

Grid Resource Discovery Based on Web Services

Deniz Cokuslu^{1,2,3}, Abdelkader Hameurlain², Kayhan Erciyes³

¹*Izmir Institute of Technology, Department of Computer Engineering Gulbahce, Turkey*

²*IRIT, Paul Sabatier University, France*

³*International Computer Institute, Ege University Bornova, Turkey*
denizcokuslu@iyte.edu.tr, hameur@irit.fr, kayhan.erciyes@ege.edu.tr

Abstract

The size of grid systems has increased substantially in the last decades. Resource discovery in grid systems is a fundamental task which provides searching and locating necessary resources for a given process. Various different approaches are proposed in literature for this problem. Grid resource discovery using web services is an important approach which has resulted in many tools to become de facto standards of today's grid resource management. In this paper, we propose a survey of recent grid resource discovery studies based on web services. We provide synthesis, analysis and evaluation of these studies by classification. We also give a comparative study of different classes proposed.

1. Introduction

Grid system is a large scale dynamic environment which provides a high number of powerful resources to its users [12]. The resources that grid systems provide can be defined as computing resources (CPU cycles, storage, network bandwidth, memory etc.), services and metadata.

Web services in this area have become very popular software entities that offer an easy and standard way to access functionalities of different kinds of platforms [21]. In this sense, web services can be used as a bridge between complex grid middleware systems and their users. To realize this, many standards such as UDDI [1], WSDL [7] and SOAP [17] are developed and already in use. Since grid services can easily be mapped to web services, their convergence leads researchers to develop grid services as abstract sub classes of web services. The mapping is used in many grid problems such as resource discovery which is one of the most common problems in both domains. Resource discovery

problem in grid systems can be defined as searching and locating resource candidates which are suitable for a job in a reasonable time, in spite of the dynamicity and large scale of the environment. In this perspective, grid resource discovery can be treated as a special case of discovery of web services. Although some studies use web services in this manner, some others use them for building easier user interfaces.

Today many grid management tools which are based on web services are already in use along the ongoing research. GlobusToolkit [10], Alchemi [15], Unicore [20] and gLite [3] are the most known examples of such systems. Especially GlobusToolkit (GT) is broadly accepted to be the current de facto standard in grids. Globus provides many grid services such as resource discovery, allocation and monitoring by extending standard service interfaces [11].

Many web service based grid studies benefit from the services of GT. In this paper, we provide a survey of recent grid resource discovery methods which are based on web services [23, 14, 18, 9, 11, 19]. We classify the methods according to the topology of the resource information. In centralized methods, the resource information is stored in a central server, whereas in hierarchical methods the resource information is divided and partially distributed on several locations.

The main contribution of this paper is the detailed analysis of recent grid resource discovery methods based on web services within a specified classification, and comparison between the classes. We give detailed analysis for each study. We also evaluate them according to important qualitative criteria namely, scalability, dynamicity, reliability and support for multi-attribute, dynamic-attribute and range queries. We think that those criteria are very important in analyzing grid algorithms. More

precisely, scalability is a very important measure since grids are large scale environments in their nature. Dynamicity is another important measure because of the nodes in grid systems might be highly dynamic in terms of joining and leaving the system, mostly without any notice. Reliability should also be considered because in some cases erroneous query results may cause irrecoverable faults. And lastly, support for different types of queries is a decisive criterion in selecting the methodology in most cases. The main contribution of this paper is the detailed analysis of recent grid resource discovery methods based on web services within a specified classification, and comparison between the classes. The rest of the paper is organized as follows: Section 2 gives analysis and evaluation of grid resource discovery methods based on web services which rely on centralized topology. Section 3 analyzes and evaluates grid resource discovery methods based on web services which rely on hierarchical topology. Section 4 compares two classes of RD methods based on web services. Finally, in Section 5, we conclude the paper by giving an overall summary and remarks about the examined methods and express our future plans.

