Running Head: MATURITY OF GRID COMPUTING

MetaTech Consulting, Inc.

White Paper

An essay on the maturity of grid computing

Jim Thomas

October 5, 2003

An essay on the Maturity Grid Computing

Computational grid was first used as a term by Foster and Kesselman (1998) where they defined it as the "hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities" (p. 3). Their notion of the computational grid as an infrastructure for computing was founded on an analogy of the power grid that had provided ready access to electricity in the early years for the twentieth century. Since publishing that early influential work, the authors, together with others in the research community, have furthered the notions of the computational grid through a number of subsequent papers.

Architectures for computational grids have been envisioned and have been realized through large scale functional research demonstrations. This has led to greater understanding of the viability of the concepts and has establish a new computing paradigm – *grid computing* – as being sufficiently distinct from traditional computing and having promise for practical application in the near future. The purpose of this essay is to provide a concise summary of the state of maturity of grid computing based on a review of available peer-reviewed papers. The fundamentals of the computational grid are provided first through a brief discussion of its structure and function. This is followed by a concise treatment of the grid computing maturation roadmap to include the computational grid as an enabling infrastructure. A discussion of ongoing related research activities and a discussion of issues likely to challenge its continued maturity will be provided. The paper will conclude by summarizing the maturity of grid computing as an emerging paradigm for computing and its interrelationship to existing computing paradigms.

Fundamentals of Grid Computing

The seminal work on grid computing was authored by Foster and Kessleman and published in 1998. They employed an analogy of electrical power grid to illustrate their notion of a computing infrastructure that would bring to bear appropriate computing resources for the computational problem at hand. Their approach, discussed in the paragraphs below, would provide an increase in computational power of five orders of magnitude to users within a decade. This was predicated on innovations being made in the areas of a) technology improvements, b) increase in demand-drive access to computational power, c) increased utilization of idle capacity, d) greater sharing of computational results, and e) new problem solving techniques and tools. Furthermore, they asserted “it is the combination of dependability, consistency and pervasiveness that will cause computational grids to have a transforming effect on how computation is performed and used” (Foster & Kesselman, 1998, p. 3). More recent comparisons (Chetty & Buyya, 2002) largely support the comparison the computational grid to the electrical power grid while drawing attention to the fact that the computational grid lacks the regulatory oversight imposed on the power grid.

The problem at which grid computing is targeted is “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations” (Foster et al., 2001, p.2). Central to the problem as stated is the notion of the *Virtual Organization* (VO) which represented a set of entities that are interested in participating to a common set of sharing rules applicable to given resources (e.g. CPUs, sensors, data, etc.). There is no necessary assumption of common geographic presence for any of the entities or resources. Though this problem does overlap to some extent with other technology trends (i.e. peer-to-peer, enterprise computing, distributed computing, and the internet), it is unique in that the other trends do not aspire to address the problem in whole.

Structure

Four views of the grid computing infrastructure that describe its structure are a) protocols, b) standards, c) application programming interfaces, and d) software development kits. The protocols provide the foundation for entities within the VO and the available resources to negotiate, establish, and maintain the sharing relationship (Foster et al., 2001). It is through an standards-based open architecture that it will be possible to achieve extensibility, interoperability, portability, and code sharing. Development and adoption of new protocols and standards are necessary to address necessary attributes of the grid that are absent from other computing trends such as quality of service and dynamic optimization of resources.

Architecturally, the grid infrastructure has been described in a layered model akin to the Internet protocol model (Foster et al., 2001). Figure 1 provides a comparative illustration of the grid protocol architecture to the Internet model. The layers of the grid protocol architecture are named a) fabric, b) connectivity, c) resource, d) collective and e) application. The fabric layer is the most abstract layer and has the most direct interfaces with the concrete resources on the grid while the application layer is necessarily the most transparent layer to developers and users. The need for brevity precludes even a modest treatment of each of the layers. Interested readers are encouraged to reference the cited works for a complete understanding.

