Towards the Next-Generation Grid: A Pervasive Environment for Knowledge-Based Computing

Mario Cannataro[1] and Domenico Talia[2]
[1]*University "Magna Græcia" of Catanzaro, 88100 Catanzaro, Italy*
[2]*DEIS-University of Calabria, 87036 Rende, Italy*
cannataro@unicz.it, talia@deis.unical.it

Abstract

The Grid is an integrated infrastructure for coordinated resource sharing and problem solving in distributed environments. A main factor that will drive the development and evolution of the Grid will be the necessity to face the enormous amount of data that any field of human activity is producing at a rate never seen before. This position paper attempts to forecast the ongoing evolution of computational Grids towards what we name next-generation Grids. We describe where we are now and where, most probably, we will go in the next years, and propose a general architecture for the emerging next-generation Grid.

1. Introduction

The *Grid* is an integrated infrastructure for coordinated resource sharing and problem solving in distributed environments. Grid applications often involve large amounts of data and/or computing, and are not easily handled by today's Internet and Web infrastructures [1].

Since their birth, *Computational Grids* have traversed different phases or generations [2]. In the early 1990s, first-generation Grids allowed to interconnect large supercomputing centers to obtain aggregate computational power not available in none of the participating sites (see the I-WAY testbed), or to decompose and coordinate distributed computations over thousands of workstation owned by their users (see the FAFNER experiment). This early generation of Grid systems was a first real implementation of a metacomputing model, and gave the basis for the second-generation Grids. Namely, FARNER originated projects such as SETI@Home, whereas I-WAY was the forerunner of Globus and Legion.

Second-generation Grids are characterized by their capability to link more than just few regional or nation-wide supercomputing centers, and by the adoption of standards (such as HTTP, LDAP, PKI, etc.) that enable the deployment of global-scale computing infrastructure, linking and allowing for the collaboration of virtual organizations. From an architectural point of view, second-generation Grids use a Grid middleware as glue between heterogeneous distributed systems, resources, users, and local policies. Grid middleware targets technical challenges in such areas as communication, scheduling, security, information, data access, and fault detection.

A milestone between first- and second-generation Grids is posed by [3], where the Grid is defined as *"flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources—what we refer to as virtual organizations"*. Main representatives of second-generation Grids are Globus that evolved from the Globus Toolkit 1 (GT1) through Globus Toolkit 2 (GT2), and Legion.

The main goal of the paper is to describe the ongoing evolution of computational Grids towards what we name *next-generation Grids*. After a brief review of the history and main milestones in the Grid evolution, we describe where we are now and where, most probably, we will go in the next years.

A main factor that will drive the development and evolution of the Grid will be the necessity to face the enormous amount of data that any field of human activity is producing at a rate never seen. The so called "*data tombs*", i.e. data stores where data is stored and, with high probability, never accessed again, is a trend in current database field. Although we can imagine larger and more powerful databases and data warehouses where to store data, only a small portion of it will be accessed by humans or programs: the obstacle is not the technology to store and to access data, but perhaps what is lacking is the ability to transform data tombs in useful data, extracting knowledge from them [4].

The effective and efficient use of stored data and its transformation into information and knowledge is not the only driver in Grid evolution. So the paper reviews some of the current and future technologies that will impact on architecture, computational model and applications of the next-generation Grid. The architecture of this next-generation Grid, in our vision, will be derived considering both the technologies and methodologies that most probably will impact on (and will integrate into) current Grid solutions, and the major requirements emerged in many sector of computing, apparently far and unaware of Grids, such as mobile

and pervasive computing, ontology-based reasoning, peer-to-peer, and knowledge management.

The motivation for third-generation Grids is to simplify and structure the systematic building of Grid applications through the composition and reuse of software components and the development on knowledge-based services and tools. So, following the trend emerged in the Web community, the *service-oriented* model has been proposed. Another key aspect that more and more will characterize third- and next-generation Grids is the systematic adoption of metadata to describe resources, services, data sources and each Grid component, and the use and transmission of such metadata to enhance, and possibly automate, processes such as service discovery and negotiation, application composition, information extraction and knowledge discovery.

