Grid Computing: Enabling a Vision for Collaborative Research

Gregor von Laszewski

Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL 60439
gregor@mcs.anl.gov, http://www.mcs.anl.gov/~gregor

Abstract. In this paper we provide a motivation for Grid computing based on a vision to enable a collaborative research environment. Our vision goes beyond the connection of hardware resources. We argue that with an infrastructure such as the Grid, new modalities for collaborative research are enabled. We provide an overview showing why Grid research is difficult, and we present a number of management-related issues that must be addressed to make Grids a reality. We list projects that provide solutions to subsets of these issues.

1 Introduction

The Grid approach is an important development in the discipline of computer science and engineering. It is making rapid progress on several levels, including the definition of terminology, the design of an architecture and framework, the application in scientific problems, and the creation of physical instantiations of Grids on a production level. In this paper we outline important management issues influencing current Grid computing efforts. A strong overlap between past, current, and future research in other disciplines influences this new area and makes answers to some of the questions complex. Nevertheless, we hope that readers will be encouraged to contribute to the ongoing Grid efforts, either by enhancing the infrastructure or by using it to provide solutions for applied parallel computing.

The paper is structured as follows. First, we motivate the creation of Grids based on a vision for a scientific collaboratory. We list a number of management issues that need to be addressed to make a Grid a reality. Next we list several tools that address a subset of these problems. We list a number of production Grids that can be used to prototype Grid based applications. Last, we summarize future work.

1.1 Vision for an Open International Scientific Collaboratory

First, we identify what motivates us to develop a Grid approach. We simplify our presentation by providing an example for a particular scientific domain, meteorology. The ingredients for an accurate weather prediction are a *model* allowing calculations based on *observations* for the upcoming weather (see Fig. 1).

L. F. Richardson expressed the first modern vision for numerical weather prediction in 1922. Within two decades, the first prototype of a predictive system was implemented by von Neuman, Charney, and others on the first generation of computers [21]. With

the increased power of computers, numerical weather prediction became a reality in the 1960s and initiated a revolution in the field that we are still experiencing today.

But what vision promotes a Grid-like scenario for weather prediction?

In contrast to these early weather prediction models, today the scientific *community* understands that complex chemical processes and their interactions with land and sea have to be considered. The information based on observations is still incomplete, and international efforts are under way to improve this situation. Thus, we see that one of the ingredients for a successful weather forecast is a sophisticated sensor network. Another important ingredient is accurate models. A group of interdisciplinary scientists is necessary to derive such models while sharing the intellectual property of their contributions with the community. A third ingredient is high-end distributed computers. We believe that although today's supercomputers offer enormous power, predictive climate and weather modeling will require distributed computing, exploiting diverse computational resources at dispersed locations.

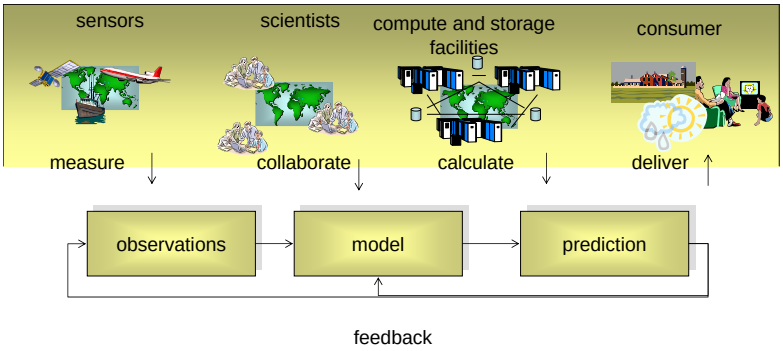

Fig. 1. Weather forecasting is a complex process that requires a complex infrastructure. The result is delivered appropriately to consumers.

Thus, we have identified the need for an *infrastructure* that allows us to create a dynamic, dispersed set of sensor, data, compute, collaboration, and delivery networks. Such a Grid-like infrastructure is now being developed, enabling an international community to formulate forecasts as a collaborative and interdisciplinary effort while providing proper delivery to consumers. In summary, the *Grid approach promotes a vision for sophisticated international scientific and business oriented collaboratories.*

1.2 Historical Perspective: Making the Vision a Reality

When we look at why it is now possible to develop very sophisticated forecast models, we see an increase in understanding, capacity, capability, and accuracy on all levels of our infrastructure.

