Performance Evaluation of a Resource Discovery Scheme in a Grid Environment Prone to Resource Failures

Konstantinos I. Karaoglanoglou, and Helen D. Karatza
Aristotle University of Thessaloniki
kkaraogl@csd.auth.gr

Abstract

This paper studies the problem of discovering the most suitable resource for a specific request in a Grid system. A Grid can be seen as an environment comprised by routers and resources, where each router is in charge of its local resources. In our previous works we enhanced the routers of the system with matchmaking capabilities in order to determine an appropriate set of resources capable of satisfying a specific request. Moreover, we presented an efficient resource discovery mechanism called Re-routing Tables that directs the requests to the resources capable of satisfying them in a dynamical Grid system, where resources are not statically online. In this paper, we present an expansion of our resource discovery scheme in order to cover the cases of consecutive resource failures, and we emphasize in the performance evaluation of our resource discovery scheme by providing new sets of simulation tests in Grid environments that are prone to resource failures.

1. Introduction

A Grid can be defined as "a large-scale, geographically distributed, hardware and software infrastructure composed of heterogeneous networked resources owned and shared by multiple administrative organizations which are coordinated to provide transparent, dependable, pervasive and consistent computing support to a wide range of applications. These applications can perform distributed computing, high throughput computing, on-demand computing, data-intensive computing, collaborative computing or multimedia computing" [1].

One of the main capabilities a Grid infrastructure needs to support is a resource discovery mechanism [2]. The base of Grid technology is the concept of resource-sharing. Resources shared in a Grid system could be desktop systems, clusters, storage devices and large data-sets. A remote user should be able to gain access to a remote resource either to execute a job or to have access in the resource's data. An effective resource discovery mechanism is responsible of supporting this operation in a Grid system.

In a Grid environment, there are certain factors that make the resource discovery problem difficult to solve. These factors are: the huge number of resources, distributed ownership, heterogeneity of resources, resource failure, and resource evolution (upgrades changing a resource's technical characteristics). An efficient resource discovery mechanism should take into consideration the above factors.

Combining matchmaking and routing approaches, we propose a resource discovery scheme that can guarantee discovering the most suitable resource for a specific request and then directing that request to the appropriate resource in a Grid environment, where resource failures are a common fact. Moreover, the proposed scheme deals effectively with the phenomenon of consecutive resource failures.

2. Related work

One of the popular approaches for the Grid Resource Discovery problem is the Matchmaking one [3, 4, 8, 11, 14, 15, 18, and 20]. The Matchmaking resource discovery framework suggests categorizing the entities comprising a Grid system into requestors, and providers. These entities advertise their characteristics and requirements to a matchmaking service, which is responsible of finding a match between the advertisements and informing the relevant parties of the match. The matched entities connect and cooperate for the service's execution.

It is a well-known fact that Grid and Peer-to-Peer systems share the same main concept. They are both resource-sharing environments. Their difference is that they have followed different evolutionary paths. Grid systems are mainly used in complex scientific applications, while Peer-to-Peer systems are developed around mainstream services such as file-sharing.

978-1-4244-3750-4/09/$25.00 ©2009 IEEE

Several research papers suggest the application of already known Peer-to-Peer resource discovery mechanisms into the Grid [5, 6, 13, 16, 19, and 21].

Expanding the Matchmaking approaches, another notable approach to the Resource Discovery problem is the Semantic Communities one [7, 10, 12, and 17]. The main concept behind this approach is that Grid communities and human communities consist of members that are engaged in sharing and communication. Grid communities are formed based on similar-interests policies, allowing community nodes to learn of each other without relying on a central meeting point.

Another interesting approach to the resource discovery problem is the agent-based one [26]. Each agent in the Grid system acts as a representative for a local Grid resource and also cooperates with other agents to perform service advertisement and discovery. The agent-based approach also includes a matchmaking framework, in which metrics such as the deadline for the execution of an application and the earliest time at which the application will terminate play a crucial role.

To the best of our knowledge, a resource discovery framework combining matchmaking and routing concepts has never been deployed before. Using semantic concepts for describing resources and requests, and then directing the requests in the Grid environment with an efficient routing mechanism, we achieve to overcome the obstacle of resource failures bound to happen in any Grid system.

