Grid4All: Open Market Places for Democratic Grids

Ruby Krishnaswamy[1], Leandro Navarro[2],
René Brunner[2], Xavier León[2], Xavier Vilajosana[3]

[1] Computer Architecture Department, Polytechnic University of Catalonia, Spain
{leandro,rbrunner,xleon}@ac.upc.edu
[2] France Telecom R&D, France
ruby.krishnaswamy@orange-ftgroup.com
[3] Universitat Oberta de Catalunya, Spain
xvilajos@uoc.edu

Abstract. The Grid4All project is focused on the provision of the benefits and opportunities of Grids for everyone, including small organizations such as schools, families, non-governmental organizations, or small businesses. This involves multiple relevant and related aspects despite the scale of the global system: (i) self-management of applications as they adapt to environmental changes, (ii) the complexity of developing and using applications in that situation by multiple users, (iii) and the need to organize, govern, and regulate the community. Grid4All [4] promotes the concept of a democratic Grid, virtual organizations and self-management systems, based on decentralized overlays. Thereby, the providers offer resources and services either for a shared-interest within a virtual organization (pooling) or for an open market across virtual organizations. These two models of distributing resources that co-exist locally, contribute to achieve global regulation. We propose an architecture according to these ideas, which are inspired by real-world cases, which include a collection of data sharing and execution services, used by collaborative applications.

Key words: Economic-aware Grid Application, Markets and Market Mechanisms for the Grid, Democratic Grid, Large-scale Grid Market, Virtual Organizations

1 Introduction

The idea of a public utility for digital data and computing is a natural evolution beyond the widespread accessibility of Internet connections with higher capacity and reliability to support the need for social interaction, sharing, and working together. This model apply that to rent or share competitively computational services is easier than having to acquire and maintain hardware and applications in advance. Therefore, The Grid or The Cloud opens an opportunity not

[4] More information at: http://www.grid4all.eu/

only to build a ”virtual supercomputer”, necessary to tackle scientific or industrial grand challenges, but also an opportunity for citizens, for the rest of us, to democratize this distributed computing global network in the sense of opening up participation by lowering the cost, facilitating the usage, supporting collaboration, and data sharing, increasing the flexibility of computing resources to adapt to demand, facilitating the transparency and control on how services and resources are shared and used.

There is an huge potential for empowering many people or small organizations equipped with low-performance PCs connected to the Internet as shown by many recent Internet-based services and applications such as Google Search, Yahoo Mail, SETI@Home, Flickr, and Google Docs. Moreover, commercially available platforms such as Amazon Web Services, SUN's Network.com, and Google Application Engine are expected to emerge.

The Grid4All project promotes the concept of a democratic Grid, accessible to modest groups of end-users such as schools, families, non-governmental organizations (NGO), or small businesses. It enables to put together people and computing resources to form Virtual Organizations (VO), a virtual collection of users or institutions that pool their resources into a single virtual administrative domain for a common purpose. Virtual Organizations can also trade resources among different VO on a decentralized market place.

Therefore, our use cases involve schools, learning institutions, families, and small businesses. Our scenarios do not only include distributed execution of bag-of-tasks computing-intensive applications, but they are also oriented towards facilitating collaborative work. In the area of data services, compared to previous Grids, the Grid4All architecture provides with a minimal administration enhanced support for content sharing and collaboration within groups. Semantic search and ontologies are used to locate and select among diverse resources and services.

The objective of this paper is to present the opportunity, the challenges, and the design for a Grid for domestic users or small organizations and how a scalable market place for resources is a key enabler. The rest of the paper starts describing an educational VO as one typical scenario. Based on the requirements it presents the Grid4All architecture. The corresponding Grid4All infrastructure is based on a component model supporting a number of properties required in a democratic Grid: usability, self-administration, dynamic behaviors, and security. We present several novel or improved collaborative applications that rely on and leverage self-managed software components and services. Furthermore, they apply network-wide shared data services for storage with support for flexible concurrent modification of application data and collaborative task execution.

2 Motivating Scenario

Among multiple scenarios for modest groups of people including schools, families, NGOs, or small businesses, we present in this section an educational scenario for remote collaborative learning tasks that shares many aspects with other cooperative work activities within the focus of this work. Our scenario takes its inspiration from iEARN, the International Educational, and Research Network [3]. iEARN is a world-wide educational network of people (young students, teachers, and families), who participate in diverse collaborative educational projects within regional or world-wide collaboration networks.