Clearly, technology has advanced dramatically.. Sensor infrastructure to measure data for the input in prediction models has expanded from temperature measurements on the surface to Doppler radar, weather balloons, and weather satellites; and many

more improvements are under way to improve further coverage and accuracy.. Communication satellites and the Internet enable remote access to regional and international databases collecting the weather measurements. Collaborative infrastructures such as the Access Grid have moved exchange of information beyond the desktop. These advances have profoundly affected the way scientists work with each other.

Compute power also has steadily increased. Indeed, as for more than three decades, computer speed has doubled every eighteen months, and this trend is expected to last at least for the next decade. Furthermore, over the past five years network bandwidth has increased at a much larger rate, leading experts to believe that the network speed doubles every nine months. At the same time, the cost of production for network and computer hardware is decreasing.

Besides the increase in capability, we also observe a change in modality of computer operation. The first generation of supercomputing enterprise was concerned mostly with the development of high-end mainframes, vector processors, and parallel computers. Access to this expensive infrastructure was provided and controlled as part of a single institution within a *single administrative domain*. With the advent of network technologies, promoting connectivity between computers, and the creation of the Internet, promoting connectivity between different organizations, we observe a trend leading away from the centralized computing center to a decentralized environment. As part of this trend, it was natural to collect geographically dispersed and possibly heterogeneous computer resources, typically as networks of workstations or supercomputers. The first connections between high-end computers to solve a problem in parallel on these machines were termed a metacomputer. The term is believed to be originated as part of a gigabit testbed [20]. Much research in this area has been influential in shaping what we now term the Grid approach or concept.

1.3 The Term Grid

The term “Grid” is an analogy to the electric power grid that allows pervasive access to electric power. In a similar fashion, computing Grids provide access to pervasive collections of compute-related resources and services. Already in 1965, the designers of the Multics operating system envisioned and named requirements for a computer facility operating like a power company or water company [26], and others anticipated Grid-like scenarios [19].

We emphasize that the concept of the Grid goes far beyond just sharing compute resources in a distributed fashion. Besides supercomputers and compute pools, Grids include access to information resources (such as large-scale databases) and access to knowledge resources (such as collaboration between colleagues). Additionally, we expect that multiple Grids will exist and be supported by multiple organizations. This is also analogous to the electrical power grid where multiple power companies may maintain their own grids while providing persistent services to the user community accessed as part of a *community production Grid.* To manage such a community production Grid, it is necessary to define sharing rules that govern membership and operation

2 Grid Management Facets

Reviewing Fig. 1 we observe that the envisioned Grid infrastructure raises significant control issues. For example, we have to deal with the control of communities, information, data, tasks, hardware, services, and applications or software. Within each of these management challenges, we find a number of issues that must be addressed, such as security, heterogeneity, quality, distribution, disparity, dynamicity, unpredictability, interoperability, and compatibility. In each case we must consider the dynamic, unpredictable properties of the Grid, while at the same time try to provide a reliable and persistent infrastructure. Additionally, we want to enable open collaborations, while at the same time protect the collaboration with appropriate security restrictions. These apparent contradictions -desire for reliability vs. a potential unreliable infrastructure, or restricted vs. unrestricted access to information - provide complex challenges for Grids. In order for Grids to become a reality, we must develop infrastructures, frameworks, and tools that address these challenges. Several state-of-the-art projects try to provide solutions to a subset of these issues. We hope, as part of the Grid community, that a large number of issues can be addressed while learning from existing solutions.

