International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

Resource Discovery in Web-services based Grids

Damandeep Kaur, and Jyotsna Sengupta

***Abstract*—**A Web-services based grid infrastructure is evolving to be readily available in the near future. In this approach, the Web services are inherited (encapsulated or functioned) into the same existing Grid services class. In practice there is not much difference between the existing Web and grid infrastructure. Grid services emerged as stateful web services. In this paper, we present the key components of web-services based grid and also how the resource discovery is performed on web-services based grid considering resource discovery, as a critical service, to be provided by any type of grid.

***Keywords*—**Web Services, resource discovery, Grid Computing, OGSA.

I. Introduction

GRID is a system that coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service. Web-services based Grid [1], can be build, containing the features & functionalities of the Web services, inherited or extended from the Web services class. In this approach, the Web services are inherited (encapsulated or functioned) into the same existing Grid services class. The Grid services can be used to extend the basic Web services by defining a two-layer naming scheme that will enable support for the conventional distributed system transparencies, by requiring a minimum set of functions and data elements that support discovery, and by introducing explicit service creation and lifetime management.

A. Resource Discovery

The grid resource discovery problem [2] can be defined as the problem of matching a query for resources, described in terms of required characteristics, to a set of resources that meet the expressed requirements. The problem is complicated by the fact that some resource information (e.g., CPU load or available storage) changes dynamically. Resource discovery techniques maintain the resource attribute and status information in a distributed database and differ in the way they update, organize, or maintain the distributed database. The challenge is to devise highly distributed discovery techniques that are fault tolerant and highly scalable. Matching the needs of an application with available resources is one of the basic and key aspects of a Grid system. Resource Discovery is systematic process of determining which grid resource is the best candidate to complete a job with following trade-offs.

- In shortest amount of time.
- With most efficient use of resources.
- At minimum cost

Damandeep Kaur is with Computer Science and Engineering Department, Thapar University, Patiala, India (e-mail: damandeep@tiet.ac.in).

Jyotsna Sengupta, PhD, is with Department of Computer Science, Punjabi University , Patiala,India(e-mail: jyotsna.sengupta@yahoo.com).

II. Grid Services and Web Services

'Service-Oriented Architecture' (SOA)[3] is an architectural style whose goal is looser coupling among interacting software systems compared to earlier forms of distributing computing architecture such as RPC [7], CORBA [8], Java RMI [9] and DCOM [10]. It allows great flexibility in the interconnecting protocol and the underlying platform for clients and servers, all of which are advantageous in constructing Grid systems.

A service is defined in terms of the messages one uses to interact with it and the behaviour expected in response. This behaviour may depend on the state of resources such as files; databases or sensors that are encompassed by the service and the state may change in response to messages from one or more clients, or on internally generated events such as timers or external physical events. The messages, state and other behaviour must be described by a service definition.

Web Services is a particular type of Service-Oriented Architecture whose interfaces are defined using the Web Services Description Language (WSDL)[12] and which focuses on simple, Internet-based standards (such as eXtensible Markup Language (XML) and the http protocol) to enable heterogeneous distributed computing.

OGSI [4] describes Grid Services using Web Services as a basis. This means exploiting:

• The mechanisms for encoding message transmission protocols that are described as bindings in Web Services. There are many ways of encoding messages, and Web Services separates these concerns from the XML definitions of application interfaces: OGSI standardizes interfaces at the application level, independently of the bindings.

• The conventions used in Web Services to separate the primary application interfaces (function calls and their parameters) from functions which can be managed by standardized, generic middleware. This includes issues such as authentication and access control.

• The techniques used in Web Services to separate service-

and network-management functions from the application interface. The management issues include workload balancing, performance monitoring, and problem diagnosis.

A. What is needed to provide and use Grid Services?

In this section we introduce the requirements for the implementations of clients and servers participating in a Grid system [4]. A client may, of course, also be a Grid Service in its own right, but the two aspects can be separately addressed.

i). The Client Side

There are two meanings to the term 'client'. The end-users of a Grid system are its ultimate clients, but they may be shielded from the internal workings and requirements of the underlying systems by intermediaries such as Web portals, whose main objective is ease of use for the end-user.

