On Business Grid Demands and Approaches

Carsten Franke, Adolf Hohl, Philip Robinson, and Bernd Scheuermann

SAP Research CEC Belfast, Karlsruhe
{carsten.franke, adolf.hohl, philip.robinson, bernd.scheuermann}@sap.com

Abstract. This paper addresses necessary modification and extensions to existing Grid Computing approaches in order to meet modern business demand. Grid Computing has been traditionally used to solve large scientific problems, focussing more on accumulative use of computing power and processing large input and output files, typical for many scientific problems. Nowadays businesses have increasing computational demands, such that Grid technologies are of interest. However, the existing business requirements introduce new constraints on the design, configuration and operation of the underlying systems, including availability of resources, performance, monitoring aspects, security and isolation issues. This paper addresses the existing Grid Computing capabilities, discussing the additional demands in detail. This results in a suggestion of problem areas that must be investigated and corresponding technologies that should be used within future Business Grid systems.

Keywords: Grid Size, Software Landscapes, Application Data, Execution Characteristics, Autonomy, Service Level Agreements.

1 Introduction

The paradigm of Metacomputing [6] and later Grid Computing originated in the early 1990s and refers to the coupling of geographically dispersed computers, storage systems, scientific instruments etc. [13,10,12]. This enabled the execution of a wide range of scientific applications like e.g. large scale simulations, collaborative engineering or computer-aided instrumentation [6,7].

In the majority of businesses, the ability for their organization and technical infrastructure to adapt to dynamically changing business environments has become a key component in their success. Many businesses are searching for proven, technological solutions that enable them to execute with these new levels of adaptability. In the world of large-scale scientific problem solving, similar adaptability challenges have been faced and already addressed by allocating computational jobs to aggregations of distributed nodes. Although this seems a logical avenue to pursue for solving the business agility problem, such solutions are still not fully compliant with the operational constraints imposed by a business environment. The term *Business Grids* is hence introduced to distinguish the effort towards achieving this compliance in comparison to what currently supports scientific problem-solving.

D.J. Veit and J. Altmann (Eds.): GECON 2007, LNCS 4685, pp. 124–135, 2007.

We consider the current approaches to supporting Scientific Grids as having the ultimate goal of aggregating as much computational power as possible. Examples of toolkits and technologies that enable this goal are Globus [11], GRIA [24], and gLite [8]. These approaches can be used as foundations for building Business Grids but, due to the business-related constraints that we explain, there are some hindrances to direct usage.

This paper introduces some of the major requirements on Business Grids and discusses several technologies and concepts that might be used to overcome the current limitations of Scientific Grids. Note that the paper is not intended to describe particular business models, pricing model, market economies etc., as these are business-domain specific [9,25]. Moreover, such models can only be realized when the necessary technologies are in place. In our opinion, this is not the case, such that true evaluation of these models depends on the right technical frameworks and infrastructure being in place.

The remainder of this paper is structured as follows: In Section 2, we introduce and compare the characteristics of Scientific Grids and Business Grids. Afterwards, the extended, business-driven requirements for using Grid Computing are discussed in Section 3. Section 4 reviews some related work and technologies, deriving the challenges that need to be solved to fulfill the business demands. In Section 5, we identify key technologies and steps that may pave the way towards Business Grids. Finally, we summarize and conclude the paper in Section 6.

2 Scientific Grids and Business Grids

Within this section we elaborate on the applications that are executed on Scientific Grids today and the applications intended for hosting and execution in a grid-like infrastructure we call Business Grids. Note that the term "Scientific Grid" is meant to refer to Grid computations for scientific projects like e.g. the Large Hadron Collider (LHC)[4] or SETI@home [1], in the context of this paper.

2.1 Scientific Grids

A Scientific Grid is considered as a computing model for parallel processing across heterogeneous resources belonging to multiple geographically dispersed administrative domains. The computing problems considered require large amounts of either computing time or data, such that their reduction into several small parallel processes, with only little inter-process communication and a limited execution time (as opposed e.g. to interactive applications), sees computational improvements through parallelization. The majority of Scientific Grid computations can be characterized as stateless batch jobs mainly performing file-based input and output operations. Such jobs can be deployed in a relatively short time as they are submitted in a self-contained manner along with all input data files and executables required. The jobs usually do not depend on locally available license files or user interactions and are therefore highly mobile.