2.1 A Role-Based Layered Grid Architecture

The secure access to a collectively controlled set of physical resources reused by applications motivates a role-based layered architecture that is outlined in [13] and [14]. Within this architecture, it is easy to identify fundamental system components, specify the purpose and function of these components, and indicate how these components interact with one another. The architecture classifies protocols, services, application programming interfaces, and software development kits according to their role in resource sharing. It identifies five layers: fabric, connectivity, resource, collective, and application layers. Interoperability is preserved by using a small standard set of protocols assisting in the secure exchange of information and data amongst single resources. These resources are managed by collective services in order to provide the illusion of a single resource to application designers and users. The layers within the architecture are defined as follows:

- The *fabric layer* contains protocols, application interfaces, and toolkits that allow development of services and components to access locally controlled resources, such as computers, storage resources, networks, and sensors.
- The *connectivity and resource layer* includes the necessary Grid-specific core communication and authentication support to perform secure network transactions with the resources within the Grid fabric. This includes protocols and services allowing secure message exchange, authentication, and authorization. It is beneficial to develop a small set of standard protocols and services to provide the means of interoperability. The resource layer contains protocols that enable secure access and monitoring by collective operations.
- The *collective layer* is concerned with the coordination of multiple resources and defines collections of resources that are part of a virtual organization [13]. Popular examples of such services are directories for resource discovery and brokers for distributed task and job scheduling.

– The *application layer* comprises the users' applications that are used within a virtual organization.

Each of these layers may contain protocols, application-programming interfaces (APIs), and software development kits (SDKs) to support the development of Grid applications and services. A benefit of this architecture is the ability to bootstrap a complex Grid framework while successively refining it on various levels. We emphasize that our architecture can be supported with an immensely rich set of already defined application interfaces, protocols, toolkits, and services provided through commodity technologies and developments within high end computing. Reuse and extension of these standards, based on Grid specific requirements, will support the development of Grids.

2.2 Grid Services

Over the next few years, we will observe a shift within information technologies toward the service concept. From the perspective of Grid computing we define a service as a platform-independent software component, which is described with a description language and published within a directory or registry by a service provider. A service requester can locate a set of services with a query to the registry., a process known as resource discovery. This mechanism has been used by many Grid-related projects. Examples are and, more recent, the Globus Metacomputing Directory Service . A suitable service can then be selected and invoked, a process known as binding.

The usefulness of the service-based architecture within Grid can be illustrated by scheduling a task on a computer cluster. First, we locate a set of possible resources. Next, we select a compute resource from this set where we would like to schedule our task. A criterion to select such a resource could be cost or load balance among the resources. Once a suitable resource is selected, we bind the task of execution to this resource. An important aspect of services is the possibility to easily compose new services while using existing ones. This is enabled by the standard description not only of the protocol, but also of the behavioral description of such a service.

Clearly, it is possible to develop complex flows between services. As this service model deals with the use of asynchronous services, it will be important to deal appropriately with service guarantees in order to avoid deadlocks.

This concept has been in wide use not only by the Grid community, but also by the business community. These realizations led to recent collaborative efforts between the Grid and the business community. An example of such an activity is the creation of the Open Grid Service Architecture .

2.3 Grid Security Management Aspects

Since the Grid approach deals with heterogeneous and dispersed resources and services, security aspects within Grids play an important role. Most commodity security services available today enable the interaction between two peers. The concepts used to enable this are authentication, authorization, encryption, and nonrepudiation.

Authentication deals with the verification of the identity of an entity within the Grid. Though this is commonly associated only with identification of a Grid user, the Grid also requires authentication of resources and services provided as part of the Grid.

Authorization deals with the verification of an action an entity can perform after authentication was successfully performed. Thus, policies must be established that determine the capabilities of allowed actions. A typical example is the use of a batch queue byuser A between 3 and 4 o'clock, but by user B only from 5 to 6 o'clock. In general, policies determine *who* can do *what* and *when* at *which* resource.

Encryption provides a mechanism for protecting the confidentiality of messages in transit between two peers.

Nonrepudiation deals with issues that provide data or message integrity, such as verifying that data was not changed accidentally or maliciously during message transmission.

Besides these general security issues, the Grid infrastructure poses unique requirements. For instance, it is unfeasible to authenticate via password challenges for a user on thousands of different resources.

