

A 3-tier Grid Architecture and Interactive Applications Framework for Community Grids

Oscar Ardaiz, Kana Sanjeevan, and Ramon Sanguesa

Polytecnic University of Catalunya
Campus Nord, Barcelona 08034 Spain
{oardaiz,sanji}@ac.upc.es, sanguesa@lsi.upc.es

Abstract. Grids originated within the scientific community where the benefits of utilizing an infrastructure that connects and shares resources that are geographically and organizationally dispersed was first realized. Community grids have very different characteristics and requirements to that of scientific grids – the most important being the heterogeneous resource offer and the application demands of the users. The users of this type of grid does not expect to know each and everyone of the resources available, nor does he expect to provide and install his own applications. Additionally, the applications that are usually run by community grid members have one special characteristic – they are very interactive in nature and require quick responses and a communication protocol between the user and the grid. Since the 2-tier architecture of current grids is unable to deal with these special requirements, we propose a 3-tier architecture and an interactive application framework for community grids. Our architecture and framework is able to effectively overcome the problems of application deployment, resource management and interactive execution.

1 Regional Community Grids

Many communities, besides the scientific community, can benefit from an infrastructure that connects and shares resources geographically and that are organizationally dispersed, such as the grid [6]. For example, a regional community can create a grid connecting its resources and share them for the execution of applications. These types of grids are different in many aspects compared to grids belonging to the scientific community. Differences are due to the heterogeneous resource offer, and different applications demanded by its members. Firstly, a regional community grid has very heterogeneous resources: some members might provide supercomputing resources while others provide small personal computers, some may provide their computing resources only at certain hours of the day, while others provide them any time provided their users can continue to run their local applications with no loss in the quality of service. Such heterogeneous resource offer requires an adaptive resource management that may employ ‘agents’ technology [8]. Secondly, regional community grid users have different application requirements to that of the scientific community. Applications are not domain or problem specific, on

the contrary many different groups use the same applications. Application providers will develop and provide such applications as a service for an economic incentive. Moreover, regional community grid members demand applications for business, education or local administration purposes. These applications have different requirements to that of scientific applications, the most important one being: interactivity. Business, educational and community applications produce results for rapid consumption by users, as compared to scientific applications whose results have to be carefully analyzed by scientists. Current grid architectures do not support application providers effectively and are not suited for interactive applications.

1.1 Application Provision and Access Requirements

Current grids have a 2-tier architecture. Such an architecture has three main problems. Firstly, each resource is configured so that each user has permission to access such resource, therefore it will not scale with the number of users. Secondly, each user must know every single resource he has access to, it is not scalable since every community member has to maintain a list of every grid resource. And thirdly, each user must provide his own applications; though one can reuse some one else's code, he must transfer, install and execute it. It is not realistic that every community member provides his own applications, because it is a task that requires complex technical skills. Such an architecture is not appropriate for a regional community grid. What is required is a grid architecture that permits resources to be configured to scale with the number of users; that permits users to access large number of resources in a transparent manner, and that permits application developers to provide applications and give access to end users in a straightforward way.

1.2 Interactive Applications Requirements

Computational grids are designed so that grid users minimize the time it takes them to obtain results from applications being executed in a grid. A distributed computational capacity-scheduling algorithm achieves such execution time reduction by selecting resources with large computational capacity. Data grids have an equal goal, but in addition have to deal with large data transfers among different computers. Transfer time must be taken into account, and included as a parameter of the scheduling mechanisms [2]. Interactive applications have other requirements. Users require an interactive application to provide results in a very short time, values that are in the order of hundreds of millisecond. Such a requirement demands interactive grid applications to be scheduled in resources that should be near to users, so as to minimize network latency. Also, scheduling resources in advance can reduce access time. Finally, an interactive protocol is required so that users can interact with grid applications.

2 Related Work

Grid portals aim at providing grid users with a simple way to access many resources. A grid portal provides a unifying interface for grid users to gain access to all of its resources from a centralized location, thereby users do not have to discover and maintain a list of every resource available. Several portals are being developed [10]. The GRASP project has proposed a grid business model for grids and application providers, though no implementation is [3]. An architecture for running interactive applications in grids has being proposed by Kumar [7], though its requirements have not been defined, and it has not been implemented.