The rest of the paper is organized as follows. Section 2 is an overview of the current and ongoing technologies that will affect the development of the Grid. Section 3 proposes a comprehensive software architecture for the emerging next-generation Grid, as the integration of currently available services and components in Semantic Web, Semantic Grid, Peer-To-Peer, and Ubiquitous systems. Finally, Section 4 concludes the paper.

2. Emerging technologies for future Grids

This Section discusses current and future technologies that will play a major role in the development of future Grids.

2.1. The Semantic Web

The *Semantic Web* is an emerging initiative of World Wide Web Consortium (W3C) aiming at augmenting with semantic the information available over Internet, through document annotation and classification by using ontologies, so providing a set of tools able to navigate between concepts, rather than hyperlinks, and offering semantic search engines, rather than key-based ones. The Semantic Web is defined as *"…an extension of the current Web in which information is given well defined meaning, better enabling computers and people to work in cooperation…"* [5]. The main goal is to allow Web entities (software agents, users, programs) to interoperate each others, dynamically discovering and using resources, extracting knowledge and solving complex problems.

Although the use of metadata to annotate and describe Web contents is the key requirement to allow machine-to-machine operation, complex automated processing is needed to give semantic to each web resource. A layered model of the Semantic Web comprises:

- A set of web resources, that are characterized by a unique global identity and are described by metadata expressing knowledge about them, in a common and shared formalism and rules for inferring new metadata and knowledge, through ontologies.
- A set of basic services, such as reasoning and querying over metadata and ontologies, semantic search engines, etc. These services represent a great evolution with respect to current Internet services, such as DNS and key-based search.
- A set of high level applications developed using basic services.

At this stage major efforts regard the development of languages and technologies for the standard and agreed modeling and implementation of metadata and ontologies (XML Schema and RDF Schema, DAML+OIL and OWL). Tools and techniques for ontologies manipulation and navigation are in their early stage and these are some examples (see www.daml.org, www.ontoweb.org):

- *ontology building tools*, that allow users to define and build ontologies (e.g. DUET, OilEd, OntoEdit)
- *ontology-based annotation tools*, for annotating web resources according to an ontology (e.g. UBOT DAML);
- *ontology learning tools*, for learning ontologies from natural language documents (e.g. Corporum, Text-To-Onto);
- *ontology manipulation tools* (e.g. DAML API), allowing to navigate and manipulate ontologies.

2.2. The Semantic Grid

The *Semantic Grid* is an initiative of the UK EPSRC/DTI Core e-Science Program that aims to integrate and bridge the efforts made in the Grid and in the Semantic Web communities [6]. The Semantic Grid vision is to incorporate the Semantic Web approach (systematic description of resources through metadata and ontologies, and provision for basic services about reasoning and knowledge extraction), into the ongoing Grid. The Semantic Grid statement says *"As the Semantic Web is to the Web, so is the Semantic Grid to the Grid. Rather than orthogonal activities, we see the emerging semantic web infrastructure as an infrastructure for Grid computing applications*", that forecasts, for the Grid, a similar evolution of the Web towards the Semantic Web. The Semantic Grid is supported by the Global Grid Forum through its Semantic Grid Research Group (SEM-GRD).

The research issues of the Semantic Grid covers many aspects of the next-generation Grid:

- full support of the three recognized layers composing a Grid, i.e. computation/data, information (where data produces information) and knowledge layer (where knowledge can be used to take decision);
- provision of seamless, pervasive and secure use of resources.

Although the Semantic Grid initiative is in its early stage, we think that it will be a significant component of the next-generation Grid.

2.3. OGSA and Globus Toolkit 3

The Open Grid Services Architecture (OGSA) introduces service orientation in Grids, leveraging the results of Web Services [7]. A Grid Service is a Web Service that conforms to a set of conventions for the controlled, fault resilient and secure management of stateful services and exposes capabilities via standard interfaces [8]. The OGSA represents a border between second- and third-generation Grids.

To satisfy the new requirements of Grids, Grid Services extend significantly Web Services because of:

- Gris Services may be *dynamic* and *transient*, i.e. some or all services (i.e. software, sensors, computing resources) participating in computation can be switched on or off, changing their availability.
- Grid Services are *globally distributed*, without a central control and without a globally-agreed trust relationship.
- Grid applications can involve tens to hundreds of Grid services, which have to be coordinated in an efficient way.
- Grid applications are often *long-lived* and this impacts on the live requirement for Grid services.

