

Perspectives on Grid Computing

Uwe Schwiegelshohn¹ Rosa M. Badia² Marian Bubak³ Marco Danelutto⁴
Schahram Dustdar⁵ Fabrizio Gagliardi⁶ Alfred Geiger⁷ Ladislav Hluchy⁸
Dieter Kranzlmüller⁹ Erwin Laure¹⁰ Thierry Priol¹¹ Alexander Reinefeld¹²
Michael Resch¹³ Andreas Reuter¹⁴ Otto Rienhoff¹⁵ Thomas Rüter¹⁶ Peter Sloot¹⁷
Domenico Talia¹⁸ Klaus Ullmann¹⁹ Ramin Yahyapour²⁰ Gabriele von Voigt²¹

Abstract

Grid computing has been subject of many large national and international IT projects in the past. However, not all goals of these projects have been achieved. Particularly, the number of users are lagging behind the forecast of many grid projects. This underachievement may have caused the claim that the grid concept is on the way to be replaced by Cloud computing and various X-as-a-Service approaches. In this paper, we try to analyze the current situation and to identify promising directions for future grid development. Although there are shortcomings in current grid systems, we are convinced that the grid concept is still valid and can benefit from new developments like Cloud computing. Further, we strongly believe that some future applications need this concept and that in turn more research is required to turn this concept into reliable, efficient and easy to use implementations.

Key words: Grid computing, virtual research environment, grid applications, grid middleware, Cloud computing

1. Introduction

During the last sixty years the general impact of computing has increased continually. On the one hand, computing has become ubiquitous by using microcontrollers to embed compute agents into many devices of our daily life [50]. On the other hand, the execution of large scale simulation tasks and the use of large data bases have become key elements for problem solving in many disciplines, from natural sciences via engineering even to the humanities. It is obvious that networks are indispensable in ubiquitous computing as each individual device only possesses a very limited view of the world and it is only by sharing information with other devices that many desired goals, like increasing energy efficiency in a building, can be achieved. Networks also play an important role in high performance computing as there are only a limited number of

¹ TU Dortmund University, corresponding author
² CSIC and BSC, Barcelona
³ AGH University of Science and Technology Krakow PL, and Universiteit van Amsterdam NL
⁴ University of Pisa
⁵ Technical University of Vienna
⁶ Microsoft Research
⁷ T-Systems Sfr
⁸ Slovak Academy of Sciences Bratislava
⁹ Ludwig-Maximilians-Universität München
¹⁰ Royal Institute of Technology (KTH) Stockholm
¹¹ INRIA Rennes
¹² Zuse-Institute Berlin
¹³ HPC Center Stuttgart (HLRS)
¹⁴ EML Research and TU Kaiserslautern
¹⁵ Georg-August-University Göttingen
¹⁶ IBM, Germany
¹⁷ University of Amsterdam
¹⁸ ICAR-CNR and University of Calabria
¹⁹ DFN, Germany
²⁰ TU Dortmund University
²¹ Leibniz Universität Hannover

supercomputers in the world and in general, users cannot be expected to relocate close to these supercomputers in order to local gain access to them. Therefore, remote computing via networks is the method of choice for high performance computing. Further, a significant number of companies also uses networks in a similar fashion when outsourcing their IT tasks to an external provider with the goal to reduce IT expenses. This approach is often based on the observation that large compute installations administered by experts can be run more efficiently than scattered installations of the same aggregated size.

The performance and reliability of our IT networks has led to the idea of extending the concept of remote computing to a significant larger set of users similar to the provisioning of electric power to every user even in the most remote place in a country with the help of power nets. Borrowing the name from the electrical power infrastructure, grid computing was born. Particularly for users who do not know or cannot afford to manage suitable IT systems by their own and who have no access to a sufficiently large local compute center, grid computing is particularly appealing. The idea was so convincing that for more than ten years, grid computing has been promoted as the global computing infrastructure of the future [24,5]. For instance, *Jeremy Rifkin* considered it as one of sources of the impact of scientific and technological changes on economy and society [37].

This more or less generally accepted potential of grid computing has convinced many countries to provide significant investments into the generation of a grid computing infrastructure during the past years, like, for instance, the D-Grid initiative in Germany²² [42,25], Grid'5000 in France [8], DAS in The Netherlands [4], PL-Grid in Poland²³, NAREGI in Japan²⁴, or Open Science Grid²⁵ [31] and TeraGrid²⁶ in the USA, to name just a few. Some specific variants of grid computing that are closely related to distributed computing, like Enterprise grids²⁷, have already reached production status. The same is true for a few large scale grids spanning many different administration domains in communities that require huge amounts of data and use a hierarchical storage concept, like the IT infrastructure for parti-

cle physics²⁸ [29,6]. But in general, the number of users is still small compared to the forecasts often publicly announced in project descriptions and vision publications. This leads to the question whether the originally claimed goals are achievable at all and if yes what is required to achieve the goals.

Faced with this mood of uncertainty, the authors of this paper feel that it is time to reflect on the foundations of grid computing in order to determine the most promising directions of grid research and development. They met in a Dagstuhl perspectives workshop in February 2009 to discuss the current situation and the likely future evolution of grid computing. This article is based upon the results of this meeting.

The paper is organized as follows: After presenting some definitions of computing grids in Section 2, Section 3 compares the development of grids to the development of the Internet, the most prominent IT success story, showing many similarities to grid computing at first glance. Then Section 4 describes current and future applications that are most likely to benefit from grid computing. Section 5 discusses the most suitable architectural and organizational structure in connection with grids while challenges in the design of grid middleware, the key software component of grids, are subject of Section 6. The relationship between grid and Cloud computing is addressed in Section 7. Finally, we conclude with a summary and a brief outlook in Section 8.

2. What is a Computing Grid?

Any analysis of grid computing is handicapped by the lack of a generally accepted definition of grid computing. In Section 1, we have already mentioned improved efficiency and cost reduction as reasons to introduce grid computing. The focus on these reasons has recently led to the *infrastructure as a service* (IaaS) variant of Cloud computing, see Section 7. In addition, there are other advantages that people expect from grid computing. To discuss these advantages, we start by citing several descriptions of grid computing: A computing grid

- provides ... *coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations* [22].
- ... *coordinates resources that are not subject to centralized control ... using standard, open,*

²² www.d-grid.de

²³ www.plgrid.pl

²⁴ www.naregi.org

²⁵ www.opensciencegrid.org

²⁶ www.teragrid.org

²⁷ www.opengroup.org/ges/

²⁸ www.eu-egee.org/

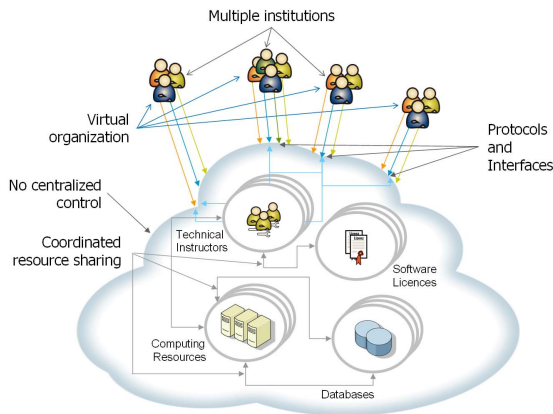


Figure 1. Foster's Model of Grid Computing.

general-purpose protocols and interfaces ... to deliver nontrivial qualities of service [20].