Data management in Scientific Grids focuses on data modeling, data movement, and handling of distributed and replicated files. Typically, data is modeled in comparatively flat data structures, and data access is performed in a non-transactional fashion. Resource management puts the main emphasis on high throughput and high resource utilization whereas low response times are often considered as less important. Security issues are commonly of minor interest since confidentiality and integrity of data are not critical in real-time, and computation resources and networks are assigned exclusively to scientific project members.

2.2 Business Grids

For the economic success of companies, it becomes more and more essential to be able to dynamically adapt to new business demands, knowledge and constraints. Consider a simple scenario for business agility where customer or application transaction load needs to be shifted or redirected to different application service providers, given variations in costs, availabilities and reputations. As a step towards such levels of flexibility, several companies host computing infrastructures for dedicated business tasks, billing on a per-use basis, labeling them as *e-business on demand*, *business on demand*, *utility computing* and others. However, such solutions do not support the levels of management automation required in order to respond effectively to dynamically changing business needs. Nevertheless, direct deployment of business applications to an existing scientific Grid infrastructure is still insufficient and too risky. Firstly, Business Grids should be able to restrict the sites providing resources to build the execution environment, requiring more complex resource and provider selection logic. One of the main reasons is that the participating parties need to have standing, legal contracts with each other regarding resource usage. In addition, they need to beware of global compliance concerns that could have damaging effects if breached as a result of transferring data to different sites. Furthermore, in order to obtain maximum flexibility that reflects the way in which ambitious businesses operate, technical security mechanisms with no assumed pre-existing trust relationships or anchors have to be assumed. Determining and initializing these on-demand still adds to the overhead of agility and requires more attention.

In contrast to Scientific Grids, which mainly interchange and process large, flat data files, Business Grids data processing must assume the existence of large relational databases, given the legacy of business systems. In such cases it is not possible to hide the transport of data to the executing node by pre-fetching all data before the job execution starts. In Business Grids, small sets of data must be loaded and stored from a database with a random access pattern. This does not only effect the execution time of the application itself, but also the scheduling decisions as the node that executes a certain application cannot be allocated too far from the database. Depending on its size the database remains static at a certain location. Additionally, the main applications within Business Grids are interactive, where interactions take place within open and stateful sessions, small data packets are exchanged frequently with the database, and no delays in data transmissions can be tolerated. Application components can be migrated but the interactive session must remain open. Even so, security levels of business applications are high in both the online and offline modes, in comparison with applications in Scientific Grids. The execution of the application itself must be secured and also all the related data before, during, and after the processing. Such security constraints also limit the allocation possibilities and the mobility of data.

If these business-related requirements were to be fulfilled, Business Grids could be used in various scenarios. For example, data centers could make high use of Business Grid technologies in order to increase the in-house adaptability to changing customer demands. The Grid technologies would be used as mechanisms for dynamic in- and outsourcing of computational capacity, sizing and re-sizing the deployment landscape, ensuring that all given business constraints are fulfilled. Additionally, the resources from different providers could be logically bound and identified as a new, collective functional capability that did not previously exist. This is today known as a Virtual Organization (VO) [10]. However, there remains a gap between the technical resource-sharing notion of a VO and the corporate management-sharing notion of business, which needs to be reconciled before Business Grids come into full existence.

3 Business Grids Requirements

In this section, we elaborate on the increasing demand for Grid technologies in business scenarios followed by a discussion of the various differences that upcoming Grid technologies must address in order to be compliant with the business demand. A comparison of Business and Scientific Grids is provided using identified problem areas, including resource composition, software landscape, application data, execution characteristics, execution characteristics, and service level agreements.

3.1 Grid Size

A fundamental difference between Business and Scientific Grids considers the scope and distribution of the computational nodes that form the infrastructure. Business Grids have been motivated by needs to serve single or very few administrative domains. They therefore have very closed assumptions about where data and applications are physically located, executed and managed. Scientific Grids are nevertheless developed with open-world assumptions and seek to acquire more and more resources without the hard constraints on the administrative domains involved [3]. This adds an additional constraint on the sizing and distribution properties of Business Grids that is currently not a concern for Scientific Grids. Furthermore, the structure of an organization places constraints on the topology and pricing model for resource distribution and usage.

