Survey: Grid Computing and Semantic Web

Belén Bonilla-Morales[1], Xavier Medianero-Pasco[2] and Miguel Vargas-Lombardo[3]

[1, 2, 3] **Technological University of Panama**
Panama City, Panama

Abstract

Grid Computing is a platform for coordinated resource sharing and problem solving on a global scale among virtual organizations.

Grid uses Grid Services to access and use a set of Grid resources. Subsequently, these Grid Services need to be discovered, selected and invoked quickly and efficiently to satisfy the needs of a demanding environment such as Grid.

In this paper, we introduce how the Grid Computing takes advantage of the benefits of Semantic Web to manipulate Grid Services obtaining better searches, results and performance.

Keywords: *Grid Computing, Web Services, Grid Services, Semantic Web.*

1. Introduction

The Grid is defined as a secure and coordinated resource sharing (direct access to computers, software, data, and high performance equipment) among groups of individuals, institutions and resources that have rules which define what is shared, who is allowed to share resources and how resources can be shared. These groups of individuals, institutions and resources are organizationally and geographically dispersed, and they are called virtual organizations (VO) [1].

The benefit of increasing the computational power by combining geographically diverse systems and high performance computational equipment raises the issues of heterogeneity and scalability [2]. The use of a middleware, such as Globus Toolkit [2, 3], addresses heterogeneity, providing uniformity through standard interfaces to shared resources.

Through a middleware such as Globus Toolkit, we can obtain a set of services and software libraries to support the resources of Grids and Grid applications [2].

Because Grid computing uses Internet to enable resource sharing of geographically diverse computational resources, we can use Web Services (WS) to ensure the communication between Grid applications.

But WS do have certain limitations and they would not be sufficient for building and communicate Grid applications. To solve these limitations Grid Services (GS) [2] are introduced.

Similarly to WS [4], GS need to apply resource discovery to be functional. However, sometimes the conventional search of services does not provide good results, in these cases, Semantic Web tries to solve the problem adding meaning to GS and thus, the search is improved and better results are obtained [5].

The Semantic Web is composed of a set of technologies, and it can be defined as a symbiosis of Web technologies and knowledge representation [6] constituting what is known as web of data where some human intelligence is integrated to the Web, making the search easier and more productive [7].

Through this paper, we will focus on demonstrating the importance of Semantic Web for Grid Computing, specifically for GS.

The rest of this paper is organized as follows: Section II describes the concept of WS, their characteristics and limitations when using Grid computing. Section III examines the concept of GS. Section IV explains the convergence between WS and GS; Section V describes Semantic Web technology and section IV introduces semantic GS.

2. Web Services

Web Services describe a distributed computing paradigm focused on Internet-based standards to facilitate heterogeneous distributed computing [2].

A WS is a way to expose the functionality of an information system or application, and to make it available via web technologies [8].

WS are composed of software components or computational functions which have a set of operations which define a specific functionality; moreover, they have methods to allow access to the components or functions of any service provider from any computer, and discovery methods that permit us to identify a specific service that the operations we need.

However, WS are more than a remote procedure invocation; WS allow separating interface from implementation, this means that the interface is written in a programming language neutral independently of the programming language and platform that was used to create them. For this reason, WS allow the communications among heterogeneous systems.

WS are defined as a set of standards created by the World Wide Web Consortium (W3C) such as the Simple Object Access Protocol (SOAP), Web Services description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI) [2].

SOAP is a XML based communication protocol that permits to exchange XML (Extensible Markup Language) messages among consumers. WSDL describes the functionality and interface of WS; but it is a XML document that brings the WS information [2, 9]. The interface of a WS is usually called portType; a normal WS can only have a portType. UDDI protocol is another standard of WS that allows to register and make Web Services available to consumers [9].

In a typical WS invocation and discovery, the first step consists in discovering a WS that meets our requirements. Then, the UDDI registry will reply telling us what servers can provide us the WS we require. At this stage, we know the location of the WS we need, but we do not know how to invoke it, so we need the description of the Web Service. The WS described itself using WSDL. This information indicates to us the location of the WS and how to invoke it. The invocation is done through the SOAP language; sending a SOAP request we can ask the Web Service what we need, and the WS answers our question with a SOAP response [8].

There are some limitations to the use of WS in Grid Computing.

WS are stateless [10]. This means that WS cannot remember what the end user has done from one invocation to another. If the end user wants to perform a chain of operations, he has to get the result of one operation and send it as a parameter to the next operation.