3 Three-Tier Grid Architecture: Application Clients, Mediators, and Resources

A grid that accomplishes the three previously mentioned objectives of: scaling with number of resources, scaling with number of users and does not require each user to provide its applications must have a three-tier architecture, as show in figure 2. Such an architecture incorporates a mediator between the application clients and resources, acting as a user portal and application deployer. Such a mediator would facilitate the following: Firstly, the application user need not be aware of which resource are being used; they need not discover and monitor individual resources. Secondly, resources should not be aware of every application user - resources are not configured to give access to individual users but to the application provider.

3.1 Application Deployment

Application deployment encompasses every action required from the time an application is created till the users gain access to it [1]. It includes resource discovery, resource reservation, resource allocation, application code transfer, and application initialization. If the number of applications is low compared to the number of users it will be more efficient to deploy application once, since it will be used many times by different clients. Therefore initial set up costs due to resource discovery and allocation, code transfer, and application initialization will be shared. The mediator of a 3-tier grid has this role of interfacing with application providers to permit them to deploy new applications.

3.2 Users Portal

Users access an application through a portal. A mediator portal controls which users are allowed to use an application. It also maintains accounting information so that clients are charged for usage of applications and resources. Application providers and resource providers each receive a share of this income.

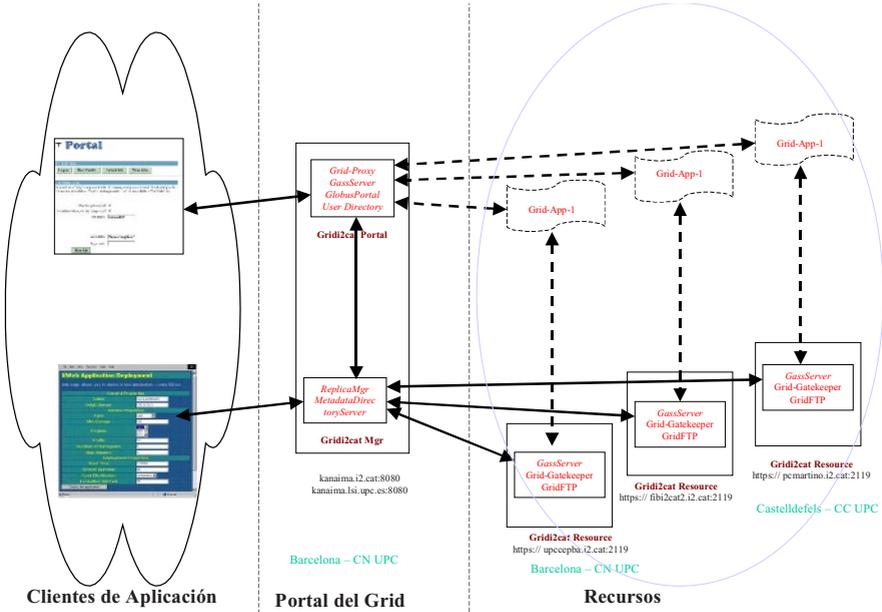


Fig. 1. 3-tier grid architecture: applications providers deploy applications once; multiple clients access the application through the portal many times

4 Interactive Applications on Grids

A grid infrastructure that provides interactive applications must meet a number of requirements: First, resources should be assigned physically close to the user so as to minimize network latency. Second, resource scheduling and application installation and start up time should be decreased as much as possible, it should preferably be made in advance. Also an interactive protocol is required so that users can interact with applications.

4.1 Proximity Based Scheduling and Advance Deployment

A grid that provides interactive applications must schedule resources which provide enough computational capacity, storage capacity and which are nearer to users. We have studied and simulated resource allocation mechanisms that take into account distance between resource and clients in a previous work. Our algorithms are also based in an economic model, which is being proposed as a basic model for resource allocation in grids. Results show that proximity based scheduling is feasible [4].