- ... enables resource virtualization, on-demand provisioning, and service (resource) sharing between organizations [34].
- constitutes ... a service for sharing computer power and data storage capacity over the Internet²⁹.

Although we are aware that there are more definitions or descriptions of grid computing we decided to use the list above in our paper as we feel that this list provides a reasonable collection of descriptions. Fig 1 depicts the grid model of Foster et al. [22,20].

These descriptions indicate that grid is not only an infrastructure but in addition or even primarily a paradigm of sharing, selecting, and aggregating of large scale geographically distributed and (possibly) virtualized resources (people, software, data, computers, experiment equipment) depending on availability, capability, cost, and QoS requirements for solving large-scale problems [24]. This paradigm is strongly related to a new approach of organizing researchers within a discipline: the virtual organization (VO) and the virtual research environment (VRE). Therefore, we use the following definition:

A computing grid is a distributed system that supports a virtual research environment across different institutions.

This definition is closely related to Foster's definition [22]. However, it remains vague regarding the degree of VRE support. For instance, video conferences via the Web constitute some form of VRE support. Therefore, the Web already provides some

minimal support for a VRE and can formally be considered a computing grid using our definition. But if we require more extensive support, no other presently used version of a distributed system satisfies our definition.

3. Lessons to Be Learned from the Internet

Note that the last of the four cited definitions in Section 2 emphasizes the role of the Internet in grid computing. But the Internet is not only a vehicle for grid computing, the Web already satisfies some of mentioned characteristics of the grid, see Section 2:

- (i) We already share resources over the Internet, like, for instance, information, music, and movies.
- (ii) Via the Internet, we have access to a multitude of Web services based on general purpose protocols and interfaces.
- (iii) The Internet is highly dynamic and its resources are not subject to centralized control.

Therefore, it is not really surprising that some people simply consider grid computing as the next evolutionary step in Internet development and have selected the name *Next Generation Internet* [26] for grid computing. But in academic research, Internet technology has become indispensable nowadays while many grid projects still lack a sufficient number of users. Hence, is the name *Next Generation Internet* really appropriate? To this end, we compare the development of the Internet with the present state of grid computing. This comparison may also help us in determining appropriate directions for future grid research and development.

3.1. Initial situation

The initial situations of Internet and grid computing development differ significantly: The Internet started nearly from scratch while grid technology is based on substantial experiences in distributed computing and on a well established network infrastructure. Due to the lack of previous experiences during the first days of the Internet, even small advances were welcome although some design decisions during the development of the Internet technology later proved to be unsuitable and had to be amended. In such cases, older technology was typically replaced step-by-step with new and more suitable components instead of possibly waiting a long time for the perfect solution. We have stated already that expect-

²⁹ cdsweb.cern.ch/record/976156/files/it-brochure-2006-002.pdf

tations in grid technology were too high at the beginning, maybe due to previous experiences with distributed systems and the Internet. Moreover, compatibility with older systems is often required due to an existing relationship to other systems. The combination of these expectations and requirements has produced a high threshold that is almost impossible to be reached by any new system immediately. Therefore, we feel that a step-by-step approach of the early Internet would also be appropriate for developing grid technology despite the different initial situation. To this end, we first need to identify the key properties and requirements of grid technology and their future potentials. Then we may continue by enhancing existing technologies for networks and distributed computing with components implementing these properties and requirements.

3.2. *Infrastructure*

One of the key success factors of the Internet is its stable infrastructure that serves as a foundation for numerous Internet services today. This infrastructure provides a vehicle for service delivery which, from a user's point of view, is **reliable, easy to access**, and through appropriate market developments **relatively cheap**. A similar infrastructure will also help in the adoption of grid computing as grid technology is supposed to perform an intermediate delivery function for applications. Unfortunately, most grid infrastructures of today do not yet satisfy the characteristics mentioned above. We believe that even excellent results in grid research are not sufficient to convince grid providers and grid users if an appropriate infrastructure is not available. Therefore, specific activities must be devoted towards establishing such an infrastructure even if its initial functionality does not satisfies all properties of grid computing. In our view, Cloud computing is a step in this direction.

3.3. *The use of standards*

Internet technology is based on (quasi-)standards on all technical levels. For the lower (physical) level, like the Ethernet, standardized specifications have cleared the way to mass products that are both reliable and cheap. On higher levels, standards are necessary to achieve interoperability with the help of professionally produced and maintained middleware. As grid middleware is likely to be more com-

plex than Internet middleware, the observation of standards is even more important. Therefore, we agree with Foster [20] and believe that it is important to leverage the standardization process for grid systems and to use standards in future grid applications in the most flexible way.

3.4. *Scaling*

It is well established that Internet technology scales with respect to the number of users, available bandwidth and others aspects while the actual usage numbers do not allow validating such a claim for grid computing. However, scalability is a key issue for widespread use of grid technology. Therefore, it is crucial to further address this issue especially regarding important properties of the infrastructure, like stability and reliability for users. To this end, we need a careful analysis of the few existing production grids and commercial IaaS installations to determine potential problems.

3.5. *Sharing*

The Internet is a shared resource. This sharing is executed on the technical level and usually invisible to a user in modern networks. Although our definitions point out the importance of the resource sharing paradigm it turned out that boundlessness of this feature is not desired by everyone: Most grid users only want to share their resources, like data and compute power, with a well defined and restricted set of users and therefore may be reluctant to join grid computing. Hence, we believe that it is important to modify the old sharing paradigm to concepts that support well-defined collaborations on all scales and to develop new tools that support this selective resource sharing.

4. **Grid Applications**

The world wide acceptance of the Internet is based to a large extend on the multitude of Internet and Web applications. We are convinced that a successful establishment of large scale computing grids similarly requires key applications that benefit from the grid. Many grid projects exist that support a multitude of different applications from different disciplines including art and humanities [7]. For instance, in *Open Science* data, results, methods and

software tools are freely available enabling massively distributed collaboration supporting further development of the virtual research environments. myExperiment³⁰ [39], originally developed to support workflow applications on grid systems is considered as an example of *Science 2.0* tools. Further with respect to production status, there is one area that already today heavily uses grid computing and requires a large scale production grid: the evaluation of particle collision experiments particularly the Large Hadron Collider (LHC). The corresponding applications are characterized by a very large volume of scientific data that is shared among many research groups. These research groups need a significant amount of computing power to evaluate the data and produce new data that are made available to other groups as well. To support the researchers, an compute environment named the Worldwide LHC Computing Grid (WLCG) has been established. This environment monitors the sites participating in the grid and assures some quality of service. Although the CERN description of a grid used in Section 3 strongly resembles the IaaS Cloud computing variant, the WLCG also satisfies grid properties as defined by Foster et al. [20]. Moreover, this concept is not singular, as many new ESFRI (European Strategy Forum on Research Infrastructures) projects of the EU plan to establish similar research infrastructures. Looking at these projects, we conclude that a typical grid application is related to modeling and simulation of technical systems and real world processes and accesses very large databases.

