Silvan Kaiser, Collin Müller, Sahin Albayrak

Seamless, Intelligent Grid Services for e-Science

DAI-001-2004

Zuerst erschienen:

DAI-Labor, Berlin, March 2004

Technische Universität Berlin
Faculty IV - Electrical Engineering & Computer Science
DAI-Labor

Seamless, Intelligent Grid Services for e-Science

Silvan Kaiser, Collin Müller, & Sahin Albayrak

{Silvan.Kaiser|Collin.Mueller|Sahin.Albayrak}@dai-labor.de

White Paper

March 2004

Summary – This white paper serves as a discussion basis for all participants in the TU Berlin managed Grid computing activities, especially those within the Sun Center of Excellence (CoE) for ”Seamless, intelligent Grid services for e-Science”.

Computational Grids provide easy and relatively cheap access to supercomputing power by plugging together distributed computing resources. We believe that they will play a major part in the global e-Science initiatives and will help to make high-performance computing affordable for almost all research institutions.

There are already two families of middleware that allow for transparent user interaction and comfortable management for the Grid. However, those products have their shortcomings and are barely interoperable with each other. Therefore, we propose an additional architectural layer that integrates the heterogenous approaches and provides the foundation for a truly global Grid. Further we identify the research efforts, which have to be undertaken in order to let our vision become reality.

Keywords – e-Science, Grid Computing, Meta-Grid Service Engine, Agent Technology, JIAC IV Serviceware Framework, Grid Middleware.

Contact

Prof. Dr.-Ing. Sahin Albayrak
Technische Universität Berlin
Fak. IV, Sekr. GOR 1-1
DAI-Labor
Franklinstrasse 28/29
D-10587 Berlin, Germany

Phone: +49.30.314.24943

Email: Sahin.Albayrak@dai-labor.de

Web: www.dai-labor.de

Contents

1 Introduction

1.1 Background

During the last few years the new possibilities delivered by Internet technology have not only changed the way we gather information, communicate, and shop but also the way scientific research is done. In order to speed up scientific research, to deal with ever growing amounts of data, and to provide a better quality of scientific results, ”there is an increased emphasis on collaboration between large teams, an increased use of advanced information processing techniques, and an increased need to share results and observations between participants who are not physically co-located [DeRoure, Jennings, ShadboltDeRoure et al.2003]”.

New technologies as the World Wide Web and e-mail are a first step to satisfy these upcoming needs of scientists. However, computer and telecommunications technology is in many aspects not only the solution to some of the above mentioned problems but also the cause. The ”data generated from sensors, satellites, high-performance computer simulations, high-throughput devices, scientific images and so on will soon dwarf all of the scientific data collected in the whole history of scientific exploration.” [Hey TrefethenHey Trefethen2003] Without new technologies beyond WWW and e-mail this data explosion will lead to less instead of more insight because of an *information overload*. Those innovative technologies which could enable the global collaboration of scientists, handle large amounts of data, and provide mechanisms for an efficient sharing of resources are generally subsumed by the term e-Science.

With the described challenges in mind, Foster et al. sketched the vision of a global network of distributed computing resources that should be as easily used as the electric power grid, which is available in almost every household and can be utitilized by simply plugging in any device that is compatible with it [Foster, Kesselman, Tsudik, TueckeFoster et al.1998]. This gave rise to a growing interest in the topic of Grid computing, which is now one of the most promising technologies for the solution to the above mentioned problems. Simply put, Grid computing means easy access to distributed supercomputing resources.

Grid computing will help to connect distributed processing power, which currently remains unused for most of time, to a global supercomputing network that will help to use data processing resources more effectively and efficiently. We believe that Grid computing will unleash a vast amount of computing power and make it available to many new users, and will therefore not only tremendously enhance the speed with which new innovations will be produced, but it will also enhance the quality of the results produced by the global scientific community.

1.2 Goal and Methodology

This white paper will form a basis for the discussion about the upcoming Grid computing activities at the Technische Universität Berlin and the planned *Sun Center of Excellence (CoE)* for *Seamless and Intelligent Grid Services for e-Science*. Our goal is to find a common understanding of the term *Grid Computing* and the related technologies. Additionally we identify some research challenges that have to be tackled in order to enhance Grid computing technologies for the deployment in global production environments.

Therefore we give a brief overview over the current state of Grid research and about the anticipated results for the next years in section 2. Further we define the term *Grid Computing* for the context of this paper and for our research plans. The paper shows how the already available Grid engines work and how they provide their services to the user and developer. We refer to the *Sun Grid Engine* and to the *Globus* counterpart as well as *Condor* and the *IBM* products related to the Grid. The section is concluded by our vision of an agent-based grid middleware. A brief overview of the *JIAC IV Serviceware Framework* with respect to its already existing Grid functionality is given.

Section 3 introduces our view of a *Meta-Grid Service Engine (MGSE)*, which implements an additional software layer between proprietary Grid engines and the applications and therefore integrates the different approaches. It will make a global Grid across OEM-boundaries become reality. The section also describes our view of the basic Grid services provided by our MGSE and some value-added Grid services that significantly extend the functionality of the underlying proprietary Grid engines. The section closes by identifying research challenges to be targeted by the agent and Grid communities with respect to the proposed Meta-Grid services.