3. Depiction of the Grid environment

A Grid system can be seen as an environment comprised by matchmaking-routers and resources. Each router is in charge of its local resources and also connects with other routers within the Grid system. Figure 1 presents a Grid system based on the matchmaking-router model. The system is comprised by three matchmaking-routers, where each one controls its local resources.

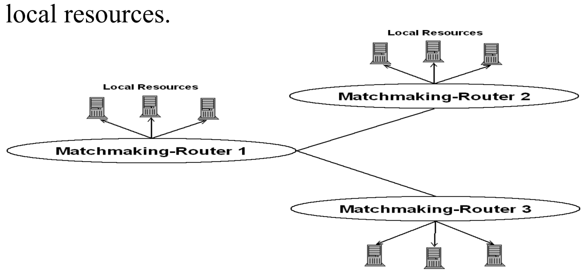

Figure 1. A Grid system comprised by matchmaking routers and resources

4. Matchmaking

4.1. Descriptions of requests and resources

A request created in a matchmaking-router at some point of time describes four basic characteristics [23] in order to get satisfied. These required characteristics are: architecture, operating system, disk, and memory. Resources that are available in the Grid system also use descriptions of their characteristics. A Grid resource is fully described using the four characteristics mentioned above.

After a request is created in a matchmaking-router, the router uses the descriptions of Grid resources to find an appropriate match. Since all the descriptions of resources are available to the routers, the matchmaking process is considered relatively easy.

4.2. Matchmaking rules

The matchmaking process in the resource discovery scheme has to obey certain rules. The matchmaking rules determine the set of candidate resources that can satisfy a specific request.

The basic matchmaking rules in the framework are the following:

- The architecture and operating system characteristics of the request must match the architecture and operating system characteristics of the resource.
- The minimum disk size required by the request must be smaller or equal to the available disk size of the resource.
- The minimum memory space required by the request must be smaller or equal to the available memory space of the resource.

An example of a request created in a matchmaking-router is shown in Figure 2. The resources available to the supposed Grid system are four and their description types are shown in Figure 3. Resources of types 3 and 4 are capable of satisfying the request due to matches in the architecture and operating system characteristics. Note that the available disk and memory characteristics also conform to the request's minimum disk and memory requirements.

Architecture: Intel
Operating System: Linux
Minimum Disk: 35000
Minimum Memory: 1024

Figure 2. An example of a request's requirement

Resource Type 1
Architecture: Intel
Operating System: Solaris26
Available Disk: 20000
Available Memory: 512

Resource Type 2
Architecture: SGI
Operating System: Irix6
Available Disk: 40000
Available Memory: 1024

Resource Type 3
Architecture: Intel
Operating System: Linux
Available Disk: 40000
Available Memory: 2048

Resource Type 4
Architecture: Intel
Operating System: Linux
Available Disk: 35000
Available Memory: 1024

Figure 3. Available Grid resources

The matchmaking-router based on the matchmaking rules concluded to the set of resources that are capable of satisfying the request. The question now is which of the two resources of type 3 and 4 is the most suitable to satisfy the request? Resource of type 3 has larger disk size and memory space from the request's requirements and its capabilities could be needed to a more demanding future request. On the other hand, resource of type 4 fits best with the request's requirements.

Given a set of resources that are capable of satisfying a request, the Best Fit resource is the one that has:

- a smaller or zero difference between available disk size and required minimum disk size,
- and a smaller or zero difference between available memory space and required minimum memory space.

For the example request in Figure 2, the request would be directed to the most suitable resource of type 4. The difference of the available disk size and memory space of resource type 4 from the minimum required disk size and memory space of the request is zero.

5. Directing requests

5.1. Routing Tables

Assuming that at a point of time a request for a specific resource is created in one of the matchmaking-routers in the Grid system, an efficient resource discovery mechanism should be able to find an appropriate resource in the system. In the simplest case, the matchmaking-router checks if one of its local resources meets the request's requirements. If not, the router should forward the request to its neighbors randomly. This random forwarding could be sufficient in a small network, but would not satisfy a large system's needs. The Routing Tables mechanism is used in order to forward the requests in the Grid system in a non-random way.

