

Comparison of Global Computing with Grid Computing

Babak Behsaz¹, Pooya Jaferian¹ and Mohammad Reza Meybodi¹

¹Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez ave., Tehran, Iran

{behsaz, jaferian, meybodi}@ce.aut.ac.ir

Abstract. In recent years, Global and Grid Computing emerge as two powerful technology trends. In this paper, we compare these two approaches of distributed computing. First, we present a definition for Global Computing that accentuates the key point in this trend. This key point distinguishes Global Computing from other trends and covers many such systems. Second, we contrast two approaches in general characteristics. Then, by comparing them in technical issues, we show that the key point in our definition of Global Computing is the main source of many technical differences between these methods. Finally, we present our opinion about the probable future of Global and Grid Computing and suggest a generic framework that satisfies our views.

Keywords: Distributed Computing, Global Computing, Grid Computing

1 Introduction

Dramatic progress of network technology has led to growing interests in distributed computing approaches in recent years. As a result, Grid Computing and Global Computing came out as appealing research areas. Both of these approaches try to address the problem of utilizing idle distributed resources connected via a network. Various resources of computers from computational sources to software licenses are idle most of the times and they can be used more suitably.

Grid computing is a distributed computing approach to create an illusion of simple yet large virtual computer from a large set of heterogenous computers sharing various resource types to benefit a (virtual) organization. This definition includes a wide range of computing methods and has intersection with many other computing approaches. But, according to the current practical uses of Grid Computing, fulfilling requirements of an organization (or virtual organization) is the key point of Grid Computing [1, 2]. Therefore, many Grid Computing systems suppose existence of some sort of control (most of the time high control) on the entities of the system.

Due to lack of a clear distinction between Global Computing and Grid Computing, a need for comparison of these approaches has raised. This comparison can reveal motivations, strengths and weaknesses of each approach that may help better understanding and selecting of current methods to apply on real world problems. In

this paper, we aim to address this issue from a practical perspective. This means we will compare Global Computing and Grid Computing in the way they are in real systems, not in the way they are theoretically assumed to be. The only work that is to some extent close to ours has expressed some general similarities and differences between Peer-to-Peer (P2P) and Grid Computing [3].

This paper is structured as follows. In the next section, we specify and fix the definition of Global Computing. In section 3, we state general differences and similarities of Global Computing and Grid Computing. These approaches will be compared in technical issues in section 4. Section 5 addresses the opinion of authors about probable future of Grid Computing and Global Computing and suggests an appropriate generic architecture to fulfill this opinion. Finally, last section discusses our concluding remarks.

2 Global Computing

To have a logical comparison of Global Computing and Grid Computing, we should firstly specify what we mean by Global Computing. Unfortunately, in contrast with Grid Computing [4], there are few attempts to present a precise definition of Global Computing until now. Therefore, we will try to present an acceptable definition covering most of the current Global Computing systems. To do this, we reviewed the meaning of this term in proposed and existing Global Computing systems.

Interestingly enough, we observed this term is used in various meanings which are contrasting in some cases. But with a careful inspection, it can be seen that most of the proposed and existing systems have a characteristic in common. Some systems considered Global Computing as a computing approach for computing by computers distributed over world [5, 6, 7, 8]. Some considered it a method for computing in a wide area network [9, 10, 11, 12]. Others assumed it an approach that can be used for computing in the Internet [13, 14, 15, 16, 17]. All these considerations about the meaning of Global Computing have assumed explicitly or implicitly the system will be used for computation in a network of computers scattered world wide. This common point leads to the following definition of Global Computing: "Global computing is a distributed computing approach for computing with a large collection of heterogeneous computers which are *essentially scattered over the world* and sharing various kinds of resources."