Finally, we conclude this paper in section 4 with a summary of our results and provide a research route for the years to come.

2 Grid Overview

2.1 Definition and Overview

2.1.1 Definition of the Grid

As described in section 1, there is a tremendous need for high performance information processing capacity, not only for a few selected research institutions and companies but for the broad mass of computer users. In their seminal work Foster and Kesselman describe Grid Computing as a way ”to provide users [. . .] with substantially more computational power” [Foster KesselmanFoster Kesselman1999]. Their initial definition of the Grid also gives some first keywords regarding the requirements, that have to be met by Grid computing solutions:

> *”A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.” [Foster KesselmanFoster Kesselman1999]*

In 2001 Foster and his co-authors refine and complement their definition by referring to the social influence of the Grid. They state that Grid computing means ”coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations [Foster, Kesselman, TueckeFoster et al.2001]”. But this does not only implicate the exchange of information, ”but rather direct access to computers, software, data, and other resources [Foster, Kesselman, TueckeFoster et al.2001]”. In this context a virtual organization (VO) is any set of individuals and institutions who share computing resources in a controlled manner defined by sharing rules.

One of the most prominent examples for the shared use of processor cycles for scientific research is the SETI@home project[1]. This project aims to find non-stochastic patterns in the data received from the Arecibo radio telescope. The amount of information to be analyzed is so huge that even supercomputers would need years in order to get along with it. Therefore, the dataset is split into millions of pieces and sent to the PCs of users spread all over the world who are willing to share their unused processing power.

2.1.2 The general Grid architecture

Traditional peer-to-peer computing usually relies on the TCP/IP-Protocol family for connectivity and supports relatively autonomous nodes in the net. The Grid is notably different to these conventional forms of distributed computing, ”since there is a concern of proposing a shared infrastructure for direct access to storage and computing resources [Binder, Di Marzo Serugendo, HulaasBinder et al.2002]”. For the application developer, who could so far ”often assume a target environment that was [. . .] homogeneous, reliable, secure, and centrally managed [Foster, Kesselman, Nick, TueckeFoster et al.2002]”, the Grid imposes a new way of thinking as well. More and more we face collaborative computing, shared data and shared memory on heterogeneous systems, and other interactivity between distributed resources.

[1]http://setiathome.ssl.berkeley.edu/

Therefore, Foster et al. propose a suitable high-level architecture as a discussion basis for Grid research. [Foster, Kesselman, TueckeFoster et al.2001]. Figure 1 gives an overview of this basic grid architecture in opposition to the traditional Internet architecture.

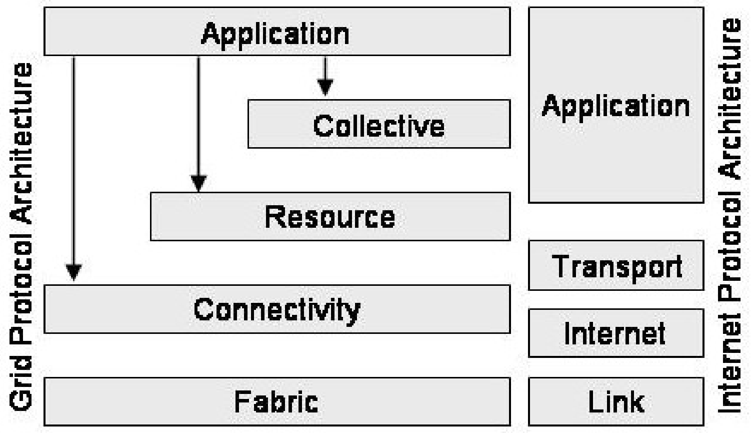

Figure 1: Grid Architecture according to (Foster et al. 2001)

The *Fabric* level implements the specific access mechanisms to the logical and physical resources of the Grid. These could be distributed file systems, databases, processors, etc. At a minimum, resources should implement enquiry mechanisms that permit discovery of their structure, state, and capabilities as well as resource management mechanisms for e. g. quality of service and reservation.

The basic communication and authentication mechanisms are provided by the *Connectivity* layer. They allow for data exchange between different resources on the Fabric layer.

Based on the Connectivity layer, the *Resource* layer implements protocols for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources. This layer is only concerned with local resources and does not take the global state of the Grid system into account.

The ”real” Grid functionality is provided by the *Collective* layer that manages a set of resources and therefore has some kind of global view on the system.

The Collective layer offers services as brokering, scheduling, directory services, monitoring and diagnostics, replication as well as load management.

Applications, languages, and development frameworks reside on the *Application* layer of the Grid architecture. Applications comprise of calls to services provides by any of the lower layers.