This scenario involves a network of schools engaged in collaborative activities. A collaboration consists typically of a course or a course project. It may involve students and instructors from multiple sites, spanning different departments, schools, countries, time zones or cultures. The member institutions may be schools at any educational level (primary, secondary or university). Participants work either from home or at their school.

A collaborative network pools resources from different providers for its activities. The providers include the schools, the students, the parents, and outside partners. The resources may be physical or virtual and include computational, storage, and other resources. Educational institutions already rely on specific digital tools, for instance to consult external experts, to perform digital experiments, and simulations, to communicate with other students, and to work remotely. These tools should be available in the collaboration and should be made aware of the collaboration context.

Collaborative work tools should allow a group of users (e.g., a class, or a set of instructors from different schools), to share computational or storage resources, to share and modify content together (e.g., to author a document collaboratively), and to make common decisions. However, the system must ensure that users access resources or information only if authorized.

Collaborative groups are fluid, short-lived or long-lived, and the membership of a group changes over the time. Groups are likely to overlap: some users will belong to several groups; the same content could be shared in different groups. However, content belonging to a group should be isolated from outside access. Content must be persistent and modifiable (subject to authorization).

Existing infrastructures of schools may vary significantly as it changes the demand imposed by users on it. For instance, one laptop per child (OLPC) based projects [6] rely on an ad-hoc network of personal computers without dedicated servers. In contrary, PlanetLab [7] consists of shared servers in a P2P manner but provides only best-effort quality service. Grid5000 [2] is similar but is a more stable environment with resource reservation.

Despite the differences, networks and computers usually remain unreliable given the usual lack of technical capacity to properly manage systems. Therefore, churn (computers connect and disconnect without notice), failures, and service degradation are arbitrarily frequent.

3 Main Requirements

The main challenges in this setting are:

- The work environment is dynamic and unpredictable because participants come and go. The load varies: just before an exam deadline, many students

run simultaneously resource-intensive activities. Computing resources can fail. Nonetheless, the system should remain manageable, in order to ensure some level of dependability. For instance, a class project should not miss its deadline because of a glitch in the system. Otherwise, schools will be reluctant to use it.
- The responsiveness of the system should not degrade despite partial failures, network problems, and disconnected-mode work.
- The system must appear simple to its users and manual administration cost must be very low. The system should adapt automatically to problems, while collaboration should not tax the schools (usually very limited) support staff.
- Collaboration among distributed partners requires computer support. In that case, activities in a distributed school environment are not viable face-to-face (at the same place) and given the number and diversity of participants, there is need to use applications to support sharing, coordination and collaboration.
- A collaboration incorporates rules and policies. For instance the members may need to abide to predefined rules for that community, or to make contributions, either monetary or in kind (e.g., computational resources), and system may need to support or enforce this.
- Participants may move to multiple locations: in the school, at home, in a library, on the move or at work in other cities or countries elsewhere in the world.
- Participants must be able to work remotely in disconnected mode. It is not reasonable to expect that all participants are connected all the time.

4 A Democratic Grid Architecture

To overcome these challenges, the Grid4All architecture is based on the concept of virtual organization and it uses a component framework, which provides autonomic management mechanisms.

Virtual Organizations. A Virtual Organization (VO) consists of a set of resources and a set of users. Both resources and users may belong to different institutions; thus these sets may have cross institutional and administrative boundaries. The VO concept combines two related, but relatively orthogonal sets of mechanisms:

- Grouping and virtualisation of a set of resources (objects) that we call a virtual resource.
- Grouping and naming a set of users (subjects) that we call a user group.

The VO management system and the security infrastructure constitute one link between these two sets of concepts. The VO management system maintains the security associations between user groups and virtual resources, as prescribed by the VOs security policy. After a user executes an operation on a resource, the security infrastructure ensures that the operation is permitted according to the relevant policy.

The other link between the two concepts of virtual resources and users is that the VO management system also monitors users to ensure that they fulfil their obligations as stipulated by the appropriate policy. Indeed, the placement of both users and virtual resources within the VO is motivated by the two-way interaction with the VO management system indirectly monitoring both virtual resources and users.

Component Framework. Components are an effective approach for building and managing complex software systems. In this approach, all system elements are constructed or wrapped as components and management involves using primitives of the underlying component model (e.g., setting properties of components). We have adopted a component model that extends Fractal, a general and reflective component model intended for building dynamically reconfigurable systems, with remotely accessible components and over different machines distributed entities. The model adds composite bindings that support remote method invocation and group communication. This is particularly useful for building decentralized, fault-tolerant applications. Group bindings enable invocations to be delivered either to all group members (one-to-all style) or to any, randomly-chosen group member (one-to-any style).