The ongoing implementation of OGSA is Globus Toolkit 3 (GT3). GT3 extends GT2 providing open source implementation of OGSI (Open Grid Services Infrastructure). OGSI addresses detailed specifications of the interfaces that a Grid Service must implement in order to fit into the OGSA architecture. GT3 offers several OGSI-compliant services corresponding to usual GT2 services, and the ability to create new OGSI-compliant services. These services are provided through standard OGSI mechanisms that enable a consistent way of querying any Grid Service about its configuration and status information.

The Grid Service Specification specifies the way in which a client interacts with a Grid Service. OGSI software provides mandatory Grid Service features, such as service invocation, lifetime management, a service data interface, and security interfaces that ensure a basic level of interoperability among all Grid Services. A Grid Service executes inside a Hosting Environment that supports the language in which the service is written (e.g. C, Java, Python, .NET, etc.), but the Hosting Environment insulates clients of Grid Service from its particular implementation language. Access to services is through the standard Web Services Definition Language (WSDL). Thus, the clients of a Grid Service can be written in any language for which bindings to WSDL are available. The GT3 implementation includes support for the WSDL and SOAP W3C standards.

2.4. Data Grids and Knowledge Grids.

Grids can be used today as effective infrastructures for distributed high-performance computing and data processing. Grid application areas are shifting from scientific computing towards industry and business applications. To meet those needs *Data Grids* have been designed to store, move, and manage large data sets located in remote sites. Data Grids represent an enhancement of Computational Grids that allow handling of large data sets in distributed data-intensive applications.

As an advancement of the Data Grid concept, it is imperative to develop knowledge-based Grids that may offer tools and environments to support the process of analysis, inference and discovery over data available in many scientific and business areas. These environments will support scientists and engineers in the implementation and use of Grid-based problem solving environments (PSEs) for modeling, simulation and analysis of scientific experiments. The same can occur in industry and commerce, where analysts need to mine the large volumes of information generated by industrial processes to support corporate decision making.

Recent works [9, 10] claimed that the creation of *Knowledge Grids* on top of Computational Grids is the enabling condition for developing high-performance knowledge discovery processes and meeting the challenges posed by the increasing demand of power and abstractness coming from complex problem solving environments. Knowledge Grids offer high-level tools and techniques for the distributed mining and extraction of knowledge from data repositories available on the Grid, thus they realize the higher layer of the Grid architecture (computation/data, information, and knowledge layers).

Main issues for the development of the knowledge layer in Grids are: i) synthesizing useful and usable knowledge from data, and ii) leveraging the Grid infrastructure to perform sophisticated data-intensive large-scale computation. The integration of knowledge discovery techniques in Grid environments must pass through a unambiguous representation of the knowledge base (through metadata and ontologies) needed to translate moderately abstract domain-specific queries into computations and data analysis operations able to answer such queries by operating on the underlying systems.

The KNOWLEDGE GRID is a joint research of ICAR-CNR, University of Calabria, and University of Catanzaro, Italy, aiming at the development of an environment for geographically distributed high-performance knowledge discovery applications [11]. The KNOWLEDGE GRID can be used to perform data mining on very large data sets available over Grids, to make scientific discoveries, improve industrial processes and organization models, and uncover business valuable information. Other examples of Knowledge Grids are shortly described in [12].

2.5. Peer to Peer computing

Peer to Peer (P2P) is, at the same time, a set of protocols, a computing model, and a design philosophy

for distributed, decentralized, self-organizing systems. In other words, P2P is a set of methodologies and technologies to allow two or more computers to collaborate in a network of equals (peers), without central coordination [13]. The technologies and challenges faced in P2P systems are not new: in P2P peers stay at the edge of a network in which everyone creates as well as consumes, that is the original formulation of Internet [14]. The basic elements of P2P are:

- the *action* that is taking place at the edge of the network (e.g. computing, resource sharing, communication);
- *resources* that are being shared between peers (e.g. CPU idle cicles, disk space, computing power, network bandwidth, content, etc.);
- direct *communication* between peers, that takes place without great assumptions about the underlying network and protocols (e.g. DNS).