Advance deployment is an attractive option so that interactive applications do not suffer an initial delay due to application transfer and start up. However advance

reservation and allocation mechanisms have an important drawback due to non-utilization of some allocations. But if the ratio between the number of application users and number of applications is high, then due to statistical multiplexing such cost will be decreased.

4.2 Interactive Protocols

Finally the last and most important requirement for a grid that supports interactive applications is an interactive protocol between grid nodes and application clients. The grid model assumes that applications employ minimal clients as possible, i.e. computational and data grids can operate with a simple asynchronous email client. Interactive protocols however require more complex clients, though there are different levels of complexity that come into play. The least demanding client-side interactive protocols are Virtual Network Computer VNC [9] and the streams protocol. VNC is designed to provide remote graphics visualization and remote control with the least demanding client. Streams provide a reliable transport mechanism for transfer of simple data.

5 Implementation

GridCat is the research prototype of the regional community grid of Catalonia. It consists of a testbed implementing the 3-tier grid architecture, and an interactive application framework that provides basic building blocks for gridifications of interactive applications. GridCat is a regional community grid, based on a local community, Catalonia. Its resources are being provided by all kinds of institutions of the Catalan society: universities, local companies, local administrations and civic associations.

5.1 GridCat Testbed

An intermediary tier, implemented as a mediator, facilitates a grid where application providers can provide applications and application users access such applications. The implementation of such a 3-tier testbed is based on the Globus Toolkit v2 implementation. It has been extended to facilitate application providers to deploy applications, and users to access applications transparently with a mediator module that functions as a user portal and application deployer.

The Globus Resource Manager GRAM has been installed at each resources node. Resources have being configured so that certain applications are permitted for execution. To provide a high level of security, X.509 certificates and GSI Globus Security Infrastructure are being used. Only grid execution request from authorized application providers are be permitted. Application providers delegate to the mediator such certificates at application deployment time.

Several applications have been "gridified": where the functionality of an application has been divided so that it can be executed in several grid nodes. For each application a certificate has been created. Resource nodes have been configured to accept deployment and execution requests from different applications identified by a Distinguished Name DN. For example "O=Proust, CN=MoleculeSearcherApp".

Mediators implement a web based portal interface so that application users access and execute applications. Application users authenticate to portals with a username and password or with a proxy certificate provided by a proxy certificates server implementation of the Globus Portal Development Kit GPDK. Intermediates check which applications each user is allowed to use, and calls a resource manager to select which resources should be assigned to such a user.

Application providers request the mediators to deploy applications for usage by an estimated number of users. Mediators deploy applications in a number of resource nodes using GSI-FTP and GRAM Globus services. Intermediates maintain a table of nodes where an application has been deployed.

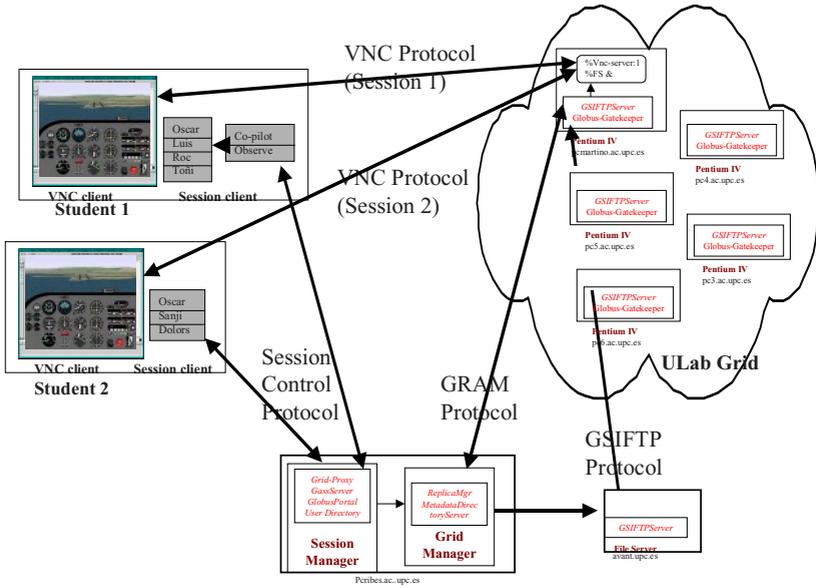


Fig. 2. Grid interactive applications framework: users access a flight simulator session that is being executed in grid nodes through VNC sessions