2.1.3 The Grid in the e-Science Context

One application domain that is especially affected by the impending ”information overload” already sketched in section 1 is the science community. Modern scientific methodologies and applications heavily rely on rapid access to large quantities of data as an input factor. Analysis of this data is currently only possible with state-of-the-art computing resources. And the demand for processing power for scientific applications is likely to rise. The problem even increases, because ”both resources and data are often distributed in a wide-area network with components administered locally and independently. Computations may involve hundreds of processes that must be able to acquire resources dynamically and communicate efficiently.”[Foster, Kesselman, Tsudik, TueckeFoster et al.1998]

The scientific community has reacted to the ever increasing need for computing power. We observe a ”a dramatic pluralization of computing and storage resources. [Frey, Tannenbaum, Livny, Foster, TueckeFrey et al.2001]”. But much of the available resources remain idle for most of the time, because usually only the owning organization is able and allowed to access them. Although a cross-institutional usage of the processing capabilities is possible in theory, Frey et al. identify a ”significant ’potential barrier’ associated with the diverse mechanisms, policies, failure modes, performance uncertainties, etc., that inevitably arise when we cross the boundaries of administrative domains.”[Frey, Tannenbaum, Livny, Foster, TueckeFrey et al.2001]

Overcoming this potential barrier requires new methods and mechanisms that meet the following three *key user requirements* for computing in a Grid that comprises resources at multiple locations: [Frey, Tannenbaum, Livny, Foster, TueckeFrey et al.2001]

- They want to be able to discover, acquire, and reliably manage computational resources dynamically, in the course of their everyday activities.
- They do not want to be bothered with the location of these resources, the mechanisms that are required to use them, with keeping track of the status of computational tasks operating on these resources, or with reacting to failure.
- They do care about how long their tasks are likely to run and how much these tasks will cost.

If all these user requirements can be met then very likely the Grid is ”the middleware infrastructure that is currently being developed to support global e-Science collaborations. [Hey TrefethenHey Trefethen2003]” Allen et al. even go a step further when they postulate that ”the Grid [...] will enable us to perform new kinds of science and engineering which would remain out of reach without this new technology [Allen, Goodale, Russell, Seidel, ShalfAllen et al.2003]”.

2.2 Applications and Requirements

However, it is important to note that Grid computing may of course not only be applied to the e-Science domain, but rather to any application domain that requires or benefits from a global collaboration of institutions or people. This could be in engineering, gaming, multimedia, and many more. The Grid will allow for establishing ”dynamic markets for computing and storage resources, hence overcoming the limitations of current static configurations. [Foster, Kesselman, TueckeFoster et al.2001]” This may result in new business opportunities for telecoms and other players in the information and communication technologies sector. They could provide processing power or network storage like they already offer voice communication or Internet access and then bill that by the second.

If there is a generally accepted standard for the sharing of distributed resources, institutions will be able to outsource their IT operations much more easily than today. Only specialized providers will have to deal with ordering and deploying the latest hardware and can establish a much more efficient yield management. We believe that this will significantly reduce the cost of employing high-performance computing and will therefore give nearly every institution the possibility to benefit from the already described advantages.

But before our vision of the Net as one almost unlimited supercomputer with the anticipated user experience (see page 6) will come true, many technical challenges have to be met. Performing a computation on resources that belong to different sites can be difficult in practice for the following reasons [Frey, Tannenbaum, Livny, Foster, TueckeFrey et al.2001]:

- Different sites may feature different authentication and authorization mechanisms, schedulers, hardware architectures, operating systems, file systems, etc.
- The user has little knowledge of the characteristics of resources at remote sites, and no easy means of obtaining this information.
- Due to the distributed nature of the multi-site computing environment, computers, networks, and subcomputations can fail in various ways.
- Keeping track of the status of different elements of a computation involves tedious bookkeeping, especially in the event of failure and dependencies among subcomputations.

Johnston and Brooke's collection of core technical requirements for Grid systems is still work in progress but it can serve as a very helpful starting point in the search of interesting research challenges. The following issues have to be solved [Johnston BrookeJohnston Brooke2002]:

- **Resource Discovery**: Every Grid needs a service that provides information about the available Grid resources. It must inform about location, capabilities, and eventually prices of those resources. The service should not rely on user maintained lists, but collect the information itself.

- **Resource Scheduling**: The scheduling within a Grid should be separately managed from process initiation. It must take all kinds of resources into account, some of which do not involve processes (e. g. communication resources). The service should provide schedules on the meta-institutional level, i. e. within Virtual Organizations, and collect the necessary information for negotiating a common Quality of Service (QoS) accross institutional boundaries.

- **Uniform Computing Access**: It should be possible in a global Grid to access heterogenous computing resources via some common interface. Therefore, a mechanism for job submission must be established, which lets the user ignore the underlying individual hardware and operating systems.

- **Uniform Data Access**: A Grid comprises of persistent storage systems based on different technologies and concepts such as flat files, relational databases, and object-oriented databases. In order to guarantee a transparent access to the storage, we need an abstract protocol that integrates access, updates, and results from and to heterogenous systems. It should be tightly integrated with the security mechanisms (see below).

- **Asynchronous Information Sources**: An Asynchronous Information Source is any source of XML formatted objects that can publish its existence and object content characteristics, and then support subscription based delivery of those objects. Those could be:

  - Events (system, application, workflow scripts)
  - Monitoring (nodes, network, application)
  - Accounting records

  - Logging information

- **Remote Authentication, Delegation, and Secure Communication**: The purpose of this 'service family' is to establish trust relations between the different objects within a Grid. This is accomplished in a sequence of trusted steps, each one of which is essential in order to get from accepting a remote user on a Grid resource back to a named entity. It is quite clear that the security system will rely on a public key infrastructure, which provides organization, user, application and resource keys. All of the standard security goals such as non-repudiation, data integrity, confidentiality, and accountability must be achieved in order to provide an effective AAA-infrastructure.