Secondly, there are the direct clients of Grid Services, which may be part of such a Web portal. OGSI specifies many things that are required of services in order for them to be used by clients in a standard way.

Since OGSI is based on Web Services standards, a client which is capable of interacting with a Web Service can also interact with a Grid Service provided the interface description has been translated from GWSDL into WSDL.

ii) The Server side

The provider of a Grid Service can take many forms, and the OGSI Specification is designed to impose the least possible restriction on its implementation, which can range from dedicated hardware components through operating system services and custom-built applications.

However, since the OGSI standard is built on Web Services, a platform for hosting Grid Services may be built on existing systems for hosting Web Services, such as those based on the J2EE [14] component model. Such a system can provide many of the basics of a Grid Service via standard layers and services known as the 'hosting environment', and exploitation of such an environment and associated development tools is an efficient way of developing Grid applications.

The details of the interfaces between a Grid Service implementation and a hosting environment depend on the particular system, but some aspects of such systems can be described in generic terms.

Firstly, the application-specific operations and state of the service can be described using an interface description language or tools specific to that system. Clients can translate this interface description into a form described by Web Services and OGSI for consumption.

Secondly, the runtime system creates the interface between the service implementation and the hosting environment. This may include management of the state of the service, converting messages to and from the implementation language, and provision of standard operations for Grid Services.

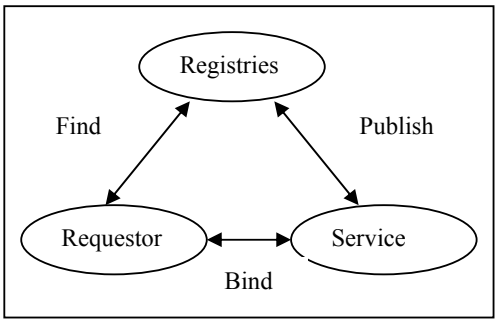

Fig. 1 The basis of Web Services computing [4]

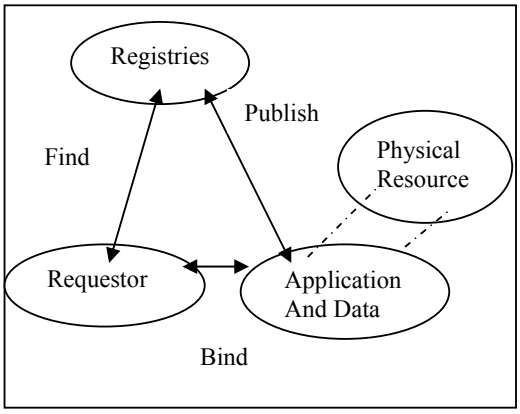

Fig. 2 Web Services facilities used in a Grid [4]

III. Resource Discovery in Web-Service Based Grid

A. Web-Service based Grids

Web-services based Grid extend and enhance the capabilities of Web services by providing common and powerful mechanisms that service consumers can rely upon across all services, instead of the service-specific, often ad-hoc approaches typically employed in standard Web services.

TABLE I
VARIOUS FEATURES OF WSGA

Feature	**Web-Services based Grid**
Data Model	SOAP message exchange model
Client-Server Coupling	Loose Coupling
Type System	XML
Service Discovery	UDDI + rich Query Model
Location Transparency	URL
Parameter passing	By value only
Description Language	GSWDL
Connection Protocol	HTTP

UDDI+ rich Query Model
GWSDL
SOAP
HTTP
Grid Middleware(e..g Globus)
Grid resource

Fig. 3 Web-services based grid Architecture

B. Key Components of Web-Service Based Grid

i). UDDI + rich Query Model:

At present UDDI Universal Description Discovery Integration) is an open standard for publishing and discovering the software components of service-oriented architecture. It allows for a standard interoperable platform that enables companies and applications to quickly, easily and dynamically (at run-time) find and use Web services over the Internet. In the Grid context it can be used to find and use grid services over the grid.