Single sign-on is a mechanism must exist to support authentication to a large number of Grid resources on behalf of the user or resource while *delegating* the task of authentication to a service acting on behalf of the user (also called a proxy service). Such a service will typically create a temporary credential (often referred to as a *secure proxy*) that is used for authentication. An important factor to consider within single sign-on is that different domains may provide different local security mechanisms. Thusany solution must be able to deal with different identity mappings, such as Unix accounts accessible through PKI or Kerberos.

Delegation is the process of one Grid entity acting on behalf of another Grid entity. Delegation must be performed carefully as it is possible to create delegation chains. A simple example of such a chain is the initiation of a process on a resource D, initiated by a resource A, and subsequently delegated through B and C ($A \rightarrow B \rightarrow C \rightarrow D$). Delegation must be designed and evaluated carefully in order to minimize the risk for exploits. In general, we observe the longer the chain, the greater the risk for misuse. Accordingly, it is desirable to create what we term *limited delegation*. This includes procurements for authentication restriction with more sophisticated Grid services. Thus, we can create a limited proxy that includes for example restrictions on the usage of the Grid resource to be used.

Community authorization provides mechanisms for a virtual organization to define policies for groups of users that can be applied to enabling access control to resources by a community. This service is needed in case it is impossible or impractical to keep track of the access to a resource on a user-by-user basis. An authority that establishes trust between the peers regulates inclusion in such a community. In this sense community authorization enables single-sign-on to resources while being delegated to a trusted authority.

Secure execution is desired in environments where the user community becomes too large to handle. In these cases, it is important to provide a service that can run untrusted applications (those submitted by the users) in a trusted environment (the compute center or cluster); the concept of virtual machines essential for such a service.

We must consider the user community when designing a security infrastructure for applications and services running in a Grid environment. Many users are unwilling to deal with obtrusive security procedures, but at the same time expect a reasonable level of security. Hence, it is of utmost importance to present the security mechanisms to the users in an easy and mostly transparent way. A minimum level of understanding by users is necessary, so that they can specify their own security requirements and understand what security guarantees or risks the Grid provides. Thus, it is necessary to include an *educational service* as part of the strategy of production Grids. This can provide the necessary explanations and guidance for accessing Grid resources and developing secure service.

2.4 Grid Information Management Aspects

Within Grids, information about the users and the system is critical. User information will help to establish collaborative sessions and system information will help us to select the appropriate resources and applications for a problem solving process. The availability of such information enables us to maintain, configure, and use the heterogeneous and dynamically changing Grid infrastructure. Required characteristics that must be imposed on such an information service to support Grids are uniform, flexible access to information, scalable, efficient access to dynamic data, access to multiple information sources, and decentralized maintenance.

The creation of such an information service must be a central part of each Grid toolkit and application. In the past, we have seen the use of distributed directories to enable such a service. Often it is possible to use a centrally maintained relational database to serve the same purpose. In any case, the design of a scalable information service must consider the distributed nature of the Grid. Furthermore, it is often ignored that the resource owners may not wish to export the information about their system to unauthorized users. Though the restricted access to information is already possible, it is not adequately addressed within the first generation of prototype production Grids.

2.5 Grid Data Management Aspects

Each program executed in a Grid is dependent on data, and the data requirements typically are enormous. For example, a high-energy experiment requires storing petabytes of data a day. To compensate for limited storage capacities at remote sites, services that perform delivery on demand may augment the data with a lifetime to limit the amount of actual data in the Grid. In case the calculation cannot be performed on the server where the data is located, we must be able to efficiently replicate that data elsewhere. Thus, a reliable file transfer service must be provided to move the data between source and destination on behalf of the issuing client. To reduce the amount of data during a transfer, appropriate filters may be needed.. In case the data can be created with less effort than the actual data transfer, it may be advantageous to augment data with pedigree information about how to generate the data instead of storing the actual data itself. Data caches at remote sites may be used to reduce the need for data replication even further.

2.6 Grid Execution and Resource Management

Calculations on resources within the Grid are controlled by execution services. The simplest form involves execution services that are part of the operating system and allow execution of jobs and tasks on a single resource. A Grid security infrastructure must be in place that provides authentication and authorization mechanisms to govern the use of this resource. Batch queuing systems provide a convenient way to extend such an execution service to a cluster, a parallel, or a supercomputer. In case we would like to use multiple instances of such resources, a resource co-allocation mechanism is needed. First, we have to identify a suitable set of resources based on the Grid information service. Then, we have to verify that the selected resources are available; if they are not, an algorithm determines how to fulfill the user's request. Once we have a set of suitable resources, we reserve them and finally execute our tasks on this agglomeration of resources.