Each router in the Grid system maintains a Routing Table [9] with size equal to the number of different resources in the network. Each data element in that table is the minimum distance measured in hops from that router to all the resources available in the network.

This procedure is effective when all the available resources in the system are statically online. A failure to a random resource in the system causes that specific resource to exit the Grid system, therefore to get in an offline state. Because the Routing Tables mechanism is not capable of directing requests in an environment where resources could get in an offline state at any point of time, we deployed the Re-routing Tables mechanism.

5.2. Dealing with Resource Failures

Figure 4 presents a supposed Grid system comprised by twelve matchmaking- routers (MR stands for matchmaking-router). Each router in this system controls a number of local resources between three and five. At some point of time a request for the resource of type 9 is created in the MR11. Based on the information maintained in its Routing Table (shown in Table 1), MR11 forwards the request through MR6 to MR7, where the requested resource exists locally. The problem is that resource of type 9 in MR7 has gotten in the offline state due to a resource failure. Due to this offline resource event, an alternative resource of type 9 has to be discovered somewhere in the Grid system. Such a resource type exists locally in MR5.

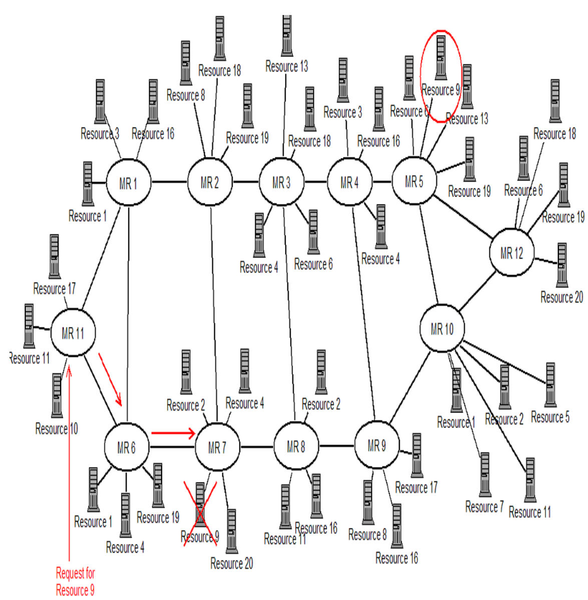

Figure 4. An example of a re-routing event

We compute the new minimum distance from MR7 to resource of type 9 (5 hops). The problem is that there are still routers in the network who still think that Resource 9 is online in MR7 and they will keep forwarding requests for that specific resource to MR7. Even though an alternative resource exists in the system, the matchmaking-routers do not have the new information in their Routing Tables.

Table 1. Routing Tables available in Matchmaking-Routers. Distance in hops from each router to the resource of type 9.

Matchmaking-Routers	Resource 9
1	3 hops
2	2 hops
3	3 hops
4	2 hops
5	Local
6	2 hops
7	Local
8	2 hops
9	3 hops
10	2 hops
11	3 hops
12	2 hops

When the fact that a resource failure happened somewhere in the Grid system, we can re-compute the minimum distances for all routers in the network in order to update the information in the Routing Tables regarding the offline resource. Of course this solution is computationally intensive especially in cases of large systems. We can effectively reduce the computational cost of the above solution by noticing that not all matchmaking-routers forward requests to the matchmaking-router where the offline resource event occurred.

Examining the Grid system in Figure 4, it is obvious that not all routers forward requests for resource of type 9 to MR7. Therefore updating the Routing Tables for all routers is unnecessary. A proper solution is to find the routers that forward requests for resource of type 9 to MR7 and only update their Routing Tables creating their Re-routing Tables. From a total of 12 routers in Figure 4 only the following forward requests for Resource 9 to MR7: MR1, MR2, MR6, MR8, and MR11. All the other matchmaking-routers forward requests to the alternative resource of type 9 existing locally in MR5.