Modeling of systems and processes for scientific purposes started around 1960 with rather simple mathematical tools. Methodological shortcomings, limited computer resources and underestimation of complexity were the key reasons for the failure of this approach at that time. The advent of High Performance Computing tools and advanced IT resources revived the idea to model complex systems and helped to produce new scientific methods. Now, the modeling world is governed by parallel computing and networking of resources: It is not a coincidence that the fastest computer for some years was called Earth Simulator [40]. Further, recent advances in experimental techniques such as detectors, sensors, and scanners have opened up new windows into physical and biological processes on many levels of detail. However, it is expensive to

generate and maintain all these data as the LHC experiments clearly demonstrate. In order to obtain the largest possible benefit for these expenses, these data are made available to many research groups that participate in the exploitation of the data. To this end, virtual research environments (VRE) and virtual organizations (VO) must be established [41,38]. Therefore, it is not sufficient to merely provide more compute power combined with storage and data management by establishing more clusters and supercomputers. In addition, we need a computing infrastructure that supports VOs and is part of a VRE to be well suited for these problems. According to our definition in Section 3, such an infrastructure is called a computing grid.

But the availability of such an infrastructure is only a prerequisite to address the mentioned scientific problems. For instance, the forth coming data explosion also requires sophisticated techniques, to register, transport, store, manipulate, and share the data. Further, the management of data quality, the ontologies of the descriptors, and the handling of errors in the data will develop into important prerequisites for valid research processes using these data. As clearly indicated by the current controversy in evaluation of climate data, the long-term storage of original data and their availability is a precondition for transparency of and trust in the these processes. The goal of a suitable VRE can only be achieved once all these problems have been resolved.

Further, we believe that more disciplines will use modeling and simulation to address their problems leading to a increased demand in new methodologies that exploit the available IT infrastructure. For instance, consider the combination of analysis in the biomedical sciences with phenotype data including biosignals. There, we have data from virtually all levels between *molecule and man* and yet there are no models to allow studying these processes as a whole. It is a real complex system: from a biological cell, made of thousands of different molecules that work together, via billions of cells that build our tissue, organs and immune system, to our society, six billion unique interacting individuals. The complete cascade from the genome, proteome, metabolome, physiome to health constitutes multi-scale, multi-science systems and is an example of crossing of many temporal and spatial scales [46]. Therefore, researchers face the challenge to study not only the fundamental processes on all these separate scales, but also their mutual coupling through the scales in the overall system, and the resulting emergent

³⁰ myexperiment.org

properties. Understanding, quantifying and handling these processes is one of the biggest scientific challenges of our time [45,47,27] requiring more advanced, collaborative problem solving environments based on workflow systems [51] and virtual laboratories [9]. The challenge includes understanding how one can reconstruct multi-level systems and their dynamics through computational simulation that connect models to massive sets of heterogeneous and often incomplete data. Conceptual, theoretical and methodological foundations are necessary in understanding these multi-scale processes, dynamic networks, and the associated predictability limits of such large scale computer simulations. An example where new IT methods support new ways to understand complex systems comes from the European ViroLab project ³¹, where novel distributed computing techniques are combined with novel computational methods in order to provide medical doctors with a decision support system for drug ranking, see Sloot et al. [44,43]. Such a virtual laboratory is part of a virtual research environment that is not provided by present distributed and high performance computing concepts.

Looking further into the future, we expect the additional inclusion of environmental data to be a potential next step. However, it will substantially add to the complexity of the model and is therefore not likely to start in the immediate future. Looking at the Web applications of today, we observe that social networks like Facebook ³² [1] or MySpace ³³ [49] are particularly appealing because of the aggregated information provided by links which build complex network structures that reveal previously unknown relationships between nodes representing people or organizations. Other popular services, like Earth ³⁴, mainly exploit one dimension: they serve a single purpose such as finding a restaurant or the shortest route from A to B. Their usefulness increases by adding further spatial and temporal information from many different sources resulting in a multidimensional data space that allows us to address various real world problems. A simple but striking example is the recent tracing of flue activities by analyzing the keywords typed into search engines by users. Present Web applications already offer some

features of a virtual environment but do not access very large data sets.

In general, the benefit of databases increases exponentially with the amount and dimensionality of the available data. Based on access to different databases, sophisticated algorithms [28] will provide various services, up to a one-to-one mapping of real world phenomena into a *virtual world environment*.

5. Architecture of Grids

In Section 1, we mentioned that the model of a computing grid is based upon the electrical power grid. Undoubtedly, there is a lot of similarity between the electrical power grid and computing grids:

- (i) the availability of an efficient network infrastructure reaching almost everyone
- (ii) the efficient generation of (electrical or computing) power in large installations
- (iii) the simple integration of additional distributed providers (solar or wind energy or special hardware)

But while electrical power is simply provided in the elementary form of current and voltage there is a large diversity of resources in the IT world. Even if we restrict ourselves to hardware and operating systems, there exist already a multitude of different options. Moreover, the mere provisioning of hardware and system software is only sufficient for standard applications or for users with significant IT knowledge that program their own applications. Users with little experience in programming need additional support to exploit nontrivial qualities of service [20]. For them, the idea of Software as a Service (SaaS) dedicated to their need is particularly appealing.

To consider this diversity within computing grids, we need to take a closer look into their structure. Based on our already mentioned observations, a grid system consists of the following major groups of components:

- (i) Different hardware resources like processors, storage, networks, and possibly sensors including the corresponding system software
- (ii) Discipline-independent software to manage, for instance, virtual organizations, data, and access to the various resources
- (iii) Application software dedicated to the needs of the various virtual organizations within a virtual research environment

³¹ www.virolab.org

³² www.facebook.com

³³ www.myspace.com

³⁴ earth.google.com

This classification of grid components strongly suggests to represent the organizational structure of a grid by using layers: Hardware forms the bottom layer encapsulated by a management software layer. The third layer consists of the discipline-specific application software and uses the services of the management software layer while any user-specific application forms the top layer and only accesses the services of the discipline-specific software layer, see Fig. 2.

