

Toward a Synergy Between P2P and Grids

Domenico Talia and Paolo Trunfio

University of Calabria, Italy

Peer-to-peer (P2P) networks and grids are distributed computing models that enable decentralized collaboration by integrating computers into networks in which each can consume and offer services. P2P is a class of self-organizing systems or applications that takes advantage of distributed resources – storage, processing, information, and human presence – available at the Internet's edges. A grid is a geographically distributed computation platform comprising a set of heterogeneous machines that users can access through a single interface.

Both are hot research topics because they offer promising paradigms for developing efficient distributed systems and applications. Unlike the classic client-server model, in which roles are well separated, P2P and grid networks can assign each node a client or server role according to the operations they are to perform on the network – even if some nodes act more as server than as client in current implementations.

In analyzing both models, we discover that grids are, in essence, P2P systems. Although many aspects of today's grids are based on hierarchical services, this is an implementation detail that should be removed in the near future. As grids used for complex applications increase from tens to thousands of nodes, we should decentralize their functionalities to avoid bottlenecks. The P2P model could thus help to ensure grid scalability: designers could use the P2P philosophy and techniques to implement nonhierarchical decentralized grid systems.

In spite of current practices and thoughts, the grid and P2P models share several features and have more in common than we perhaps generally recognize. As Ian Foster and Adriana Iamnitchi point out (dsl.cs.uchicago.edu), a broader recognition of key commonalities could accelerate progress in both communities. It is time to consider how to integrate these two models. A synergy between the two research communities, and the two computing models, could start with identifying the similarities and differences between them.

Basics

In the past few years, P2P has attracted enormous media attention and gained popularity by supporting two main classes of applications:

- *file sharing*, in which peers share files with each other (Napster and Gnutella for music, for example)
- *highly parallel computing*, in which an (inherently) parallel application runs on available nodes (SETI@home and FightAIDS@home, for example).

Apart from these well-known systems, the P2P model is emerging as a new distributed paradigm because of its potential to harness the computing, storage, and communication power of hosts in the network to make their underutilized resources available to others. P2P

shares this goal with the Grid, which was designed to provide access to remote computing resources for high-performance applications, data-intensive applications, or both.

Although originally intended for advanced scientific applications, grid computing has emerged as a paradigm for coordinated resource sharing and problem solving in dynamic, multi-institutional, virtual organizations in industry and business. Grid computing can be seen as an answer to drawbacks such as overloading, failure, and low QoS, which are inherent to centralized service provisioning in client-server systems. Such problems can occur in the context of high-performance computing, for example, when a large set of remote users accesses a supercomputer.

Grid nodes typically make their own resources available at the same time they are accessing resources on other nodes. The grid model thus removes the definite distinction between client and server machines. However, current grid environments delegate specific management or coordination functions to certain nodes that are required to take "major responsibility." Some recently developed P2P systems also require nodes to act as servers, at least when joining the network.

P2P comprises several kinds of applications with different design goals, such as anonymity (typically in file-sharing applications), scalability (typically in highly parallel computing applications), or availability (in both application classes). Moreover, P2P systems are based on several different designs:

- systems such as Napster use centralized resource indexes,
- systems such as Gnutella use flooding-based search,
- some experimental systems such as Gridella use structures with distributed resource indexes, and
- hybrid networks, such as the super-peer model (described later), combine the P2P and client-server models.

As mentioned before, the identification of similarities and differences between grid and P2P systems is a good starting point for finding a convergence.

Similarities and Differences

In analyzing the P2P and grid models, we must consider several significant aspects and issues. Here we discuss some of the main issues that determine features of distributed computing models. The techniques that the P2P and grid models use to handle those issues are key to finding a common foundation.

Security

Security is a central theme in grids, and several efforts are devoted to integrating relevant mechanisms for authentication, authorization, integrity, and confidentiality in grid platforms. Nevertheless, such mechanisms are designed mainly for "closed communities," in which

designers have devoted some effort to letting users participate without accounts or trust relationships. By their nature, such security mechanisms allow anonymity of neither users nor resources.

In contrast, P2P systems originate in "open communities," in which users share more generic goals (such as retrieving music from the Internet), rather than specific objectives (such as participating in high-energy physics simulations). For this reason, security mechanisms in the most widespread P2P systems generally don't address authentication and content validation, but rather offer protocols that assure anonymity and censorship resistance. Although the two models currently handle security differently, it should be interesting to analyze how to exploit the approaches to create a security model for P2P grids.

Connectivity

Grids generally include powerful machines that are statically connected through high-performance networks with high levels of availability. On the other hand, the number of accessible nodes is generally low because access to grid resources is bonded to rigorous accounting mechanisms.

Conversely, P2P systems are composed mainly of common desktop computers that are connected intermittently to the network, remaining available for a limited time with reduced reliability. The number of nodes connected in a P2P network at a given time is much greater than in a grid. Thus, the grid connectivity approach is still too stiff for new nodes and user access and accounting; it could benefit from the more flexible connectivity models used in P2P networks today.

