ranking
Service Provider A	0,03	0,09	0,13	0,14	0,39
Service Provider B	0,02	0,05	0,09	0,07	0,23
Service Provider C	0,03	0,08	0,12	0,15	0,38

The results with the consolidated weights are presented in Table 7.

Table 7 allows us to deduce that the Services Providers A and C had the highest and a very close rating, while the Service Provider B had the lower rate among the servers and it needs to improve it. Table 7 also allows us to see that there is a clear preference for issues related to security and reliability in all three providers, but in the case of providers A and C security aspect had greater importance while reliability is the most important aspect of the provider B.

When analyzing the data separately, it is possible to notice, within the Security criterion (Table 4), the criterion with the lower level of awareness is Resource Management capacity by Providers B and C and Physical Security by the provider A.

7. Conclusion

The use of Service Level Agreements, together with a proper Economic Model that helps the charging process, proved to be an efficient way to provide Quality of Service, not only for Grid Computing Applications, but for any kind of computer services. On the other hand, the adoption of Grid Indicators came to improve the precision of decisions needed to be taken on the two main problems of Quality of Service provision: the definition of the types of services based on these indicators and the level of quality that a service has.

The use of AHP method was important to allow the assessment of the relative importance of the criteria (Performance, Reliability, Availability, Security) for grid managers and service providers in a community grid. These values were obtained from a qualitative and quantitative assessment of the grid sub-criteria and elements. From the analysis of these results, it was possible to establish the relative levels of the provided services offered in the grid. The AHP Method had also shown itself as an effective tool on the identification process of criterion that must be improved by Services Providers Managers.

The indicators analysis presented in section 6 not only allowed a benchmarking between the quality of service made available by the providers, sorting them, but they also helped to draw a profile of each one. Based on these indicators and on the results of the previously analysis, it is possible to establish goals and requirements within the SLA, which will guide and encourage improvement in the services made available by the providers, pointing the items that should be improved to a best service provision. Finally, the main favored with this type of approach is the end user who has a transparent vision of the Quality of Service provided by each one of the several service providers that compose the Grid.

Furthermore, this work presents a mechanism to assess computing characteristics and sort them as needed. This is not only applicable for a Quality of Service perspective, but can also be used to measure what services are the most accessed and which one needs to be improved.

Acknowledgment

The authors would like to thank the Distributed Scientific Laboratory (ComCiDis) at the National Laboratory for Scientific Computing (LNCC) for the provision of the computing resources and for the information gained from the resource management system VMS (VCG Management System). The authors would also like to thank the Department of Computer Engineering of the Military Institute of Engineering (IME) on the acomplishment of the work reported in this paper.

References

Al-ali, R. J., Amin, K., Laszewski, G. V., F, O., Walker, D. W., Hategan, M., and Zaluzec, N. (2004). Analysis and provision of qos for distributed grid applications. *Journal of Grid Computing*, 2:163–182.

Bandini, M., Mury, A. R., Schulze, B., and Salles, R. (2008). Pré–escalonamento com qos em grids computacionais utilizando economia de créditos e acordos em nível de serviço. In *WCGA 08 – SBRC 2008*.

Burchard, L.-O., Linnert, B., Heine, F., Hovestadt, M., Kao, O., and Keller, A. (2005). A quality-of-service architecture for future grid computing applications. In *19th International Parallel and Distributed Processing Symposium*. IEEE, IEEE Computer Science.

EPCC (2003). Apples: Application–level scheduling system.

Foster, I. (2005). Globus toolkit version 4: Software for service-oriented systems.

Foster, I., Kesselman, C., Lee, C., Lindell, R., Nahrstedt, K., and Roy, A. (1999). A distributed resource management architecture that supports advance reservations and co-allocation. In *Proceedings of the International Workshop on Quality of Service*.

Foster, I., Kesselman, C., Nick, J., and Tuecke, S. (2002). The physiology of the grid: An open grid services architecture for distributed systems integration.

Foster, I., Kesselman, C., and Tuecke, S. (2001). The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150:1–25.

ISO (1995). International organization for standardization/international eletrotechnical commitee. “qos: Basic framework”. draft international standard (dis) 9309.

Katchabaw, M. J., Lutfiyya, H. L., and Bauer, M. A. (1998). A quality of service management testbed. In *SMW ’98: Proceedings of the IEEE Third International Workshop on Systems Management*, page 57, Washington, DC, USA. IEEE Computer Society.

Li, M. and Baker, M. (2005). *The grid core technologies*. John Wiley & Sons.

Nemeth, Z., Gombas, G., and Balaton, Z. (2004). Performance evaluation on grids: Directions, issues, and open problems. In *in Proceedings of the Euromicro PDP 2004, A Coruna*, pages 290–297. IEEE Computer Society Press.

Nimrod/G, T. N. P. (2005). Nimrod: Tools for distributed parametric modelling.

Office, U. S. G. A. (2002). Applied research and methods: Assessing the reliability of computer–processed data, external version 1.

Roy, A. J. (2001). *End-to-end quality of service for high-end applications*. PhD thesis. Adviser-Ian Foster.

Sahai, A., Graupner, S., Machiraju, V., and van Moorsel, A. (2002). Specifying and monitoring guarantees in commercial grids through sla. In *Internet Systems and Storage, HP Laboratories Palo Alto*.

Schulze, B. (2006). Workgroup proposal: (vcg - virtual community grid). https://vcg.lncc.br.

Sun, Y., He, S., and Leu, J. Y. (2007). Syndicating web services: A qos and user-driven approach. *Decis. Support Syst.*, 43(1):243–255.

West-Brown, M. J., Stikvoort, D., Kossakowski, K.-P., Killcrece, G., Ruefle, R., and Zajicek, M. (2003). Handbook for computer security incident response teams (csirts).

Whitaker, R. (2007). Validation examples of the analytic hierarchy process and analytic network process. *Mathematical and Computer Modelling*, 46(7-8):840–859.

Wu, C. and Chang, E. (2007a). Intelligent web services selection based on ahp and wiki. In *WI ’07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 767–770, Washington, DC, USA. IEEE Computer Society.

Wu, C. and Chang, E. (2007b). A method for service quality assessment in a service ecosystem. In *Digital Ecosytems and Technologies Conference 2007 DEST 07*.

Wu, Y., Liu, L., Mao, J., Yang, G., and Zheng, W. (2007). An analytical model for performance evaluation in a computational grid. In *CHINA HPC ’07: Proceedings of the 2007 Asian technology information program’s (ATIP’s) 3rd workshop on High performance computing in China*, pages 145–151, New York, NY, USA. ACM.