UDDI[5] provides its functionality through four principle entities: the *businessEntity*, *businessService*, *tModel*, and the *bindingTemplate*. The businessEntity is the largest container within UDDI. It contains information about any organizational unit which publishes services for consumption. The designers of UDDI envisioned businessEntities as being UDDI representations of actual businesses that choose to offer services over the Internet [capitalize?]. In the Grid context, businessEntities can be used to hierarchically separate and form relationships between different organizational groups within a Grid or multiple Grids.

A businessService object describes each service that a businessEntity offers. These objects provide information about the service name, categorization, and any other useful details added in a variety of attribute lists. The information is purely descriptive and does not include any instructions for accessing or using the service.

The bindingTemplate object represents the link between abstract businessService descriptions and actual endpoints at which these services may be accessed. Like all objects in UDDI, bindingTemplates are given universally unique identifiers, known as UUIDs, which are used as a key of reference. Each businessService object stores the UUID keys of bindingTemplates within the same businessEntity that provide instances of that service.

BindingTemplates may provide specialized information about a particular businessService endpoint through use of tModels. The tModel is the standard way to describe specific details about a particular businessService or bindingTemplate in UDDI. TModels contain lists of key-value pairs used for description and may be associated with multiple objects in the UDDI database. They may also contain placeholders for URIs, which point to descriptive information external to the UDDI registry.

UDDI designed as a business directory system and has some limitations that complicate resource discovery in Grid computing, for example difficulties in handling dynamic information (such as CPU load) that requires frequent updating. These limitations can be overcome by having a rich Query model, which can be based on either Agent-based resource discovery or Query-based resource discovery

ii). GWSDL:

OGSI adds the features of Grid Services to basic Web Services by redefining the WSDL portType element. Grid Services are described using an extended form of WSDL known as GWSDL. The main structure of a GWSDL document [4], which is very similar to the WSDL structure

• In OGSI, a portType can be constructed by referencing an existing portType or portTypes, with extends attribute, adding new definitions as required. This enables standard behaviour to be defined authoritatively, and to be referenced by service definitions that need to incorporate it. Standard behaviour is important to clients that may need to use independently implemented services.

• While a WSDL portType contains only operations, an OGSI portType can also contain Service Data Declarations (SDDs), which help describe the identity and interfaces of the service, and how a client can view its state.

iii). SOAP:

The standard Simple Object Access Protocol (SOAP) was developed to enable the transmission and reception of messages [1] for accessing distributed resources or objects. The Grid specific Resource Discovery (RD) service, which is one of the most important operations/services on Grids, is used not only to access but also to track, match, select and request the geographically distributed resources or objects. This Grid RD service is a set of many sub-services or instances, and one of its instances can be glued to the SOAP Web service for the simplification of the larger resource discovery problem.

iv). HTTP

Accessing a grid service is much like accessing anything else on the Internet in that it requires a URL When we surf the Web , we provide a browser application with a URL, which sends an HTTP request to the grid service provider associated with that URL, and retrieves an HTTP response containing the HTML to display. Using SOAP over HTTP is ideal for grid services.

The following illustrates how HTTP works:

1.Most Web pages contain *hyperlinks,* which are specially formatted words or phrases that enable you to access another page on the Web. Although the hyperlink usually doesn't make the address of this page visible, it contains all the information needed for your computer to request a Web page from another computer.

When you click the hyperlink, your computer sends a message called an HTTP request. This message says, in effect, "Please send me the Web page that I want."

The Web server receives the request, and looks within its stored files for the Web page you requested. When it finds the Web page, it sends it to your computer, and your Web browser displays it. If the page isn't found, you see an error message, which probably includes the HTTP code for this error: 404, "Not Found."