Let us look closer into the various layers: In general, the hardware layer contains a large number of heterogeneous resources. But not all users are allowed access to all resources. For instance, medical data is available to users with special access permission to guarantee data privacy protection of patients. It is one task of the management layer to guard this access. Further, the management layer must guarantee seamless interplay between the resources while hiding most heterogeneity from the user. The discipline-specific software layer provides tools that are particularly useful for the target group of users. This layer constitutes the visible part of the virtual research environment (VRE) of a discipline. In an ideal world, the user only needs to configure the tools of the VRE to create the desired application. It is important to realize that there will be many different implementations of the first and the third layers due to the large number of different resource types in the grid and due to the different disciplines participating using grid computing.

Note that many existing resources come with their own (legacy) software and many users also rely on existing software. The described diversity in combination with legacy systems creates a design challenge particularly for the second layer as it is almost impossible to find designers and administrators who overlook a large system like a computing grid. This design problem is significantly larger than its counterpart during the development of the Internet that does not face so much diversity nor many legacy components. To address this challenge, teams are typically formed with different responsibilities.

These teams must cooperate to establish such a layered structure. To support such cooperation, we consider the use of standard, open, general-purpose protocols and interfaces to be indispensable, see also Foster et al. [20]. There are already a few examples for successful grid standards, see [36,21,2]. But as all these protocols and interfaces must be carefully maintained it is also necessary to control their number and their complexity. This is particularly true

in a grid containing many heterogeneous resources and VREs. Unfortunately, the attempt to address the heterogeneity of resources and applications in a broad way has produced large and complex grid middleware stacks that often exhibit a high operational complexity as well, see also Section 6. To show the reason for this complexity we present a simplified grid job submission process:

At the beginning of the submission process, a user must authenticate himself to gain access to the required resources. To support user migration between different VREs, the authentication service is common to all VREs and should be part of the management software layer. After authentication another management software service determines VO membership information using data of the appropriate VRE. Now the user is able to access the software services of his VRE and to build a job request. His credentials are added to the grid object representing this job request. This object is then again forwarded to the management layer where a service uses the properties of the grid object to initiate a data query possibly followed by a computation request in some other resource. Another service of the management software layer is responsible for the data transfer to the computing resource and finally back to the VRE. There yet another service handles visualization of the results and embeds it into the software of the user.

The example shows that the process may flow back and forth through the layers. Of course this flow depends on the **assignment of components and functionality to layers**. This assignment determines **performance** and **manageability** of the architecture. The flow is also important for service level agreements (SLA): In case of a failure, the responsible component must be determined as the various grid components are likely to be owned by different organizations. Therefore, SLAs typically require a **tracing functionality** that allows a reliable failure localization. Unfortunately, it is generally difficult to reverse an assignment decision once made. Therefore, we suggest the extensive evaluation of **use cases** to determine appropriate assignments and interfaces in the architecture.

Further, the described complexity is frequently an obstacle to the integration of new resources or applications and therefore reduces flexibility. We see two main alternatives to address this problem: On the one hand, we can strive for less complex middleware systems with reduced functionality, see Section 6. On the other hand, we can try to manage this complexity by enforcing strict guide lines during

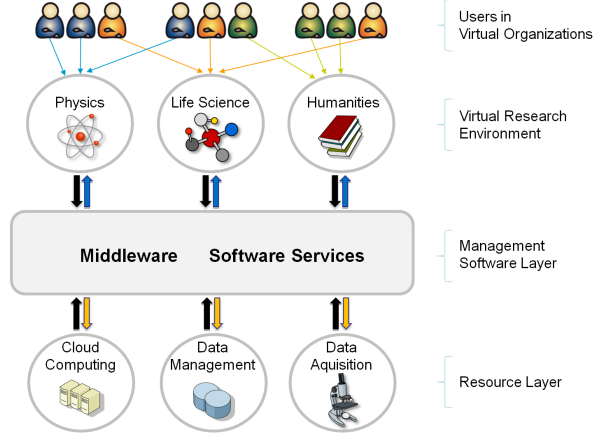


Figure 2. Layer Structure of Grids.

development and operation. However, as the latter approach bears the risk that these guidelines negatively affect development of grid technology, we prefer the first one.

6. Grid Middleware

The middleware plays a prominent role in distributed systems as it connects the different parts of such a system and thus generates a useful environment. Therefore, the grid middleware is the major part of the management software layer. It is often based upon the middleware of distributed systems, using some or all of the following components [35]:

- Communicating processes that enable the exchange of messages (sockets, MPI, PVM). These processes are often supported by application programming interfaces (APIs).
- Remote procedure calls that transfer control from one process to another one. They include an interface definition and implement a programming language entity (a function).
- Objects within the Distributed Object Oriented Architecture, that enable a transparent access to objects within a distributed system. They implement a programming language entity (an object) as well.
- Components using the Component Oriented Architecture that allow a modular system regarding composition, configuration, and deployment. They are supported by programming language entities (extensions of objects).
- Services within a Service Oriented Architecture that provide functionality independent of a spe-

cific platform. Services do not directly map to a programming language entity.

Due to the existent middlewares of distributed systems and different VREs looking for grid solutions, there are currently multiple versions of grid middlewares often named middleware stacks, like gLite [30], Globus Toolkit [23], UNICORE [18], ARC [16], and others. These stacks generate a new type of heterogeneity as they provide different functionalities and serve different application domains although most of them support the Open Service Grid Architecture (OGSA) that was developed within the Global Grid Forum (GGF), now the Open Grid Forum (OGF)³⁵. But even the development of the OGSA was not straight forward: First, it was based on the Open Grid Service Infrastructure (OGSI). Later, the OGSI was more or less replaced by the Web Service Resource Framework (WSRF). Now, the new Globus Toolkit Version 5 does not include anymore the Web service components of Globus Toolkit Version 4. Unfortunately, none of the available stacks is generally accepted by all user communities. As they do not yet support common interfaces the European Grid initiative (EGI)³⁶ intends to initiate a standardization effort called unified middleware distribution (UMD).

Moreover, in their present state, we must consider these middleware stacks to be still in the state of **experimental software**, that tries to solve problems with a best effort approach. This is even true for gLite that is used in the WLCG. Although it is not

³⁵ www.ogf.org

³⁶ www.egi.eu

possible to identify a single reason for the current state and diversity of grid middleware implementations we believe that past funding approaches, lack of sufficient basic research, and partial use of technologies that were not yet mature enough significantly contributed to the present situation. In our view, a new approach to design future grid middleware is required.

6.1. *Grid applications and past experiences*

During the past ten years, different implementations of grid middleware have been tested with various applications. To satisfy the claim of applicability of grid technology to many different disciplines while preserving existing structures, developers sometimes simply integrated new functionalities into existing software systems. This has led to a loss of structure within and between the layers and increased complexity. We believe that in the future, the definition of **a minimal set of grid middleware components** is necessary. These components must be based on commonalities between different application functionalities and provide a sufficient foundation to support these applications. Specific requirements of the application must be satisfied by separate services within the management software layer or the VRE layer. To design such a middleware - either from scratch or as a modification of existing grid middleware - we should exploit experiences from past grid projects, carefully analyze grid applications and determine how to cluster them according to the functionalities they require from a middleware system. Similarly, the requirements of resource providers must also be taken into account.