Furthermore, WS are persistent [10] meaning they outlive all their clients. They are available from the moment the server starts until the server stops. It can be potentially insecure because when a client is using a WS, all the information the WS is remembering can be accessed by the next client using this WS. In fact, while a client is using a WS, another client could access the same WS and ruin the operations of the first client.

Due to these obstacles, a new concept arises: Grid Service [2]. GS are a kind of WS that emerge to address the limitations of working with WS within Grid computing.

3. Grid Services

Grid Services are software components which provide access to a set of Grid resources such as data sources, high performance equipment and computational resources [11]. GS are the base of the Open Grid Services Architecture (OGSA) [12], which aims to standardize all services on a Grid application. OGSA defines the GS concept based on technologies of Grid computing and WS. It also defines standard mechanisms for naming, creating and discovering GS [13, 14].

Open Grid Services Infrastructure (OGSI) defines a set of conventions and specifications for the use of WSDL and XML Schema to enable GS [12].

GS solve the problems WS have for their use within Grid Computing (stateless and persistence) using a factory/instance approach to WS. It means that we could have a central service factory which maintains a set of service instances instead of having one big stateless service shared by all users. When a client needs to invoke an operation of the service, it would talk to the instance, not to the factory. In the same way, if the client wants to create or destroy an instance, it will talk to the factory. Each client could have access to more than one instance of the service, and one instance could be shared by many clients. These instances are transients (the opposed to persistent) because they have a limited lifetime which is not bound to the lifetime of the GS container. It means that one client can create and destroy instances of a GS instead of having one persistent service permanently available. It avoids the risk that another client might destroy the operations performed over the GS by the first client.

Thus, the main improvements that GS introduce are:

GS support portType extension: It means we can define a portType as an extension of a previously existing portType [2].

GS are stateful services: It means they retain the state after each invocation and subsequent execution [14].

GS are transients: It means they are not linked to the lifetime of the server, but they can be created and destroyed as required [14].

GS have Service Data: Service Data allows to include a set of structured data to any service, which can be accessed through its interface. Service Data is an extension that allows including not only operations in the WSDL, but also attributes and any type of data (classes, arrays, etc.) [8].

4. Convergence between Web Services and Grid Services

Even though, the goal was the convergence of the standards for WS (WSDL) and GS (OGSI), this convergence has not achieved.

The main drawbacks of OGSI are as follows: its long and dense specifications, it is difficult to integrate it with existing WS tools, and its excessively object oriented [12].

To overcome these drawbacks and encourage a true convergence between the two types of services, a new standard was created to replace OGSI. In contrast to OGSI, this new standard called Web Services Resource Framework (WSRF) is based on Web Services [12]. Thus, OGSA would be based directly on WS instead of being based on an improved version of OGSI Grid services, and WSRF would be part of a collection of WS standards rather than a patch.

WSRF is not a significant change but an evolution of OGSI. The OGSA standard is not affected because changes affect the infrastructure only and not the architecture [12]. WSRF provides a set of operations that WS may implement to become stateful.

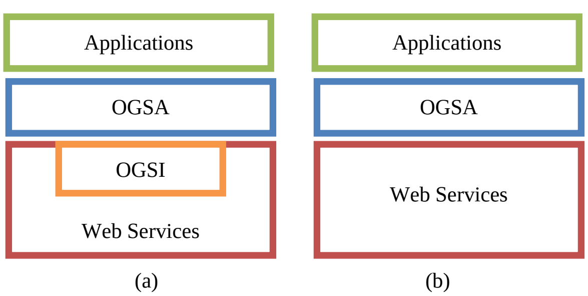

Fig. 1 (a) Architecture based on OGSI. (b) Architecture based on WSRF

Figure 1 (a) shows the approach based on OGSI where WS are manipulated to satisfy the needs of Grid Computing aggregating functions and new specifications and creating what we have called GS, while part (b) shows the architecture based on WSRF where OGSA works directly over WS.

5. Semantic Web

Semantic Web is defined as an extension of Web technology in which the information has its own meaning. It makes possible for the web to understand and satisfy the requests of people and machines using web content. Besides, Semantic Web allows obtaining effective discovery, integration, automation and reuse of applications [10].

Semantic Web was originally proposed by Tim Berners-Lee, James Hendler and Ora Lassila in 2001, with the objective of extend to the Web some of the intelligent behavior of humans [7].

Semantic Web technologies are based on open standards. If a computer understands the semantics of a document, it does not only interpret the series of characters that make up that document but also understands the meaning of the information in the document.