Autonomic management. The Grid4All scenario is very dynamic: members and resources can be constantly changing (we call this churn) as individual computers can join and leave continuously as members go off-line and on-line. Over the time, the total amount of available resources and members also changes; we call this evolution.

Within the democratic Grid, virtual organization will be created, grow, shrink, and die off as the members interest into the collaboration change. New applications and services will be added to an existing VO. Some VOs will prove popular, some will not. The democratic Grid evolves and changes at a high rate. Churn rates are expected to be high as different populations of user pools exist and share their resources.

Autonomic computing aims to automate system administration and management. By taking humans out of the loop, labour costs decrease, response to problems is faster and the availability improves. Grid4All takes the control loop approach to autonomic management. High-level management policies are expressed in a high-level language. These are translated into rules for the management runtime.

The runtime monitors the system and triggers events, which execute the associated rules. For instance, when a machine crashes, other interested machines receive the corresponding events and can recover.

The elements of the overall architecture of Grid4All VOs are presented in Figure 1. By an analogy with Foster et al.s decomposition of Grid systems [1], the Grid4All VO architecture can be decomposed as follows, from bottom to top:

The **Fabric** includes computers, storage capacities, files, application binaries and sources as well as other basic resources, which are provided to the VO by its

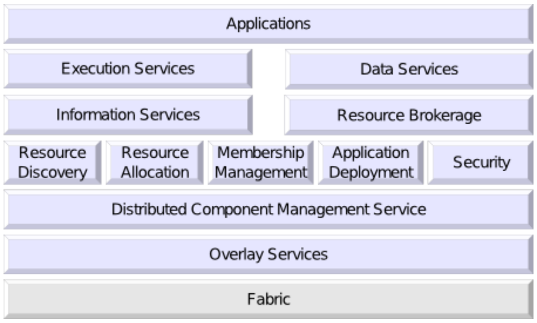

Fig. 1. Overall Grid4All architecture.

members. In the following, the term resource without further qualification refers to a fabric resource.

The Core VO Support Architecture comprises the following basic services:

- Overlay Services deal with connectivity of VO resources and elements. Participation of nodes (i.e., computers) form a logical network is called an overlay. We use structured overlay networks because of their inherent self-management properties and scalability. These services provide basic naming, communication, and data storage services that are used by upper layer services.
- The Distributed Component Management Service (DCMS) supports distributed VO-aware self-* services. DCMS allows the application developer to program self-* functionality of component-based services. In the following, we refer to the latter as self-* code. DCMS is overlay-based and it delegates resource discovery and allocation to the VO management services.
- The VO Management Services include Resource Discovery and Allocation, Membership Management, Application Deployment, and Security services. VO Management Services maintain information about VO users and decide on sharing of VO resources between VO users. It also provides the basic mechanism to start new services within the VO. VO Management Services are overlay-based.

The **Information Service** provides match-making between semantic service descriptions and client requirements.

The **Resource Brokering Service** facilitates interaction between VOs. This service enables applications and VOs to find external resources as they need them. The brokering facility is a resource market place where providers and consumers can meet and trade Grid resources.

Our support for collaborative and federative VOs focuses on data services for sharing storage capacity and content, i.e., a **VO-oriented File System (VOFS)**, a semantic consistency middleware **Telex**, and an **Execution Service** for deploying computation tasks, which are based on a bag-of-tasks model.

Supporting flexible and dynamic VOs requires a novel approach to information sharing. The Grid4All data storage architecture provides three different levels of flexibility. (i) VO users can pool disk space over the network: either offered to each other from their own disks, or acquired on an open market. (ii) A VO has an associated virtual file system, a flexible federation of file names and file contents, exposed by VO users to one another. (iii) A semantic middleware layer allows users to update shared data in a flexible manner. Users may share content in either on-line or off-line mode; the semantic store resolves conflicting updates according to data semantics.

Applications use the lower layers, primarily to obtain computation, storage, and content resources but also to achieve self-* behavior.

5 Resource Allocation for Democratic Grids

Applications require resources model computational and storage elements to execute. The available resources on a VO have to be managed internally but they can be also be brokered with other VO at the market place. This adds another dimension to the adaptability of VO to a changing demand or changing conditions of resources beyond the incorporation of additional participants. Resource allocation is a key issue in this dynamic environment that intends to self-manage. This issue permeates mostly all services at all levels and even end-user applications. We need to distinguish between resource management (how resources are contributed and then used) and resource brokerage (when and how resources are offered and obtained from the market place).