3.2 Software Landscape

Business applications exhibit characteristics that differ from scientific or high performance computing applications. Since business applications implement both, standardized and individual business processes, they vary from enterprise to enterprise. They have to react on unschedulable/spontaneous events, triggered by user input or connected infrastructure components (e.g. logistic supporting RFID sensor infrastructure) as well as scheduled workload (e.g. payroll accounting in components supporting human resource tasks). From a birds eye perspective business solutions are deployed in a three tier architecture: The business application itself providing the business logic and user interfaces. This business logic is executed on a middleware platform (application server), with the third component in this architecture being a database. Since the transactions of almost all business applications access a central database, the database can be considered as a major medium for information exchange. Deploying such an infrastructure requires an individual configuration of each system and differs in the complexity to a Scientific Grid application where one and the same software has to be deployed multiple times. The size of the software which has to be deployed varies from a few to several hundred GB.

3.3 Application Data

Due to business and legal reasons, data and information of enterprises need to be accessible and readable for several years. Thus a lot of data and aggregated information has to be stored in a structured way. The size of used databases rises up to several TB. However small sets of data in the database are accessed frequently in a non-predictive manner. The implementation of sensor-triggered solutions (e.g. in the field of logistics or mass production) increases the demands on data management due to frequent transactional data access.

The application data has to be consistent and reflect the modeled reality. To achieve this, locks to subsets of data are applied and synchronous logs are written to guarantee this property. In contrast to the high performance computing, a relaxed consistency model is sufficient and computations are mostly performed independently. Usually, the problem space can be structured in an easy way and is divided into several parts. This is then sent to computation nodes. After finishing the computation, the result is returned and composed with others.

3.4 Execution Characteristics

Another main difference between Scientific Grids and Business Grids concerns application execution [16]. As already described, business applications are often interactive and frequently access databases. Such applications often have a very long runtime (up to years) in comparison to single, batch, scientific jobs. Furthermore, the deployment of these applications is rather long compared with the majority of scientific applications as it normally includes the installation and configuration of an entire multi-tier system including application servers and databases with up to several Gigabytes of data.

The comparatively longer execution times lead to the demand for mechanisms to migrate running business applications dynamically, as it is very likely that the underlying infrastructure must be maintained during the application runtime. However, the mobility of business application is limited due to the fact that each application and the corresponding data often has very strong security requirements and therefore is critical for the business itself. In contrast, the mobility of data is often not restricted within Scientific Grids.

Additionally, the business applications are in most cases split into several components and layers with separate executables. This also leads to quite complex start procedures for business applications. Here, the various dependencies between the operating system, the middleware, the data, and the application must be incorporated. In contrast, the majority of scientific applications consist of a single executable that can be started independently.

The majority of business applications are stateful, especially when databases are available for transaction data and transaction persistence. Thus, if cases of failures a simple restart of the application is not possible. This motivates the usage of concepts to either duplicate the different application executions or to develop methods that allow a very precise checkpointing of the applications such that states are not lost.

3.5 Autonomy

As the business processes are changing quicker than ever before, the underlying information technology must be able to adapt and to follow these changes. In the past, the required system modifications were mainly performed by human beings. This is expensive and also often not quick enough. Thus, business solutions drive the demand to reduce the amount of work that has to be spent in the management of the Grid. Hence, self-provisioning of applications and self-management of the Grid infrastructure are two major longer term goals, see Franke et al. [14]. Furthermore, self-healing strategies need to be developed that enable stateful applications to recover after failures. Additionally, business usage scenarios require the ability of self-optimization on different levels, like the application level, the middleware level, and the hardware level. All the above-mentioned requirements are in contrast to the existing Scientific Grid approaches. In most cases, restarting an application after a failure is sufficient in theses cases.

3.6 Service Level Agreements

One of the main differences between the scientific and the business usage of Grid technologies is the purpose itself. In the scientific environment, the main goal can be described as getting as much computing power as possible with a reasonable amount of management tasks for the participating scientific institutes. In the business context, Grids are used to flexibly execute applications that are provided to customers paying for this kind of service. Thus, one major goal is to fulfill the business requirements with all the corresponding constraints by automating as much as possible.

All the constraints of the application execution are defined within a Service Level Agreement (SLA) which reflects a legal contract between the service provider and the customer. Usually, an SLA is a bidirectional contract that specifies in detail what service the provider must deliver and which rules the customer agrees to follow. SLAs also describe the agreed reliability of the service, the billing and other business-critical issues. Furthermore, the security constraints are detailed within SLAs by specifying the kind of isolation that the service provider must deliver (user isolation, application isolation, performance isolation). As the business is directly connected to such SLAs, a very sophisticated SLA management is required. Therefore, industry demands Business Grids to vertically integrate an automated SLA management across all layers of the execution stack.