The technologies that integrate the Semantic Web are:
Uniform Resource Identifier (URI): It is a global naming scheme.

Resource Description Framework (RDF): It is a standard syntax for describing data. RDF is a specification that defines a model for representing the world, and syntax for serializing and exchanging that model [15].

RDF Schema: It is a data typing model for RDF which allows creating properties and classes. It could also be defined as a vocabulary for describing properties and classes of resources based on RDF and aggregating semantics descriptions [16].

Web Ontology Language (OWL): It is designed to be used by applications that need to process the content of information rather than just presenting information to humans. It adds more vocabulary to describe properties and classes than RDF or RDF Schema [15, 16].

OWL-based Web Service (OWL-S): It supplies Web service providers with a set of markup language constructed to describe the properties and capabilities of their WS in a computer-interpretable form [17].

Semantic technologies represent meaning through the use of ontologies[18]. Ontologies, in terms of Semantic Web, are defined as meta-data schemas that provide a controlled vocabulary of concepts, each of them with a complete and

structured semantic [19, 20]. They are descriptions of concepts for the purpose of allowing knowledge sharing and provide machine readable and understandable data [16].
OWL Web Ontology Language works with ontologies defining and instantiating them. OWL is also used to represent the meaning of terms in vocabularies and the relationship existing among these terms [16].
The use of Semantic Web in our applications has its benefits. The Semantic Web provides the ability to tag all content on the Web, describes each piece of information, and gives semantic meaning to each resource. Thus, search engines become more effective, and users can find the precise information they are looking for. Organizations providing services can tag services with meaning; using software agents based on the Web, these services can be found and used dynamically [21].

6. Discussion: Semantic Grid Services

As mentioned above, through WSRF, GS can be seen like WS. Moreover, we have to remember that WS are described using WSDL definitions and advertised in UDDI registries.

The capability of a WS is defined through several aspects, such as their automated discovery. The current discovery mechanism supported by UDDI is not adequate for automated discovery. Its main limitations are the lack of semantics in the discovery process and that UDDI does not use information in the service descriptions during discovery; this reduces the effectiveness of UDDI [18, 22]. Because GS are treated as WS through the WSRF standard, these limitations also affect them.

The semantic discovery of Web/Grid services is achieved adding semantics to WSDL and UDDI, and then we can find the required services using semantic matching algorithms.

In order to add semantics to WSDL of a service and find relevant operations, these operations must be mapped to concepts in appropriate OWL ontologies. The use of ontologies allow to represent service descriptions (WSDL) in a machine-interpretable form like OWL-S [17, 18].

Additionally, WSDL contains message parts defined as input and output operation parameters, thus, we can add ontologies and RDF concepts to annotate message parts in WSDL, and later we can perform more effective searches for Grid/Web Services and obtain better results.
On the other hand, we can improve semantic service discovery and storing semantics in the existing structures of UDDI. UDDI contains the technical information that is necessary to invoke a service, information about companies that own the service, and business entities [22]. If we add semantics to UDDI, we could also obtain good results in our search.
The Semantic Web approach to GS, not only facilitates services discovery, but could also improves the selection, invocation, execution, composition and interoperation of Grid/Web Services as it does for services discovery[23] through the application of Semantic Web technologies.

Consequently, we can observe that the use of Semantic Web and ontologies provide an appropriate mechanism to represent GS and to enhance their discovery, composition, selection, invocation and interoperation.

6. Conclusions

In this paper, we have defined Grid Services as software components which permit us to access a set of Grid resources such as data sources, high performance equipment and computational resources. Grid Services are an approach to Web Services; they have their own architecture, OGSA, and infrastructure, WSRF, which allows us to manipulate Grid services as Web services.

Furthermore, we have demonstrated that Grids have taken advantage of the technologies and standards developed for Web Services such as WSDL, UDDI and SOAP, and it has begun to use Semantic Web - defined as an extension of the Web technology where information has its own meaning- and its technologies (ontologies, RDF, OWL, RDF-Schema, OWL-S) to aggregate semantics to the Grid Services descriptions and enhance their discovery, composition, selection, invocation and interoperation.

These are significant aspects because a Grid Computing environment needs to have an efficient search service and quick response time in addition to adequate and accurate results. Subsequently, through the convergence between Grid and Semantic Web these needs can be fulfilled.

