

# A Loosely Coupled Vision for Computational Grids

**Ignacio M. Llorente**, *Universidad Complutense de Madrid*  
**Rubén S. Montero**, *Universidad Complutense de Madrid*  
**Eduardo Huedo**, *Centro de Astrobiología*

The ability to have applications draw computing power from a global resource pool to achieve high performance has become a new challenge for distributed-computing and Internet technologies. Several research centers share their computing assets in grids, which dramatically increase the number of processing and storage resources applications can access. Different grid infrastructures are being deployed in the context of growing national and international research projects. The coexistence of those different infrastructures opens an interesting debate about their coordinated operation.

From our point of view, it's debatable whether some of these projects embrace the Grid philosophy, and to what extent. This philosophy, proposed by Ian Foster, posits three main requirements:<sup>1</sup>

- A grid isn't subject to centralized control.
- A grid is based on standard, open, and general-purpose interfaces and protocols.
- The interfaces and protocols provide some level of quality of service, in terms of security, throughput, response time, or the coordinated use of different resource types.

Here we assess several current grid architectures that comply, more or less, with these requirements. The tendency exists to ignore the first two requirements to get higher levels of

quality of service—mainly performance and reliability—for a given application scope. However, we believe that these two requirements are the key to a grid's success. The *loosely coupled* model provides the required trade-off to foster the diffusion of grid technologies.

## An Internet-like architecture for grids

A grid infrastructure usually comprises four layers:<sup>2</sup>

- grid applications and portals,
- user-level grid middleware,
- core grid middleware, and
- grid fabric (resources).

The two internal layers are called middleware because they connect applications with resources.

By following the Grid philosophy, we can create *loosely coupled grids*, computational environments whose architecture resembles the Internet's. This architecture is based on the *end-to-end* principle, which has fostered the spectacular development and diffusion of the Internet and, in particular, Web technologies in the past decade:

The basic argument is that, as a first principle, certain required end-to-end functions can only be performed correctly by the end-systems themselves.<sup>3</sup>

The Globus (<http://www.globus.org/>) toolkit, a de facto standard in grid computing, follows the end-to-end principle. The Globus architecture takes an hourglass approach, in which the bottom of the hourglass is the resources, the middle is the core Globus services, and the top is higher-level Globus services and applications. So, instead of succumbing to the temptation to tailor the core grid middleware to our needs (which would result in an application-specific infrastructure) or homogenize the underlying resources (which would result in a highly distributed cluster), we propose to strictly follow the end-to-end principle. Clients should have access to a wide range of resources provided through a limited, standardized set of protocols and interfaces. The Globus core grid middleware provides this set. Just as in the Internet, the protocols and interfaces are provided through TCP/IP.

In a loosely coupled grid, a limited, well-defined set of protocols and interfaces separates and interrelates the layers. Loosely coupled grids have four main characteristics: scalability,

autonomy of the multiple administration domains, dynamism, and heterogeneity.<sup>2</sup> These characteristics determine how scheduling and execution must be performed on grids. For example, scalability and autonomy prevent the deployment of centralized resource brokers, which maintain total control over client requests and resource status. In addition, job scheduling and execution must be adaptable to resource dynamics, such as changing availability, capacity, and cost. Finally, the management of resource heterogeneity implies a higher degree of complexity.

Workload management middleware is required on the client side to provide the end user with portable programming paradigms and common interfaces. On the server side, resource management software is advisable in the grid fabric to provide system administrators with tools to determine the amount of resources they're willing to devote to a grid, thereby avoiding a saturation of grid jobs. Such software will aid in the grid's expansion, leading resource owners to embrace grid technologies and share their resources with more confidence, because the performance for local users will always be assured.

Grid computing involves not only the technical challenge of constructing and deploying this vast infrastructure but also issues related to resource-sharing policies. Undoubtedly, an approach that gives administrators full control of their resources and security policies could help overcome these sociopolitical difficulties.<sup>4</sup> Moreover, the end-to-end principle minimizes the firewall configuration, which is also a welcome advance for security administrators. The more confident resource owners are, the more nodes they'll add to the grid, overcoming the typical scenario where administrators share only a small fraction of their hosts owing to their mistrust of the grid.

## **The Grid philosophy in some existing projects**

The EGEE (*Enabling Grids for E-Science*, <http://www.eu-egee.org>) project is creating a larger production-level grid infrastructure, which provides a new level of performance and reliability. Participating organizations must adhere to a restrictive set of requirements. EGEE defines the user-level grid middleware, the core grid middleware, and the grid fabric as being tightly related. EGEE uses the LCG, <http://lcg.web.cern.ch> (*Large Hadron Collider Computing Grid*) middleware, LCG-2. This limits heterogeneity somewhat because LCG-2 has a fixed configuration for clusters. The scalability of its deployment is also limited because the middleware should be installed on the compute nodes, which should have network connectivity. The LCG focuses mainly on particle physics applications connected with CERN (the European Organization for Nuclear Research). However, the new EGEE middleware, gLite, <http://www.glite.org>, should overcome some of these limitations.

