Grid Computing Standards

Daniel Stahr

Forschungs- und Lehreinheit Informatik V
Lehrstuhl für Informatik mit Schwerpunkt Wissenschaftliches Rechnen
Technische Universität München
Seminar Cluster Computing – SS2005

Abstract—In this paper we present most accepted standardization approaches concerning grid computing. Standardization has always been a key issue when talking about widespread commercialization and enterprise applications. In centre of interest are standards published by the Global Grid Forum and OASIS. These standards are OGSA, OGSI and WSRF. They base on Web Services. To complete the overview a standard conform implementation is presented: Globus Toolkit. In order to give a practical relation two examples will be briefly discussed. The first one is a sample implementation with Globus Toolkit and the second one is a real life example about scientific computing.

I. INTRODUCTION

This document should give a compact overview of most accepted standards in Grid Computing. These standards are mainly published by Global Grid Forum and OASIS who will be introduced in chapter two. Global Grid Forums published OGSA and OGSI which will be discussed in chapter three. Together with OASIS the GGF put out a new standard, WSRF, accepted by both worlds Web Services and Grid Computing. To relate the theoretical with practical elements in chapter four the Globus Toolkit will be introduced, an implementation that considers discussed standards. Chapter five shows a real life example about scientific computation and finally there will be an outlook on further developments.

II. IMPORTANT ORGANIZATIONS AND CONCEPTS

A. Important Organizations

“The Global Grid Forum (GGF) is the community of users, developers, and vendors leading the global standardization effort for grid computing.” [10]. This community has two goals:
- building an infrastructure for communication and exchange of ideas
- creating widely accepted standards and interoperable standards

Momentary there are over 400 organizations in more than 50 countries joining the community. Important standards are OGSA (Open Grid Service Architecture), OGSI (Open Grid Service Infrastructure) and the WSRF (Web Services Resource Framework). WSRF is created together with OASIS (Organization for the Advancement of Structured Information Standards). Besides these documentations the GGF publishes “best practises” implementation guidelines and over two dozens of working groups were defining grid standards in areas such as applications and programming models, data management, security, performance, scheduling, and resource management [15 S. 102].

B. Web Services (WS)

Web services are used for service sharing among different systems and over the internet. WS are platform independent and based on XML. Important for Grid computing are Simple Access Protocol (SOAP), Web Service Description Language (WSDL) and Web Service Introspection Service (WSIL). SOAP defines a Remote Procedure Call with XML language that is platform and protocol independent. This enables the possibility that different services on different platforms can work together. Web Service Description Language is a language which describes a Web service and all specific interfaces and APIs. Part of these interfaces is the name, parameters, failures, descriptions etc. WSIL is an XML-based service. It is used to simplify WS discovery and aggregation of WS description. Advantages against classical services as
UDDI are that WSIL is lightweight and decentralized. This makes it valuable for distributed computing. Practical WS features are Service Discover, Service Description, Service Invocation and Transport. With respect to Grid computing WS also have several disadvantages, as they are stateless, communication is not very efficient and they are not very flexible. Those problems were partly solved by OGSI and later WSRF.

C. Virtual Organizations (VO)

VOs are a key concept of OGSA. A VO defines a logical group of members within the grid, which can be geographical distributed. The idea of VOs is used for many years in business and engineering to form temporal working groups to reduce costs. Usually all members have a common interest, for example one institute, a group of scientist or a commercial enterprise. Within these VOs all resources are shared and the use is strictly defined by policies.

III. STANDARDS

A. OGSA (Open Grid Service Architecture)

The aim of OGSA is to standardize grid computing and to define a basic framework of a grid application structure. Some of the key concepts are first presented by Ian Foster who still leads the OGSA working group. This Architecture combines different aspects from grid computing with advantages from Web Services [9].

The Global Grid Forum published two main guidelines, the Open Grid Service Architecture and the Open Grid Service Infrastructure. OGSA represents a service oriented architecture. It defines mechanisms for creating, managing and exchanging information among grid services [15]. The architecture is based on Grid Services. A Grid Service is defined by ‘a Web Service that provides a set of well-defined interfaces and that follows specific conventions’ [1]. This was a logical choice, as similar problems arise in both worlds, such as distributed service discovery and use.

OGSA was first presented in 1998, but it lasted until June 2004 that OGSA 1.0 was released. Momentary the working group is building version 2.0 which is expected to be finished in summer 2005.

OGSA main goals are [15]:
- Resources must be handled in distributed and heterogeneous environments
- Support of QoS-orientated (Quality of Service) Service Level Agreement
- Partly autonomic management
- Definition of open, published interfaces and protocols that provide interoperability of diverse resources
- Integration of existing and established standards

B. OGSA Services

The OGSA specifies services which occur within a wide variety of grid systems. They can be divided into 4 broad groups: core services, data services, program execution services, and resource management services.