This definition is very similar to the definition of Grid Computing and some other computing approaches like Volunteer and P2P Computing, but in fact, it has an important distinction with them. Firstly, we are emphasized that computing entities must be scattered over the world. Secondly, we have not posed any other constraints in this trend of computation like benefiting a (virtual) organization in Grid Computing, having essentially volunteer entities in Volunteer Computing or having decentralized control in P2P computing. Therefore, despite vast intersection of Global Computing with some similar technology trends, none of them cover Global Computing or is covered by it. Thus, every effort for generalizing other similar computing technologies to cover Global Computing will result in loss of efficiency at

least in some cases. Finally, it should be mentioned that there are two major motivations behind world wide scope of these systems: reaching capabilities equal to existing distributed systems with less cost in some cases and reaching more capability than existing systems in other cases. As we discuss in section 4, the mentioned key point causes some remarkable differences in technical issues (or even creates new technical issues) to reach high performance, which is without any doubt the major cause of recent researches in this area.

3 Comparing in General Characteristics

In this section, we compare Global Computing and Grid Computing systems in general qualities. By general qualities, we mean characteristics which can be observed in current proposed and deployed systems without going deep in their behind technical counter parts. In other words, these traits show how current systems look like not how they are intended to be. These characteristics are listed as below and are summarized in table 1.

Knowledge of users: Typically, users of Global Computing systems have less professional knowledge than users of Grid Computing Systems. Because most of them are anonymous individuals that want to use the Global Computing system for their personal will. In contrast, users of Grid commonly have an organizational relationship and some extent of knowledge as members of a (virtual) organization. For example, in well-known SETI@home Global Computing project [18], most of the users were individuals who participated in project for fame and hadn't any special skills for working with their software.

Resources: On the one hand, most of the resources involved in a Grid Computing system are more powerful and better connected than the common Global Computing resources. For example, a cluster or Grid may be nodes of another Grid. In comparison with resources of anonymous users in Global Computing systems, resources in (virtual) organizations using a Grid are distinctly more powerful and better connected to provide the professional requirements of the organizations. On the other hand, number of resources of a Global Computing system is much more than number of resources in a Grid. In most recent Global Computing projects like folding@home [19] and EON [20] over millions of users are involved. Furthermore, although it is theoretically assumed that computers in Global Computing will share various types of resources, but in practice the systems almost always only share computational and informational resources.

Applications: Currently, successful applications on a Global Computing system have two main characteristics. They have limited dependency to data transition and require great computation power. The first characteristic is the result of weak connectivity of resources which leads to great delay in communications. The second characteristic is concluded from the great number of resources which provide enormous computation ability. For example, in SETI@home Classic project finished in December 2005, the largest computation in history is done.

Services and Standards: First, in contrast to Grid Computing, Global Computing systems provide smaller number of services to their users. Also, these services are

very simple in most of the cases. Second, many works are done for standardization of Grid technologies like Open Grid Services Architecture [21] used in Globus Toolkit [22]. The lack of such standardization can be sensed in Global Computing systems. Thus, many Global Computing systems like GIMPS [23] and SETI@home have their own protocols to provide their required functionality. Moreover, lack of standardization in Global Computing has led to little generality of these systems and most of them are special-purpose.

Algorithms: Parallel Algorithms which are used in Global Computing systems are typically asynchronous and simple (in most of the cases, they just partition the problem space). This is a consequence of remarkable delay in communication that makes a synchronous algorithm impractical. In addition, these asynchronous algorithms are mostly very simple, because conventional parallel algorithms rely on some sort of knowledge about the topology and delay in network. Since the network is very dynamic in these systems, this knowledge is not available.

Table 1. Comparison of general characteristics.

Characteristics	Comparison with grid
Users' Knowledge Resources	primitive and less professional weaker, more numerous, and just computational and informational
Applications	low dependence on data transition and more computational
Services and Standards Algorithms	less sophisticated and less numerous mostly asynchronous and very simpler

