

What is the Grid? A Three Point Checklist

Ian Foster

Argonne National Laboratory & University of Chicago
foster@mcs.anl.gov

July 20, 2002

The recent explosion of commercial and scientific interest in the Grid makes it timely to revisit the question: What is the Grid, anyway? I propose here a three-point checklist for determining whether a system is a Grid. I also discuss the critical role that standards must play in defining *the* Grid.

The Need for a Clear Definition

Grids have moved from the obscurely academic to the highly popular. We read about Compute Grids, Data Grids, Science Grids, Access Grids, Knowledge Grids, Bio Grids, Sensor Grids, Cluster Grids, Campus Grids, Tera Grids, and Commodity Grids. The skeptic can be forgiven for wondering if there is more to the Grid than, as one wag put it, a “funding concept”—and, as industry becomes involved, a marketing slogan. If by deploying a scheduler on my local area network I create a “Cluster Grid,” then doesn’t my Network File System deployment over that same network provide me with a “Storage Grid?” Indeed, isn’t my workstation, coupling as it does processor, memory, disk, and network card, a “PC Grid?” Is there any computer system that isn’t a Grid?

Ultimately the Grid must be evaluated in terms of the applications, business value, and scientific results that it delivers, not its architecture. Nevertheless, the questions above must be answered if Grid computing is to obtain the credibility and focus that it needs to grow and prosper. In this and other respects, our situation is similar to that of the Internet in the early 1990s. Back then, vendors were claiming that private networks such as SNA and DECNET were part of the Internet, and others were claiming that every local area network was a form of Internet. This confused situation was only clarified when the Internet Protocol (IP) became widely adopted for both wide area and local area networks.

Early Definitions

Back in 1998, Carl Kesselman and I attempted a definition in the book “The Grid: Blueprint for a New Computing Infrastructure.” We wrote:

“A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.”

Of course, in writing these words we were not the first to talk about on-demand access to computing, data, and services. For example, in 1969 Len Kleinrock suggested presciently, if prematurely:

“We will probably see the spread of ‘computer utilities’, which, like present electric and telephone utilities, will service individual homes and offices across the country.” [\[link\]](#)

In a subsequent article, “[The Anatomy of the Grid](#),” co-authored with Steve Tuecke in 2000, we refined the definition to address social and policy issues, stating that Grid computing is concerned with “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.” The key concept is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose. We noted:

“The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a *virtual organization*.”

We also spoke to the importance of standard protocols as a means of enabling interoperability and common infrastructure.

A Grid Checklist

I suggest that the essence of the definitions above can be captured in a simple checklist, according to which a Grid is a system that:

- 1) *coordinates resources that are not subject to centralized control ...*
(A Grid integrates and coordinates resources and users that live within different control domains—for example, the user’s desktop vs. central computing; different administrative units of the same company; or different companies; and addresses the issues of security, policy, payment, membership, and so forth that arise in these settings. Otherwise, we are dealing with a local management system.)

- 2) *... using standard, open, general-purpose protocols and interfaces ...*
(A Grid is built from multi-purpose protocols and interfaces that address such fundamental issues as authentication, authorization, resource discovery, and resource access. As I discuss further below, it is important that these protocols and interfaces be *standard* and *open*. Otherwise, we are dealing with an application-specific system.)

- 3) ... *to deliver nontrivial qualities of service.* (A Grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts.)

Of course, the checklist still leaves room for reasonable debate, concerning for example what is meant by “centralized control,” “standard, open, general-purpose protocols,” and “qualities of service.” I speak to these issues below. But first let’s try the checklist on a few candidate “Grids.”

First, let’s consider systems that, according to my checklist, do not qualify as Grids. A cluster management system such as Sun’s Sun Grid Engine, Platform’s Load Sharing Facility, or Veridian’s Portable Batch System can, when installed on a parallel computer or local area network, deliver quality of service guarantees and thus constitute a powerful Grid resource. However, such a system is not a Grid itself, due to its centralized control of the hosts that it manages: it has complete knowledge of system state and user requests, and complete control over individual components. At a different scale, the Web is not (yet) a Grid: its open, general-purpose protocols support access to distributed resources but not the coordinated use of those resources to deliver interesting qualities of service.

On the other hand, deployments of multi-site schedulers such as Platform’s MultiCluster can reasonably be called (first-generation) Grids—as can distributed computing systems provided by Condor, Entropia, and United Devices, which harness idle desktops; peer-to-peer systems such as Gnutella, which support file sharing among participating peers; and a federated deployment of the Storage Resource Broker, which supports distributed access to data resources. While arguably the protocols used in these systems are too specialized to meet criteria #2 (and are *not*, for the most part, open or standard), each does integrate distributed resources in the absence of centralized control, and delivers interesting qualities of service, albeit in narrow domains.

The three criteria apply most clearly to the various large-scale Grid deployments being undertaken within the scientific community, such as the distributed data processing system being deployed internationally by “Data Grid” projects (GriPhyN, PPDG, EU DataGrid, iVDGL, DataTAG), NASA’s Information Power Grid, the Distributed ASCI Supercomputer (DAS-2) system that links clusters at five Dutch universities, the DOE Science Grid and DISCOM Grid that link systems at DOE laboratories, and the TeraGrid being constructed to link major U.S. academic sites. Each of these systems integrates resources from multiple institutions, each with their own policies and mechanisms; uses open, general-purpose (Globus Toolkit) protocols to negotiate and manage sharing; and addresses multiple quality of service dimensions, including security, reliability, and performance.

The Grid: The Need for InterGrid Protocols

My checklist speaks to what it means to be “a Grid,” yet the title of this article asks what is “*the* Grid.” This is an important distinction. The Grid vision requires protocols (and interfaces and policies) that are not only open and general-purpose but also *standard*. It is standards that allow us to establish resource-sharing arrangements dynamically with *any* interested party and thus to create something more than a plethora of balkanized, incompatible, non-interoperable distributed systems. Standards are also important as a means of enabling general-purpose services and tools.

In my view, the definition of standard “InterGrid” protocols is the single most critical problem facing the Grid community today. Fortunately, we are making good progress. On the standards side, we have the increasingly effective [Global Grid Forum](#). On the practical side, six years of experience and refinement have produced a widely used de facto standard, the open source [Globus Toolkit](#). And now, within the Global Grid Forum we have major efforts underway to define the [Open Grid Services Architecture \(OGSA\)](#), which modernizes and extends Globus Toolkit protocols to address emerging new requirements, while also embracing Web services. Companies such as [IBM](#), [Microsoft](#), [Platform](#), [Sun](#), [Avaki](#), [Entropy](#), and [United Devices](#) have all expressed strong support for OGSA. I hope that in the near future, we will be able to state that for an entity to be part of *the* Grid it must implement OGSA InterGrid protocols, just as to be part of the Internet an entity must speak IP (among other things). Both open source and commercial products will interoperate effectively in this heterogeneous, multi-vendor Grid world, thus providing the pervasive infrastructure that will enable successful Grid applications.

Thanks for reading this far. I expect to be writing further columns for Grid Today, so please feel free to contact me if there are issues that you would like to see raised in this forum.