The central aspect of P2P is the management of a large number of peers, in a usually unstable environment (where peers appear and disappear continuously) without a central coordination. In contrast, today Grids are based on a persistent service infrastructure that often uses a central or a distributed, but hierarchical coordination. Almost all the P2P applications belong to the following categories:

- distributed computing, such as SETI@home;
- content sharing, such as Napster (that is not a pure form of P2P since it was based on a central directory service) and Gnutella;
- collaboration, such as instant messaging.

The main potentiality of P2P is the ability of exploiting idle resources (computing), facilitating the exchange of information (information discovery and content distribution). From the Grid point of view the main interesting aspects of P2P are scalability, self-configuration, autonomic management, dynamic resource discovery and fault tolerance. On the other hand, current P2P systems lack in some fields that are crucial to the deployment of production-quality services, such as QoS negotiation, persistent and multi-purpose service infrastructure, complex services (beyond the simple file-sharing), robustness, performance and security. We think that P2P and Grid communities can benefit each other sharing their key technologies.

2.5. Pervasive and Ubiquitous computing

Ubiquitous Computing, but the terms "*pervasive*" or "*ambient*" are used interchangeably, regards the description of distributed computing devices, such as personal devices, wearable computers, sensors in the environment, and the software and hardware infrastructures needed to support applications on these computing devices. As Mark Weiser described in its well known article [15], Ubiquitous Computing is about interconnected hardware and software that are so ubiquitous and so spread in the environment that no one notices their presence. A more deep characterization can be obtained if we better consider the main dimensions of Ubiquitous Computing [16]:

- *mobility* of users, devices (PDA, phones, etc.), and software (e.g. mobile agents);
- *embeddedness* of devices into the environment.

So, mobile computing is about the ability to physically move computing services with users, but the computing model does not considerably change while the user move. In pervasive computing the device has the capability to obtain information from the environment in which it is embedded and to adapt its behavior, e.g. by dynamically building (choosing) a suitable model of computing. In ubiquitous computing, we have a combination of high mobility and high embeddedness: any device, while moving with the user, can build incremental models of the visited environments and configure its services accordingly. On the other hand, the software (the environment) can adapt itself to the currently available devices. The integration in Grid systems of large-scale mobility and pervasive computing functionality poses new challenges and requirements to the underlying architecture:

- Ontology-based semantic modelling, (user preferences, devices characteristics, context) enables reasoning about user's needs and the required adaptation of services.
- Adaptable and composable software infrastructure, able to find, adapt, and deliver appropriate applications (services) to the user's computing environment (devices) on the basis of context. To execute a user task the computing platform should dynamically find and compose the appropriate components and services, and, once instantiated, the application may need to move between devices and environments.

3. A comprehensive architecture for the next-generation pervasive Grid

This section discusses a promising architecture of the future next-generation Grid. The main (new) requirements of future Grids are (will be):

- knowledge discovery and knowledge management functionalities, for both user's needs (intelligent exploration of data, etc.) and system management;
- semantic modeling of user's tasks/needs, Grid services, data sources, computing devices (from ambient sensors to high-performance computers), to offer high level services and dynamic services finding and composition;
- pervasive and ubiquitous computing, through environment/context awareness and adaptation;
- advanced forms of collaboration, through dynamic formation of virtual organizations;
- self-configuration, autonomic management, dynamic resource discovery and fault tolerance.

These requirements are partly fulfilled by some of the previous emerging technologies, and we envision that next-generation Grids will be based on their integration and composition. Figure 1 shows an overview of these technologies with respect to two fundamental dimensions: knowledge and distribution.

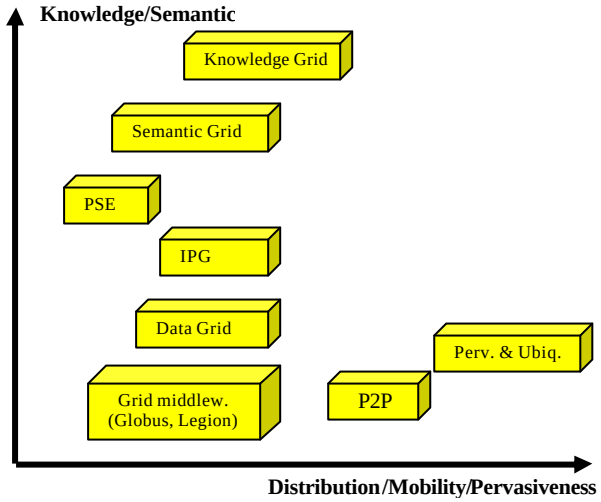

Figure 1. Key technologies for Grids.