4 Related Work

There are various technologies and concepts in existence and development that might contribute to the generation of Business Grids. A review of some of these possible concepts shows that there are important strategic and technical challenges that need to be solved before Business Grids can be fully realized.

Information Systems Outsourcing: Information Systems Outsourcing is the contracting of various systems to external information system providers for operational and/or maintenance purposes [18,15]. The simple argument for outsourcing is that the customer focuses on their business domain, while the provider relieves them of significant overhead and risks associated with maintaining large-scale, complex technology. From the customer's perspective they run their business applications without hiring additional, specialists staff or worrying about financing additional utility costs. From the provider's perspective they offer specialists services at a price that is less than the potential hiring, training and utilities costs faced by their customers, while having control over how they manage and reuse their resources. These sorts of arguments are not particularly of relevance for Business Grids for three reasons: (1) Business Grids seek to keep the customer in control of resource management, (2) customers and users of a Business Grid determine the levels of infrastructure transparency they require, and (3) Business Grids allow to enforce that providers fully respect the resource selection, deployment and scheduling constraints of customers.

Service-Oriented Computing: Service-oriented Computing provides software integration constructs and mechanisms that allow distributed processing and storage units to be flexibly linked together in a loosely-coupled manner [20]. That is, services maintain their autonomy and encapsulation and interact strictly using message passing, following the fundamental definition of a distributed system architecture. A service specification acts as descriptor and invocation interface for specific, well-defined and encapsulated functionality, which can be composed with others to form more complex systems and workflows. Depending on the protocols and accessibility of services, resultant systems can be distributed across multiple administrative domains, allowing dynamic discovery and replacement. Many of the technical capabilities and facilities for Service-Oriented Computing can be applied in the context of the Business Grid. However, the description concept of service oriented computing only allows for the use of static attributes (service names, functions and categories) during resource discovery, whereas these attributes need to be more dynamic and varied for the purposes of Business Grids.

Utility Computing: Utility computing [21,22] is rather a business concept than a technological advance, where users of a large-scale computing resource pay only for the computational power, storage and software that they use, similar to the way in which electricity, telephones and water are charged. Users therefore subscribe to computational utility providers and agree on their terms of usage, which may be measured based on volume, time or quality. While the Business Grid requires a similar model of subscription, pricing and resource reservation, the underlying computational infrastructure must be leveraged to flexibly adapt to dynamically changing business requirements.

Virtual Clusters: The term *Cluster Computing* is often misused as a synonym for Grid Computing. While clusters are typically bound over short-range, high-speed communication links and controlled by a centralized scheduler, Grid nodes are interconnected over possibly wide-area, variable-speed networks, with sophisticated scheduling methods for the actual resource allocation over time. The applications deployed on clusters therefore differ from those enacted on a Grid. Secondly, a Grid can be composed of several clusters, making it a larger-scale and potentially more complex computational infrastructure for supporting distributed applications. Therefore, the hosting of business applications on a cluster does not constitute the concept of Business Grids that we propose. Virtual Clusters provide an abstraction for regulating computational resource usage for different groups of users. Selected nodes of a cluster are reserved for different users or groups by the specification of resource sharing rules, such that the groups are not aware of the full power available. From our perspective, Virtual Clusters (VCs) can be seen as complimentary to the Virtual Organization (VO) concept, where VCs technically refer to how resources are bound across domains, while VOs provide a management model. Business Grids also require mechanisms derived from Virtual Clusters for resource regulation, yet the means by which they are enforced need to be more sophisticated than message interception using reference monitors.

5 Steps on the Way to Business Grids

This section describes different approaches and technologies that can be used in order to make a Grid ready for the execution of business applications.

First the technologies which we think are of major interest are briefly introduced. Since there is no one-to-one relationship of the discovered requirements and the proposed technologie their concrete exploitation is mentioned in the following.

- Virtual Machines: Virtualization of a machine such as [2,17] multiplexes multiple virtual instances to a physical machine. It can improve utilization but more importantly provide flexibility in the management of systems. From a birds eye perspective most technologies provide the same primitives independent of their technological implementation. Process checkpointing technologies can provide similar primitives too and are also of interest.
- Distributed Filesystem: Distributed filesystems can from a birds eye perspective seen as a multiple instantiation of network filesystems such as NFS. Information is accessed through multiple pathes which promise a higher throughput when multiple clients accessing data. Usually data and its metadata is stored in separately, which can improove metadata access. A replication policy controls which and where data is replicated to. Representatives are CEPH, PVFS [19,5].
- Deployment Infrastructure: Deployment infrastructure focuses on the rollout and management of software and its configuration. Provided a description of the interdependencies of the software, it can be deployed and started in the right order. One representative implementation is SmartFrog [23].