5.2 Grid Interactive Application Framework

We have selected two different interactive applications that are computationally intensive and that could take advantage of a grid. One of the applications is a flight simulator that has been "gridified". Flight simulators have high computational

resource requirements because three dimensional graphics rendering is computationally intensive. The simulator used is FS [5], which is an open source flight simulator that renders graphics using the OpenGL library. The other application is a molecule finder and visualization application that has also been “gridified”. Such applications search for molecules containing certain submolecules selected by users in a database, and present the results in a three dimensional view.

We have implemented a framework that provides all basic functionality for the “gridification” of future interactive applications. Such a framework consists of 3 modules: a VNC client and server, a stream client and server, and a connector that coordinates every interactive session of each application session.

A flight simulator session requires the grid mediator to bind a flight engine and a rendering engine to a VNC server, the VNC client is then connected to this VNC server. A molecules search session requires the grid mediator to bind the molecular databases and molecule search engines to a stream session, at the other end of the stream session there is a molecule rendering engine. Such an engine is bound to a VNC server.

6 Conclusions and Future Work

This work presents three contributions, first, the community grid, a distributed computational system built from very heterogeneous resources provided by a regional community, and with very specific application demand requirements. Second, the 3-tier grid architecture, that makes an efficient use of resources by employing a strategy of deploy once and run multiple times an application. And the third contribution is the framework for interactive applications running on grids.

We are currently taking measurements to evaluate the effectiveness of this approach, and also we plan to adapt this framework to Globus Toolkit version 3.

References

1. O. Ardaiz, L. Navarro "Xweb a Framework for Application Network Deployment in a Programmable Internet Service Infrastructure", 12-th Euromicro Conference on Parallel, Distributed and Network based Processing. February 2004, A coruña Spain.
2. W. H. Bell, D. G. Cameron, R. Carvajal-Schiaffino, A. Paul Millar, K. Stockinger, and F. Zini. "Evaluation of an Economy-Based File Replication Strategy for a Data Grid" In Intl. Workshop on Agent based Cluster and Grid Computing at CCGrid 2003, Tokyo, Japan, May 2003.
3. Theo Dimitrakos, Matteo Gaeta, Pierluigi Ritrovato, Bassem Serhan, Stefan Wesner, Konrad Wulf "Grid Based Application Service Provision", Euroweb 2002, Oxford, UK, Dec 2002

4. T. Eymann, M. Reinicke, O. Ardaiz, P. Artigas, F. Freitag, L. Navarro, "Decentralized Resource Allocation in Application Layer Networks", In Intl. Workshop on Agent based Cluster and Grid Computing at IEEE 4CCGrid'2003, May, 12th-15th, 2003. Tokyo, Japan.
5. "Flightgear Simulator", www.flightgear.org , 2003.
6. I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International J. Supercomputer Applications, 15(3), 2001.
7. Raj Kumar, Vanish Talwar, Sujoy Basu: "A Resource Management Framework For Interactive Grids". Middleware Workshops 2003: 238-244.
8. Luc Moreau. "Agents for the Grid: A Comparison with Web Services (Part 1: the transport layer)". Second IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002), pages 220-228, Berlin, Germany, May 2002. IEEE Computer Society.
9. Richardson T., Staord-Fraser T., Wood K.R., Hopper A. "Virtual network computing". IEEE Internet Computing, 2(1):33-38, January-February 1998.
10. M. Thomas, M. Dahan, K. Mueller, S. Mock, C. Mills. "Application Portals: Practice and Experience. Grid Computing environments": Special Issue of Concurrency and Computation: Practice and Experience. Winter 2001.