Resource Discovery in a Grid System based on Matchmaking-Routers

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

Abstract

This paper studies the Resource Discovery problem in a Grid system based on Matchmaking-Routers. The proposed framework suggests that a Grid can be seen as an environment comprised by matchmaking-routers and resources. Each matchmaking-router is in charge of its local resources. The goal is to discover the appropriate resource for a specific request and then effectively direct the request to the resource within that environment. Matchmaking-routers are responsible of providing the set of resources that can satisfy a specific request and then directing the request to the resource that is capable of satisfying it.

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 [20, 22], high throughput computing, on-demand computing, data-intensive computing, collaborative computing or multimedia computing" [1].

The base of Grid technology is the concept of resource sharing. The types of resources shared in a Grid infrastructure could be desktop systems, clusters, storage devices and large data-sets. The question is what happens when a remote user requests access to a remote resource either to execute a job or to have access in the resource's data? A resource discovery mechanism provided by the Grid infrastructure should be available to discover an appropriate resource for a request [2].

We developed two approaches in the resource discovery framework: a Minimum Distance approach and a Best Fit approach. After a match between a request and a resource is accomplished, the first approach suggests directing the request to the nearest resource capable of satisfying it. The second approach suggests directing the request to the resource that best fits the request's requirements.

2. Related Work

The Matchmaking framework [3] was designed to solve real problems encountered in the deployment of Condor, a high throughput computing system. Several other research papers make use of the Matchmaking framework trying to add new aspects in the existing mechanism [4, 11, 14, and 18]. According to this framework, requestors and providers (resources) in a Grid system advertise their characteristics. A matchmaking service is responsible of finding a match between the advertisements and informing the relevant entities of the match.

Peer-to-Peer systems and Grids are both resource-sharing environments. Research papers, concerning that field, suggest the use of existing protocols developed for Peer-to-Peer systems into Grid systems [5, 6, 13, 16, 19, and 21].

Another notable approach to the Resource Discovery problem is the Semantic Communities one [7, 8, 10, 12, 15, and 17]. Main target in this approach is to create Grid communities based on similar-interests policies allowing community nodes to learn of each other without relying on a central meeting point.

The proposed framework makes use of the matchmaking process and semantic-based descriptions of resources and requests in order to provide sets of resources capable of satisfying a specific request. We deployed a Routing Tables mechanism [9] for directing a request to the resource that will satisfy it. To the best of our knowledge, a resource discovery mechanism combining matchmaking and routing concepts has never been deployed before.

IEEE
computer
society

3. The Matchmaking-Router Model

3.1. 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. Fig. 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. Fig. 1 also shows the way that the matchmaking-routers are connected (Router 1 connects directly to Routers 2 and 3).
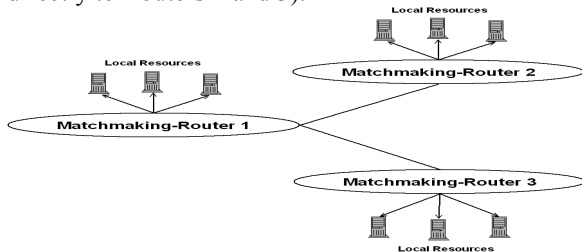

Fig. 1 The Matchmaking-Router model

3.2. Routing Tables

Each matchmaking-router in the Grid network maintains a Routing Table 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 (Fig. 2 presents such a Table assuming that the number of available resources in the system is twenty).

	Resource 1	Resource 2	...	...	Resource 20
Distance in hops from router to resource					

Fig. 2 An example of a Routing Table available in a matchmaking-router

It is obvious that a shortest distance algorithm plays a central role in creating the Routing Tables for each router in the system. We have deployed such an algorithm for our simulation needs. Due to space limitations, the algorithm is not presented in the paper.

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 in order to get satisfied. These required characteristics are: architecture, operating system, disk, and memory. For example a request could require a resource with the following characteristics: Intel architecture, Solaris26 operating system, minimum disk required 25000MB and minimum memory required 512 MB.

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. For example a resource description in the Grid system could be the following: SGI architecture, IRIX6 operating system, available disk 35000MB and available memory 1024MB.