This analysis must lead to a definition of grid middleware functionalities that may then be split into components and implemented individually following common software engineering principles. As already mentioned this process must be accompanied by standardization efforts to provide well-defined interfaces between the different components of the grid architecture. Of course, due to further development of grid applications, grid middleware must be able to consider changing requirements over the course of time.

6.2. *Experimental and production middleware*

We have already mentioned that experimental grid middleware is used to run production grid sys-

tems often resulting in user disappointment as the expected production quality of the grid systems could not be achieved. We see a main reason for this problem in funding strategies that have put significant resources towards development of production grids while separate software research projects were supported on a smaller scale. Therefore, these research projects were rarely exposed to real life input data preventing to test scalability and performance issues. Of course, it is generally impossible to carry out experiments on a production grid middleware itself without generating threats to the reliable execution of user applications. But we strongly believe that these experiments are necessary to improve the maturity of grid middleware.

To handle this dilemma, we suggest to establish a development process of a future grid middleware that consists of two parts: a **research part** where computer scientists seek the next breakthrough in technology with the help of prototype implementations and experiments, and an **engineering part** that ensures production quality of the grid middleware used for grid applications and provides feedback to the researchers. Both parts must be granted their share of access to the resource. Further, a close collaboration between researchers and engineers is required to identify problems when operating a production infrastructure and to generate solutions that have been intensively and rigorously tested using experimental large scale grid platforms. To our knowledge, only few initiatives have been started so far to overcome this situation, like DAS in the Netherlands or the French Grid'5000.

6.3. *Middleware interoperability*

In general, interoperability is the ability of a system to cooperate with or to exploit functions or services of another system possibly belonging to a different organization. More particular, the *Grid gurus website*³⁷ states:

Interoperability is the native ability of grids and grid technologies to interact directly via common open standards.

Other definitions of grid interoperability have been given by Field and Schulz [19]. While grid interoperability can be achieved in different layers it is usually assumed that the middleware is responsible for

³⁷ gridgurus.typepad.com/grid_gurus/2008/06/grid-interopera.html

it. To describe the present situation of grid middleware regarding interoperability, we again employ a comparison between the Internet and grids, see Section 3:

Anyone is capable of addressing, but not always accessing for obvious security reasons, any physical node or software service available on the Internet. This property is achieved by a single distributed DNS service that is made available to all operating systems and middlewares. Now consider an alternative Internet with several distinct and incompatible DNS systems controlled by various organizations that manage their own set of routers. It is highly unlikely that this alternative approach would have produced the same success.

Yet, this scenario of the alternative Internet resembles the current grid situation where several grid middleware stacks co-exist today managing their own infrastructures and resources. These stacks create isolated silos instead of providing a global pool of addressable resources. There is still a lack of interoperability between existing stacks although several attempts have been made to investigate this issue in various EU-funded projects like UNIGRID, GRIP or DataTAG [14]. Grid interoperability was also the subject of an intense collaboration between European and US teams managing grid infrastructures such as the Open Science Grid and EGEE. But most of these projects only consider interoperability between no more than two existing grid middleware stacks instead of addressing it in a more global way. D-Grid uses another approach and tries to address this problem by implementing the compute elements of the middlewares UNICORE, gLite, and Globus on each cluster such that the worker nodes can be accessed by each middleware [42].

But is interoperability really as important in grids as it is in the Internet? To answer this question, it is necessary to determine the benefits of interoperability in grids.

- **User interoperability** allows users to seamlessly migrate from one VRE to another. However, we expect that the VRE hides middleware functionality from the user. Therefore, user interoperability can be achieved on the VRE level and does not necessarily need middleware interoperability.
- **Resource interoperability** enables users from different VREs to use the same resources without requiring the resource provider to install and maintain different grid middlewares. Resource interoperability is certainly beneficial if resources

are really used by applications from different VREs. This is likely to be true for compute resources and corresponding system software. However, it is doubtful that high energy physicists will really use the databases of climate researchers.

- **Design interoperability** permits the transfer of a design concept from one VRE to another. As there are significant differences between the state of grid development for the various disciplines VRE, development is likely to happen in a staggered fashion. Here interoperability will help to exploit synergies and reduce development cost.

We believe that currently user interoperability is not yet a convincing reason but it may become more important in the future with the further advance of interdisciplinary research. Then, for instance, a physician may be interested to evaluate past climate data to study the spread of a disease over time. Resource interoperability regarding compute and storage resources is usually achieved with the help of virtualization. The above mentioned advance of interdisciplinary research in the future may also increase the importance of data interoperability. It is generally acknowledged that many grid applications need to access and share multiple data sources that are widely distributed [33]. Due to the variety and heterogeneity of both applications and data-related resources particularly in interdisciplinary research, the middleware must support higher-level models and services that assist users in exploiting several data repositories within an application. Frequently, these data repositories have been created by different organizations that often reserve the right to make local, independent decisions about their best data management paradigm. We believe that these organizations do not want to implement interoperating software nor that grid users want to address these interoperability problems. In order to allow middleware to handle them, common data formats as well as a common metadata format and structure must be defined. Finally, design interoperability is particularly important for the development of grid systems for new disciplines and particularly concerns service interoperability. This service interoperability mainly uses two basic mechanisms: standard interface description possibly with proper metadata carrying associated non functional information, and reflection, supporting inspection of the interfaces and of the information associated to these interfaces. A specific part of service interoperability is security and authentication interoperability. The agreement on a common, shared security infrastructure is actu-

ally a condition *sine qua non* for any interoperability in grids. Interoperability simply cannot be achieved without an agreement concerning the security procedures. This agreement must preserve the disjoint security and authentication mechanisms of the co-operating grids.

6.4. Middleware realization

We have already stated that in our view, the size and the functionality of existing middleware stacks are the main hurdle to achieve production status and interoperability. Therefore, we suggest to reduce the functionality of these stacks, see Section 6.1. In our view, there are several components of current grid middleware stacks that can be considered for *promotion* outside of the actual grid middleware as **independent services** of the management software layer such that the corresponding policies can be accessed by all interacting VREs. Similarly, the DNS is something separate from the Internet protocol layers but actually all the actors in the Internet scenario agreed to it. We have identified the components of the following list to be promising candidates for such promotion:

- VO management,
- accounting and billing,
- resource and job monitoring,
- security (certificate management as a first step).

For instance, the accounting and billing systems should be run in each individual VRE by exploiting the primitive accounting and billing facilities of the management software layer. These primitives provide the data necessary to implement global, inter-grid accounting and billing activities.