- **System Management and Access**: System management and system access functionality must be provided that allows for comfortable, efficient, and interactive usage of the Grid. One of the biggest issues in this context could be a single sign-on functionality for the system.

- **Architectural Constraints**: The Grid architecture must ensure that the trusted security features cannot be circumvented. Only secure data communication channels should be optional, because they may be impractical or unnecessary under certain circumstances.

2.3 Current Grid projects

In this section we present an overview of several proprietary grid engines available today. These include commercially as well as freely available software. As this overview shows the different projects are quite closely connected to each other, the two main players being Sun microsystems, IBM and HP.

2.3.1 SUN Grid Engine

In July 2000 Sun microsystems aquired a company named Gridware along with the company's main product which was released as an open source project under the name of GridEngine[2]. Based on the code developed by this project Sun features two products, the N1 Grid Engine and the Sun Grid Engine Enterprise Edition. The first product is aimed at single departments planning to establish a computing grid, the second allows department spanning grid systems and includes advanced scheduling mechanisms and grid management tools.

The Sun Grid Engine is available for several platforms, currently featured are Solaris, Linux, Mac OS/X and several other Unix derivates. Jobs are scripted and then scheduled through commandline or GUI tools and sent to the executing nodes in the grid. There the script is started and the job executed. The management software allows to submit not only single but queues of jobs and automatically spreads the jobs throughout the grid based on its load balancing mechanisms. Exit status and output of the jobs are returned after the scheduled scripts have finished.

This approach allows the distribution of arbitrary jobs, there is no need for a specialized API as this concept distributes the call of the scheduled job, not the job itself. All nodes in the grid have to be able to execute the scheduled call, though. Mainly suited for 'fire and forget' scripts the jobs should use no interaction unless they contain the integrated ability to find their interaction partners from their dynamically assigned location. This concept ist well suited for mathematical equations requiring high computational power.

2.3.2 Condor

In concept quite similar to the Sun Grid Engine, this open source project[3] from the University of Winsconsin-Madison was created by researchers mainly for running CPU intensive jobs that do not need user interaction, like e. g. molecular simulations. Jobs are specified in a Condor configuration file and distributed to the different nodes of a Condor grid. Again all nodes in the grid have to be able to execute the commands specified in the job description. A special version labeled Condor-G implements several Globus specifications for security and management.

[2]http://gridengine.sunsource.net/

[3]http://www.cs.wisc.edu/condor/

2.3.3 IBM, HP and the Globus Alliance

IBM[4] and HP[5] feature grid products based on the OGSA implementations provided by the Globus Alliance. These are extended by management tools. The IBM Grid Engine is available for AIX and Linux, as well as other operating systems.

Driven by the Globus Alliance[6] this extensive framework aims to feature a global standard for sharing computing resources securely over institutional and corporate boundaries. The Globus Alliance is a partnership of Argonne National Laboratory Mathematics and Computer Science Division, the University of Southern California's Information Sciences Institute, the University of Chicago's Distributed Systems Laboratory, the University of Edinburgh in Scotland, and the Swedish Center for Parallel Computers. Main achievements of the Globus Alliance are the OGSI (Open Grid Service Infrastructure) and OGSA (Open Grid Service Architecture) specifications, the latter being a comprehensive framework that, besides OGSI, includes a range of APIs and further specifications, with an especially strong security focus. As the framework aims at sharing resources across corporate boundaries authentication and authorization are important aspects of the concept.

Globus relies on Web Services for accessing remote nodes, in fact Globus defines Grid Services as Web Services providing a minimum set of management functionalities[7]. The technical overview describes the OGSA as "...a standards based distributed service system (...) that supports the creation of the sophisticated distributed services required in modern enterprise and interorganizational computing environments" ([Foster, Kesselman, Nick, TueckeFoster et al.2002], page 4). Many other projects are developing interfaces to this OGSA framework. FTP based protocols allow data transfer and different APIs provide access to server side Web Services, mainly using Java implementations. This Java core runs on

[4]http://www-1.ibm.com/grid/

[5]http://www.hp.com/techservers/grid/index.html

[6]http://www.globus.org/about/default.asp

[7]Currently the new WSRF (Web Services Resource Framework) replaces parts of the Grid Service specifications, some major work is under way in this area and changes are to be exspected.

all Java enabled platforms, other Globus implementations are available for Unix only.

Unlike the concepts of the SUN Grid Engine the Globus architecture does not spread jobs on a relatively homogeneous grid of nodes but works on a variety of different architectures. However jobs cannot be spread throughout the grid on arbitrary nodes but have to call for specific addresses of the provided Grid Services. There are scheduling tools controlling access to different instances of specified services, yet the main focus of the Globus Alliance is the management of interorganizational service access and OGSA lacks dynamic abilities but requires fixed service installations.