C. Resource Discovery as an Example of Grid Service

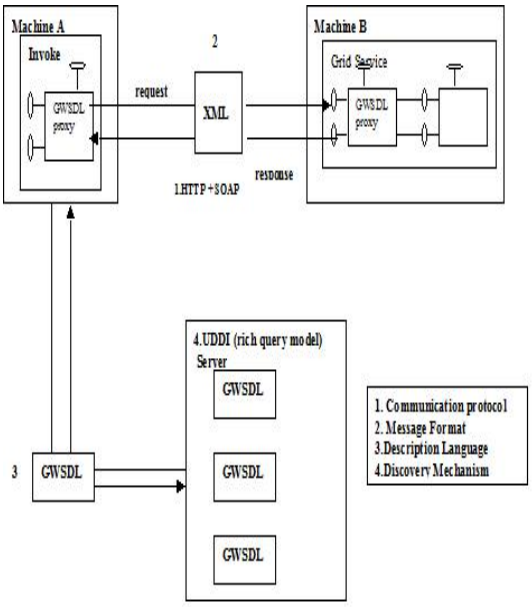

Fig. 4 Resource discovery as Grid service invocation in Web-services grid

The following steps depict how resource discovery as grid service invokes:

1. The first step will be to *discover* a Grid Service (resource discovery) that meets our requirements.
2. The discovery service will reply, telling us which grid resources can provide us the service we require.
3. Ask the Web Service to *describe* itself (i.e. tell us how exactly we should invoke it).
4. The Web Service replies in a language called GWSDL.
5. Finally know where the Grid Service is located and how to invoke it. The invocation itself is done in a language called SOAP. Therefore, we will first send a *SOAP request* asking for the specific resource requirement.
6. The Grid Service Provider will kindly reply with a *SOAP response* which includes desired resources we asked for, or maybe an error message if our SOAP request was incorrect.

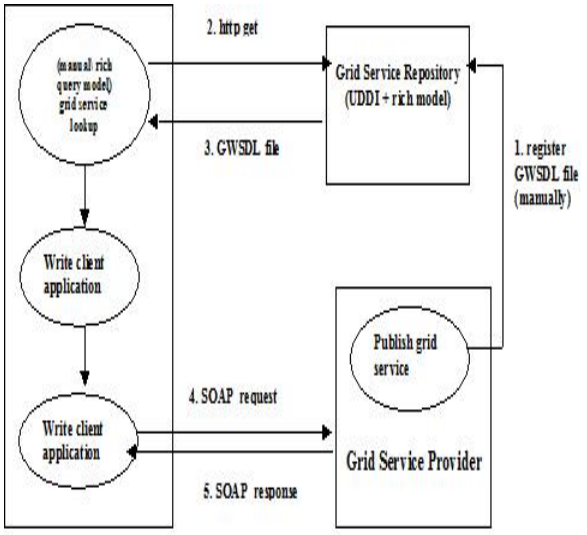

Fig. 5 Steps for Resource discovery in Web-service based Grid (* resource discovery is referred to as grid service)

Web-based Grid approach in a purely Grid environment may be more effective because in this way we have the entire Web service class packaged and integrated into the Grid service class and each Grid service object, thus instantiated, would have all the capabilities of a Web service and some additional specialized capabilities of its own.

IV. Challenges to Resource Discovery in Web-Services based Grid

Grid services provide a means for discovering other Grid resources. The Grid community has not yet been able to fully resolve the issue of Resource Discovery due to various technical constraints and geographical limitations such as autonomous, heterogeneous resources, dynamic nature and status of resources, geographical dispersion of resources, large number of users and large distributed networks, different operating systems/platforms, different administrative domains, lack of portability, availability status of resources and different technology policies. It is not easy to track or locate the right resource in a vast interconnected environment. The mechanism of resource discovery can be viewed through different lenses in various domains; it is a multidisciplinary task and is one of the most important issues to be dealt with in the future Grid technology. Basically there are three challenges [3] in integrating web services with Grid for resource discovery:

1) Lack of explicit data typing: The ability to associate data types with resource metadata is fundamental for resource discovery in Grid environments. Data typing allows not only more strict matching of resource information with resource requirements, but

2) Dynamic Service Data: Web services use static data. They have no built-in notion of dynamic service information or any other mechanisms in which the

context of stored data changes over time. The implications of this inability to the Grid community are quite large. This makes it difficult for web service to make available such important time varying information as CPU load.

3) Semantic Search Model: For web service to be successful as a resource discovery service in Grid environments, it must be able to respond to requests such as "find a resource that can perform task X on a machine with properties W, Y, and Z with guarantees A and B." Thus semantic search model should be preferred over syntax-based search.