2. Grid Resource Discovery Using Centralized Web Services

Grid resource discovery using centralized web services is excessively studied and implemented in grid environments. Most of those studies profit from existing information provider systems such as LDAP or UDDI. In this section, we examine some recent studies in this class, and evaluate them according to the selected evaluation criterions.

2.1. Synthesis and Analysis

In [23], Yu et al. developed a web services based grid service publication directory system (GMD). The system provides service for resources and clients via web by using XML formatted messages. The GMD consists of two main components. The first component, GMD Portal Manager, provides a web based platform to manage registration and publication of resources. A user which would like to share its resources uses the portal in order to advertise its resources to the central repository. The updates of resource information are accomplished by the resources' owners. Therefore, to compensate the dynamicity of grid, users need to update resource information either manually or with the help of a third party software. The second component, GMD Query Web Service is a query processing web service which is used to discover required resources that meet the query requirements. GMD is a centralized grid resource management system. It uses a central resource repository database and central

query processing engine. Because of its centralized nature, the system may suffer from bottleneck and single point of failure problems in a large scale grid system. Moreover, it is not suitable for dynamic grid environments since the dynamicity requires manual updates. On the other hand, the system supports multi-attribute and range queries since the repository is a database system in which those types of queries can easily be supported. But it does not support dynamic-attribute queries. Kaur and Sengupta presented a web-services based resource discovery mechanism for grids [14]. Their study consists of four main components. The UDDI rich query model uses UDDI standard [2] to discover grid services. The standard is used to hold grid resource information as key-value pairs in the UDDI database. The second component GWSDL is used to describe grid services in an extended version of Web Services Description Language. GWSDL has OGSi support and plays an important role on standardizing the communication between OGSi and web services. The third component is SOAP which accomplishes communication between web services in the grid environment. The last component is HTTP which provides easy to use interface to post requests and get responses. The web service processes the requests by querying UDDI database. The system uses central servers and databases to run the web service. Therefore, it contains bottlenecks and single point of failures. Moreover, since the resource information is held in a repository database, the dynamic attribute queries are not supported. On the other hand; the proposed web service supports range and multi-attribute queries. In [18], Moltó et al. proposed a metascheduler grid service that can be accessed through the network by users interested in task allocation and scheduling in computational grids. Metascheduling is the process which takes a job demand, allocates suitable resources, runs the job and returns the result to the user. The proposed system is a service oriented architecture which can be used via the web without any platform requirements. It is developed on top of Globus Toolkit which uses Metacomputing Directory Service (MDS) for the resource discovery. The proposed system in this study fills the gap between a simple user interface and complex Globus resource management primitives. In terms of resource allocation, the proposed system additionally implements a transparent caching component which stores static resource information, which is assumed to be unchanged, for the future queries. The resource discovery phase is held by first interrogating a GIIS (Grid Index Information Service) or a BDII (Berkeley Database Information Index) service. The proposed system uses standard protocols like SOAP and XML for data exchange and the whole process is held by a central server. Therefore, the central server may easily become a bottleneck and single point of

failure in highly dynamic and large scale grid environments. Moreover, since dynamic attributes are updated on the database in discrete intervals, dynamic-attribute queries are not supported. On the other hand the system supports multi-attribute and range queries.

2.2. Evaluation

The grid resource discovery using centralized web services provide users an easy to use interface to manage grid resources. They keep grid resource information by using centralized databases. In a large scale grid environment, the centralization of the service may easily create bottlenecks on the central servers. The bottleneck problem may arise both because of frequent resource updates or large number of query requests waiting to be processed. The centralization causes another important problem in dynamic grids as being a single point of failure. Failure of one of the central servers in the system may cause the whole system to become unavailable. In some approaches, the idea of replication of central servers is depicted in order to eliminate single point of failures. But replication of servers in a large scale dynamic grid may be very expensive in terms of communication costs. The proposed systems support the multi-attribute and range queries since the resource information is stored in databases which are capable of processing complex queries. But since the update of dynamic resource attributes are held in discrete intervals, most of these systems do not support dynamic-attribute queries.