Once an offline resource event due to a resource failure is established in a router of the network, we have to determine which matchmaking-routers are going to update their Routing Tables. The idea is to start from the matchmaking-router where the offline resource event occurred and visit all nodes in the system to examine which of them do forward requests for the offline resource to that specific router. Counting hops from the offline resource router and checking the Routing Tables of the visited nodes will determine whether or not a router should update its Routing Table. If the calculated hops are equal to the distance in hops included in a matchmaking-router's Routing Table for that offline resource then this router has to update its Routing Table creating its Re-routing Table.

Starting from MR 7, in which the offline resource event occurred and with the knowledge included in the Routing Tables shown in Table 1 we conclude to the following regarding MR1 and MR4. The distance of resource 9 in MR7 from MR1 is 3 hops, equal to the distance represented in MR1's Routing Table. So MR1 forwards requests for resource 9 to MR7. The distance of resource 9 in MR7 from MR4 is 4 hops, not equal to the distance represented in MR4's Routing Table (2 hops). This means that MR4 does not forward requests for resource 9 to MR7 and forwards them to another router controlling locally a resource of that type. Following the above procedure for all the routers in the system we conclude to the matchmaking-routers that need to update their Routing Tables.

A significant problem lies in MRs 3 and 9. The distance of resource 9 in MR7 from MR3 and MR9 is 3 hops, equal to the distance represented in both of the routers Routing Tables. The distance of the alternative resource of type 9 in MR5 from MR3 and MR9 is also 3 hops. So it is not clear to which matchmaking-router the MR3 and MR9 forward requests for resource of type 9. In order to avoid problems in future requests we update their Routing Tables creating their Re-routing Tables anyway.

We concluded that MRs 1, 2, 3, 6, 8, 9, and 11 including MR7 must update their Routing Tables in order to satisfy future requests for the offline resource of type 9. Computing the minimum distances for these routers has as a result the creation of Re-routing Tables shown in Table 2. Note that the distances shown here are different for MRs 1, 2, 3, 6, 7, 8, 9, and 11 from those in Table 1 because of the update. The distance for all the other routers is the same because they were unaffected from the departure of resource 9 in MR7.

Table 2. Re-routing Tables for all matchmaking-routers after the update regarding resource of type 9. Bold rows are the matchmaking-routers that update their information regarding offline resource 9

Matchmaking-Routers	Resource 9
1	**5 hops**
2	**4 hops**
3	**3 hops**
4	2 hops
5	Local
6	**6 hops**
7	**5 hops**
8	**4 hops**
9	**3 hops**
10	2 hops
11	**6 hops**
12	2 hops

5.3. Consecutive Resource Failures

An important aspect of the proposed resource discovery scheme is that it covers the cases of consecutive resource failures. As mentioned before, when a request is created in a matchmaking-router, the router is responsible of providing the Best Fit resource capable of satisfying the request. If the Best Fit resource is in an offline state due to a resource failure, the Re-routing Tables mechanism provides an alternative resource somewhere in the Grid system. There are rare cases when the alternative resource is also offline due to another resource failure.

In Figure 5, a case example of consecutive resource failures is presented. At some point of time a request for resource of type 4 is created in MR1. The request is forwarded in MR4, where resource of type 4 exists locally, but it is currently offline due to a resource failure. The update phase in the Routing Tables would suggest directing the request to the neighbour Grid node MR5, where an alternative resource of type 4 exists locally. Due to a resource failure, this specific resource in MR5 is also offline. The update phase in the Routing Tables would suggest directing the request to MR6, where a resource of type 4 is online and available. After two consecutive resource failures and travelling through three intermediate routers (MR4, MR5, and MR3), the request is finally satisfied.

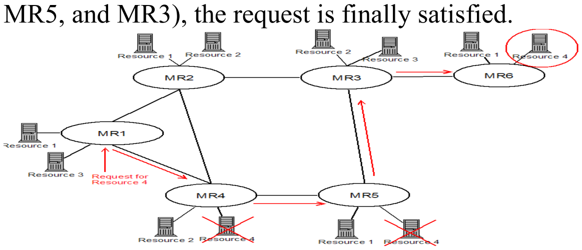

Figure 5. An example case of consecutive resource failures

Considering the nature of a Grid system, one may suggest that a phenomenon of consecutive resource failures is extremely rare. Though rare, the case of consecutive resource failures cannot be ignored. If this phenomenon is not dealt effectively, it could lead to discarded and unsatisfied requests. The Re-routing Tables mechanism is able of coping with consecutive resource failures in the Grid system. For consecutive resource failures, the Re-routing Tables procedure works in a recursive way in each matchmaking-router, where a resource failure occurred.