References

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", Proceedings of the International Journal of Supercomputers Applications, Vol. 15, 2001.
[2] I. Foster, J. Nick, C. Kesselman, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", 2002.
[3] I. Foster, and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Int. Journal of Supercomputer Applications, 1997, pp. 115-128.
[4] E. Al-Masri, and Q. Mahmoud, "Discovering the Best Web Service: A Neural Network-based Solution", Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, 2009, pp. 4250- 4255.

[5] M. Muhammad, L. Yuan, and Z. Jianqiu, "Web 3.0: A real Personal Web", 2009 Third International Conference on Next Generation Mobile Applications, Services and Technologies, 2009, pp. 125 – 128.
[6] O. Lassila, and J. Hendler, "Embracing Web 3.0”, IEEE Internet Computing, Vol. 11, 2007, pp. 90-93.
[7] S. Chun, J. Geller, and Y. Jung, "Toward the Semantic Deep Web", Computer, Vol. 41, pp. 95-97, 2008.
[8] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services: Concepts, Architectures, and Applications", Springer Verlag, 2004.
[9] N. Park, and G. Lee, "Agent-based Web Services Middleware", 2003.
[10] C. Goble, D. DeRoure, "The Grid: An Application of the Semantic Web", SIGMOD Record, Vol. 31, pp. 65-70, 2002.
[11] M. Younas, K-M. Chao, R. Anane, S. Y. Yan, P. J. Lovett, and A. N. Godwin, "Grid Services Mediation", Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA’05), Vol. 1, pp. 879-884, 2005.
[12] I. Foster, K. Czajkowski, D. Ferguson, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke "Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF", Proceedings of the IEEE, Vol. 93, pp. 604-612, 2005.
[13] D.Talia, "The Open Grid Service Architecture: Where the Grid Meets the Web", IEEE Internet Computing, Vol. 6, 2002, pp.67-71.
[14] W. Liu, Z. Liu, and W. Tong, "Agent-Oriented Modeling for Grid Service", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, 2007.
[15] V. Muppavarapu, and S. Chung, "Semantic-based Access Control for Grid Data Resources in Open Grid Services Architecture - Data Access and Integration (OGSA-DAI)", 2008 20th IEEE International Conference on Tools with Artificial Intelligence, Vol. 2, 2008, pp. 315- 322.
[16] S. Sane,and A. Shirke, "Generating OWL Ontologies from Relational Databases for the Semantic Web", International Conference on Advances in Computing, Communications and Control, 2009, pp. 157-162.
[17] D. Martin, M. Burstein, D. Mcdermott, S. Mcilraith, M.Paolucci, K. Sycara, D. L. Mcguinness, E. Sirin, and N. Srinivasan, "Bringing Semantics to Web Services with OWL-S", World Wide Web, Vol. 10, 2007, pp. 243-277.
[18] K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller, "Adding Semantics to Web Services Standards", Proceedings of The 2003 International Conference on Web Services (ICWS'03), 2003, pp. 395-401.
[19] A. Maedche, and S. Staab, "Ontology Learning for the Semantic Web", IEEE Intelligent Systems, Vol. 16, 2001, pp. 272-279.
[20] J. Hendler, "Agents and the Semantic Web", IEEE Intelligent Systems, Vol. 16, 2001, pp. 30-37.
[21] M. d'Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc,V.Lopez , and D.Guidi, "Toward a New Generation of Semantic Web Applications", IEEE Intelligent Systems, Vol. 23, 2008, pp. 20-28.
[22] S. Banerjee, S. Basu, S. Garg, S. Garg, S. Lee, P. Mullan, and P. Sharma, "Scalable Grid Service Discovery Based on UDDI", ACM International Conference Proceeding Series, Vol. 117, 2005, pp. 1-6.
[23] S. McIlraith, T. Son, and H. Zeng, "Semantic Web Services", IEEE Intelligent Systems, Vol. 16, 2001, pp. 46-53.

Belén Bonilla-Morales is a student of the Master of Science in Information and Communication Technology at the Technological University of Panama. She was awarded a Bachelor’s degree in Engineering and Computer Science from the Technological University of Panama in 2009. Her research interests include Software Engineering, Semantic Web, Grid Computing and other topics.

Xavier Medianero-Pasco is a student of the Master of Science in Information and Communication Technology at the Technological University of Panama. He was awarded a Bachelor’s degree in Engineering and Computer Science from the Technological University of Panama in 2009. His research interests include Aspect Oriented Programming Framework, Grid Computing and other topics.

Miguel Vargas-Lombardo is a professor at the Technological University of Panama, PhD awarded by the Polytechnic University of Madrid.