3. Grid Resource Discovery Using Hierarchical Web Services

In recent years, as the size of the grid environments grows, researchers directed their attention to hierarchical systems to overcome the problems caused by centralized systems in resource discovery. In this section, we analyze and evaluate some recent grid resource discovery methods which use hierarchical web services.

3.1. Synthesis and Analysis

In [9], Elmroth and Tordsson proposed a grid resource broker and job submission system based on web services. Their algorithm is a complete grid management middleware. In their design, they take advantages of many existing tools by utilizing and extending them. For the resource discovery phase, they benefit from Grid Laboratory Uniform Environment (GLUE) project which provides information related to the resources. The server side module which is called job submission module is composed of seven components. The first component, in which we are interested in, is the

InformationFinder component. The InformationFinder component discovers the grid resources and provides information about them. The InformationFinder component works in cooperation with index servers which are responsible for indexing resource information. The index servers are organized in a hierarchical topology. Once a query is received, the InformationFinder performs the resource discovery by querying each index server. Then, the resources which are found to be suitable for the query are contacted to get more detailed up-to-date resource information. Both static and dynamic information about the resources are retrieved in this step. The InformationFinder also retrieves usage policies, allowing it to discard resources where the user is not authorized to submit jobs. Then the extracted resource information is passed to the higher level components of the system to accomplish job submission processes. The system stores the resource information in information servers which are organized in a hierarchical structure. The hierarchical organization reduces the probability of bottleneck problem. But it cannot solve single point of failure since failure of a server in the hierarchical organization may result in a large part of the resources to be excluded from the queries. On the other hand, since the resource information is held without any hash function and since it is verified by contacting the candidates directly, the system supports all multi-attribute, range and dynamic-attribute queries. In [19], Ramos et al. proposed a web service for resource discovery in grids, based on Globus Toolkit (GT3). They proposed a hierarchical topology in which the grid environment is divided into virtual organizations (VO). In each VO, there are master and slave nodes. The master nodes are responsible for updating the resource database and the slave nodes are responsible from retrieving resource information from each machine which composes the grid system. The resource discovery is realized by the preparation of a configuration file which includes the requested resource information. Upon detecting the file, the resource discovery mechanism generates an XML file which is going to be manipulated by resource discovery service. After this new configuration is distributed to all slave nodes, the service is restarted in all nodes. By doing this, a new service containing the new search is deployed. When a slave machine receives such a query, it checks if the requested resources are satisfied and returns results to its master node. Then the resource discovery service on the master machine processes the retrieved files and returns the results to the requester. Since the proposed system is a hierarchical web service based system, it decreases the possibility of bottleneck problem. On the other hand; since the resource discovery task is held by master machines, failure of a master machine may result in a large number of

slave machines becoming invisible to the system. This system supports all multi-attribute, range and dynamic-attribute queries since the resource discovery is held by resources on-the-fly without any hashing function.

3.2. Evaluation

The hierarchical web services based grid resource discovery algorithms provide a more scalable platform than the centralized ones and still provide a simple user interface to manage grid resources. In a large scale grid environment, the hierarchical topology of the service decreases the probability of bottleneck problem. But single point of failure problem still exists since failure of one of the master servers in the system may cause a large part of the nodes become invisible to the queries. All proposed algorithms in this class support the multi-attribute and range queries since the resource information is stored in databases which are capable of processing complex queries. And since after querying the databases, the information is verified in the resource nodes, the proposed algorithms also support dynamic-attribute queries.

4. Comparison

The summary of comparison between resource discovery using centralized web services and resource discovery using hierarchical web services can be seen in Figure 1. Both centralized and hierarchical methods behave closely against nearly all the evaluation criterions. The main difference is in scalability and reliability. The centralized systems suffer from the bottleneck problems in large scale. There also exists the single point of failure problem. Even though some studies propose to replicate the centralized index server, this procedure might be very expensive in terms of messaging complexity in large scale. On the other hand, hierarchical systems distribute the load into many locations instead of one central server. This property increases the scalability of the system by distributing load on index servers. They also decrease the effect of single point of failures. In case of a failure of an index server, a part of the system becomes unreachable instead of the whole. Support for dynamic attribute queries requires that the query is processed within the resource nodes. Since the idea in the examined solutions is indexing the resource information in central locations, they do not support dynamic attribute queries. Some of the examined algorithms propose solutions for this problem, but the solutions are independent from the classification that we propose. By taking the evaluations into consideration, we can say that the centralized methods are not suitable for the large scale environments. But they might be well suited to the systems in which the scale is small and indexing