5.1 Resource Management

The resource management service enables members to contribute resources to the VO and to discover and allocate VO resources. In the context of this service, a resource is a share of computation and storage capacity that supports component deployment; a component is deployed on a single resource.

Contributing resources involves configuring and adding the local Grid4All container or a remote container to the VO. Resource discovery relies on specifying requirements on resource properties; e.g., specifying a minimum CPU speed or memory size. Resource allocation allows reserving discovered resources or parts thereof. Reserving a part of a resource creates a new resource with capacity deducted from the original resource. The service includes support for publishing resource discovery events (e.g., a new resource has joined) and resource parameter change events (e.g., the CPU load of a specific resource has changed). Resource management is used in a VO to regulate the resource usage between services prioritizing based on a policy upon research scarcity.

5.2 Resource Brokerage

Brokering concerns arbitration and allocation of Grid resources to virtual organizations. VOs lease resources on need by negotiating at the resource market place.

The Grid4All model of virtual organizations is similar to peer-to-peer networked applications. Members of a VO use resources that belong to the VO. In Grid4All we allow VOs to incorporate resources from non-members. Such resources are expected to be leased from resource brokers that select and match of consumers and suppliers.

The motivation is to adapt to fluctuations in supply and demand with mechanisms to arbitrate which requests for resources should be satisfied and by whom. Priority based, proportional sharing allocations are mechanisms that are useful when the different consumers (applications or users requiring resources) belong to the same organization. In that case, consumers may be prioritized or allocated to shares. This is not the case if applications and users from multiple independent VOs compete for resources. Market based brokering with pricing mechanisms provide fair arbitration, gives incentives and is decentralized.

The minimal GRIMP (Grid4All Market Place) addresses:

- Selection of suitable resources.
- Decentralized feedback from market to aid traders in their negotiation.
- Mechanisms to allocate and establish prices.
- Protocol to establish agreements between consumers and provider.

The resource brokering problem is decomposed into multiple replaceable and extensible components. This allows different scenarios to use a subset of these services to realize their objectives. The platform provides tools that allow traders (consumers and providers) and mediating agents to create ad-hoc auction markets, where they can negotiate with resources. Divergence of prices and hot-spots are risks for market places with simultaneous auctions. For that reason, the GRIMP provides a decentralized publish/subscribe-based information service. This allows trading agents to react to market situations by publishing and subscribing to aggregated (summarized) global information (such as average prices) of the market place. The minimal platform consists of tool-kits and basic services and may be extended to provide added-value, higher-level brokerage, and mediating services. The market-place itself will use the VO run-time services for deployment, monitoring, and other self-management.

5.3 Services and Functionality

Auction servers. Its service is accessed through a common market interface. Participants send bids specifying the resources, quantities, time constraints, and budget. The auction determines allocations and the transaction prices. Behind a unique facet (interface), auction servers may implement different mechanisms (rules and policies for bidding, price and allocation determination). The server is designed as a set of Fractal components that separate economic and system concerns. The server is designed as a tool-kit, within which specific allocation and pricing policies and rules may be plugged in. Auction servers are advertised at the Semantic Information System (SIS) which may be queried by traders to obtain candidate lists of active auction markets.

Within this framework two proof-of-concept economic mechanisms have been implemented:

- K-pricing double auction: This mechanism is used to acquire storage resources to satisfy the scenario described in Section 2. This mechanism is computationally efficient for the trading of a resource type amongst multiple consumers and providers.
- Combinatorial auction: This mechanism allows allocation of bundles of computational and storage resources.

Market factory. This is a repository service provided by the market place and offers two sets of interfaces:

- To designers and developers to register implementations for a new type of auction mechanism.
- A mechanism for the selection allows the traders (consumers and providers) to choose a specific auction format and request instantiation of a new auction server.

Market Information Service. Obtaining synthetic and summarized information is important in a decentralized market place with a large number of participants and where multiple auctions are executed simultaneously. Examples for such aggregated information are average prices and average demand. This service collects, routes, aggregates, summaries, and delivers global information in a scalable manner to any market participant.

Currency Management Service. This is a decentralized account and banking service, which provides basic functions to create accounts and transfer funds between accounts. Technically, it implements a transactional mechanism on top of a DHT to implement decentralized accounts.

