Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 655–661
ISBN 978-83-60810-27-9
ISSN 1896-7094

Service level agreements for job control in high-performance computing

Roland Kübert
High Performance Computing Center Stuttgart
University of Stuttgart
Stuttgart, Germany
Email: kuebert@hlrs.de

Stefan Wesner
High Performance Computing Center Stuttgart
University of Stuttgart
Stuttgart, Germany
Email: wesner@hlrs.de

***Abstract*—A key element for outsourcing critical parts of a business process in Service Oriented Architectures are Service Level Agreements (SLAs). They build the key element to move from solely trust-based towards controlled cross-organizational collaboration. While originating from the domain of telecommunications the SLA concept has gained particular attention for Grid computing environments. Significant focus has been given so far to automated negotiation and agreement (also considering legal constraints) of SLAs between parties. However, how a provider that has agreed to a certain SLA is able to map and implement this on its own physical resources or on the ones provided by collaboration partners is not well covered. In this paper we present an approach for a High Performance Computing (HPC) service provider to organize its job submission and scheduling control through long-term SLAs.**

***Index Terms*—Service Level Agreements, High Performance Computing, Cloud Computing**

I. Introduction

FOR High Performance Computing resources scheduling of jobs is still realized in most cases using simple batch queues. While batch queues like OpenPBS [1], TORQUE [2] or others offer a quite comprehensive set of functionality for placing jobs in appropriate queues and optimizing the load of the cluster systems, also across sites, there is no mapping from business level requirements down to the low-level specifications. Low-level specifications are typical elements of a job description, for example desired number of CPUs, maximum wall- or run-time. The provision of such low-level properties requires a high level of expertise of the user and can only be specified if the target platform is pre-determined as different node and CPU architectures require different values. Additionally the number of queues is limited and therefore requirements have to be mapped to a particular queue. While advanced reservation, allowing a pre-defined start time, can be specified the drawback of potentially significantly reduced efficiency due to fragmentation of the schedule is not mapped to potential business penalties such as dynamically adapted pricing for such requests depending on the concrete loss in efficiency.

If an HPC provider wants to offer its services as utilities and aims to map different possible flavors of the services on different queue structures, the following problems can occur:

- Typically the use of certain queues is mapped to Unix credentials and groups. So all users of a certain group can or cannot use e.g. the express queue. However, depending on time of day or load situations, the “express queue service” might not be available to the same group of users all the time.
- While queues for specialized nodes (for example with graphics processing units (GPUs) or high memory nodes) are underutilized and normal nodes are oversubscribed there is no way to allow clients and providers to agree on special “discounts” for them. An automatic movement from normal to premium node queues would require interaction with accounting services.
- Customers might want to differentiate quite fine-grained about the treatment of their jobs. In such cases nowadays manual movement of jobs within the queues to “prioritize” them might be agreed beyond existing queue structures and group memberships. Such manual interactions cannot scale.
- Not all elements describing the Quality of Service (QoS) or Quality of Experience (QoE) can be mapped on queue properties and parameters. The overall service covers a wider range of properties such as the availability of a certain compute environment, application versions and licenses, proper treatment of data or specific configurations of the cluster system such as “require logical partition to isolate from other users”.

This limitation that only a few functional parameters can be specified when submitting a job (also reflected in standards like the Job Service Description Language (JSDL) [3]) means that there is basically no way for the user to express his requirements on a QoS or QoE level. Considering that the HPC provider is offering a utility potentially replaceable with other providers there is a clear gap between the demands from the user side and current offerings.

As a result a significant amount of work has been spent on realizing SLA frameworks allowing to mutually agree on the terms of the service between provider and consumer. However, while these frameworks cover well the necessary steps to realize SLAs also as a legally binding agreement, the concrete content of such an SLA and more important how these terms

can be guaranteed and provided from the service provider side are not adequately addressed.

So far we have the possibility for the consumer to express the requirements and agree the terms with the provider but

- terms within the SLA are not on the desired business level but mimic the low-level properties of the underlying queuing systems and
- the agreement process is typically detached from the underlying infrastructure such as current load situation of different resources, priority and importance of the consumer in a Customer Relationship Management (CRM) system and the accounting and billing services.