4.2. Matchmaking Rules

The basic matchmaking rules that determine the set of candidate resources that can satisfy a specific request in the framework are the following:
1. The architecture and operating system characteristics of the request must match the architecture and operating system characteristics of the resource.
2. The minimum disk size required by the request must be smaller or equal to the available disk size of the resource.
3. 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 Fig. 3. The resources available to the supposed Grid system are four and their description types are shown in Fig. 4. The matchmaking-router has to provide the set of resources that are capable of satisfying the request complying to the basic matchmaking-rules mentioned above. Resource of type 1 cannot satisfy the request due to a mismatch in the operating system characteristic. Resource of type 2 also cannot satisfy the request due to a mismatch in the architecture characteristic. Finally, 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: Solaris26
Minimum Disk: 35000
Minimum Memory: 512
```

Fig. 3 An example of a request's requirements

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 4 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 3 fits best with the request's

requirements but its distance from the router that created the request could be unacceptable.

The two developed approaches for the proposed Grid resource discovery framework take into consideration the concept of the best suitability (best fit) and the factor of distance when there is more than one resource capable of satisfying a specific request.

5. Best Fit Approach

The Best Fit approach suggests directing the request to the most suitable resource. The most suitable resource is the one that fits best to the requirements of the request.

The Best Fit resource is identified as 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 Fig. 3, the Best Fit approach would suggest directing the request to the most suitable resource of type 3. The difference of the available disk size and memory space of resource type 3 from the minimum required disk size and memory space of the request is zero.

```
Resource Type 1
Architecture: Intel
Operating System: Linux
Available Disk: 20000
Available Memory: 128

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

Resource Type 3
Architecture: Intel
Operating System: Solaris26
Available Disk: 40000
Available Memory: 1024

Resource Type 4
Architecture: Intel
Operating System: Solaris26
Available Disk: 35000
Available Memory: 512
```

Fig. 4 Available Grid resources

6. Minimum Distance Approach

The Minimum Distance approach suggests directing the request to the nearest available resource. Given a set of resources, the matchmaking-router uses the information maintained to its Routing Table in order to determine the nearest resource. The resource chosen for satisfying the request does not have to be the Best Fit resource.

7. An example of Resource discovery

In Fig. 5, a Grid system comprised by five matchmaking-routers and four different types of resources is presented. At some point of time, a request is created in Router 1. This request requires for a resource with the following characteristics: Intel architecture, Linux operating system, 30000MB disk size, 1024MB memory space. Fig. 5 also presents the descriptions of characteristics of the four different types of resources available in the system.

The matchmaking-router (Router 1) based on the matchmaking rules is responsible of providing the set of resources capable of satisfying the request. From the four different types of resources available in the system, only two are appropriate for the request's requirements. These are: Resource of type 3 and Resource of type 4.