2.4 An Agent-based Grid Engine

2.4.1 General Vision

As early as 1999 Foster and Kesselman consider in their seminal paper agent-oriented software technologies as a useful tool for Grid applications: ”Agent-based programming models represent another approach apparently well suited to grid environments; here, programs are constructed as independent entities that roam the network searching for data or performing other tasks on behalf of the user.” [Foster KesselmanFoster Kesselman1999]

Later Foster and his co-authors state that a service integration across distributed, heterogenous, dynamic virtual organizations in both e-business and e-science environments must be achieved [Foster, Kesselman, Nick, TueckeFoster et al.2002]. De Roure et al. carry on this thought when they observe that ”a natural way to conceptualise the service owners and the service consumers are as autonomous agents [DeRoure, Jennings, ShadboltDeRoure et al.2003].” Obviously, the service provisioning within a Grid is well suited to be managed by software agents.

New services will be continuously offered and existing ones may be removed at any time. The same is true for nodes and other resources within the Grid.

”This means the system is in a state of continual flux and never reaches a steady state. [DeRoure, Jennings, ShadboltDeRoure et al.2003]” This leads to another challenge in Internet-scale distributed systems: Ideally they should scale from a few to several thousands or even millions of nodes. Overeinder et al. propose agent technologies as an effective solution for this problem, as agent systems avoid the scalability shortcomings of centralized approaches. They go even further when they state that ”peer-to-peer interaction strategies as embraced by agent technologies seems to be the most promising approach to provide scalable and adaptive services.” [Overeinder, Wijngaards, N., Steen, BrazierOvereinder et al.2002]

Overeinder et al. also tackle the complex high-level task of coordinating collective resources, which ”integrates the multiple resources into a wide-area distributed system [Overeinder, Wijngaards, N., Steen, BrazierOvereinder et al.2002]”. Furtherly, the co-ordinating processes in the Grid require ”intelligence, autonomy, and social capabilities [Overeinder, Wijngaards, N., Steen, BrazierOvereinder et al.2002]”. All these characteristics are constitutive features of software agents.

Furthermore, the challenge of ever growing data must be solved by Grid systems. Especially in cross-institutional virtual organizations it is often unfeasible to move data to the analysis software due to legal and security concerns as well as because of the mere volume to be transferred. So if we cannot move the data to the computation, why not move the computation to the data. This again points at employing software agents with an in-built functionality of migrating from node to node.

Additionally, mobile agents provide ”sufficient abstraction at the application level” in order to cope with ”the continuing evolution of individual machines [Allen, Goodale, Russell, Seidel, ShalfAllen et al.2003]”. This not only solves the problem of increasing computing power but also deals with the problem of heterogenous hardware and operating systems, as agent architectures can offer a standardized runtime environment on different platforms.

Fukuda et al. summarize the advantages of employing software agents as a Grid middleware with the following statement: ”Mobile agents are capable of autonomously navigating over networks and launching network tasks, with which a user job can mine available resources and migrate from one computer to another whenever the current resource becomes unavailable. A central technical challenge is user process migration, extending its target to multiple processes working together in parallel, thus communicating with each other, which requires not only capturing a process state but also forwarding messages to migrating processes." [Fukuda, Tanaka, Suzuki, Bic, KobayashiFukuda et al.2003]

2.4.2 JIAC IV as a basis for agent-based Grid Services

In [Sesseler AlbayrakSesseler Albayrak2001] JIAC IV is described as "...a generic agent toolkit which is intended as a service-ware framework for the realisation of applications in the areas of telecommunications, mobile services and electronic commerce.". This toolkit includes a scalable component architecture, AAA services, security infrastructure and a wide range of development tools. The software development process is led by an agent oriented development methodology.

The JIAC IV toolkit is based on the Java programming language and therefore largely platform independent: "Java runtime systems are available for most hardware platforms and operating systems. Because of the heterogeneity of the hardware and of operating systems employed by Internet users, it is crucial that a platform for large-scale Grid computing be available for a large variety of different computer systems. Consequently, a Java-based platform potentially allows every computer in the Internet to be exploited for distributed, large-scale computations, while at the same time the maintenance costs for the platform are minimal ("write once, run everywhere)."[Binder, Di Marzo Serugendo, HulaasBinder et al.2002]

Driven by the included AOSE (Agent Oriented Software Engineering) methodology JIAC IV applications are composed of Services provided by agents which can be distributed on different computers running agent places, the runtime environment. Communication between agents can take place either on the local agent place or over the network utilizing different protocols like TCP, SSL or HTTP.

Services are implemented in JADL (JIAC Agent Description Language), a language based on declarative concepts. For handling peripheral access Java components can be added to JIAC IV agents, e.g. for accessing standard APIs like Java Mail. Special agents ("Managers") provide management functionalities and a wide range of runtime tools supports the administration of JIAC IV applications.

JIAC IV features mobile agents with strong migration [Fricke et al.Fricke et al.2001] realizing mobile code providing a key feature to aforementioned requirements of future Grid technologies. Furthermore JIAC IV agents can use Web services which is useful when thinking about integration with OGSA implementations. In Addition the toolkit contains a wide range of security functionalities for securing communication and applying authentication and authorization services. The generic accounting framework included in JIAC IV relies on these implementations.