6. Testing

6.1. Simulation Tests and Performance Parameters

In order to emphasize in the performance evaluation of the proposed resource discovery scheme, we performed two types of simulation tests. In the first type of tests, we used Grid environments where resource failures occur in a rare way. We characterize these environments as “friendly” Grid systems. In the second type of tests, resource failures occur in a vast, rapid, and common way. We characterize these environments as “hostile” environments. Categorizing the simulation tests in two types (“friendly” and “hostile” Grid systems) is crucial for evaluating the performance of our scheme. A resource discovery scheme ought to behave well in both “friendly” and “hostile” environments.

As presented in Section 3, the Grid environment is comprised by matchmaking-routers and resources, where each router controls its local resources. In such a network-like Grid system, the performance parameter for our proposed scheme is the distance in hops a request traveled to get satisfied [25]. A small distance in hops indicates better performance.

6.2. Environment

Grid Graph generator produced the Grid systems [22] for our simulation needs. The Grid Graph generator produced the backbone of the systems, meaning the matchmaking-routers. After this we allocated a certain number of resources to each matchmaking-router in the system. The number of resources that a matchmaking-router can control locally is three to five and the different types of resources available in the system are twenty. Note that a resource of a specific type can exist locally in the same matchmaking-router more than once. Taking into consideration that a resource discovery framework should perform well in both small and large Grid systems, simulation tests were conducted in “friendly” and “hostile” Grid systems of 202, 402, 602, 802 and 1002 matchmaking-routers. Due to space limitations not all results and tests are presented here. The results presented here concern the cases of “friendly” and “hostile” Grid systems of 1002 matchmaking-routers.

Requests are created in random matchmaking-routers somewhere in the system. The matchmaking-routers using the available descriptions of resources, and the matchmaking rules are responsible of providing the Best Fit resource capable of satisfying the request. After the Best Fit resource is identified, the request is forwarded in the system using the Routing Tables mechanism. For the cases of resource failures the forwarding of the request happens with the support of the Re-routing Tables mechanism.

Independently of the creation of requests, the offline resource events due to resource failures happen in a random way also. At some point of time a single random resource in a random router gets in the offline state. For the cases of “friendly” Grid systems of 1002 matchmaking-routers, a random resource gets in the offline state every 1 time unit of the simulation. For the cases of “hostile” Grid systems of 1002 matchmaking-routers, four random resources get in the offline state every 1 time unit of simulation.

We have used the ability of the Grid Graph generator of producing different topologies for the same size of network. For every size of a Grid system, the proposed resource discovery scheme was tested in 4 different topologies of that exact size. The results presented here are the final averages of the four different executions in four different topologies for a “friendly” and a “hostile” Grid system of 1002 matchmaking-routers.

6.3. Hybrid Matchmaking Resource Discovery

In order to compare the behaviour of the proposed resource discovery scheme, we deployed another resource discovery mechanism called Hybrid-Matchmaking Resource Discovery. The hybrid mechanism works in a semi-random way, based on the popular random-walk approach [25]. When a request is created in a matchmaking-router, the router is again responsible of providing the Best Fit resource capable of satisfying the request. The forwarding of the request happens with the following procedure. The matchmaking-router checks in its local resources for the Best Fit resource. If the resource is not found there, the request is randomly forwarded to neighboring nodes. To avoid unnecessary cycles in the network, nodes that are visited from a request are marked, so they’re not chosen in a future forward of the same request.

6.4. Results for the “friendly” Grid system

Results presented here are the final averages of the four different executions in four different topologies in a “friendly” 1002 Routers Grid system. A random resource gets in the offline state due to a resource failure every 1 time unit of simulation. The Matchmaking-Routers approach is the one that makes use of the Re-routing Tables mechanism when a resource failure event is acknowledged in some router of the system. The Hybrid Matchmaking-Routers approach is the one described in Section 6.3.