server is reliable. In such cases centralized systems can be used effectively. On the other hand, hierarchical methods are more suitable for environments in which scale is bigger since the load is distributed to many locations. But even the load is hierarchically distributed; those methods may still suffer from bottleneck problem in large scale.

	RD Using Centralized Web Services	RD Using Hierarchical Web Services
Scalability	Not scalable due to bottleneck problem	Better scalable because of the hierarchical distribution of load
Dynamicity	Tolerant to node dynamicity, but not tolerant to indexing mechanism's dynamicity	Tolerant to node dynamicity, better tolerant to indexing mechanism's dynamicity
Reliability	Reliable in terms of query correctness, but not reliable in terms of single point of failure	Reliable in terms of query correctness, better reliable in terms of single point of failure
Range Queries	Supports since queries are resolved within the resource nodes without any hashing	Supports since queries are resolved within the super-peers without any hashing
Multi-attribute Queries	Supports since queries are resolved within the resource nodes without any hashing	Supports since queries are resolved within the super-peers without any hashing
Dynamic-attribute Queries	Does not support since queries are resolved by central servers	Does not support since queries are resolved by central servers

Figure 1. Comparison summary between two resource discovery methods using web services

5. Conclusion and Future Works

In this paper, we have synthesized and analyzed some recent grid resource discovery methods which are based on web services. We evaluated them by defining some qualitative criterions, and compared different classes of methods with each other. The resource discovery algorithms based on web services seem to have many disadvantages in large scale dynamic grid environments since most of them rely on centralized resource discovery approaches. But their simple and standard user interface brings them to the foreground and makes them suitable platforms to design and implement grid management tools.

Most web service based methods use grid resource discovery techniques which are not suitable for large scale. Therefore, with regards to the examined methods, we can say that the web service based approaches are suitable for small scale grid environments in which the dynamicity of nodes is low. In such environments, they provide the users a very simple, standard and platform independent grid management middleware without requiring a deep knowledge of grid management primitives. There is vast amount of ongoing research in this topic which is aimed at solving problems mostly caused by scalability issues. P2P based [4, 6, 16] and Agent based [5, 8, 13] approaches in grid resource discovery are two of the most promising approaches in this field. A very comprehensive and detailed survey about those grid resource discovery approaches can be found in [22, 10].

We believe that combining the advantages of the different RD approaches would result in useful studies for resource discovery in grids. For example by using web services, we can provide a simple interface to the users, while for the RD module, by using mobile agents or P2P techniques, we can design a scalable, reliable and also easy to use resource discovery middleware. To the best of our knowledge, there is no research work which is directly focused on this subject. For this purpose, we are planning to propose a new cross-layer design approach in resource discovery which combines topology control, efficient multicasting and web services together. Firstly, we are planning to build virtual clusters in the grid environment which are represented by their clusterheads. Then we are planning to build a spanning tree between clusterheads in order to ease routing of the messages between peers. After building the spanning tree, we are planning to deploy an efficient routing algorithm on top of this spanning tree. Finally we are planning to join web services in order to enable integration of our study to the current grid middleware systems.