V. Conclusion & Future Work

The designing and deployment of Web-Services based Grid infrastructure is the subject of a relatively new area, which considers Web services framework as its basis. Many of the problems associated with the implementation of Grids could be resolved if the Web is used as the underlying ("backbone") infrastructure for the development of Grids, thus resulting in the formation of a Web-services based grid. In this paper, we discussed working of grid services and also discussed Web-services based architecture in detail. Resource discovery is an important service for any type grid. The paper presented how resource discovery as a grid service is invoked in Web-services based Grids. Since Web-services based grid consider web-services as their basis, so using web-services framework resource discovery can be achieved by adding a rich model in UDDI. However above type of grid infrastructure is evolving so there are some challenges in considering resource discovery that will be solved in the future works.

References

[1] Aisha Naseer and Lampros K. Stergioulas , "Integrating Grid and Web Services: A Critical Assessment of Methods and Implications to Resource Discovery".
[2] Iamnitchi and I. Foster, "On Fully Decentralized Resource Discovery in Grid Environments," IEEE International Workshop on Grid Computing,Denver, CO, 2001.
[3] Srinivasan L., Treadwell J. An Overview of Service – Oriented Architecture, Web service and Grid Computing, HP Software Global Business Unit, 2005.
[4] Open Grid Infrastructure Primer, GFD-I .031, August, 2006.
[5] Edward Benson, Glenn Wasson, Marty Humphrey, Evaluation of UDDI as a Provider of Resource Discovery Services for OGSA-based Grids, IEEE Conference, 2006.
[6] Ian Foster and Carl Kesselman (editors), The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, USA, 1999.
[7] RFC 1831. RPC: Remote Procedure Call Protocol Specification Version 2.Available at http://www.ietf.org/rfc/rfc1831.txt
[8] The Object Management Group CORBA specification. Available from the OGM Web Site: http://www.omg.org/corba/
[9] Java RMI specification. Available at: http://java.sun.com/j2se/1.4.2/docs/guide/rmi/spec/rmiTOC.html
[10] *DCOM Architecture*. Markus Horstmann and Mary Kirtland. Available from http://msdn.microsoft.com
[11] Grimshaw, A., and Tuecke, S. Grid services extend Web services. Web Services Journal, 3(8):22-26, 2003.
[12] Hypertext Transfer Protocol. RFC 2616. Available at http:// www.w3.org/Protocols/rfc2616/rfc2616.html
[13] Web Services Description Language (WSDL) 1.1 W3C Note 15 March 2001Available at http://www.w3.org/TR/wsdl
[14] J2EE Platform Specification. Available at http://java.sun.com/j2ee/1.4/download.html
[15] Twardoch, A. Web Services Technologies: XML, SOAP and UDDI. .Net and Web Services Seminar, European University Viadrina Frankfurt (Oder), Germany, http://www.twardoch.com/webservices, 2003.
[16] Talia, D. The Open Grid Services Architecture: Where the Grid Meets the Web. IEEE Internet Computing, 6(6):67-71, Nov-Dec 2002.
[17] Foster I., Kesselman, C., Nick, J. M., and Tuecke, S. Grid Services for Distributed Systems Integration. IEEE Computer 35(6):37-46, Jun 2002.
[18] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Snelling, D., and Vanderbilt, P. (2003) Open Grid Services Infrastructure (OGSI) Version 1.0. Global Grid Forum, www.ggf.org.
[19] Foster, I., Czajkowski, K., Ferguson, D.E., Frey, J., Graham, S., Maguire, T., Snelling, D., Tuecke, S. Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF. Proceedings of the IEEE 93(3):604-612, Mar 2005.

Damandeep Kaur, 1980 holds M.E (Software Engineering) degree form Thapar University, Patiala. She is currently working as lecturer in Computer Science and Engineering Department. Her area of interests includes Parallel and Distributed Computing, Grid Computing, Resource Management, Resource Discovery and Peer-to-Peer Networks.

Jyotsna Sengupta holds PhD degree from Thapar University, Patiala. She is currently reader in Department of Computer Science, Punjabi University,, Patiala. Her area of interests includes Parallel and Distributed Computing, Grid Computing, Adhoc Networks.