4 Comparing in Technical Issues

To compare the technical issues in Grid and Global Computing systems, we should first define quality attributes that should be satisfied in each system. Grid and Global Computing share many quality attributes such as performance, heterogeneity, scalability, fault tolerance, security, etc. Yet, respecting trade-offs between mentioned qualities, the main attributes of two systems differ significantly. For Grid [2], the main quality attributes are performance, availability, and interoperability along with providing user with as much functionality as possible. In contrast, respecting our definition of Global Computing, its main quality attribute is scalability and consequently, Global systems must have usability, security, and fault tolerance. These differences have a significant impact on architecture and technical issues of each system. Because a good architecture for a software system should satisfy main quality attributes of that system [24], we first compare general architectural differences of Grid and Global systems. Then, we examine differences relating to design and implementation of mentioned quality attributes and some related technical issues to these attributes. In addition, we show the current solutions to satisfy the required quality attributes in Global Computing systems. We summarize technical differences in Table 2. Notice that detailed analysis of technical differences is beyond the scope

of this paper and we just present general ideas for achieving some parts of important mentioned quality attributes.

Architecture: Currently, most Global Computing systems are based on reference architecture of Grid systems and have not presented a significantly different architecture. Also, the major works on architectures for Global Computing systems are concentrated on coordination architecture [6, 13, 17]. In fact, it can be seen that all these works follow the five layer architecture proposed in [2] to some extents, but the ingredients of each layer differ in two systems according to the differences between quality attributes.

First, in the fabric layer, Grid systems goal is to make many kinds of resources (CPU, Storage, Network, Software, etc.) as transparent as possible (because of the high level of functionality required). But in Global computing systems, doing this makes the Global Computing application very large in size, and therefore it contradicts usability quality attribute of Global Computing systems. Second, in the connectivity layer, Grid systems implement some authentication and encryption mechanisms. But due to the scale of the Global Computing systems, implementation of authentication mechanisms is not feasible, and some other techniques should be used to maintain security requirements. Also in connectivity layer, Global Computing systems should implement some fault tolerance mechanisms. This mechanism should manage high node volatility of Global environments by showing the clients a collection of stable resources. Third, in resource layer, with respect to high performance and functionality required by Grid systems, Grid architecture embodies different techniques for advance resource monitoring, reservation, and managing quality of service. In Global environments, due to lack of control on participating resources, the mentioned techniques are hard to implement. Therefore, Global Computing systems are usually implementing simple resource monitoring and management techniques. Fourth, in collective layer, both systems implement scheduling, brokering, and directory services. But in Grid Computing environments, other services such as data replication, workload management, diagnosis, etc. is implemented. These services have not been implemented on famous Global Computing systems due to limitations of their lower layers. Also mechanisms for scheduling and resource allocation should be much simpler in Global Computing in comparison to Grid. Finally, the application layer in two systems does not differ significantly, except that application development in Grid Computing environments is more systematic as a result of well-defined protocols in lower layers.

Usability: In Grid Computing environments, users are members of professional communities or (virtual) organizations. Therefore, they are forced to share their resources and also learn how to work with the Grid system in order to achieve the virtual organization's goal [3]. In contrast, in Global Computing, because we are seeking as many resources as possible, we want to attract more users even whom are less professional and have their own self interests. Therefore, usability is one of the most important attributes, which leads to three dominant properties. First, the client application of Global Computing systems must be easy to use and have a user friendly environment. Second, most parts (installation, configuration, etc.) of the system must work automatically. For instance, since the underlying network is unknown, a Global Computing system should deal with firewall/NAT/proxy settings automatically [25]. Third, the client application for these systems must be highly portable. Therefore,

many deployed systems use Java programming language to fulfill high portability [13, 17, 25].

Security: This quality attribute is important in both systems. But, tactics for achieving security in Global Computing differ from Grid. The virtual organization nature of Grid makes it possible to identify each participant and maintains moderate level of trust in the environment. Yet, in Global environment due to the scale and usability requirements of the system, it's very hard to identify each user. Therefore, absence of trust is the main security problem in Global environments. Therefore, in famous Grids such as OGSA based Grids [26] and Legion [27] the main security concerns are identification of users, privacy of data and enforcement of virtual organization wide security policies. These concerns are usually addressed by using cryptographic algorithms and authentication and authorization mechanisms. In contrast, in Global computing environments, two major security issues exist as a result of untrusted environment. On the one hand, participant computers privacy and integrity should be maintained and they must be protected from any sort of attacks. In [13, 25], Sandboxing technique and in [14, 17] mechanisms such as software fault isolation and Secure Remote Helper Applications [28] are used to achieve this goal. On the other hand, it is not tolerable for some clients that untrusted hosts have access to their data and algorithms. To deal with this problem [13] suggests using encrypted computing [29] and decomposition of pieces of data and code that should be executed in Global environments. In [30, 31, 32, 33, 34] more techniques are discussed about security in untrusted environments.