Algorithms to control the collective use of such resources may be quite complex. Since the algorithmic implications for scheduling in such an environment are an NP complete problem, heuristics may be used to solve the scheduling problem and to guarantee the execution of the tasks. Researchers are currently exploring the use of combinatorial optimization strategies, stochastic sampling, economic models, and agent based systems to study this quality-of-service (QoS) problem.. In order to address the scheduling problem, smart services are necessary that can deal with deadlock prevention, avoidance, and QoS guarantees on local and global scale. Often, complicated workflows must be formulated as part of the complex interdisciplinary applications run by scientists on Grids. Thus, it is necessary to provide workflow management services that allow control of the flow of data and applications as part of the problem-solving process.

2.7 Grid Software Management

Deployment of applications, components, and services in a distributed heterogeneous environment is a tremendously challenging problem. In particular, we must guarantee interoperability between different versions of software and libraries on already installed and operational software and services. The use of the Grid service model described earlier offers a partial solution to this problem by providing metadata to each application and service installed on the Grid that can be queried through the Grid information service. In this way, it is possible to include portability data within the infrastructure, which will be used as part of an authorization service to verify whether services or applications can interoperate [25].

2.8 Grid Hardware Management

The resource providers are responsible for hardware management on the Grid. Notifications about downtimes and maintenance upgrades must be available through the information service in order to simplify finding suitable resources with service guarantees to the user. In general, hardware management must be augmented with an appropriate infrastructure on the hardware service provider side. Quality-of-service augmentations on the hardware level (for example, in networks) could provide a profound advantage for future Grid infrastructures.

3 The Grid Communities

As apparent from the diversity of the Grid approach, a variety of communities are supported by Grids. Each of these communities has its own requirements that have a profound impact on the development of Grids. Within today's Grid community, we identify three basic classes of communities dealing with many Grid related activities:

Development: Grid programmers that develop services in a collaborative fashion to for deployment in the Grid.

Application: Scientific or application users that access the services provided as part of the Grid.

***Community Building*:** Administrators that deploy services and applications in production Grids in order to make them accessible to others.

While today's Grid users include mostly large scale scientific application users and developers, we expect that with the availability of robust Grid toolkits the community will expand to the financial sector, the health care sector, small scale industries, and even the common household user needing access to services resources accessible through the Grid. Thus the Grid will be instrumental in furthering the scientific discovery process [15] while developing the next generation of *community* problem solving environments. Critical to the establishment of these Grid communities will be the development of interoperability standards and policies.

3.1 Global Grid Forum

The Global Grid Forum (GGF) is an international community-initiated forum of individual researchers and practitioners working on various facets of Grids. The mission of the GGF is to promote and develop Grid technologies and applications via the development and documentation of "best practices," implementation guidelines, and standards with an emphasis on "rough consensus and running code." The objective is to support with such standards the creation of production Grids; address infrastructure obstacles inhibiting the creation of these Grids; perform educational outreach; and facilitate the use of Grid technologies within diverse application communities. Based on the IETF model, the GGF contains several area groups and, within these areas, working groups dealing with a particular Grid-related problem. The current areas include information services, security, scheduling and management, performance, architecture, data, and applications and models. Regular meetings are held in which over two hundred organizations from more than thirty countries are represented [10].

3.2 Production Grids

For application scientists it is important to have access to production Grids that allow the integration of their codes within an existing infrastructure. Thus, a number of production Grid efforts have been created that spawn various administrative domains.