A first solution proposal to the discovered deficiencies is described next. Table 1 shows the technical mechanisms which are considered and how they are exploited with respect to the various *problem areas* introduced in Section 3.

Table 1. Assessment of the various technical mechanisms with respect to Grids aspects

Problem Areas	**Virtual Machines**	**Distributed File Systems**	**Deployment to Infrastructure**
Grid Size	-	-	Affects
Software Landscape	Affects	Affects	Affects
Application Data	Affects	No	Affects
Application Execution	Affects	No	Affects
Autonomy	No	No	No
SLA	Affects	No	Affects

Software Landscape: The deployment routines for deploying a business application have to be aware of system dependencies. Frameworks such as SmartFrog are promising to provide a solution to this. Nevertheless, the installation procedure can be very time consuming. Instead of using traditional software installers the use of virtualization technologies can accelerate this task by deploying ready to use disk and virtual machine images.

Application Data: The ability to deploy the database at a location different from the other parts of the business application can be supported by the use of advanced caching mechanisms to compensate network latency above local networks. Access to often used pages to the database file are cached to local disks in the vicinity of the application and thus replicated incrementally. A distributed file system which replicates to the most used deployment sites may also provide a solution to the migration issue of huge data files. Although there are a lot of distributed file systems available their suitablity for database typical workload has to be investigated.

Application Execution: The constraints for the application execution can be fulfilled by the combined usage of several new technologies and the slight adaptation of existing tools. In general, virtualization techniques can be used to enable checkpointing of existing legacy software with only little need to modify these software solutions. Furthermore, the virtualization techniques would enable the migration of applications during the runtime by migrating the virtual execution container. This would solve the problem of infrastructure maintainance for applications with long runtimes. The problem of database connectivity to the business application can only be solved by an adapted deployment. This deployment must take care that the application and the database are located close enough such that the database access from the application has a reasonable response time. This response time should be specified in the corresponding SLA. Additionally, the deployment of business applications must incorporate the various application layer and all corresponding constraints. Note that checkpointing or migration can also be performed on the application directly (see e.g. BLCR, CHPOX, CRAK, HPC4U, UCLiK). However, the problem with such approaches is that entire goups of processes belonging to the application along with all open communications and files have to be captured and tranferred to a compatible hardware architecture.

Autonomy: This issue is not addressed in existing Grid approaches. Thus, the Grid management itself must adapt functionalities that enable a kind of self-management and self-adaptability. All the related issues are at the moment open research topics.

Service Level Agreements: This topic is addressed in several Grid approaches. However, the current state is primarily dominated by scientific requirements and mainly focuses only on the physical infrastructure. Thus, current approaches mainly specify hardware characteristics. However, the requirements of businesses are different in the sense that the higher level SLAs between the service providers and the customers must be broken down onto the infrastructure. To this end, the complex software landscape and virtualization technologies must be incorporated. Furthermore, the SLAs must consist of functional and non-functional service specification. Theses issues are also open research topics that must be addressed soon in order to make Grid technologies applicable for businesses.

6 Conclusion

Business applications were developed and further improved for decades on mainframes and client/server infrastructures. As the businesses need to react more dynamically than ever before, the underlying information technology also needs to adapt dynamically to changing environments. Thus, the application of Grid technologies for business solutions seems reasonable.

However, making all these applications fully aware of a Grid Computing infrastructure is not feasible. Therefore, this paper starts to find a greatest common divisor between what is provided by current Grid technologies and related approaches and the business application demands.

In this paper, we pointed out differences between the Grid and the business application domains as well as technical deficiencies of current Grid approaches from the business perspective. Furthermore, technical mechanisms are proposed to address most of these issues. One of the major interests is the investigation of technologies which alleviate the binding of applications to dedicated physical resources. In general, business applications could benefit most from Grid infrastructures if these applications could dynamically be deployed in a distributed manner.

Of course, the business context restricts the flexibility during the distribution which must be incorporated at the deployment time. Ongoing efforts to build business application on a service oriented architecture provide the technical precondition to further increase the potential benefit of using Grids in business environments. Services allow for a finer-grained deployment on Grid nodes than traditional applications thereby increasing the flexibility of business solutions and leading to a higher adaptability of the Grid.

