Is Oracle Database Moving Toward Grid Computing?

Christian Antognini
Trivadis AG
Zürich, Switzerland

Introduction

Over the past year Oracle has been promoting Grid computing as the next major revolution in information technology after the advent of the Internet. The Grid computing campaign started simultaneously with the announcement of their new software infrastructure which is composed of the Database 10*g*, the Application Server 10*g* and the Enterprise Manager 10*g* (a.k.a. Grid Control). According to Oracle this is the first complete Grid software infrastructure [1].

The first part of this article explains what the Grid is, or rather, what the Grid, according to the people who "invented" it, should be. The second part shows what Oracle has done in the Grid area in the last couple of years, the focus will be given to Oracle's flagship product: the database server. The aim is to show to what extent the Grid is available in Oracle Database 10*g*.

A Brief History of the Grid

The ancestor of the Grid is Metacomputing. This term was coined in the early eighties by NCSA Director, Larry Smarr [2]. The idea of Metacomputing was to interconnect supercomputer centers in order to achieve superior processing resources. One of the first infrastructures in this area, named Information Wide Area Year (I-WAY), was demonstrated at Supercomputing 1995 [3]. This project strongly influenced the subsequent Grid computing activities. In fact one of the researchers who lead the project I-WAY was Ian Foster who along with Carl Kesselman published in 1997 a paper [4] that clearly links the Globus Toolkit [5], which is currently the heart of many Grid projects, to Metacomputing.

The Foster-Kesselman duo organized in 1997, at Argonne National Laboratory, a workshop entitled "Building a Computational Grid" [6]. At this moment the term "Grid" was born. The workshop was followed in 1998 by the publication of the book "The Grid: Blueprint for a New Computing Infrastructure" by Foster and Kesselman themselves. For these reasons they are not only to be considered the fathers of the Grid but their book, which in the meantime was almost entirely rewritten and re-published in 2003, is also considered the "Grid bible".

What Is the Grid?

In 1998 Foster and Kesselman defined the Grid as follows: "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

In fact one of the main ideas of the Grid, which also explains the origin of the word itself, was to make computational resources available like electricity. One remarkable fact of the electric power grid infrastructure is that when we plug an appliance into it we don't care where the generators are located and how they are wired. We are only interested in getting the electric power, and that's all! Unfortunately, in practice, the similarities between the electric power grid and the computational Grid are very few. Actually from a computational Grid we cannot draw on computational resources, instead we have to provide the Grid with the program to be processed along with the access to the data needed for the computation.

According to a Foster's check list the minimum properties of a Grid system are the following [7]:

- A Grid coordinates resources that are not subject to centralized control (e.g. resources owned by different companies or under the control of different administrative units) and at the same time addresses the issues of security, policy, payment, membership, and so forth that arise in these settings.
- A Grid uses standard, open, general-purpose protocols and interfaces that address such fundamental issues as authentication, authorization, resource discovery and resource access.
- A Grid delivers nontrivial service qualities, i.e. it is able to meet complex user demands.

Therefore, a Grid should have a middleware that integrates distributed and heterogeneous computational resources in a large, virtual computer that can be used to solve a single problem at a given time. Of course, to achieve this result, the applications must be completely decoupled from the physical components, i.e. an application, instead of directly accessing a physical component of the Grid, has to request it through a middleware.

Who Needs a Grid?

The Grid was originally conceived to provide vast computational resources to high-performance applications. For this reason scientific applications were and are the most important exploiter of Grids.

Another class of application that is suitable to run very well on Grids, is an application that can be divided and therefore processed in many different independent tasks (a.k.a. high-throughput applications). Such applications usually exploit distributed idle resources (often idle workstations). An example of this approach has been used by the microprocessor manufacturer Advanced Micro Devices (AMD) since the development of K6 [8]. Since 1995, AMD has been using Platform LSF software to cluster over a thousand individual computers together as a single, virtual resource. By marshalling the computational resources needed to run leading design simulation and verification applications, Platform LSF ensured the right person had access to the right computing resources at the right time to handle any complex job. And by making effective use of unused resources, AMD optimized the use of existing IT equipment, and reduced the need to purchase additional hardware.