Incentive issues: In Grid Computing environments, the users' incentive is achieving virtual organization's goal. Therefore, there is usually no need for charging a user for using the Grid and reward a user for sharing his resources. Yet, in many Global Computing environments, a user's self-interests should be satisfied. Therefore, a need for incentive mechanisms exists in these environments. The general idea for providing an incentive mechanism is to use economic models. For example, in SuperWeb [13], the need for economic model is addressed using an accounting module in the broker. Also, using market-based resource allocation in Global Computing environments is suggested in [14]. Another idea is to attract users by fame, competition, etc. As an illustrative example, sending feedback to users by letting them know their status and rank in computation is another incentive mechanism suggested in [19].

Correctness: This issue is one of the aspects of fault tolerance. Comparing to Grid Computing environments, in Global environments, there is a doubt about the correctness of the result sent by the workers, as a result of untrusted environment. There are two reasons for such behavior: First, economic incentive nature of many Global Computing environments, raises the probability of cheating of the workers in generating results. Second, unknown and even non-standard computing resources of some workers (e.g. old Pentium processors), or changed applications may lead to generating faulty results from workers. In [13] some solutions for detecting such behavior are proposed. Also, [35] suggest methods for eliminating such workers in scheduling algorithms.

Resource allocation: In Grid environments, the availability quality attributes is satisfied. Therefore, we can suppose that the resources are available (if a resource is shared, it must delivers its promised quality of service) and reliable (if a resource is

shared, it must deliver the correct and expected result). In contrast, in Global environments, the resources are not available and reliable. As a consequence, allocation techniques in Global environments should deal with the problem of availability and reliability. Thus, the allocation strategies of Grid system do not address main Global environments concerns. For instance, in Global environments, in contrast to Grid, scheduling techniques should not put a high overhead on the system because of usability and scalability quality attributes. Therefore, adaptive scheduling techniques [36, 37, 38, 39, 40, 41] and methods based on statistical modeling [42, 43, 44, 45, 46] which are introduced for Grid systems, are not applicable for Global systems. To deal with the problem of scheduling, most of the efforts are concentrated on finding easy to measure metrics for resource classification. For example, some simple metrics for availability and reliability is presented in [35]. In addition, some metrics for measuring performance of workers are introduced in [16].

Table 2. Comparison of technical characteristics.

Attribute	Grid Computing	Global Computing
Architecture	Should satisfy performance, availability, interoperability	Should satisfy scalability, usability, security, fault tolerance
Usability	No need for high degree of usability	Client programs should be easy to use, work automatically, and be portable
Security	cryptographic algorithms, authentication, and authorization mechanisms are used	Mechanisms to maintain security in untrusted environment should be used
Incentive Issues	No need for incentive mechanisms in most of the cases	Market mechanisms or other rewarding methods are used to deal with incentive issues
Correctness	No need for checking the correctness of results	Result checking and elimination of faulty results should be performed
Resource Allocation	Adaptive and statistical based methods are used	Simple and easy to measure metrics are used to allocate resources

5 Future of Global Computing and Grid Computing

In previous sections, we compared Global Computing with Grid Computing. Due to expressed differences, it seems that Global Computing and Grid Computing are going to the different ways. But, the abilities of each system are so attractive that it will cause a gradual movement of these technologies toward each other. On the one hand, Global Computing communities seek more sophisticated services and standards, and higher quality of service. On the other hand, Grid Computing communities want to reach scalability of Global Computing systems and consequently, their large computational power. For example, in one of the ongoing Global Computing systems, XtremWeb, every evolution of system conducts it closer to Grid systems [25].