Figure 6 presents the evolution of simulation in terms of distance in hops for 1000 requests based on the Matchmaking-Routers. Resource discovery for the most cases of requests happened in small level of hops around 2 and 3 hops. Of course, there are slightly higher results above 5 hops, but considering the large size and the dynamicity of the Grid system, results are acceptable. It appears that the proposed resource discovery scheme guarantees discovering the Best Fit resource successfully and in a small level of hops.

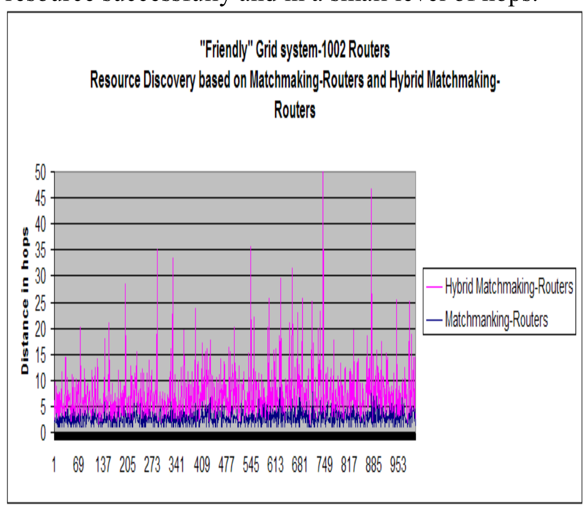

Figure 6. Resource Discovery based on Matchmaking-Routers and Hybrid Matchmaking-Routers in a 1002 Routers Grid system

Figure 6 also presents the evolution of simulation in terms of distance in hops for 1000 requests based on the Hybrid Matchmaking-Routers. Results are bad since for the most cases of requests the discovery of the Best Fit resource demands around 4 and 8 hops. For some cases of requests the distance in which a request gets satisfied could even reach to levels up to 50 hops. Results for the Hybrid Matchmaking-Routers were partially expected to be in these levels. The technique’s bad behaviour is mainly caused to the semi-random walk approach combined with the large size and the dynamicity of the Grid system. These factors lead to these unacceptable results.

6.5. Results for the “hostile” Grid system

Results presented here are the final averages of the four different executions in four different topologies in a “hostile” 1002 Routers Grid system. Four random resources get in the offline state due to a resource failure every 1 time unit of simulation.

Figure 7 presents the evolution of simulation in terms of distance in hops for 1000 requests based on the Matchmaking-Routers. For the first 400 time units of simulation, resource discovery for the most cases of requests happened in small levels of hops, around 2 and 5 hops. For the other 600 time units of simulation, results are higher, around 5 and 10 hops. It appears that the vastness, in which resource failures occur in the system, affects the resource discovery scheme. This fact was expected. As the simulation time proceeds, more and more resources exit the Grid system due to failures, and the resource discovery scheme has to travel larger distances in order to discover alternative resources capable of satisfying the requests. Taking in consideration the “hostility” of the environment, results are still in acceptable levels of hops.

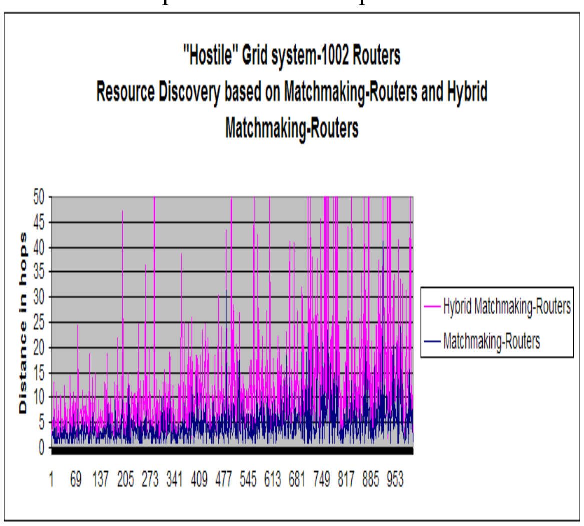

Figure 7. Resource Discovery based on Matchmaking-Routers and Hybrid Matchmaking-Routers in a 1002 Routers Grid system