Function

Foster et al. (2002) leveraged the *web services* construct in articulating the function of the grid computing environment. They describe *grid services* functioning together by dynamically aligning themselves in support of needs as they are stated. Grid services, as idealized, will differ from web services in that they will not be tightly coupled with a single (deployed) platform. That is the code that provides a given set of functionality will be mobile. As such, the identification of the software to be applied to a given processing need will be made independent of the selection of suitable hardware. The code will then be transported to the platform and executed in support of the task (e.g. workflow). Central to such large scale distribution of computing is *quality of service (QoS).* There is a need to provision resources (e.g. CPUs, sensors, data, and communications bandwidth) according to some quantifiable schedule. One proposed QoS scheme recognizes distinction between categories of applications users will need and the characteristics of their computing needs (Table 1).

Grid Computing Roadmap

Related Research

The preponderance of the materials available on the subject suggest that grid computing is an extension of many of the modern distributed computing schemes that have matured over the recent past (e.g. peer to peer, distributed, enterprise, etc). Foster (2001) enumerates four areas of technology with which grid computing is necessarily coupled. Those are: a) world wide web technologies such as the TCP/IP, HTTP, and SOAP protocols and HTML and XML languages, b) applications and storage service providers, c) enterprise computing systems, and d) internet and peer-to-peer computing. It is suggested that grid computing will mature by exploiting these technologies rather than supplanting them. So tightly inter-related are grid computing and peer-to-peer computing that it has been suggested that grid computing is simply a specialization of peer-to-peer (Loo, 2003) though this is not generally supported in the literature. Table 2 illustrates how many of the functions of the inter-related technologies apply to the layered model of grid computing.

Challenges

The computational grid is an infrastructure. Distinguishing it as an infrastructure imbues it with expectations of levels of quality of services, availability, reliability, and standardization exceeding what has yet been realized through the related computing paradigms. Extending the notion of infrastructure and utility still further, Yang, Gou, Galis, Yang, & Liu (2003) have added the concept of *resource on demand* which connotes the ability of the utility to automatically provision the available resources to accomplish the aforementioned objectives.

Much has been accomplished in furthering the concepts of grid computing, however many challenges remain. Though not comprehensive, an enumeration of challenges facing the continued maturity of grid computing includes a) standardization of interface and interchange protocols, b) development of robust security architectures that allow for trusted access to shared resources as well as for execution of mobile code, c) development of schemes for provisioning of resources, d) development of schemes for ensuring quality of service, e) installation of dynamically configurable network switching and routing devices (allowing auto-configuration based on mobile code), f) shift to a culture of resource sharing (e.g. allowing others to use otherwise idle CPU cycles), and g) synthesis of data models, taxonomies, and ontologies (Chervenak, Foster, Kesselman, Salisbury, & Tuecke, 1999) supporting universal access and use of data. Much work remains.

Conclusions

As demonstrated though research activities such as that conducted by the Globus Alliance (http://www.globus.org) are rapidly advancing the concepts of grid computing. Success in deploying computational grids to solve complex, data-intensive problems strengthen the interest in, commitment to, and understanding of this maturing computing paradigm. While it is certain that real, measurable improvements have resulted from the efforts directed toward grid computing, it is too early to determine if the prediction to provide an increase of computing capability of five orders of magnitude (Foster, 1998) within a decade will be realized.

References

Chervenak, A., Foster, I., Kesselman, C., Salisbury,C., & Tuecke, S. (1999). *The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets.* Retrieved October 4, 2003, from http://www.globus.org/documentation/incoming/JNCApaper.pdf.

Chetty, M. & Buyya, R. (2002). Weaving computational grids: How analogous are they with electrical grids? *Computing in Science & Engineering*, 4 (4), 61 – 71.

Foster, I. & Kesselman, C. (Eds.) (1998). *The grid: Blueprint for a new computing infrastructure*. San Francisco, CA: Morgan Kaufmann Publishing.

Foster, I., Kesselman, C., Nick, J. & Tuecke, S. (2002). *The physiology of the grid: An open grid services architecture for distributed systems integration*. Retrieved October 4, 2003, from http://www.globus.org.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*,15 (3), 1-25.