To reach the goal of combining the abilities of these two approaches, we predict some efforts will be made to obtain some general standards that unify necessary services of both approaches. These standards are general in the sense that they will relax limiting assumptions like world wide scalability in Global Computing systems or organizational control in Grids. Of course, it is more likely that these standards will be more similar to current Grid standards. After this standardization and bringing the combined approach to life, great efforts will be started to extend large volume of works in these two approaches to new combined approach. To facilitate evolution process to combined general approach, we suggest a two layered framework. The first layer of this framework provides the interfaces required by unified services in combined approach. This is the layer that developers interact with for using the system. The second layer *separately* deals with implementation of the services for the case of world wide scalability and for the case of organizational control. It should be noted we believe that to reach efficiency, implementation of most services like security must differ in each case. But this difference must be hidden from developers because it doesn't change the functionality they need. Therefore, we devoted the second layer for this purpose. To illustrate more, notice that this generic framework follows the same idea as protocol stack idea in network technology. Finally, it should be noted these layers has just a conceptual meaning and with respect to system requirements may contain some other layers.

6 Conclusion

Global and Grid Computing are distributed computing approaches which their main idea is utilization and organization of unused computer resources connected by a network. Although the main idea of these methods is similar, they are following it from different directions. Global computing systems are trying to refine their standards and Grids systems are seeking to reach more scalability. Due to more popularity and more growth of Grid systems, sometimes it is supposed that Grid Computing covers Global Computing. Therefore, according to this opinion, it is sometimes assumed that Grid technology can be applied for Global Computing tasks with some mild modifications. But, due to the definition and consequent comparison we presented in this paper, we disagree with such opinions. Because the world wide scalability of Global Computing systems leads to a great technical differences that needs new specialized methods to avoid loss of efficiency. In authors' opinion, the only logical generalization of Grid systems for covering Global Computing systems can be reached by a framework like what we proposed in the previous section.

References

- [1] Ferreira L. et al.: Introduction to Grid Computing with Globus. IBM Red Books, New York (2003)