```
Request: Intel, Linux, 30000, 1024
Resource 1: SGI, IRIX6, 25000, 128
Resource 2: Intel, Linux, 25000, 512
Resource 3: Intel, Linux, 40000, 2048
Resource 4: Intel, Linux, 35000, 1024
```
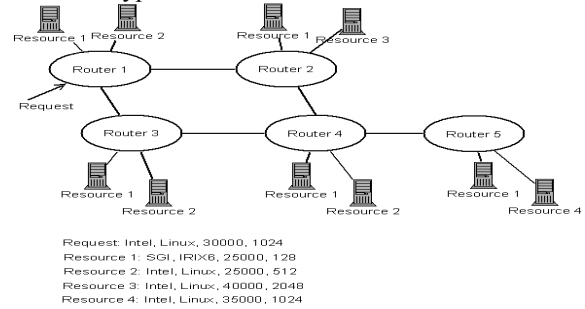
Fig. 5 An example of resource discovery

Assuming that the directing of the request happens with the Best Fit approach, the matchmaking-router 1 has to decide which one of the two resources is most suited for satisfying the request. In this case, the Best Fit resource is the resource of type 4. So, the request is forwarded to Router 5, where the resource of type 5 exists locally. Satisfaction of the request happened in a total distance of 4 hops.

Assuming that the directing of the request happens with the Minimum Distance approach, the process for the satisfaction of the request is the following. Matchmaking-router 1 has to decide which one of the two resources will satisfy the request. Based on the information maintained in its Routing Table (first row of Table 1), the matchmaking-router 1 decides to forward the request to the nearest resource. Distance from router 1 to resource of type 3 is 2 hops, which is smaller than the distance to resource of type 4 which is equal to 4 hops. So, resource of type 3 is selected for the satisfaction of the request.

Table 1 Routing Tables available in the matchmaking-routers

Router	Resource 1	Resource 2	Resource 3	Resource 4
1	1 hop	1 hop	2 hops	4 hops
2	1 hop	2 hops	1 hop	3 hops
3	1 hop	1 hop	3 hops	3 hops
4	1 hop	1 hop	2 hops	2 hops

8. Testing

8.1. Environment

Grid Graph generator [23] produced the backbone of the networks, meaning the matchmaking-routers. After this we allocated a certain number of resources to each matchmaking-router of the network. In all tests, we assume that we own twenty different types of resources. Each router of the network can control locally three to five resources. Note that a resource of a specific type could exist locally more than one time in the same router. All available resources are fully dedicated to the Grid system.

Tests were conducted in networks of the following sizes: 202 routers, 402 routers, 602 routers, 802 routers, and 1002 routers. Due to space limitations only few results and tests are presented here.

8.2. Results and conclusions

This section presents results for the approaches of Minimum Distance and of Best Fit in a Grid system of a 1002 matchmaking-routers. Results shown here are the averages of four different executions in four different topologies for a 1002 routers network.

Fig. 6 presents the evolution of simulation in terms of distance in hops for 1000 requests randomly created in random matchmaking-routers based on the Minimum Distance approach. It is obvious that resource discovery for the most cases of requests happened in small level of hops, around 1 to 3 hops. This fact was expected, because the Minimum Distance approach directs the requests to local resources or to resources that are near the matchmaking-router that created the request based on the distance information available in routing tables.

Fig. 7 presents the evolution of simulation in terms of distance in hops for 1000 requests based on the Best Fit approach. Resource discovery for the most cases of requests happened in small level of hops, around 2 to 4 hops. It appears that the numbers of hops for the Best Fit approach are small but they are slightly higher than the numbers of hops presented for the Minimum Distance approach. This can be explained by the fact that the Best Fit approach directs requests based on the suitability of the resources capable of satisfying them. Due to this, the most suitable resource does not have to be local nor near to the matchmaking-router that created the request.
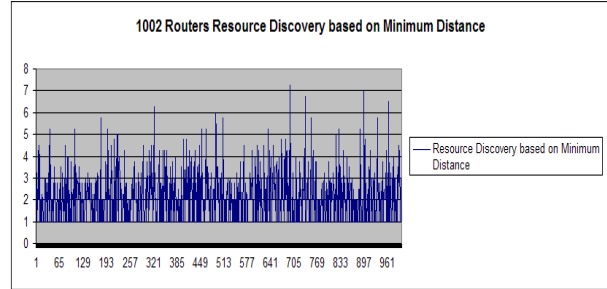

Fig. 6 Resource Discovery in a 1002 routers Grid system based on the Minimum Distance approach
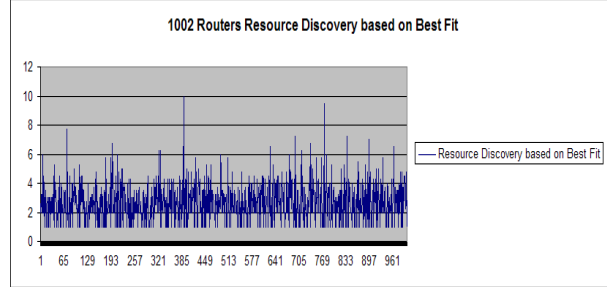

Fig. 7 Resource Discovery in a 1002 routers Grid system based on the Best Fit approach

In Fig. 8, results show the average distances for discovering all twenty types of resources during the simulation for both approaches. As expected discovery of the most types of resources with the Minimum Distance approach is extremely small, around 1 and 2 hops. Discovery with the Best Fit approach presents slightly larger numbers of hops than the Minimum Distance approach, but still acceptable, around 2 and 3 hops.
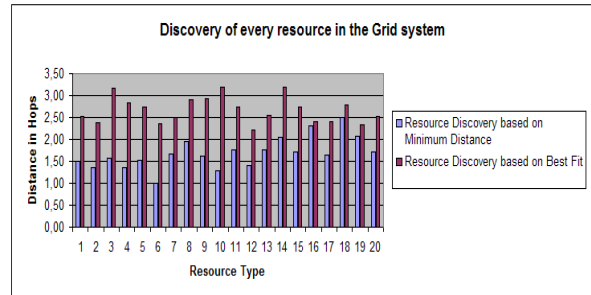

Fig. 8 Discovery of all types of resources in the Grid system

Both approaches, Minimum Distance and Best Fit, present extremely well results showing their effectiveness in discovering the appropriate resource. The Minimum Distance approach guarantees satisfying a request in the smallest number of hops. The Best Fit approach guarantees discovering the most suited for the request resource, without producing unacceptable results in terms of distance in hops.

9. 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] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing", Int. J. of Software Practice and Experience, 32(2):135-164, 2002.

[21] K.G. Zerfiridis, 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] H.D. Karatza, and R.C. Hilzer, "Performance Analysis of Parallel Job Scheduling in Distributed Systems", Proceedings of the 36th Annual Simulation Symposium, IEEE Computer Society Press, SCS, Orlando, Florida, March 30 - April 2, 2003, pp.109-116.

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