In addition, the *migration* of some basic mechanisms in Grid middleware to **operating systems** might be helpful. Promising candidates for this migration are in particular those mechanisms that are considered necessary and universal, and whose implementation is not subject to further research. As an example, the XtremOS project [11] suggests to migrate authentication mechanisms to the Linux operating system level. Note that there are different reasons for both types of migrations:

- Mechanisms are migrated to the operating system level mainly for efficiency reasons and to reduce security pitfalls while moving up policies into external inter-grid entities supports interoperability.
- Transferring a mechanism to the operating systems level requires a general consensus by manu-

facturers or communities developing the operating systems while only an agreement by the people actually using grid interoperability is necessary for promotion of a functionality into a new service.

We believe that a combination of both approaches significantly reduces the size of a grid middleware while still leaving enough space to implement in the grid middleware those mechanisms and policies that are neither fully general nor very basic to be completely independent of the grid implementation at hand.

7. Grid and Cloud Computing

Cloud computing is a new concept that has evolved recently from the *grid world* [10,3]. Cloud computing is based upon a simple business model providing on-demand service. It was started by several companies that have offered various Cloud services. More recently, academia and funding organizations alike consider Cloud computing to be interesting as it may improve flexibility and reduce fixed costs [13]. The simplicity of the business model and the commercial success of Cloud computing may have caused many companies to move away from grid computing and adopt Cloud computing instead. In order to determine the future development of grid computing in comparison to Cloud computing, it is first necessary to distinguish grid and Cloud computing. Unfortunately, such a distinction turns out to be rather difficult as there is no generally accepted definition for each approach. To describe grid computing, we cited several publications, see Section 3. For Cloud computing, we first describe its three major variants which are used in most publications covering this subject:

- (i) Infrastructure as a Service (IaaS) mainly provides compute and storage resources to users. The special features of the installation base are hidden with the help of a virtualization layer. This enables a multitude of applications to use the infrastructure. Amazon's Elastic Compute Cloud (EC2) is an example of IaaS.
- (ii) Platform as a Service (PaaS) offers hardware resources and some additional software layers. In comparison to IaaS, it provides more support for the development of applications. Microsoft's Azure is an example of PaaS.
- (iii) Software as a Service (SaaS) simply delivers software on demand to users without requiring

the user to install and run the application on his own computer. Google's Doc is one of many examples for SaaS.

The literature further distinguishes between private clouds that are operated within an enterprise and public or community clouds that are available to users from different institutions or companies. In our context of grid computing, only public clouds are relevant. Although some definitions of grid computing come very close to IaaS, see Section 3, most people agree that virtual organizations and virtual research environments are not included in the Cloud computing business model. In addition to the lack of centralized control of resource coordination [20], this constitutes the major difference between Cloud and grid computing.

Looking at current grid projects, we can realize that several concepts of Cloud computing are already included into grid computing:

- VREs have partially adopted the SaaS approach and (intend to) provide their users with application software on demand that is run on remote grid resources.
- Resource providers make increasing use of virtualization and may provide some form of IaaS within a grid.
- The importance of business models is gaining increasing attention even in research. Of course the sources of funding for research will generally be others than for commercial activities.

Nowadays, public research is under similar constraints as commercial activities. Therefore, flexibility and cost reduction have become substantial issues in the academic world as well. Hence, Cloud computing may become attractive as an alternative to academic in-house computing if academic users can actually benefit from Cloud computing. For researchers, the key aspect will be general availability and ease of use of the services. We believe that end users will be happy with having a specified and guaranteed level of service provided at reasonable cost within the standardized environment of their VRE and will not insist on a locally provided service by their compute center [15].

However, before remote on-demand services will also spread in the academic world, some obstacles must be removed. For instance, consider the management of licenses. We have already pointed out that it is not the pure cycle selling that makes the future grid attractive but the full software service. Therefore, independent software vendors (ISVs) will play an important role. Through their license poli-

cies they will be able to drive the future development. However, as of today, grid computing and VREs are often considered as a menace to existing licensing models. Future strategies must be more flexible and need to create a win-win situation for everybody: ISVs, managers of VREs, and users.

8. Summary and Conclusions

To evaluate the perspectives of grid computing, several aspects are analyzed in this paper. At first the evolution of grid computing is compared with the evolution of the Internet. Although grid computing heavily relies on the Internet and the Internet already provides some services similar to grid computing we came to the conclusion that grid computing is likely to evolve differently than the Internet. For instance, we do not believe that grid computing will rapidly be adopted by a mass market. Therefore, the name *Next Generation Internet* for grid computing is misleading in our view. We also identified various key aspects in the development of the Internet that are currently missing for grid computing but would be helpful in accelerating grid development. Altogether, we think that a comparison to the evolution of the Internet may provide valuable hints to determine promising directions of grid research.

Further, it is our strong belief that the future of grid computing will depend on key applications that require the dynamic use of large-scale compute and data resources. We already see an increasing interest in these applications in academia. For instance, the trend towards a virtual representation of the real world may provide such an application scenario. Further, the smart combination of online data from sensor networks and arbitrary archives on the one hand and computing facilities on the other hand will provide novel services. The benefit of these services may not be restricted to scientific fields but also reach into industrial and societal domains. Moreover, the use of large data-resources requires sharing to distribute cost and invariably leads to virtual (research) environments that are one of the cornerstones of grid computing. Therefore, we suggest that future grid projects shall focus on the generation of these virtual (research) environments.

The once foreseen global and open grid infrastructure did not emerge yet. So far, a general need for interoperable Grids has not been demonstrated. But we can already observe more specific forms of interoperability that can be achieved on application and

middleware levels [12,48,32]. The realization of these forms of interoperability requires a mature and reliable middleware. Unfortunately, current Grid middlewares implementations do not only fail to inter-operate seamlessly but they are also too complex to allow quick appropriate modifications. For the sake of reducing this complexity, we suggest that some of the necessary grid functionality should be moved from the middleware to other components. For instance, security and data integration are key requirements which can be integrated into the operating system. Other functions, like meta-scheduling and brokering, can become independent services that are possibly deployed on an application level. To solve these middleware related problems, we feel that the mere investment in new productions grids is not sufficient. In addition, support of more basic research in the grid domain is required.

The use of Cloud computing and virtualization techniques may generate new ways to provide and to utilize resources. Virtual research environments may heavily use the SaaS concept. Therefore, we believe that it would be a big mistake to consider Cloud computing as a competitor to grids as both approaches can benefit from each other. We strongly urge not to waste our time in simply redefining terms or key technologies: clusters, grids, Clouds... what is in a name? Here we simply refer to Ian Foster who recently³⁸ quoted Miron Livny saying: "*I was doing Cloud computing way before people called it Grid computing*", referring to the Condor technology [17]³⁹.

Acknowledgments. This work is the outcome of a Dagstuhl Perspectives Workshop on the Future of Grid Computing (Seminar 09082) which took place from February 15 to February 20, 2009. The authors would like to express their gratitude for the generous support by the Dagstuhl staff and everyone who contributed to the success of the seminar.