The highly scalable and flexible JIAC IV toolkit provides an excellent basis for agent-base Grid Services as it supports needed key features like platform independence, mobile code and AAA functionalities.

3 Meta-Grid Service Engine

3.1 General Overview

Section 3 introduces our view of an agent-based *Meta-Grid Service Engine (MGSE)*, which implements an additional software layer between proprietary Grid engines and the applications and therefore integrates the different approaches. It will make a global Grid across OEM-boundaries become reality. The section also describes our view of the basic Grid services provided by our MGSE and some value-added Grid services that significantly extend the functionality of the underlying proprietary Grid engines. The section closes by identifying research challenges to be targeted by the agent and Grid communities with respect to agent-based Meta-Grid services.

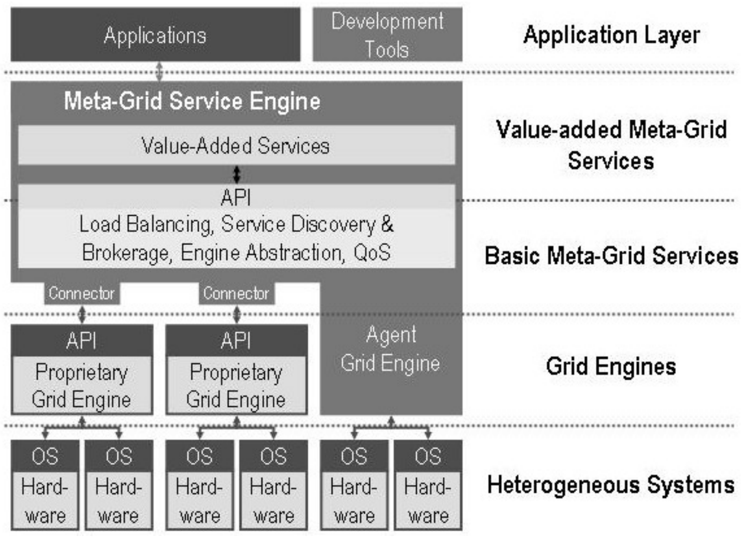

Figure 2: Meta Grid Services Architecture

3.2 Meta-Grid Services

The Meta-Grid Services should address the main issues of today's Grid Computing software. Low level Grids like the SUN Grid Engine (see page 10) provide scalable high performance Grids but have several requirements and shortages that need to be taken care of. First they need homogeneous Grid nodes because scheduled tasks contain only scripts designed to call the programs needed for the tasks. Therefore all nodes in the Grid need to be able to run the defined programs as well they need a common data access. This severly limits the Grids ability to function with heterogeneous nodes.

Furthermore the low level design does not handle AAA aspects too well. When applying the ideas of interorganizational Grids this becomes a very important issue, as perceived by the Global Alliance. Focusing heavily in this partial area the OGSA uses Web Services and sophisticated Authentication and Authorization techniques. Web Services allow a platform independent approach, combined with the included security mechanisms this becomes an important basis for the Global Grid vision. However OGSA strongly lacks the flexibility envisioned for future grids. As described in [Foster, Kesselman, Nick, TueckeFoster et al.2002] OGSA specifies “...a standards based distributed service system...”. The nodes of the Grid supply different services which can be used but prior to giving a task to the Grid the service has to be deployed on several nodes. Low level Grids lessen this problem as they use a common software basis anyway, thus a single service deployment empowers the whole grid with the new service. Globus only provides static remote services. We envision the use of mobile code in future Grids which allows service creation and deployment on arbitrary nodes of a Grid giving a flexibility unknown by todays Grid technology. Services can be created and distributed in a Grid through mobile code, severly reducing the need for software installation in the Grids nodes.

The main objective is on the one hand an integration of low level concepts like the SUN Grid Engine (see page 10) in wide scale management like the OGSA and on the other hand bringing more flexibility to systems based on the OGSA framework by integrating features like dynamic service distribution, mobile code, etc. The ultimate goal being the ability to distribute tasks over a secure and organization spanning dynamic Grid.

By using platform independent software Grids can be extended to large scale networks, allowing the flexible use of computing resources from a vast number of Grid nodes. Here support for dynamic Grid structures becomes an important point. Meta-Grid Services have to be able to tolerate changes in the managed Grid, like the addition or removal of nodes at runtime. Fallback mechanisms have to go to work when tasks cannot be fulfilled because nodes they where assigned to drop out without returning the appropriate task results.

Sub Grids can be put to best use by assigning tasks through intelligent Meta-Grid Services, these can differentiate the individual needs of a task and assign it to an applicable node or sub Grid, e.g. by assigning mathematical computations to high power SUN Grids while directing low priority tasks to other nodes (which possibly demand lower prices). Another interesting idea is the integration of semantic web[8] concepts into the Meta Grid Service Engine providing general insight into node stored informations.