- [2] Foster I., Kesselman C., Tuecke S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*. 15(3) (2001) 200-202
- [3] Foster I, Iamnitchi A. On Death, Taxes, and Convergence of Peer-to-Peer Computing and Grid Computing. 2nd International Workshop on Peer-to-Peer Systems. (2003)
- [4] Foster I.: What is a Grid? A Three Point Checklist. *GRIDToday*: www.Gridtoday.com. (2002)
- [5] Bahi, J. M., Vivier, S. C., Cuturier, R.: Asynchronism for Iterative Algorithms in a Global Computing Environment. *Proceedings of IEEE Sixteenth Annual International Symposium on High Performance Computing Systems and Applications*. (2002) 90-97
- [6] Baldeschwieler, J. E., Blumofe, R. D., Brewer, E. A.: ATLAS: An Infrastructure for Global Computing. In *Proceedings of the Seventh ACM SIGOPS European Workshop: Systems support for Worldwide Applications*. (1996) 165-172
- [7] Sato, M., Tezuka, H., Hori, A., Ishikawa, Y., Sekiguchi, S., Nakada, H., Matsuoka, S., Nagashima, U.: Ninf and PM: Communication Libraries for Global Computing and High Performance Cluster Computing. *Journal of Future Generation Computer Systems*. 13 (1998) 349-359
- [8] Wolski, R., Nurmi, D., Brevik, J.: Models and Modeling Infrastructures for Global Computational Platforms. *Proceedings of the nineteenth IEEE International Parallel and Distributed Processing Symposium*. (2005) 224-231
- [9] Aida, K., Takefusa, A., Nakada, H., Matsuoka, S., Nagashima, U.: A Performance Evaluation Model for Effective Job Scheduling in Global Computing Systems. *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing*. (1998) 352-353
- [10] Nakada, H., Sato, M., Sekiguchi, S.: Design and Implementations of Ninf: towards a Global Computing infrastructure. *Journal of Future Generation Computer Systems*. 15 (1999) 649-658
- [11] Sato, M., Kusano, K.: NetCDF: a Ninf CFD componenet for Global Computing, and its Java applet GUI. In *Proc. of the Fourth IEEE International conference on High Performance Computing*. (2000) 501-506
- [12] Suzumura, T., Nakagawa, T., Nakada, H., Sekiguchi, S.: Are Global Computing Systems Useful? Comparison of Client-Server Global Computing Systems Ninf, NetSolve versus CORBA. In *Proc of the Fourteenth IEEE International Parallel and Distributed Processing Symposium*. (2000) 547-556
- [13] Alexandrov, A. D., Ibel, M., Schauser, K. E., Scheiman, C. J.: SuperWeb: Towards a Global Web-Based Parallel Computing Infrastructure. *11th IEEE International Parallel Processing Symposium*. (1997) 100-106
- [14] Cappello, P., Christiansen, B. O., Neary, M. O., Schauser, K. E.: Market-Based Massively Parallel Internet Computing. *Proceedings of IEEE Conference on Massively Parallel Programming Models*. (1997) 118-129
- [15] Fedak, G., Germain, C., Neri, V., Cappello, F.: XtremWeb: A Generic Global Computing System. *Proceedings of IEEE Workshop on Global Computing on Personal Devices*. (2001) 582-587
- [16] Kondo, D., Casanova, H., Wing, E., Berman, F.: Models and Scheduling Mechanisms for Global Computing Applications. *Proceedings of IEEE International Parallel and Distributed Processing Symposium*. (2002) 216-229
- [17] Neary, M. O., P. Brydon, S., Kmiec, P., Rollins, S., Cappello, P.: Javelin++: Scalability Issues in Global Computing. *ACM Java Grande*, (1999) 171-180
- [18] Anderson, D. P. et al.: SETI@home: An Experiment in Public Resource Computing. *Communications of ACM*. 45(13) (2002) 56-61

- [19] Larsen, S. M. et al.: Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology. *Computational Genomics*. (2003)
- [20] Henkelman Research Group: EON Project. <http://eon.cm.utexas.edu>. (2003)
- [21] Foster, I., Kesselman, C., Nick, J., Tuecke, S.: Grid Services for Distributed System Integration. *IEEE Computer Journal*. 35(6) 2002 37-46
- [22] Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. *International Conference on Network and Parallel Computing*, Springer-Verlag Lecture Notes in Computer Science. (2005) 2-13
- [23] Great Internet Mersenne Prime Search Project. <http://www.mersenne.org>. (2005)
- [24] Bass, L., Clements, P., Kazman, R.: *Software Architecture in Practice*. 2nd edn., Addison-Wesley. (2003)
- [25] Cappello, F., Djilali, S., Fedak, G., Heralut, T., Magniette, F., Neri, V., Lodygensky, O.: Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with Grid. *Future Generation Computer Systems*. 21 (2005) 417-437
- [26] Kunszt, P.Z., Guy, L.P.: The Open Grid Services Architecture, and Data Grids. In : Berman, F., Fox, G., Hey, T., (ed): *Grid Computing: Making the Global Infrastructure a Reality*. John Willey & Sons Inc (2003) 385-408
- [27] Grimshaw, A.S., Natrajan, A., Humphrey, M.A., Lewis, M.J., Nguyen-Tuong, A., Karpovich, J.F., Morgan, M.M., Ferrari, A.J.: From Legion to Avaki: The Persistence of Vision. In : Berman, F., Fox, G., Hey, T., (ed): *Grid Computing: Making the Global Infrastructure a Reality*. John Willey & Sons Inc (2003) 265-298
- [28] Goldberg, I., Wagner, D., Thomas, R., Brewer, E. A.: A Secure Environment for Untrusted Helper Applications: Confining the Wily Hacker. In *Proceedings of the 1996 USENIX Security Symposium*. (1996)
- [29] Feigenbaum, J.: Encrypting Problem Instances - Or, ..., Can You Take Advantage of Someone Without Having to Trust Him?. In *Proc. of the 5th IACR Conf. on Crypto* (1985) 477-488.
- [30] Bolosky, W. J., Douceur, J. R., Ely, D., Theimer, M.: Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs. In *Proc. International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2000)*. ACM Press, New York (2000) 34-43.
- [31] Kubiawicz, J. et al.: OceanStore: An architecture for global-scale persistent storage. In *Proc. of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*. (2000)
- [32] Zhuang, S. Q., Zhao, B. Y., Joseph, A. D., Katz, R. H., Kubiawicz, J. : Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination. In *Proc. the Eleventh International Workshop on Network and Operating System Support for Digital Audi and Video (NOSSDAV 2001)*. (2001) 11-20
- [33] Fu, K., Frans Kaashoek, M., Mazières, D.: Fast and secure distributed readonly file system. *Int. J. Computer Systems*. 1. (2002) 1-24
- [34] Druschel, P., Rowstron, A.: PAST: A large-scale, persistent peer-to-peer storage utility. *HotOS VIII. Germany: Schloss Elmau*. (2001) 75-80
- [35] Tauber, M., Teller, P.J., Anderson, D.P., Brooks, C.L.: Metrics for Effective Resource Management in Global Computing Environments. In *Proc. Of the 1st IEEE International Conference on e-Science and Grid Computing*. Melbourne (2005), 204-211.
- [36] Heymann, E., Senar, M.A., Luque, E., Livny, M.: Adaptive Scheduling for Master-Worker Applications on the Computational Grid. In *Proc. of the First IEEE/ACM International Conference on Grid Computing (GRID'00)*, (2000) 214-227