References

- [1] Alessandro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. *Lecture Notes in Computer Science*, 4258:36–58, 2006.
- [2] Ali Anjomshoaa, Fred Brisard, Michel Drescher, Donal Fellows, An Ly, Stephen McGough, Darren Pulsipher, and Andreas Savva. Job Submission Description Language (JSDL) Specification. Technical Report GFD-R.056, Open Grid Forum (OGF), 2005.
- [3] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report EECS-2009-28, EECS at UC Berkeley, 2009.
- [4] Henri Bal, Raoul Bhoedjang, Rutger Hofman, Ciel Jacobs, Thilo Kielmann, Jason Maassen, Rob Van Nieuwpoort, John Romein, Luc Renambot, Tim Rül, Ronald Veldema, Kees Verstoep, Aline Baggio, Gerco Ballintijn, Ihor Kuz, Guillaume Pierre, Maarten Van Steen, Andy Tanenbaum, Gerben Doornbos, Desmond Germans, Hans Spoelder, Evert jan Baerends, Stan Van Gisbergen, Hamideh Afsermanesh, Dick Van Albada, Adam Belloum, David Dubbeldam, Zeger Hendrikse, Bob Hertzberger, Alfons Hoekstra, Kamil Iskra, Drona K, Dennis Koelma, Frank Van Der Linden, Benno Overeinder, Peter Sloot, Piero Spinnato, Dick Epema, Arjan Van Gemund, Pieter Jonker, Andrei Radulescu, Cees Van Reeuwijk, Henk Sips, Peter Knijnenburg, Michael Lew, Floris Sluiter, and Lex Wolters. The distributed ASCI supercomputer project. *The International Journal of Supercomputer Applications and High Performance Computing*, 34:76–96, 2000.
- [5] Henri Bal, Cees de Latt, Seif Haridi, Keith Jeffery ad Jesus Labarta, Domenico Laforenza, Peter Maccallum, Joan Massó, Ludek Matyska, Thierry Priol, Alexander Reinefeld, Andreas Reuter, Michel Rigiguidel, Dave Snelling, Martin van Steen, and Roman Tirlir. Next generation grid(s) - european grid research 2005 - 2010. ftp://ftp.cordis.europa.eu/pub/ist/docs/ngg_eg_final.pdf, June 2003.
- [6] Ian Bird, Bob Jones, and Kerk F. Kee. The organization and management of grid infrastructures. *IEEE Computer*, 42(1):36–46, 2009.
- [7] Tobias Blanke, Mark Hedges, and Stuart Dunn. Arts and humanities e-science—current practices and future challenges. *Future Generation Computer Systems*, 25(4):474 – 480, 2009.
- [8] Raphaël Bolze, Franck Cappello, Eddy Caron, Michel Daydé, Frédéric Desprez, Emmanuel Jeannot, Yvon Jégou, Stéphane Lantéri, Julien Leduc, Noredine Melab, Guillaume Mornet, Raymond Namyst, Pascale Primet, Benjamin Quetier, Olivier Richard, El-Ghazali Talbi, and Touche Iréa. Grid’5000: a large scale and highly reconfigurable experimental grid testbed. *International Journal of High Performance Computing Applications*, 20(4):481–494, November 2006.
- [9] Marian Bubak, Tomasz Gubala, Maciej Malawski, Bartosz Balis, Włodzimierz Funika, Tomasz Bartynski, Eryk Ciepiela, Daniel Harezlak, Marek Kasztelnik, Joanna Kocot, Dariusz Krol, Piotr Nowakowski, Michal Pelczar, Jakub Wach, Matthias Assel, and Alfredo Tirado-Ramos. Virtual laboratory for collaborative applications. In Mario Cannataro, editor, *Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine and Healthcare, Information Science Reference*, chapter XXVII. IGI Global, 2009. ISBN: 978-1-60566-374-6.