The IRISGrid initiative's main objective is to create a stable national grid infrastructure. IRISGrid's first mission is to provide the necessary protocols, procedures, and guidelines for

creating a research grid in Spain. This initiative seeks to link geographically distant resources so that all the interested groups can access a research test bed. IRISGrid defines only the core grid middleware, requiring different user-level grid middleware. The first version of the IRISGrid test bed is based only on Globus, and it has been widely used through the GridWay framework, <http://www.gridway.org>.<sup>5</sup> In this sense, GridWay is user-level grid middleware, which uses Globus as the core grid middleware. So, GridWay could be used in any grid infrastructure based on Globus as the core grid middleware. On the server side, a resource performance manager<sup>6</sup> lets system administrators determine the amount of resources to devote to the grid.

The architectural requirements of the NorduGrid project (<http://www.nordugrid.org>) are close to those of IRISGrid. However, NorduGrid isn't only based on the Globus basic services. It defines the user-level and the core grid middleware, leaving some flexibility in the grid fabric. Nevertheless, it presents some interesting benefits such as scalability and no single point of failure, and resources aren't necessarily dedicated to grid jobs but are under their owners' control and have few site requirements.<sup>7</sup>

The coexistence of several projects, each with its own middleware developments, adaptations, or extensions, gives rise to the idea of coordinated harnessing of resources (from the end-user perspective) or of contributing the same resource to more than one project (from the resource owner perspective). One approach could be to develop gateways between different middleware implementations.<sup>8</sup> Another approach, more in line with the Grid philosophy, is to develop client tools that can adapt to different middleware implementations. The good news is that nearly all current projects use Globus as the core grid middleware. So, this could lead to a shift of functionality from resources to brokers or clients, letting users access resources in a standard way, making resources easier to share among organizations and projects.

## Conclusion

Loosely coupled grids allow straightforward resource sharing because resources are accessed and exploited through de facto standard protocols and interfaces, similar to the Internet's early stages. The loosely coupled model allows easier, scalable, and compatible deployment at a lower global performance expense, compared to a tightly coupled alternative. Because we can apply the end-to-end principle on both the client and server sides, we can build loosely coupled grid environments based only on Globus services and user-level middleware, while obtaining acceptable quality of service for both grid and local users, and fair resource sharing.

## References

1. I. Foster, "What Is the Grid? A Three Point Checklist," <http://www.gridtoday.com/02/0722/100136.html>, *GRIDtoday*, vol. 1, no. 6, 2002.
2. M. Baker, R. Buyya and D. Laforenza, "Grids and Grid Technologies for Wide-Area Distributed Computing," *Software: Practice and Experience*, vol. 32, no. 15, 2002, pp. 1437–1466.
3. E.B. Carpenter, *Architectural Principles of the Internet*, IETF RFC 1958, June 1996.
4. J.M. Schopf and B. Nitzberg, "Grids: The Top Ten Questions," *Scientific Programming*, vol. 10, no. 2, 2002, pp. 103–111.
5. E. Huedo, R.S. Montero and I.M. Llorente, "A Framework for Adaptive Execution on Grids," *Software: Practice and Experience*, vol. 34, no. 7, 2004, pp. 631–651.
6. O. San José et al., "Resource Performance Management on Computational Grids," *Proc. 2nd Int'l Symp. Parallel and Distributed Computing (ISPDC 03)*, IEEE CS Press, 2003, pp. 215–221.
7. O. Smirnova et al., "The NorduGrid Architecture and Middleware for Scientific Applications," *Proc 2003 Int'l Conf. Computer Science (ICCS 03)*, LNCS 2657, Springer-Verlag, 2003, pp. 264–273.
8. R.J. Allan et al. *Building Overlapping Grids*, tech. report, Univ. of Cambridge, Oct. 2003.  
(<http://tyne.dl.ac.uk/ETF/public/Deployment/Building%20Overlapping%20Grids.pdf>).



**Ignacio M. Llorente** is an associate professor of computer architecture and technology in the Department of Computer Architecture and System Engineering at the Universidad Complutense de Madrid and is a senior scientist in the Advanced Computing Laboratory at the Centro de Astrobiología, associated with the NASA Astrobiology Institute. He's also a member of the GridWay team and participates in the EGEE (Enabling Grids for E-Science) project. Contact him at [llorente@dacya.ucm.es](mailto:llorente@dacya.ucm.es).



**Rubén S. Montero** is an associate professor of computer architecture and technology in the Department of Computer Architecture and System Engineering at the Universidad Complutense de Madrid. He's also a member of the GridWay team and participates in the EGEE (Enabling Grids for E-Science) project. Contact him at [rubensm@dacya.ucm.es](mailto:rubensm@dacya.ucm.es).



**Eduardo Huedo** is a scientist in the Advanced Computing Laboratory at the Centro de Astrobiología, associated with the NASA Astrobiology Institute. He's also a member of the GridWay team and participates in the EGEE (Enabling Grids for E-Science) project. Contact him at [huedoce@inta.es](mailto:huedoce@inta.es).

**Cite this article:** Ignacio M. Llorente, Rubén S. Montero, and Eduardo Huedo, "A Loosely Coupled Vision for Computational Grids," *IEEE Distributed Systems Online*, vol. 6, no. 5, 2005.