5.4 Properties

The GRIMP architecture presents the following non-functional capabilities to address the requirements:

Scalability and reliability. The federated architecture allows the market place to scale to thousands of users. Therefore, it has to support as many auction servers as required and to provide a global information system:

- Auction servers are created spontaneously upon need and mechanisms with appropriate properties (computational complexity, efficiency) can be chosen on demand.

- The Market Information Service (MIS) collects, aggregates, summarizes, and delivers in a large-scale the global information of the market. This system allows the participants to be aware and to react on changes of the global market.

Configurability. In an open environment, the needed type of applications and resources cannot be fixed. Support for multiple auction mechanisms and tools to select appropriate mechanisms renders the market place configurable. As an example, an application requires both computational and storage resources and it requires combinatorial auctions. An auction mechanisms capable of trading single items is sufficient for an application requiring only a single type of resource (computations or storage).

The architecture based on the Fractal component model presents the advantage of being a single coherent solution to address design, deployment, configuration, and assembly.

6 Related and Future Work

There are systems that focus on some aspect of the problem. For instance Tycoon [5] uses auctions and a centralized bank to handle currency circulation. Shirako [4] is based on the idea that different entities, service managers, and brokers trade resource leases to self-manage applications. The SORMA project [8] designs a centralized market, sharing the idea of the Market Information Service, but without a Currency Management as payments are directly applied in real currency. In contrast, Grid4All is proposing a collection of integrated services and an implementation of that architecture to support collaborative, scalable, and adaptable Grid computing applications for everyone based on groups (virtual organizations). This work internally together and share resources for self-interest but they can also trade resources and eventually services across virtual organizations.

Although initial evaluation of each market mechanisms is promising, further work is required to include a wider range of situations. Particularly, the evaluation of the combined mechanisms is important for the regulation effect: auctions, market information, and currency. Another aspect to evaluate is the usefulness of market combinations with a pool of resources. Finally, it is also an open issue how to handle the Self-management of markets: the automatic and dynamic deployment of markets.

7 Conclusions

The Grid4All project presents an integrated vision, architecture, middleware, and applications of a public and large-scale Grid for everyone. Furthermore, the proposed market framework supports collaboration among groups of people (VO). On one side, the participants can interact and share work and computing resources among them. On the other side, they can trade resources with other groups. This public Grid self-adapts to the dynamics of the on-line world, which connects networks, computers, and people, who join and leave the community. Therefore, the system needs to self-manage to support applications with varying loads and resource needs.

The market place is a key element as it contains a collection of integrated resource brokerage mechanisms for the global and local regulation across virtual organizations. Limited experiments, which are based on simulations, show that the system has an equitable behavior in the allocation of resources among a range of similar or dissimilar participants. The ongoing development of end-user-oriented, prototyped applications over the Grid4All middleware show how the details of resource allocation and self-management can be partially abstracted out from application developers. Mainly, supporting services, which can gather automatically more resources from the VO and even push them to the VO, can acquire additional resources from the market place. Early results of the evaluation show the need for an integrated approach. Particularly, they show how market places can regulate resource allocation globally with the help from market aware services and self-managing services.

Acknowledgments This work would not be possible without the contributions from the members of the Grid4All project. This work has been partially supported by the EU FP6-2005-IST-5 034567, and the Spanish Ministry of Education with grant P2PGrid TIN2007-68050-C03-01.

References

1. Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. Int. J. High Perform. Comput. Appl., 15(3):200222 (2001)
2. Grid5000 project. Web site: http://www.grid5000.fr (Page visit: May 2008)
3. International educational and research network (iEARN). http://www.iearn.org (Page visit: May 2008)
4. Irwin, D., Chase, J., Grit, L., Yumerefendi, A., Becker, D., and Yocum, K. G. 2006. Sharing networked resources with brokered leases. In Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference, Boston, MA, 18-18 (2006)
5. Lai, K., Rasmusson, L., Adar, E., Zhang, L., and Huberman, B. A. 2005. Tycoon: An implementation of a distributed, market-based resource allocation system. Multiagent Grid Syst. 1, 3, 169-182 (2005)
6. One laptop per child (OLPC). Web site http://www.laptop.org/ (Page visit: 2008)
7. PlanetLab, an open platform for developing, deploying, and accessing planetary-scale services. Web site http://www.planet-lab.org/ (Page visit: May 2008)
8. SORMA - Self-Organizing ICT Resource Management. Web site: http://www.sorma-project.org. Page visit: (2008)
9. S. Zanikolas and R. Sakellariou, *A taxonomy of grid monitoring systems.* Future Generation Computer Systems 21, 1, 163-188 (2005)