Acknowledgments. We would like to thank the support from the European Commission under IST program #FP6-033576.

References

1. Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: Seti@home: an experiment in public-resource computing. Communications of the ACM 45(11), 56–61 (2002)
2. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pp. 164–177. ACM Press, New York (2003)
3. Brune, M., Gehring, J., Keller, A., Reinefeld, A.: Managing clusters of geographically distributed high-performance computers. Concurrency - Practice and Experience 11(15), 887–911 (1999)
4. Bunn, J.J., Newman, H., McKee, S., Foster, D.G., Cavanaugh, R., Hughes-Jones, R.: Bandwidth challenge - high speed data gathering, distribution and analysis for physics discoveries at the large hadron collider. In: Löwe, W., Südholt, M. (eds.) SC 2006. LNCS, vol. 4089, p. 241. Springer, Heidelberg (2006)

5. Carns, P., Walter, H., Ross, R., Thakur, R.: Pvfs: A parallel file system for linux clusters. In: Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA, USENIX Association, pp. 317–327 (2000)
6. Catlett, C., Smarr, L.: Metacomputing. Communications of the ACM 35(6), 44–52 (1992)
7. DeFanti, T., Foster, I., Papka, M., Stevens, R., Kuhfuss, T.: Overview of the i-way: Wide area visual supercomputing. International Journal of Supercomputer Applications 10(2), 123–130 (1996)
8. Dvorák, F., Kouril, D., Krenek, A., Matyska, L., Mulac, M., Pospísil, J., Ruda, M., Salvet, Z., Sitera, J., Vocu, M.: glite job provenance. In: Moreau, L., Foster, I. (eds.) IPAW 2006. LNCS, vol. 4145, pp. 246–253. Springer, Heidelberg (2006)
9. Ernemann, C., Yahyapour, R.: Grid Resource Management - State of the Art and Future Trends. In: chapter Applying Economic Scheduling Methods to Grid Environments, pp. 491–506. Kluwer Academic Publishers, Dordrecht (2003)
10. Foster, I.: What is the grid? a three point checklist. GRIDtoday, 1(6) (2002)
11. Foster, I., Kesselman, C.: The Globus project: a status report. Future Generation Computer Systems 15(5–6), 607–621 (1999)
12. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, San Francisco (1999)
13. Foster, I., Kesselmann, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, San Francisco (1998)
14. Franke, C., Theilmann, W., Zhang, Y., Sterritt, R.: Towards the autonomic business grid. In: Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe'07), pp. 107–112 (2007)
15. Goles, T.: Vendor capabilities and outsourcing success: A resource-based view. Wirtschaftsinformatik 45(2), 199–206 (2003)
16. Kenyon, C., Cheliotis, G.: Grid Resource Management - State of the Art and Future Trends. In: chapter Grid Resource Commercialization - Economic Engineering and Delivery Scenarios, pp. 465–478. Kluwer Academic Publishers, Dordrecht (2003)
17. Kolyshkin, K.: Linux virtualization (2005)
18. Nam, K., Rajagopalan, S., Raghav Rao, H., Chaudhury, A.: A two-level investigation of information systems outsourcing. Commun. ACM 39(7), 36–44 (1996)
19. Olson, C., Miller, E.L.: Secure capabilities for a petabyte-scale object-based distributed file system. In: StorageSS '05: Proceedings of the 2005 ACM workshop on Storage security and survivability, pp. 64–73. ACM Press, New York (2005)
20. Papazoglou, M.P., Georgakopoulos, D.: Introduction. Commun. ACM 46(10), 24–28 (2006)
21. Rappa, M.A.: The utility business model and the future of computing services. IBM Syst. J. 43(1), 32–42 (2004)
22. Ross, J.W., Westerman, G.: Preparing for utility computing: The role of it architecture and relationship management. IBM Systems Journal 43(1), 5–19 (2004)
23. Sabharwal, R.: Grid infrastructure deployment using smartfrog technology. In: ICNS, page 73 (2006)
24. Surridge, M., Taylor, S., De Roure, D., Zaluska, E.: Experiences with gria industrial applications on a web services grid. In: E-SCIENCE '05: Proceedings of the First International Conference on e-Science and Grid Computing, pp. 98–105. IEEE Computer Society, Washington, DC (2005)
25. Yeo, C.S., Buyya, R.: A taxonomy of market-based resource management systems for utility-driven cluster computing. Software: Practice and Experience 36(13), 1381–1419 (2006)