Figure 7 also presents the evolution of simulation in terms of distance in hops for 1000 requests based on the Hybrid Matchmaking-Routers. Results are extremely bad. The vastness, in which resource failures occur, appears to have negative effects in the Hybrid Matchmaking-Routers approach from the beginning of the simulation. For the first half of simulation, the approach presents high results, around 5 and 10 hops. Of course there are a lot cases in which resource discovery demands extremely larger distances in hops. As simulation time proceeds and more resources exit the system, the Hybrid Matchmaking-Routers approach presents catastrophic results. For the most cases of requests, resource discovery demands around 20 to 50 hops. It appears that the Hybrid Matchmaking-Routers cannot deal effectively with the “hostility” of the environment.

7. Conclusions and Future Research

We presented a resource discovery scheme capable of overcoming the obstacles of resource failures bound to happen in any Grid system. The proposed resource discovery scheme includes a simple and effective matchmaking framework in order to identify the Best Fit resource for the request’s requirements. Directing of the requests in the Grid system happens with the Re-routing Tables mechanism, which is capable of dealing with the cases of resource failures, and consecutive resource failures.

Finally, we presented two types of simulation tests: in “friendly” and “hostile” Grid systems. The proposed resource discovery scheme with Matchmaking-Routers proved its superiority against the semi-random Hybrid Matchmaking-Routers. For both types of tests, the proposed resource discovery scheme presented extremely well results, despite the offline resource events caused by resource failures.

It is in our future intentions to continue evaluating the performance of the proposed resource discovery mechanisms taking into consideration additional performance parameters, such as the reliability of the Grid resources, or directing requests to Grid resources that guarantee the minimum cost. Moreover, further research is needed in order to address the problem of “resource evolution”. The term “resource evolution” refers to the upgrades that change the technical characteristics of a resource. It is in our intentions to expand the proposed resource discovery scheme in order to cover these cases.

8. References

[1] Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis, and Eduardo Gomez-Sanchez, “Grid characteristics and uses: a Grid Definition”, *Postproc. of the First European Across Grids Conference (ACG’03)*, Springer-Verlag LNCS 2970, pp. 291-298, Santiago de Compostela, Spain, Feb. 2004.
[2] Rajkumar Buyya and Srikumar Venugopal, “A Gentle Introduction to Grid Computing and Technologies”, *Computer Society of India*, July 2005.
[3] R. Raman, M. Livny, M. Solomon, “Matchmaking: Distributed Resource Management for High Throughput

Computing", hpdc, p. 140, *Seventh IEEE International Symposium on High Performance Distributed Computing (HPDC-7 '98)*, 1998.

[4] Rajesh Raman, "Matchmaking frameworks for distributed resource management", University of Wisconsin-Maddison, 2001.

[5] Adriana Iamnitchi, Ian Foster, and Daniel C. Nurmi, "A Peer-to-Peer approach to Resource Location in Grid Enviroments", *Proceedings of the 11th Symposium on High Performance Distributed Computing*, Edinburgh, UK, August 2002.

[6] N. A. Al-Dmour and W. J. Teahan, "Peer-to-Peer protocols for resource discovery in the Grid", *Parallel and Distributed Computing and Networks*, Innsbruck Austria, 2005.

[7] Juan Li and Son Vuong, "Grid Resource Discovery Using Semantic Communities", *Proceedings of the 4th International Conference on Grid and Cooperative Computing*, Beijing, China, November, 2005.

[8]Hongsuda Tangmurarunkit, Stefan Decker, Carl Kesselman, "Ontology-based Resource Matching in the Grid-The Grid meets the Semantic Web", *International Semantic Web Conference*, 2003.

[9] Wei Li, Zhiwei Xu, Fangpeng Dong, and JunZhang, "Grid Resource Discovery based on a Routing-Trasferring Model", *Grid 2002*, LNCS 2536, pp.145-156, 2002.

[10] Juan Li, Son Vuong, "Semantic Overlay Network for Grid Resource Discovery", *Grid Computing Workshop*, 2005.