³⁸ Ian Fosters presentation at WorldComp 2009, Las Vegas, July 13th, 2009

³⁹ www.cs.wisc.edu/condor/

- [10] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, and James Broberg and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599 – 616, 2009.
- [11] Massimo Coppola, Yvon Jégou, Brian Matthews, Christine Morin, Luis Pablo Prieto, Óscar David Sánchez, Erica Y. Yang, and Haiyan Yu. Virtual organization support within a grid-wide operating system. *IEEE Internet Computing*, 12(2):20–28, 2008.
- [12] Peter V. Coveney, Giovanni Giupponi, Shantenu Jha, Steven Manos, Jon MacLaren, Stephen M. Pickles, Radhika Saksena, Thomas Soddemann, James Suter, Mary-Ann Thyveetil, and Stefan J. Zasada. Large scale computational science on federated international grids: The role of switched optical networks. *Future Generation Computer Systems*, 26(1):99 – 110, 2010.
- [13] Marios D. Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis, and Athena Vakali. Cloud computing: Distributed internet computing for it and scientific research. *IEEE Internet Computing*, 13(5):10–13, 2009.
- [14] Flavia Donno, Vincenzo Ciaschini, David Rebatto, Luca Vaccarossa, and Marco Verlatto. The worldgrid transatlantic testbed: a successful example of grid interoperability across eu and u.s. domains. In *Proceedings of CHEP 2003 La Jolla - CA, USA*, 2003.
- [15] Jorge Ejarque, Marc de Paloland Íñigo Goiri, Ferran Julià, Jordi Guitart, Rosa M. Badia, and Jordi Torres. SLA-driven semantically-enhanced dynamic resource allocator for virtualized service providers. In *Proceedings of the e-Science conference*, 2008.
- [16] Mattias Ellert, Michael Grönager, Aleksandr Konstantinov, Balázs Kónya, Jonas Lindemann, Ilja Livenson, Jakob Langgaard Nielsen, Marko Niinimäki, Oxana Smirnova, and Anders Wäänänen. Advanced resource connector middleware for lightweight computational grids. *Future Generation Computer Systems*, 23:219–240, 2007.
- [17] Dick H. J. Epema, Miron Livny, René van Dantzig, Xander Evers, and Jim Pruyne. A worldwide flock of condors: Load sharing among workstation clusters. *Future Generation Computer Systems*, 12(1):53 – 65, 1996. Resource Management in Distributed Systems.
- [18] Dieter Erwin. UNICORE—a Grid computing environment. *Concurrency and Computation: Practice and Experience*, 14(13-15):1395–1410, 2002.
- [19] Laurence Field and Markus Schulz. Grid interoperability: the interoperations cookbook. *Journal of Physics: Conference Series*, 119(1):012001 (8pp), 2008.
- [20] Ian Foster. What is the Grid? A Three Point Checklist, *Grid Today*, 1(9), 2002. www.gridtoday.com/02/0722/100136.html
- [21] Ian Foster, Andrew Grimshaw, Peter Lane, William Lee, Mark Morgan, Steven Newhouse, Stephen Pickles, Darren Pulsipher, Christopher Smith, and Marvin Theimer. Open grid services architecture basic execution service specification 1.0. Technical Report GFD-R.108, Open Grid Forum (OGF), 2008.
- [22] Ian Foster, Carl Kesselman, and Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.
- [23] Ian Foster and Carl Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputing Applications*, 11(2):115–128, 1997.
- [24] Ian Foster and Carl Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure (The Morgan Kaufmann Series in Computer Architecture and Design)*. Morgan Kaufmann, November 2003.
- [25] Wolfgang Gentzsch and Alexander Reinefeld. Special section on D-Grid. *Future Generation Computer Systems*, 25:266–267, 2009.
- [26] Douglas Heingartner. The Grid: The Next-Gen Internet? *WIRED Magazine*, 2001 www.wired.com/science/discoveries/news/2001/03/42230
- [27] Alfons G. Hoekstra, Simon F. Portegies Zwart, Marian Bubak, and Peter M.A. Sloot. Towards distributed petascale computing. In David Bader, editor, *Petascale Computing: Algorithms and Applications*, chapter 8, pages 147–164. Chapman and Hall, 2008. ISBN 9781584889090.
- [28] David Jensen and Jennifer Neville. Correlation and sampling in relational data mining. In *n Proceedings of the 33rd Symposium on the Interface of Computing Science and Statistics*, 2001.
- [29] Bob Jones. *An Overview of the EGEE Project*, volume 3664 of *Lecture Notes in Computer Science*, pages 1–8. Springer-Verlag, August 2005.
- [30] Erwin Laure, Steve M. Fisher, tAkohs Frohner, Claudio Grandi, Peter Kunszt, Ales Krenek, Ole Mulmo, Fabrizio Pacini, Francesco Prelz, John White, Maite Barroso, Predrag Buncic, Frederic Hemmer, Alberto Di Meglio, and Åke Edlund. Programming the Grid with gLite. *Computational Methods in Science and Technology*, 12(1):33–45, 2006.
- [31] Miron Livny, Ruth Pordes, Knet Blackburn, and Paul Avery. Open science grid. annual report to doe. Technical report, Open Science Grid, December 2008.
- [32] Maciej Malawski, Tomasz Bartynski, and Marian Bubak. Invocation of operations from script-based grid applications. *Future Generation Computer Systems*, 26(1):138 – 146, 2010.
- [33] María A. Nieto-Santisteban, Jim Gray, Alexander S. Szalay, James Annis, Aniruddha R. Thakar, and William O’Mullane. When database systems meet the grid. In *CIDR*, pages 154–161, 2005.
- [34] Pawel Plaszczak and Richard Wellner Jr. Grid Computing: The Savvy Manager’s Guide. Morgan Kaufmann, 2005.
- [35] Thierry Priol. Objects, components, services for grid middleware, 2005. The keynote lecture at European Grid Conference: Advances in Grid Computing - EGC 2005, Amsterdam, The Netherlands, February 14-16, 2005.
- [36] Hrabri Rajic, Roger Brobst, Waiman Chan, Fritz Ferstl, Jeff Gardiner, Andreas Haas, Bill Nitzberg, Hrabri Rajic, and John Tollefsrud. Distributed resource management application api (drmaa) specification 1.0. Technical Report FD-R-P.022, Open Grid Forum (OGF), 2004.

- [37] Jeremy Rifkin. *The European Dream: How Europe's Vision of the Future is Quietly Eclipsing the American Dream*. Penguin, 2004.
- [38] Matei Ripeanu, Munindar P. Singh, and Sudharshan S. Vazhkudai. Virtual organizations. *IEEE Internet Computing*, 12(2):10–12, 2008.
- [39] David De Roure, Carole Goble, and Robert Stevens. Designing the myexperiment virtual research environment for the social sharing of workflows. In *Third IEEE International Conference on e-Science and Grid Computing, 2007, Bangalore, India; 10-13 December 2007*, pages 603–610, 2007.
- [40] Tetsuya Sato. The earth simulator: Roles and impacts. *Parallel Computing*, 30(12):1279–1286, 2004.
- [41] Tobias Scholl, Bernhard Bauer, Benjamin Guffler, Richard Kuntschke, Angelika Reiser, and Alfons Kemper. Scalable community-driven data sharing in e-science grids. *Future Generation Computer Systems*, 25(3):290 – 300, 2009.
- [42] Uwe Schwiegelshohn. The structure of D-Grid. In Marian Bubak, Michał Turata, and Kazimierz Wiatr, editors, *Proceedings of the Cracow Grid Workshop 2008*, pages 3–10, 2009.
- [43] Peter M.A. Sloot, Alexander V. Boukhanovsky, W. Keulen, Alfredo Tirado-Ramos, and Charles A.B. Boucher. Decision support in individualized e-health. *Journal of Clinical Monitoring and Computing*, 19(4-5):1287–1307, 2005.
- [44] Peter M.A. Sloot, Peter V. Coveney, Gokhan Ertaylan, Viktor Mueller, Charles A.B. Boucher, and Marian Bubak. HIV Decision Support: From Molecule to Man. *Philosophical Transactions of the Royal Society*, 367(1898):2691–2703, 2009.
- [45] Peter M.A. Sloot, Daan Frenkel, Henk A. Van der Vorst, Antoine van Kampen, Henri E. Bal, Paul Klint, Robert M. Mattheij, Jack van Wijka, Joop Schaye, Huib Jan van Langevelde, Rob H. Bisseling, Berend Smit, E. Valenteyn, Henk J. Sips, Jos B.T.M. Roerdink, and Koen G. Langedoen. Computational e-science: Studying complex systems in silico. a national coordinated initiative. white paper. Technical report, Informatics Institute, Universiteit van Amsterdam, February 2007.
- [46] Peter M.A. Sloot, Alfredo Tirado-Ramos, Ilkay Altintas, Marian Bubak, and Charles A.B. Boucher. Decision support in individualized e-health. *IEEE Computer*, 39(11):40–46, 2006.
- [47] Peter M.A. Sloot, Geert Dick van Albada, Marian Bubak, and Anne Trefethen. *2nd IEEE International Conference on e-Science and Grid Computing*. IEEE, 2006. ISBN 0-7695-2734-5; CD.
- [48] Larry Smarr, Maxine Brown, and Cees de Laat. Special section: Optiplanet – the optiputer global collaboratory. *Future Generation Computer Systems*, 25(2):109 – 113, 2009.
- [49] Mike Thelwall. Social networks, gender, and friending: An analysis of myspace member profiles. *Journal of the American Society for Information Science and Technology*, 59(8):1321–1330, 2008.
- [50] Mark Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):94 – 104, 1991.
- [51] Zhiming Zhao, Adam Belloum, and Marian Bubak. Special section on workflow systems and applications in e-science. *Future Generation Computer Systems*, 25(5):525 – 527, 2009.