Core Services:
- Service Communication
  This category includes different services which handle communication among services. Including distributed logging, messaging, and events.
- Service Management
  Includes several subservices: installation, deployment, and provisioning; fault management; problem determination; and metering and accounting.
- Service Interaction
  This category includes services to provide interaction mechanisms between services in the grid. Including: Virtual Organisations; service group and discovery services; transactions; service domain, composition, orchestration; and workflow.
- Security
  This service handles right management within virtual organisations. Several aspects must be addressed, as: authentication, authorization, confidentiality, message integrity, policy expression and exchange, delegation, single sign-on, firewall-traversal, credential lifespan and renewal, privacy, secure logging, assurance, manageability, and security at the OGSI layer. As in most grid infrastructures the participants are not known to each other, trust has to be established dynamically. For this reason, there must be a policy for every pair of service requestor and service provider. The VOs are seen as bridge between these two.
Data Services:
The wide range of different data types, usability and transparency involve a large variety of different interfaces:
- Interfaces for caching
- Interfaces for data replication
- Interfaces for data access
- Interfaces for data transformation and filtering
- Interfaces for file and DBMS services
- Interfaces for grid storage services

Program Execution:
Main goal of this category is to enable applications to have coordinated access to underlying VO resources, regardless of their physical location or access mechanisms [15]. For this purpose a variety of factory services is used. Execution Services support task planning, observation and management of workflows.

Resource Management:
Resources need to be reserved and scheduled, orchestrated, and controlled. This group also maintains administration and deployment services for software deployment, change and identity management.

Despite Web Services where are good choice there are still several problems which must be solved. OGSA’s most important requirement is an underlying middleware that is ‘state full’. Web Services can be either state full or stateless, but there was no standard way to make them state full [17]. This is why another standard was created: OGSI.

C. OGSI (Open Grid Service Infrastructure)
OGSA defines a Grid Application and what a Grid Service should be able to do. OGSI specifies a Grid Services in detail. As mentioned, Grid Services are Web Services with special additions:
- Lifecycle Management
- Service Data:
  - State informations
  - Service metadata
- Notifications
- Service Groups
- PortType Extensions

WSRF is a derivative of OGSI. A first implementation can be found in GT4 (Global Toolkit 4, see chapter V). The framework combines 6 different WS specifications “that define what is termed the WS-Resource approach to modeling and managing state in a Web services context” [16].
WSRF describes how resources can be handled by Web Services (See fig. 2). Important property is that WS-Resources are stateful.

The reason why WSRF was necessary is that OGSI was never fully accepted by the WS-Community out of several reasons:
- OGSI is too object oriented, WS are not, WS are stateless and OGSI concepts don’t match with former WS specifications,
- OGSI has too much stuff for one specification.

The big difference is that the OGSA layer can now work directly on WS and is not dependent on “special” WS (See fig. 3).

WSRF specifications [16]:
- WS-ResourceLifetime: mechanisms for WS-Resource destruction
- WS-ResourceProperties: manipulation and definition of WS properties
- WS-Notification: event management
- WS-RenewableReference: defines updating proceeding
- WS-ServiceGroup: interface for by-reference collections of WSs
- WS-BaseFaults: standardization of possible failures

For further informations about WSRF check tutorial page at IBM Developers [20].

IV. GLOBUS TOOLKIT

A. Globus Alliance
The Global Alliance is an Open source project founded by several universities and the Swedish Centre of Parallel Computers. Their aim is to develop software behind the “Grid”, that is central to science and engineering. Their main product the Globus Toolkit is an implementation of the OGSA and includes software services and libraries for resource monitoring, discovery, and management, plus security and file management [11].

B. Globus Toolkit
While OGSI and OGSA are abstract approaches, the Globus Toolkit is a “concrete” realization of these concepts. Globus Toolkit is recently available in version 4.0. In the new version OGSI was replaced by WSRF. The toolkit is written in several programming languages and consists of various elements. A brief overview can be seen in figure 4. Main elements where already discussed in section 2.4 as the toolkit implements
OGSA. For more information about specific components please see [12]. A first stable release is available since 29 April 2005. The Java Web Service Core supports Apache Axis and is comprised by two components:

- A standard Apache Axis distribution, plus WS-Addressing and WS-Security support
- A set of libraries and handlers that implement WSRF/WSN (WS Resource Framework/WS-Notification) and GSI (Grid Security Infrastructure) behaviours, for use b service implementations that use those mechanisms [13].

In figure 5 you see how all different parts fit together with WSRF.