Integrating meta level accounting is an important Meta-Grid Service, empowering a Meta-Grid to ensure full accounting of scheduled tasks, including the use of sub Grids and other integrated legacy systems. These become immensely important when thinking about the interorganizational aspects of global Grid structures. In order to achieve this, a consistent accounting concept has to be brought to all participating Grid structures. Beginnings of the technologies required to reach the goal of realizing the introduced Meta-Grid Services do already exist but still a lot of work needs to be done

3.3 Research Challenges

Open questions providing interesting research challenges arising from the before mentioned concepts deal basically with the areas of security, accounting and mobile agents. Ensuring adequate security for mobile agents in order to protect them from unauthorized manipulation on remote grid nodes is an open issue as well as the reverted question: how can Grid nodes be protected from unauthorized manipulation by foreign agents. This has to be done without disabling the Grids flexibility.

Management functionalities for controlling mobile agents remotely need to be refined and locating running tasks in a dynamic grid is still an interesting research area when thinking about mobile agents that work on the Grid while their home node is offline, how can results be returned?

Relying on the envisioned Grid functionalities an extensive accounting concept hast to be developed for providing an industry utilizable Grid solution. The use of standards is a very important point for the widespread integration work that needs to be done, here ongoing scientific work has to evaluated (e.g. currently OGSI is evolving by the introduction of WSRF (Web Services Resource Framework) as

[8]http://www.w3.org/2001/sw/

discussed in [Czajkowski et al.Czajkowski et al.2004]. We can see the number of research challenges is quite large.

4 Outlook

As shown in the preceding sections, there are still many interesting problems that have to be solved in order to bring the benefits of Grid computing to the broad mass of users. The Technische Universität Berlin and its partners are strongly intensifying their efforts in this sector. This section lists our primary research fields in the Grid computing context for the next months.

4.1 Research Environment

The planned co-operation between Sun Microsystems and faculty 4 of Technische Universität Berlin within the Sun Center of Excellence for ”Seamless, intelligent Grid services for e-Science” does not only consist of an exchange of expertise and industry contacts. Sun Microsystems also provides a Grid testing hardware and networking infrastructure. Furthermore, Sun helps with their consulting expertise for the set-up of the testbed.

At start, the testbed consists of seven multiprocessor nodes based on heterogenous hardware configurations (see figure 3): Sun SPARC RISC Architecture, Intel-based 64-bit computers, and Intel-based 32-bit nodes. Additionally, a Sun StorEdge mass storage unit with nearly 1 Terabyte of harddisk space is part of the network. All nodes are interconnected by an Infiniband Bus system that provides – regarding our purposes – an almost unlimited bandwidth. All systems are configured to run a Linux operating system and the Sun Grid engine. The testbed can easily be extended by adding more nodes that run either Linux or Solaris.

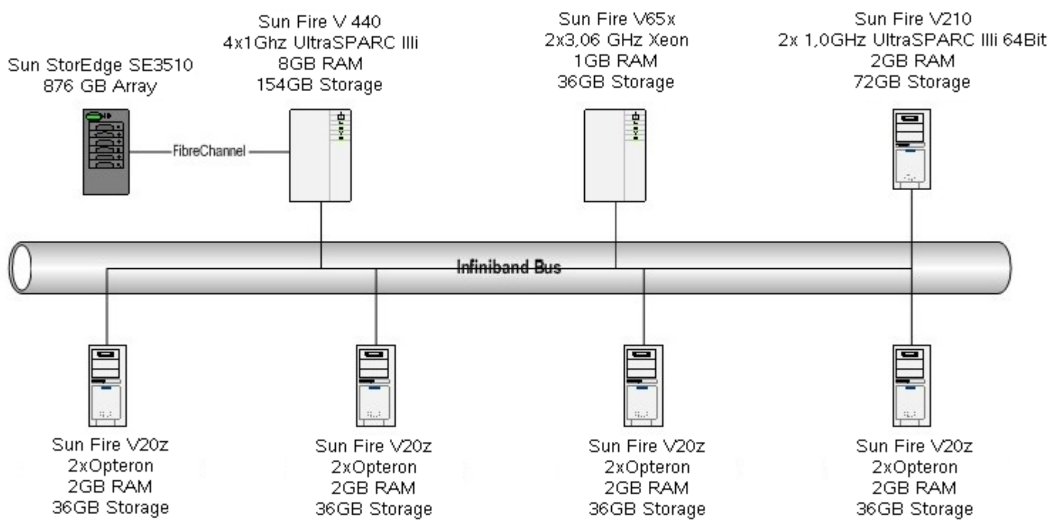

Figure 3: Sun Grid test bed at DAI-Labor

4.2 Research Topics

Our work on Grid computing within the next months will concentrate on some of the identified research challenges. We have selected five primary reseach foci, the first two aiming at the application layer, the other three concentrating on the internals of Meta Grids:

- **Research Focus I – Grid Services**
  The projects within this research focus deal with Grid services, i. e. we are going to identify business models and application fields for Grid services. Those will form the basis for actually implementing Grid services that provide an added value to the user. Primarily, the services will be settled in the knowledge extraction and analysis field, as well as in simulation applications for the automotive industry.

- **Research Focus II – Development Tools**
  Development tools that support the whole lifecycle of Grid services are needed. They must handle developer requirements in the analysis, de-

sign, implementation, testing, deployment, maintenance, and service removal phases. Therefore, we are going to create development environments, methodologies, ontologies, languages as well as service maintenance tools.