- [37] Berman, F., et al.: The GrADS Project: Software Support for High-Level Grid Application Development. *Int. J. High Performance Computing Applications*. 15(4) (2001) 327-344
- [38] Vadhiyar, S.S., Dongarra, J.J.: Self Adaptivity in Grid Computing. *Concurrency and Computation: Practice and Experience* 17(2-4): (2004) 235-257
- [39] Berman, F., et al.: New Grid Scheduling and Rescheduling Methods in the GrADS Project. *Int. J. of Parallel Programming*. 33(2) (2005) 209-229
- [40] Buyya, R., Abramson, D., Giddy, J.: Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid. In *Proc. of the 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*, Beijing, China. IEEE Computer Society Press, USA (2000)
- [41] Venugopal, S., Buyya, R., Winton, L.: A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids. In *Proc. of the 2nd International Workshop on Middleware for Grid Computing (Co-located with Middleware 2004, Toronto, Ontario - Canada, October 18, 2004)*, ACM Press, USA (2004)
- [42] Brevik, J., Nurmi, D., Wolski, R.: Automatic Methods for Predicting Machine Availability in Desktop Grid and Peer-to-Peer Systems. In *Proc. of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid'04)*, (2004) 190-199
- [43] Nurmi, D., Brevik, J., Wolski, R.: Minimizing the Network Overhead of Checkpointing in Cycle-harvesting Cluster Environment, In *Proc. of the IEEE International Conference on Cluster Computing (Cluster 2005)*. (2005)
- [44] Nurmi, D., Brevik, J., Wolski, R.: Modeling Machine Availability in Enterprise and Wide-area Distributed Computing Environments. Technical Report CS2003-28 U.C. Santa Barbara Computer Science Department. (2003)
- [45] Kondo, D., Taufer, M., Brooks, C. L., Casanova, H., Chien, A.A.: Characterizing and Evaluating Desktop Grids: An Empirical Study. In *Proc. of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS'04)*. (2004) 17-25
- [46] Kondo, D., Chien, A.A., Casanova, H.: Resource Management for Rapid Application Turnaround on Enterprise Desktop Grids. In *Proc. of the ACM Conference on High Performance Computing and Networking (SC'04)*, (2004)