[11] Ye Zhu, Junzhou Luo, and Teng Ma, "Dividing Grid Service Discovery into 2-stage Matchmaking", *ISPA2004*, LNCS 3358, pp. 372-381, 2004.

[12] Cheng Zhu, Zhong Liu, Weiming Zhang, Weidong Xiao, Zhenning Xu, and Dongsheng Yang, "Decentralized Grid Resource Discovery based on Resource Information Community", *Journal of Grid Computing*, Springer, 2005.

[13] Kashif Ali, Suprakash Datta, Mokhtar Aboelaze, "Grid Resource Discovery using Small World Overlay Graphs", *Proceedings of the 18th IEEE Canadian Conference on Electrical and Computer Engineering*, 2005.

[14] Sanya Tangpongprasit, Takahiro Katagiri, Hiroki Honda, Toshitsugu Yuba, "A Time-to-Live based Reservation Algorithm on Fully Decentralized Resource Discovery in Grid Computing", *Parallel Computing 31*, 2005.

[15] Muthucumaru Maheswaran and Klaus Krauter, "A Parameter-based Approach to Resource Discovery in Grid Computing Systems", *GRID*, 2000.

[16] Domenico Talia, Paolo Trunfio, Jingdi Zeng and Mikael Hogqvist, "A DHT-based Peer-to-Peer Framework for Resource Discovery in Grids", Technical report, TR-0048, Institute on System Architecture, *CoreGRID - Network of Excellence*, June 2006.

[17] Thamarai Selvi Somasundaram, R.A.Balachandar, Vijayakumar Kandasamy, Rajkumar Buyya, Rajagopalan Raman, N.Mohanram and S.Varun, "Semantic based Grid Resource Discovery and its Integration with the Grid Service Broker", *ADCOM 2006: Proceedings of 14th International Conference on Advanced Computing & Communications (2006)*, pp. 84 – 89.

[18] Vidal, A. C., J. R. Braga, J., Kon, F., and Kofuji, S. T., "Defining and exploring a grid system ontology", *In Proceedings of the 4th international workshop on Middleware for grid computing* (New York, NY, USA, 2006), ACM Press, p. 16.

[19] Sujoy Basu, Sujata Banerjee, Puneet Sharma, Sung-Ju Lee, "NodeWiz: Peer-to-Peer Resource Discovery for Grids", *5th International Workshop on Global and Peer-to-Peer Computing (GP2PC) in conjunction with CCGrid*, May 2005.

[20] S. Castano, A. Ferrara, S. Montanelli, G. Racca, "Matching techniques for resource discovery in distributed systems using heterogeneous ontology descriptions", *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2004.

[21] K.G. Zefiridis, and H.D. Karatza, "Centralized and Decentralized Service Discovery on a Peer-To-Peer Network - A Simulation Study", *Proceedings of the Sixth United Kingdom Simulation Society Conference, UKSim 2003*, Cambridge, England, 9th-11th April 2003, pp. 171-177.

[22] Mauricio G. C. Resende, AT&T labs, http://www.research.att.com/~mgcr/, as per January 2008.

[23] Konstantinos I. Karaoglanoglou, and Helen D. Karatza, "Resource Discovery in a Grid system based on Matchmaking-Routers", *Proceedings of the 12th Panhellenic Conference on Informatics*, PCI, IEEE Computer Society, August 28-30, 2008, Samos, Greece, pp. 73-77.

[24] Konstantinos I. Karaoglanoglou, and Helen D. Karatza, "Resource Discovery in a Dynamical Grid system based on Re-routing Tables ", *Simulation Modelling Practice and Theory*, Elsevier, Volume 16, Issue 6, Pages 704-720, July 2008.

[25] Adriana Iamnitchi, and Ian Foster, "On Fully Decentralized Resource Discovery in Grid Environments", *International Workshop on Grid Computing 2001*, Denver, CO, November 2001.

[26] Junwei Cao, Stephen Jarvis, Subhash Saini, and Darren Kerbyson and Graham Nudd, "ARMS: An agent-based resource management system for Grid computing", *Scientific Programming, Special Issue on Grid Computing*, 10(2), 135-148, 2002.