- **Research Focus III – Meta-Grid Services**
  This research focus concentrates on implementing the software, which is necessary for the Meta- Grid Service Engine proposed in section 3. Major areas of interest are intelligent task distribution, integration of services provided by different grid technologies and combined services. The integration of semantic web concepts enabling distributed data storage and retrieval provides further research topics.

- **Research Focus IV – Agent-based Grid Engine**
  Within research focus IV we are going to realize an agent-based grid engine, as described in section 2.4. There will be many synergies with the JIAC Serviceware Framework project at DAI-Labor. Essentially, some functionality will be added to the JIAC product in order to transform it into a Grid middleware.

- **Research Focus V – Interoperability**
  The last research focus will be about interoperability and connectivity between the results of the different aforementioned research foci. We are going to develop APIs, Connectors, etc. that allow for the easy but effective connection of different parts of the Grid architecture.

References

[Allen, Goodale, Russell, Seidel ShalfAllen .2003] Allen, G., Goodale, T., Russell, M., Seidel, E. Shalf, J. 2003. Classifying and enabling grid applications. B. Fran, G. Fox A. Hey (), Grid Computing. Wiley.

[Binder, Di Marzo Serugendo HulaasBinder .2002] Binder, W., Di Marzo Serugendo, G. Hulaas, J. 2002, June. Towards a secure and efficient model for grid computing using mobile code. Proceedings of the 8th ecoop workshop on mobile object systems ”agent applications and new frontiers”. Malaga, Spain.

[Czajkowski .Czajkowski .2004] Czajkowski, K., Ferguson, D., Foster, I., Frey, J., Graham, S., Maguire, T., Snelling, D. Tuecke, S. 2004, February. From open grid services infrastructure to ws-resource framework: Refactoring & evolution.

[DeRoure, Jennings ShadboltDeRoure .2003] DeRoure, D., Jennings, N. R. Shadbolt, N. 2003. The Semantic Grid. A future e-Science infrastructure. F. Berman, G. Fox A. J. G. Hey (), Grid Computing - Making the Global Infrastructure a Reality (437–470). John Wiley and Sons Ltd.

[Foster KesselmanFoster Kesselman1999] Foster, I. Kesselman, C. 1999. Computational Grids. I. Foster C. Kesselman (), The grid. blueprint for a new computing infrastructure (15–52). San Francisco, CA, USA: Morgan Kaufmann.

[Foster, Kesselman, Nick TueckeFoster .2002] Foster, I., Kesselman, C., Nick, J. Tuecke, S. 2002, June. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration (Technical Report). Global Grid Forum: Open Grid Service Infrastructure WG.

[Foster, Kesselman, Tsudik TueckeFoster .1998] Foster, I., Kesselman, C., Tsudik, G. Tuecke, S. 1998. A security architecture for computational grids. Proceedings of the 5th acm conference on computer and communications security (83–92). San Francisco, CA, USA: ACM Press, New York, NY, USA.

[Foster, Kesselman TueckeFoster .2001] Foster, I., Kesselman, C. Tuecke, S. 2001. The Anatomy of the Grid. Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 15(3).

[Frey, Tannenbaum, Livny, Foster TueckeFrey .2001] Frey, J., Tannenbaum, T., Livny, M., Foster, I. Tuecke, S. 2001, August. Condor-g: A computation management agent for multi-institutional grids. Proceedings of the 10th ieee international symposium on high performance distributed computing (hpdc-10'01). San Francisco, CA, USA.

[Fricke .Fricke .2001] Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sesseler, R. Albayrak, S. 2001, April. A tooolkit for the realization of agentbased telematic services and telecommunication applications.

[Fukuda, Tanaka, Suzuki, Bic KobayashiFukuda .2003] Fukuda, M., Tanaka, Y., Suzuki, N., Bic, L. Kobayashi, S. 2003. A Mobile-Agent-Based PC Grid. Autonomic Computing Workshop. Fifth Annual International Workshop on Active Middleware Services (AMS'03), June 25th. Seattle, WA.

[Hey TrefethenHey Trefethen2003] Hey, T. Trefethen, A. 2003. The Data Deluge. An e-Science Perspective. F. Berman, G. Fox A. J. G. Hey (), Grid Computing - Making the Global Infrastructure a Reality (809–824). John Wiley and Sons Ltd.

[Johnston BrookeJohnston Brooke2002] Johnston, W. Brooke, J. 2002, July. Core functions for production grids. (Working Draft, Grid Core Functions Architecture Working Group, Global Grid Forum)

[Overeinder, Wijngaards, N., Steen BrazierOvereinder .2002] Overeinder, B., Wijngaards, N., Steen, M. v. Brazier, F. 2002, April. Multi-Agent Support for Internet-Scale Grid Management. O. Rana M. Schroeder (), Proceedings of the AISB'02 Symposium on AI and Grid Computing (18–22).

[Sesseler AlbayrakSesseler Albayrak2001] Sesseler, R. Albayrak, S. 2001. Service-ware framework for developing 3g mobile services.