To name only a few, the NASA Information Power Grid (IPG) , the Alliance Virtual Machine Room (VMR) [23], DOE Science Grid [7], EuroGrid [3], ApGrid [2], and DataGrid [1] have made progress in developing fundamental technologies needed to build such high-end Grids [4, 24]. A well-trained administrative staff performs the deployment of services and components in such collectively maintained production Grids. Additionally, vast amounts of spare compute resources, such as workstations and personal computers, are part of a shared computing pool resource. This pool of resources can be accessed by a trusted application (as demonstrated by the SETI@home project) or by the members of the community that contribute to the compute pool (as demonstrated by the Condor project). In each case, it is important that deployment issues be carefully addressed in order to minimize the hurdle of deploying the necessary components on the local clients. For collaboration between scientists, research efforts such as the Access Grid are of importance .

3.3 Grid Middleware

Over the past few years many projects have developed middleware that enables the creation of production Grids. Most notable is the Globus Project, which has defined the current de-facto standard for Grid middleware [5]. Additionally, considerable effort has been expended to expose the functionality of the Grid middleware through so called commodity Grid kits for commodity technologies such as Java [24] and Python [17]. These commodity Grid kits also provide bridges to other commodity technologies such as Web services [12]. Other efforts are Legion and Condor . Akenti provides a security model and architecture providing scalable security services in Grids. The SDSC Storage Resource Broker (SRB) [8] is a client-server service that provides a uniform interface for connecting to heterogeneous remote data resources and accessing replicated data sets. The Network Weather Service [16] provides an initial solution to distributed monitoring of resources that periodically records and forecasts the performance of various network and computational resources over time. Many more projects exist and are represented within the GGF.

4 Portals

Future Grid efforts will need to address the problem of exposing their functionality through convenient portals that are used by a particular community. Only if the concept of the Grid is hidden from the end-user we will achieve a seamless integration of Grids in the scientific workplace. The term portal is not clearly defined within the community, as it sometimes represents integrated desktops, electronic market places, or access to information hubs. We believe the term “portal” must be used in a more general sense, providing the ability to build communities and present information relevant to the community. This also includes the integration of applications and services that benefit the community. Similar to portals for electronic business, the key is to integrate data and information with applications and services by the targeted user community (see Fig. 2).

Thus, we define a portal as a single point of entry to an integrated service providing access to information and data, applications, and services by users. Note that this

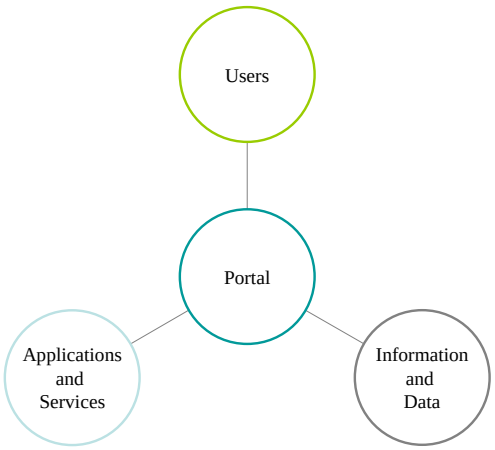

Fig. 2. Portals provide an access point of entry that helps to integrate information and data, application and services by their users.

definition does not include the use of a particular protocol such as HTTP. Most common are Web portals that build the current generation of portals based on the HTTP protocol while accessing the information through a browser. A Grid portal provides a specialized portal useful for users of production Grids. The services offered support the use of Grids and simplify access to these production Grids. Naturally, a Grid portal will include information about the status of the Grid resources and services. Commonly this includes the status of batch queuing systems, load, and network performance between the resources. Furthermore, it may include information related to a community such as the climatology community to provide a targeted access point to useful other high-end services, such as the generation of a compute and data intense parameter study for climate change.

In contrast to Web portals, Grid portals may not be restricted to simple browser technologies, as it is often the case that the data visualization capabilities exceed those of a common Web browser. Displaying data such as macromolecules or three-dimensional high-resolution weather data requires the use of specialized plug-ins or executables. These custom-designed visual components are usually installed outside of a browsers, similar to the installation of MP3 players, PDF browsers, and video conferencing tools. Fig. 3 outlines a simple architecture for Grid portals that we believe will be basis for many Grid portal activities. Special attention needs to be placed on the deployment and administrative services, as they are almost always ignored in common portal activities. A portal can function only if the administrative and deployment services are carefully worked out. In order to rapidly create such portals, it is of advantage to choose commodity technologies. The use of JavaBeans and JSP is supported in many Web portal design interface development environments. Thus it is of advantage to develop Grid portal toolkits that can be integrated in such environments.