Further applications that are able to exploit a Grid are on-demand computing, which makes it possible to meet short-term requirements for resources that are not locally available, and applications that access geographically distributed data with data-intensive applications. In the latter case, if a large amount of data is processed, it is usually more efficient to let the program run where the data reside, instead of moving the data to where the program runs.

Organizations and Standards

As suggested by the previous sections it is very important that the middleware used by the application to access the Grid physical components is based on standard protocols and interfaces. The de-facto standard in this area is the Globus Toolkit which is developed by the Globus Alliance [5].

The Globus Alliance is a research and development project focused on enabling the application of Grid concepts to scientific and engineering computing. It is principally a partnership of public institutions like universities and parallel computing centers. Major corporate partners currently include IBM.

The Globus Toolkit is a set of components that can be used to develop grid applications.

- The Globus Resource Allocation Manager (GRAM) maps requests expressed in a Resource Specification Language (RSL) into commands that local computers can understand.
- The Grid Security Infrastructure (GSI) provides authentication services.
- The Monitoring and Discovery Service (MDS) combines data discovery mechanisms with the Lightweight Directory Access Protocol (LDAP).
- The Grid Resource Information Service (GRIS) provides information about resources, e.g. configuration, capabilities and status.
- The Grid Index Information Service (GIIS) coordinates arbitrary GRIS services.
- The Global Access to Secondary Storage (GASS) implements a variety of data access strategies, enabling programs running at remote locations to read and write local data.
- The GridFTP provides a high-performance, secure and robust data transfer mechanism.

For each component, an application programmer interface (API) written in C programming language is provided to software developers. Command line tools and Java classes are provided for the most important components as well.

As already mentioned above, the need for open standards that enables interoperability between components supplied by different sources, is capital. For this reason a working group of the Global Grid Forum [9] specified in 2002 the Open Grid Services Architecture (OGSA) which is an evolution towards a Grid system architecture based on Web services concepts and technologies. Figure 1 [10] shows the OGSA main architecture and, at the same time, confirms the architectural separation between the applications and physical components such as servers, storage sub-systems and networks [11].