Loo, A. W. (September, 2003). The future of peer-to-peer computing: An economical method for pumping up computing power by taping into P2P systems using web server technologies. *Communications of the ACM*, 46 (9), 57 – 61.

Yang, K., Guo, X., Galis, A., Yang, B., & Liu, D. (2003). Towards efficient resource on-demand in grid computing. *AMC SIGOPS Operating Systems Review*, 37 (2), 37 – 43.

Figure 1

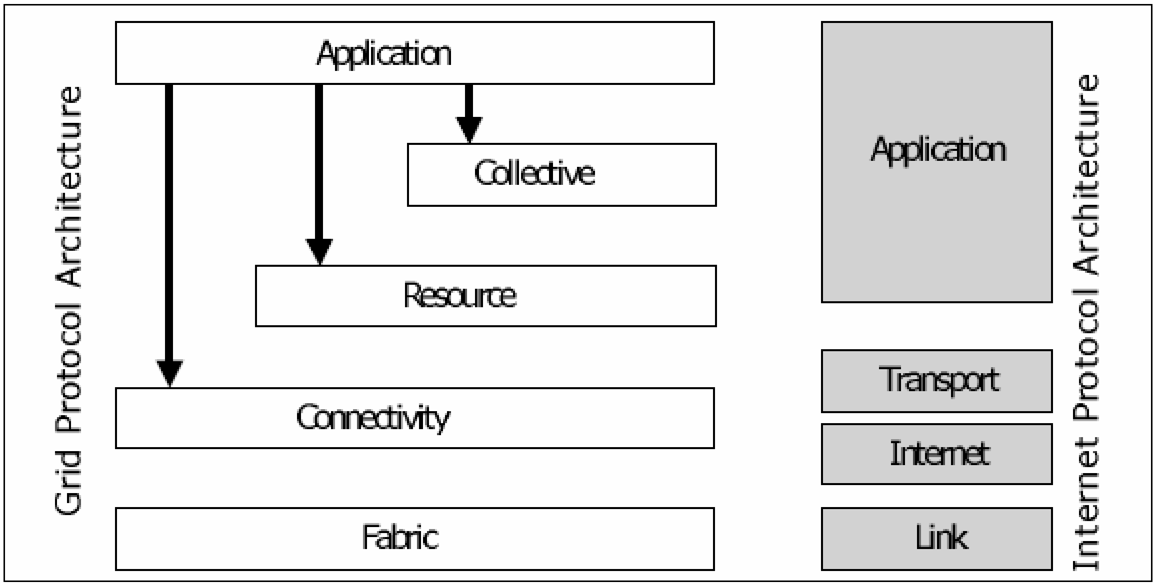

Figure 1. Comparison of the Grid and Internet protocol architectures. (Foster, Kesselman, & Tuecke, 2001, p. 7).

Table 1

Category	Examples	Characteristics
Distributed supercomputing	DIS Stellar dynamics Ab initio chemistry	Very large problems needing lots of CPU, memory, etc.
High throughput	Chip design Parameter studies Cryptologic problems	Harness many otherwise idle resources to increase aggregate throughput
On demand	Medical instrumentation Network-enabled solvers Cloud detection	Remote resources integrated with local computation, often for bounded amount of time
Data intensive	Sky survey Physics data Data assimilation	Synthesis of new information from many or large data sources
Collaborative	Collaborative design Data exploration Education	Support communication or collaborative work between multiple participants.

Table 1. Classes of grid applications. (Foster & Kesselman, 1998, p. 6).

Table 2

	Multidisciplinary Simulation	Ray Tracing
Collective (application-specific)	Solver coupler, distributed data archiver	Checkpointing, job management, failover, staging
Collective (generic)	Resource discover, resource brokering, system monitoring, community authorization, certificate revocation	
Resources	Access to computation, access to data; access to information about system structure, state, performance	
Connectivity	Communication (IP), service discovery (DNS), authentication, authorization, delegation	
Fabric	Storage systems, computers, networks, code repositories, catalogs	

Table 2. Examples of grid services in the layered model. (Foster & Kesselman, &Tuecke, 2001, p. 14).