Examples for portals are HotPage [6], Webflow and its successor Gateway [11], XCAT [18], UNICORE (UNiform Interface to COmputing REsources) [9] JiPANG(Jini-based Portal Augmenting Grids) [22] Ninf , NetSolve , and the Globus Toolkit [4] via

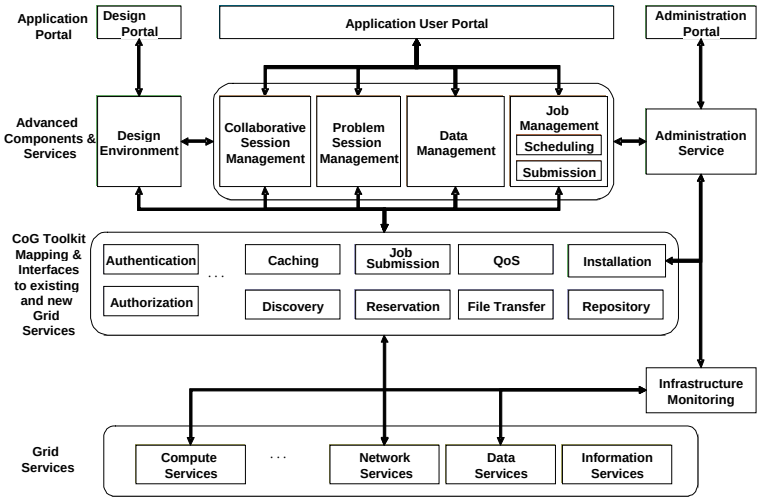

Fig. 3. A portal architecture for Grid portals.

the Java CoG Kit [24]. A simple Java API provides the user with a uniform interface to the Grids. A specialized JiPANG browser allows the interactive access to Grid resources and services.

5 Commercial Grid Activities

Many companies have successfully delivered tools that are integrated in our global vision of Grids. Batch queuing systems such as LSF, PBS, and CODINE (now part of SUN's Gridengine),are well known within the community. Much of the Grid community was based on the development of integrated services that hide the differences between the implementations. Globus Toolkit, Legion, and even earlier Webflow activities were successful in hiding such differences. Nevertheless, the current generation of middleware (often represented by Legion and the Globus Toolkit) has reached a sophistication that goes beyond the initial research projects. Legion is today marketed through Avaki, which was cofounded by the developers of Legion. The Globus Toolkit, which maintains a free open source toolkit, is now marketed by a number of companies that include support as part of their business model.

The newest development in industry and research promoting Web services through efforts such as IBM's commitment to the Web services framework, Microsoft's.Net, and SUN's Web services and JXTA framework will be major drivers for the next generation of Grid software. The development of an Open Grid Service Architecture together with companies such as IBM promises to integrate business and research models and processes in order to leverage from each others technologies. Besides such a definition of an architecture, implementations are needed that prove the validity of such an approach.

6 Conclusion

We are in the infancy of Grid development. We have identified many management-related aspects of Grids and have posed some of the problems emerging Grids will face. Point solutions do exist for a subset of these problems, but much remains in developing Grids and making the vision a reality.In particular, in addition to the development of Grid middleware, interfaces are needed that can be used by the application scientists to access Grids. Commodity Grid toolkits enabling access to Grid functionality on an API level such as in Fortran, Java, and Python are important. Portals also must be developed to hide the complex infrastructure of Grids and allow non expert scientist using this powerful infrastructure..

Besides the technical problems, we also must address the sociological aspects. We believe that the development of open standards, open source, and open communities is essential if we wish to make the Grid a reality.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Science Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. DARPA, DOE, and NSF support Globus Toolkit research and development. We thank Ian Foster, Geoffrey C. Fox, Dennis Gannon, and the members of the Computingportals working group (formerly known as Datorr), which is now the Grid Computing environment working group of the GGF for the valuable discussions leading up to this work. The Globus Toolkit and Globus Project are trademarks held by the University of Chicago.