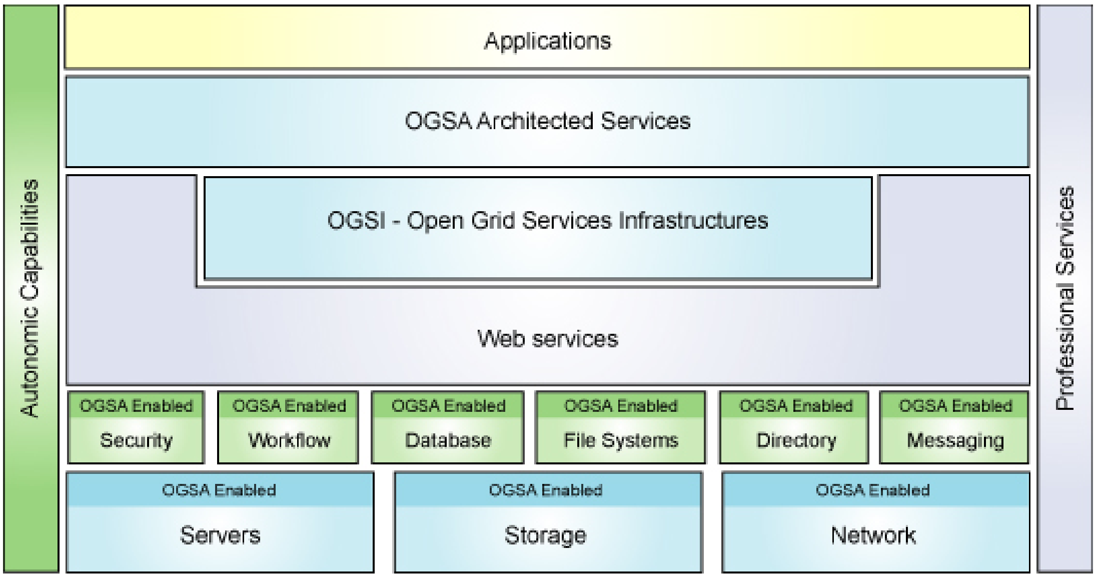

Figure 1: OGSA main architecture

As of version 3, the Globus Toolkit offers an open source implementation of Open Grid Services Infrastructure (OGSI), a part of OGSA (see Figure 1).

Grid at Oracle

In 2002 Oracle introduced the Oracle Globus Development Kit (OGDK) which allows a Globus 2.0/2.2 user to employ an Oracle9*i* database as a computation resource in a Grid [12]. OGDK supports the execution of Oracle utilities, stored procedures and SQL statements against an Oracle database. It also provides a reference implementation of Oracle Information Provider for GRIS, which returns information about all Oracle databases available in the Grid. Unfortunately no OGDK that support Globus 3 has been provided yet. Note that the current Globus release is 3.2.

It was however in September 2003, in announcing the new database version 10*g*, that Oracle started to really push Grid computing. This is confirmed by the fact that Oracle even changed its product-line name from 10*i* to 10*g*. Notice that while "*i*" stands for Internet, "*g*" stands for Grid.

Since 2003 Oracle has also been a sponsor of and a participant in working groups of the Global Grid Forum. Curiously, no press release was dedicated to this affiliation. On the other hand, on the 20th April 2004, Oracle along with EMC, Fujitsu Siemens, HP, Intel, NEC, Network Appliance and Sun announced the foundation of the Enterprise Grid Alliance [13], a consortium created to develop enterprise grid solutions and accelerate the deployment of Grid computing. Notice that Oracle presently leads the Enterprise Grid Alliance.

The foundation of this new alliance and some of Oracle's whitepapers, e.g. [1] and [14], suggests that Oracle does not have the same view as the Global Grid Forum about what a Grid is and how it should be implemented. For example, it is significant that Oracle speaks about "enterprise Grid computing" and not about "Grid computing" [1]. Anyway, let's see how much Grid is available in 10*g*…

Oracle Database 10*g* and the Grid

Oracle proclaims that 10*g* is the first complete Grid software infrastructure [1], thus, Oracle Database 10*g* is the database of choice for the Grid. According to Oracle the following 10*g* features make the database grid-aware [14][15].

- **Real Application Clusters,** with its integrated portable clusterware, enables multiple Oracle instances running on different servers to open a single database that resides on a shared storage subsystem.
- **Automatic Storage Management** simplifies the administration of database files and achieves better performance than generalized storage virtualization solutions.
- The **Resource Manager** allows the database resources which are allocated to the users to be controlled.
- The **Scheduler** provides advanced capabilities to schedule tasks.
- The **transportable tablespaces** can be used to move datafiles between databases that reside on different servers. It enables fast information provisioning.
- **Streams** is used to replicate and keep data between different databases up-to-date. It enables flexible information provisioning.
- **Instant Client** significantly simplifies the installation of an Oracle client.
- **Distributed SQL and distributed transactions** allows data stored in multiple databases to be accessed.
- **Ultra large database support** raises the maximum database size to 8 exabytes ($8 \cdot 10^{18}$ bytes).

This list of features draws our attention to two important facts that should not be underestimated.

Fact 1. Almost all these features are either already available in 9*i* or not essential for the Grid. For example ultra large database support is new in 10*g*, but for most applications the 9*i* limit of 8 petabytes ($8 \cdot 10^{15}$ bytes) is probably not a limitation. This simply means, in my opinion, that 10*g* is not Grid-aware, or at least, not more than 9*i* is. This fact is confirmed by an Oracle whitepaper written in 2002 where the same features as above, except the 10*g* new features, are listed to show that 9*i* was ready for the Grid as well [12].