Access Services

Access to remote resources was the main motivation for building grids, and it remains the primary goal today. Grid toolkits provide secure services for submitting batch jobs or executing interactive applications on remote machines; they also include mechanisms for efficiently sharing and moving data across nodes.

Current P2P systems do not support mechanisms for explicitly allocating remote cycles and storage, but they do provide protocols for sharing and exchanging data among nodes. P2P job-submission models and P2P job scheduling might thus be very attractive topics for research into applying the P2P approach to grid scheduling and job management.

Resource Discovery and Presence Management

Resource discovery in grid environments is based mainly on centralized or hierarchical models. In the Globus Toolkit (www.globus.org/toolkit), for instance, a user or an application can directly gain information about a given node's resources by querying a server application running on it or running on a node that retrieves and publishes information about a given organization's node set. Because such information systems are built to address the requirements of organizational-based grids, they do not deal with more dynamic, large-scale

distributed environments, in which useful information servers are not known a priori. The number of queries in such environments quickly makes a client-server approach ineffective.

Resource discovery includes, in part, the issue of presence management – discovery of the nodes that are currently available in a grid – because global mechanisms are not yet defined for it. On the other hand, the presence-management protocol is a key element in P2P systems: each node periodically notifies the network of its presence, discovering its neighbors at the same time. Future grid systems should implement a P2P-style decentralized resource discovery model that can support grids as open resource communities.

Fault Tolerance

The dynamic nature of grids necessitates some level of fault tolerance – especially for highly distributed code, such as parameter-sweep applications, which can fork numerous similar, independent jobs on many nodes.

Beyond simple checkpointing and restarting, reliability and fault tolerance are largely unexplored in grid models and tools. The Globus information system allows fault detection, for instance, but developers must implement fault tolerance at the application level. For greater reliability, designers of fault-tolerance mechanisms and policies for grids should consider using decentralized P2P algorithms, which avoid centralized services that can represent critical failure points.

Where We Should Go

Despite the interest in P2P and grid networks, few noteworthy research efforts are currently devoted to finding commonalities and synergies between them. In a significant exception, Fox and colleagues have sketched a P2P architecture for grid-connected resources (www.communitygrids.iu.edu), but much more remains to be done by members of both communities.

We believe a P2P approach is needed both to

- implement grid tools and services, and
- design and develop grid applications that must access and coordinate remote resources and services.

Two core Globus Toolkit components – the monitoring and discovery service (MDS) and the replica management service – could be effectively redesigned using a P2P approach, for example. If we view current grids as federations of smaller grids managed by diverse organizations, we can rethink the Globus MDS for a large-scale grid by adopting the super-peer network model (www-db.stanford.edu/~byang/pubs/superpeer.pdf). In this approach, each super peer operates as a server for a set of clients and as an equal among other super peers. This topology provides a useful balance between the efficiency of centralized search and the autonomy, load balancing, and robustness of distributed search. In a grid information service based on the super-peer model, each participating organization would configure one or

more of its nodes to operate as super peers. Nodes within each organization would exchange monitoring and discovery messages with a reference super peer, and super peers from different organizations would exchange messages in a P2P fashion.

Grid applications should be designed according to a decentralized model. This can require additional effort to develop because of the current lack of P2P-grid middleware, but P2P-grid tools and services could greatly simplify such tasks in the future we envision.

Aligning Technologies

The grid community recently initiated a development effort to align grid technologies with Web services: the Open Grid Services Architecture (OGSA) lets developers integrate services and resources across distributed, heterogeneous, dynamic environments and communities. The OGSA model adopts the Web Services Description Language (WSDL) to define the concept of a *grid service* using principles and technologies from both the grid and Web services communities. Web services and the OGSA both seek to enable interoperability between loosely coupled services, independent of implementation, location, or platform.

OGSA provides an opportunity to integrate P2P and the Grid. The architecture defines standard mechanisms for creating, naming, and discovering persistent and transient grid-service instances. It will be an interesting challenge to determine how to use OGSA-oriented grid protocols to build P2P applications. By implementing service instances in a P2P manner within such a framework, developers can provide P2P service configuration and deployment on the grid infrastructure. A peer could thus invoke a grid service by exchanging a specified sequence of messages with a service instance, which might invoke another grid service published by another peer through an associated grid service interface.

Developers and users could exploit the many contact points between P2P and grid networks by recognizing P2P's relevance to corporations and public organizations rather than viewing it as just a home computing technology. They also could exploit P2P protocols and models to face grid-computing issues such as scalability, connectivity, and resource discovery. A synergy between P2P and grids could lead to new highly distributed systems in which each computer contributes to solving a problem or implementing a system while also using services offered by other computers in the network. Enterprises, public institutions, and private companies could find it both useful and profitable to develop distributed applications on a world-wide Grid.