References

1. The datagrid project, 2000. http://www.eu-datagrid.org/.
2. Apgrid: Partnership for grid computing in the asia pacific region. 2001. http://www.apgrid.org/.
3. Eurogrid: Application testbed for european grid computing, 2001. http://www.eurogrid.org/.
4. The globus project www page, 2001. http://www.globus.org/.
5. Globus web page, 2001. http://www.globus.org.
6. Npaci hotpage, 2001. https://hotpage.npaci.edu/.
7. Scientific discovery through advanced computing (scidac), 2001. http://www.sc.doe.gov/ascr/mics/scidac/.
8. Storage resource broker (srb), 2001. http://www.npaci.edu/DICE/SRB/.
9. Unicore. 2001. http://www.unicore.de/.
10. Global grid forum web page, 2002. http://www.gridforum.org.
11. D. Bhatia, V. Burzevski, M. Camuseva, G. C. Fox, W. Furmanski, and G. Premchandran. Webflow - a visual programming paradigm for web/java based coarse grain distributed computing. *Concurrency: Practice and Experience*, 9(6):555–577, 1997.
12. Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web services description language (wsdl) 1.1, 15 March 2001. http://www.w3.org/TR/wsdl.

13. I Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, 2001. www.globus.org/research/papers/anatomy.pdf.
14. Ian Foster. The grid: A new infrastructure for 21st century science. *Physics Today*, 55(22):42, 2002. http://www.aip.org/pt/vol-55/iss-2/p42.html.
15. Geoffrey C Fox. Portals for web based education and computational science, 2000.
16. B. Gaidioz, R. Wolski, and B. Tourancheau. Synchronizing network probes to avoid measurement intrusiveness with the network weather service. In *Proceedings of 9th IEEE High-performance Distributed Computing Conference*, pages 147–154, August 2000. http://www.cs.ucsb.edu/ rich/publications/.
17. Keith Jackson. pyglobus. *Concurrency and Computation: Practice and Experience*, submitted, 2001.
18. Sriram Krishnan, Randall Bramley, Dennis Gannon, Mad husudhan Govindaraju, Rahul Indurkar, Aleksander Slominski, Benjamin Temko, Richard Alkire, Timothy Drews, Eric Webb, and Jay Alameda. The xcat science portal. In *Proceedings of SC2001*, November 10-16 2001. http://www.sc2001.org/papers/pap.pap287.pdf.
19. J. C. R. Licklider and R. W. Taylor. Scientific technology. 1968. http://memex.org/licklider.pdf.
20. P. M. Lyster, L. Bergman, P. Li, D. Stanfill, B. Crippen, R. Blom, C. Pardo, and D. Okaya. Casa gigibit supercomputing network: Calcrust three-dimensional real-time multi-dataset rendering. In *Presented at Supercomputing '92*, Minneapolis, MN, 17-20 November 1992.
21. Frederic G. Shuman. History of numerical weather prediction at the nmc. *Weather and Forecasting,*, 4, 1989.
22. T. Suzumura, S. Matsuoka, and H.Nakada. A jini-based computing portal system. 2001. http://matsu-www.is.titech.ac.jp/ suzumura/jipang/.
23. John Towns. The alliance virtual machine room, 2001. http://archive.ncsa.uiuc.edu/SCD/Alliance/VMR/.
24. Gregor von Laszewski, Ian Foster, Jarek Gawor, and Peter Lane. A java commodity grid kit. *Concurrency and Computation: Practice and Experience*, 13(8-9):643–662, 2001. http://www.globus.org/cog/documentation/papers/cog-cpe-final.pdf.
25. Gregor von Laszewski, Peter Lane, and Eric Blau. Component deployment in grids. In *Component Deployment 2002*, Berlin, June 2002. Argonne National Laboratory. http://www.globus.org/cog.
26. V. A. Vyssotsky, F. J. Corbat, and R. M. Graham. Structure of the multics supervisor. In *Joint Computer Conference, AFIPS Conf. Proc 27*, page 203, 1965. http://www.multicians.org/fjcc3.html.