6. References

- [1] T. Belwood. Uddi version 2.04 api specification. Available at <http://www.uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm>, April 2009.
- [2] E. Benson, G. Wasson, and M. Humphrey. Evaluation of uddi as a provider of resource discovery services for osbased grids. page 9, April 2006.
- [3] S. Burke, S. Campana, E. Lanciotti, P. M. Lorenzo, V. Miccio, C. Nater, R. Santinelli, and A. Sciab'a. Glite 3.1 user guide, manual series, document identifier: Cern-lcg-deis-722398, 2004.
- [4] M. Cai, M. Frank, J. Chen, and P. Szekely. Maan: A multiattribute addressable network for grid information services. In *4th Int. Workshop on Grid Computing, GRID 2003*, page 184191, 2003.
- [5] J. Cao, D. Spooner, J. D. Turner, S. Jarvis, D. J. Kerbyson, S. Saini, and G. Nudd. Agent-based resource management for grid computing. In *CCGRID '02: Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, page 350. IEEE Computer Society, 2002.
- [6] A. S. Cheema, M. Muhammad, and I. Gupta. Peer-to-peer discovery of computational resources for grid applications. In *GRID '05: Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pages 179–185. IEEE Computer Society, 2005.
- [7] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (wsdl) 1.1. Available at: <http://www.w3.org/TR/wsdl>, April 2009.
- [8] S. Ding, J. Yuan, J. Ju, and L. Hu. A heuristic algorithm for agent-based grid resource discovery. In *Intl. Conf. on e-Technology, e-Commerce and e-Service*, pages 222–225, 2005.
- [9] E. Elmroth and J. Tordsson. An interoperable, standardsbased grid resource broker and job submission service. pages 212–220, 2005.
- [10] I. Foster, N. R. Jennings, and C. Kesselman. Brain meets brawn: Why grid and agents need each other. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 8–15. IEEE Computer Society, 2004.
- [11] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11:115–128, 1997.
- [12] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 2004.
- [13] G. Kakarontzas and I. K. Savvas. Agent-based resource discovery and selection for dynamic grids. In *Proc of the 15th IEEE Intl. Workshops on Enabling Technologies*, pages 195–200, 2006.
- [14] D. Kaur and J. Sengupta. Resource discovery in webservices based grids. In *Proceedings Of World Academy Of Science, Engineering And Technology*, pages 284–288, 2007.
- [15] A. Luther, R. Buyya, R. Ranjan, and S. Venugopal. Alchemi: A .net-based enterprise grid computing system. In *6th International Conference on Internet Computing, Las Vegas*, 2005.
- [16] M. Marzolla, M. Mordacchini, and S. Orlando. Peer-to-peer systems for discovering resources in a dynamic grid. *Parallel Comput.*, 33(4-5):339–358, 2007.
- [17] N. Mitra and Y. Lafon. Soap version 1.2 part 0: Primer (second edition). Available at: <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>, April 2009.

- [18] G. Molt'o, V. Hern'andez, and J. M. Alonso. A serviceoriented wsrf-based architecture for metascheduling on computational grids. *Future Generation Computing Systems*, 24(4):317–328, 2008.
- [19] T. G. Ramos and A. C. Magalhaes. An extensible resource discovery mechanism for grid computing environments. In *CCGRID '06: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, pages 115–122. IEEE Computer Society, 2006.
- [20] M. Riedel, B. Schuller, D. Mallmann, R. Menday, A. Streit, B. Tweddell, M. S. Memon, A. S. Memon, B. Demuth, and T. Lippert. Web services interfaces and open standards integration into the european unicore 6 grid middleware. In *EDOCW '07: Proceedings of the 2007 Eleventh International IEEE EDOC Conference Workshop*, pages 57–60. IEEE Computer Society, 2007.
- [21] I. Toma, K. Iqbal, D. Roman, T. Strang, D. Fensel, B. Sapkota, M. Moran, and J. M. Gomez. Discovery in grid and web services environments: A survey and evaluation. *Multiagent Grid Syst.*, 3(3):341–352, 2007.
- [22] P. Trunfio, D. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi. Peer-to-peer resource discovery in grids: Models and systems. *Future Gener. Comput. Syst.*, 23(7):864–878, 2007.
- [23] J. Yu, S. Venugopal, and R. Buyya. Grid market directory: A web services based grid service publication directory. Technical report, Grid Computing and Distributed Systems (GRIDS) Lab, Dept. of Computer Science and Software Engineering, The University of Melbourne, 2003.