Figure 2 shows a layered architecture for the next-generation Grid: in our opinion P2P will be the transversal technology on which fundamental tasks such as presence management, resource discovery and sharing, collaboration and self-configuration will be based.

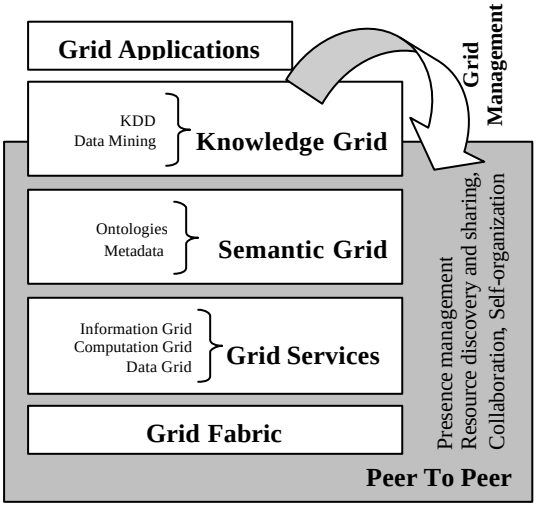

Figure 2. Main layers of the next-generation Grid.

5. Conclusions

As well as the Internet is shifting from information and communication to a *knowledge delivery infrastructure*, the Grid is moving from computation and data management, to a pervasive world-wide *knowledge management infrastructure*.

To achieve this very ambitious goal the next-generation Grids should be based on the key technologies we discussed in this paper. How those technologies will be integrated and deployed is the challenge that the research community must face in the next years.

Acknowledgements

This work has been partially supported by Project "FIRB GRID.IT" funded by MIUR.

References

[1] Foster I. and Kesselman C. (eds.), *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann Publishers, 1999.

[2] de Roure D., Jennings, N. R. and Shadbolt, N. (2003) "The Evolution of the Grid". *Concurrency and Computation: Practice and Experience*. (2003)

[3] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *Supercomputer Applications*, 15(3), 2001.

[4] Evolving data mining into solutions for insights (Special issue on), *CACM*, Vol. 45, No. 8, August 2002.

[5] Tim Berners-Lee, James Hendler, Ora Lassila, "The Semantic Web", *Scientific American*, May 2001

[6] de Roure, D. Jennings, N. R., Shadbolt, N. "The Semantic Grid: A future e-Science infrastructure". *Concurrency and Computation: Practice and Experience*, 2003.

[7] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[8] D. Talia, "The Open Grid Services Architecture: Where the Grid Meets the Web", *IEEE Internet Computing*, Vol. 6, No. 6, pp. 67-71, 2002.

[9] F. Berman. "From TeraGrid to Knowledge Grid". *CACM*, Vol. 44, N. 11, pp. 27-28, 2001.

[10] W. E. Johnston, "Computational and Data Grids in Large-Scale Science and Engineering". *Future Generation Computer Systems*, Vol. 18, N. 8, pp. 1085-1100, 2002.

[11] Cannataro M. and D. Talia, "KNOWLEDGE GRID An Architecture for Distributed Knowledge Discovery", *CACM*, Vol. 46, No. 1, pp. 89-93, January 2003.

[12] M. Cannataro, A. Congiusta, D. Talia, P. Trunfio. "A Data Mining Toolset for Distributed High-performance Platforms". *Proc. Conf. Data Mining 2002*, Wessex Inst. Press, Bologna, Italy, 2002.

[13] D. Barkai, "Technologies for Sharing and Collaborating on the Net, 1st Int. Conf. on Peer-to-Peer Computing (P2P'01), Linkoping, Sweden, 2001.

[14] D. Schoder and K. Fischbach, Peer-to-Peer Prospects, *CACM*, Vol. 46, No. 2, pp. 27-29, February 2003.

[15] Weiser M., "The computer for the 21st century". *Scientific American*,), pp. 94–104, September 1991.

[16] K. Lyytinen and Youngjin Yoo, "Issues and Challenges in Ubiquitous Computing", *CACM*, Vol. 45, No. 12, pp. 62-65, December 2002.