Fact 2. The OGDK or another API is missing, i.e. it seams that Oracle didn't recognize the importance of such a layer between the application and the physical components (as shown in Figure 1). This fact is confirmed by the following statement of Ted Farrell, Oracle's chief architect for application development: "Oracle believes there are no programming changes to take advantage of the grid." [16]. Thus, as written in the previous section, it seems that Oracle has its own view on what a Grid is and how it should be implemented.

As a corollary to these two facts, I deduce that Oracle's Grid strategy with 10*g* was to extend only the lower parts of the OGSA architecture, i.e. their clustering technology and information provisioning features.

Conclusion

Oracle Database 10*g* is not Grid-aware! At least not according to the definition used by people doing research and development in Grid computing for more than a decade. That said, some of the 10*g* features are very important steps towards a Grid-enabled database - they are going in the right direction.

If we look back, in 1999 we already had a similar situation with Oracle8*i*, the first database for the Internet. Almost no feature for Internet computing was available with the first 8*i* release.

But when we look at Oracle right now, we see a lot of features devoted to Internet computing. In my opinion, with the Grid we will have the same situation, i.e. Oracle will be really Grid-aware in version 11 or 12.

In any case, this is not a major problem. In fact, few companies need Grids right now, and Oracle Database 10*g* has a lot of very interesting features that can be exploited without a Grid, such as improved manageability and monitoring.

[1] Oracle 10g: Infrastructure for Grid Computing, Oracle Corporation (2003)
http://otn.oracle.com/tech/grid/collateral/GridTechWhitePaper_final.pdf

[2] The Metacomputer: One from Many, NCSA (1995)
http://archive.ncsa.uiuc.edu/Cyberia/MetaComp/MetaHome.html

[3] SUPERCOMPUTING '95 I-WAY Project Develops "Internet of the Future", SDSCwire (1995)
http://www.sdsc.edu/SDSCwire/v1.15/7037.iway.html

[4] Globus: a Metacomputing Infrastructure Toolkit, I. Foster and C. Kesselman (1997)
ftp://ftp.globus.org/pub/globus/papers/globus.pdf

[5] http://www.globus.org

[6] Parallel Computing Research Newsletter, Volume 5 - Issue 4 (1997)
http://nhse.cs.rice.edu/CRPC/newsletters/fal97/index.html

[7] What is the Grid? A Three Point Checklist, I. Foster (2002)
http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf

[8] AMD Relies on Platform LSF Software to Optimize Computing Resources and Accelerate Compute-Intensive Workload, Platform Computing (2003)
http://www.platform.com/pdfs/casestudies/amd_cs.pdf

[9] http://www.gridforum.org

[10] Status and Plans for Globus ToolkitTM 3.0, Globus Alliance (2003)
http://www.globus.org/toolkit/gt3-factsheet.html

[11] A visual tour of Open Grid Services Architecture, J. Unger and M. Haynos, IBM (2003)
http://www-106.ibm.com/developerworks/webservices/library/gr-visual/

[12] Oracle and the Grid, Oracle Corporation (2002)
http://otn.oracle.com/products/oracle9i/grid_computing/OracleGridWP.pdf

[13] http://www.gridalliance.org

[14] Oracle Database 10g: The Database for the Grid, Oracle Corporation (2003)
http://otn.oracle.com/tech/grid/collateral/10gdbgrid.pdf

[15] The Grid, Oracle Magazine, Volume 17 – Issue 5 (2003)
http://otn.oracle.com/tech/grid/collateral/Grid_Feature.pdf

[16] JDeveloper 10g's ADF Opens J2EE Beyond Java, Y. L. Lee, SDTimes (2004)
http://www.sdtimes.com/news/101/story6a.htm