C. Globus Toolkit Example

In this chapter we will briefly describe a simple example for a stateful Web Service that uses WSRF. For more details see Globus Toolkit 4 tutorial [13]. This example supposes an installed toolkit (See [18]). The goal is to create a MathService Web Service which performs operations like: addition and subtraction. For this you need to perform five steps:

1. Define the Web Service interface (WSDL)
2. Implement the service (Java)
3. Define the deployment parameters (WSDD & JNDI)
4. Compilation $\rightarrow$ GAR file (Ant)
5. Deploy service (GT 4 tool)

The procedure is more or less identical with the creation of normal Web Services. In the first step you have to create a WSDL document defining your service. Besides the straightforward document there are some WSRF specific additions. You have to define resource properties, WSDL preprocessor and unequal normal WSDL files you don’t define bindings. Bindings are created automatically by the GT4 tool. When implementing your service in Java you have to implement the service and the resource. In simple cases you can implement both in one class. Resources are characterized by implementing the Resource and the ResourceProperties interfaces. To make our service known to the application you need to define a WSDD and a JNDI file. These two files are also XML files which describe the deployment procedure. Finally you put all together and create a GAR (Grid Archive) file using a Java build tool, in our case ANT.
V. REAL LIFE EXAMPLE: “SCIENTIFIC SIMULATION”

In order to complete your view on Grid Computing we will present a real life example application in the area of research. The example is taken from [19] where also many other examples on Grid Computing can be found. Scientific simulation concerns different areas, as physics, chemistry and biology. In these areas you find a need for large computational infrastructures as clusters. The problem for most scientific research institutions worldwide is that they can’t afford to have a high-performance computing infrastructure. Another reason is that cluster and supercomputer are limited to an upper bound of scale. So, functional requirements would be, high performance computing, at a rather low cost and upgrade possibilities at a reasonable price. Besides there are non-functional requirements as security, ease of use and management. In order to fulfill all requirements the task is to create a grid solution. In picture you’ll find a component model diagram for this solution.

The table below lists possible technologies for all different components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Products</th>
<th>Chosen product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web-browser</td>
<td>For example IE, Firefox</td>
<td>Any product</td>
</tr>
<tr>
<td>Portal site</td>
<td>IBM WebSphere, Grid System Gateway, Tomcat</td>
<td>Grid System Gateway was chosen due to the reliability and technical support</td>
</tr>
<tr>
<td></td>
<td>OGSA Toolkit</td>
<td>Globus Toolkit</td>
</tr>
<tr>
<td></td>
<td>Platform LSF Tivoli Workload Scheduler</td>
<td>Platform LSF was chosen due to its full-compability with Globus Toolkit</td>
</tr>
</tbody>
</table>

This example shows one possible infrastructure out of many and shows different components. The big advantages against conventional solutions should be the lower costs and the possibility of scale.

VI. CONCLUSION

In our opinion GRID computing will achieve a widespread commercialization within the next few years. Therefore grid must be accepted by business and smaller companies. Until now, the use is mostly scientific.

We build our trust in statements like:

“Globus Toolkit version 4.0 (GT4) will be formally unveiled next week — not by the Globus Alliance, the scientific and academic group that develops the toolkit and decides what goes in it, but by the Globus Consortium, a group led by IBM, Intel, HP, Sun Microsystems and Nortel championing the use of open source grid technologies in enterprise environments…” [2].

Another article shows that the market for high-performance technical server grew by 30 percent in 2004 [3]. These articles are examples out of many which show the wide acceptance of GRID and the ongoing commercialisation. For this purpose, 19 technology companies founded the Enterprise Grid Alliance in April 2004 [5].

“The Enterprise Grid Alliance (www.gridalliance.org) is an open, independent and vendor-neutral community formed to address the near-term requirements for deploying commercial applications in a Grid environment, according to Donald Deutsch, president of the new group and Oracle's VP for standards strategy and architecture.” [4].
One of the key issues is the standardisation. This is why we focused on the most accepted standards in our work. As shown by technologies as internet and HTTP, a standardisation can lead to an exponential growth.

"Just as the Web has revolutionized access to information, the Globus Alliance aims to achieve a similar result in computation. New types of applications will be possible when accessing supercomputers, live satellite imagery, mass storage and other on-line resources becomes as straightforward as using the Web. That is the promise of Grid computing." [11]. Another point in favour of Grid Computing is the growing amount of data and need of computational power, for example in bioinformatics ("Cancer Research", "Human Proteome Folding Project" [7]), physics ("Nuclear Research" [8]), weather analyses and e-business. The wide support of open source brings out different interesting software products, as ActiveGrid Application Builder and Active Grid Application Server, by ActiveGrid. This tool works on top of a LAMP platform (Linux, Apache, PHP/Python/Perl, MySQL) and works as an Apache module [6]. Free solutions like this open the opportunity for small enterprises to get in contact wit Grid without economical risk.

REFERENCES