Consequently there is a gap between the demand of defining business level SLAs and their implementation using available methods and tools for the management of them on different type of computing facilities ranging from commodity of the shelf (COTS) clusters over specialized compute systems to cloud computing and storage systems.

Management systems on the provider side between the SLAs agreed with the consumer and the concrete physical resources need to interact with a range of different elements within the providers IT infrastructure and must look beyond individual SLAs to optimize the overall operation of all resources within a HPC computing service provider.

II. Related work

There are various approaches to the usage of service level agreements for job scheduling. While they differ in many respects - detail of the presentation, assumed parameters, implementation level, etc. - they all share the fact that they treat SLAs as agreements on a per-job basis. That means that, for each job to be submitted, a unique SLA is established before the job can be submitted. In [4], Yarmolenko et al., after having identified the fact that SLA-based scheduling is not researched as intensively as it could be, investigate the influence of different heuristics on the scheduling of parallel jobs. SLAs are identified as a means to provide more flexibility, increase resource utilization and to fulfill a larger number of user requests. Parameters either influence timing (earliest job start time, latest job finish time, job execution time, number of CPU nodes) or pricing. They present a theoretical analysis of scheduling heuristics and how they are influenced by SLA parameters and do not investigate how the heuristics might be integrated into an alredy existing setup. The same authors identify in [5] the need to provide greater flexibility in service levels offered by high-performance, parallel, supercomputing resources. In this work they present an abstract architecture for job scheduling on the grid and come to the conclusion that new algorithms are necessary for efficient scheduling in order to satisfy SLA terms but that little research has been published in this area. MacLaren et al. come to a similar conclusion, stating that SLAs are necessary in an architecture supporting efficient job scheduling [6].

SLAs that express a job's deadline as central parameter for deadline-constrained job admission control have been investigated by Yeo and Buyya [7]. The main findings were that these SLAs depend strongly on accurate runtime estimates, but that it is difficult to obtain good runtime estimates from job traces.

Djemame et al. present a way of using SLAs for risk assessment and management, thereby increasing the reliability of grids [8]. The proposed solution is discussed in the scope of three use cases: a single-job scenario, a workflow scenario with a broker that is only active at negotiation-time and a workflow scenario with a broker that is responsible at runtime. It is claimed that risk assessment leads to fewer SLA violations, thus increasing profit, and to increased trust into grid technology.

Dumitrescu et al. have explored a specific type of SLAs, usage SLAs, for scheduling of grid-specific workloads using the bioinformatics BLAST tool with the GRUBER scheduling framework [9]. Usage SLAs are characterized by four parameters: a user's VO and group membership, required processor time and required disk space. The work analyzes how suitable different scheduling algorithm are. Additionally, it comes to the conclusion that there is a need for using good grid resource management tools, which should be easy to maintain and to deploy.

Sandholm describes how a grid, specifically the accounting-driven Swedish national grid, can be made aware of SLAs [10]. It is presented how the architecture can be extended with SLAs and it is stated the greatest benefit would be achieved by insisting on formally signed agreements.

A comprehensive overview of resource management systems and the application of SLAs for resource management and scheduling is given by Seidel et al. [11]. The connection of service level and resource management to local schedulers is clearly shown as a gap in nearly all solutions.

In summary, it can be said that isolated aspects of the usage of SLAs have partly been investigated in detail: scheduling algorithms and heuristics, abstract architectures, parameters which are to be used as service levels, SLA negotiation etc. Gaps, however, can be easily identified: the analysis of the "big picture", that is the composition of individual aspects of SLA usage into a complete system and the integration of SLAs and SLA management with local resource management. This is not only true for the "traditional" field of high performance and grid computing but can also be extended to the field of cloud computing. Furthermore, SLAs are solely treated on a per-job basis, the analysis of SLAs as long-term contracts is not covered.

III. Benefits of SLAs for HPC service provisioning

The current operation model for high-end computing resources is conceptually still the same as fifty years ago where users placed a set of punching cards at the registry desk. The only difference is that users now can submit their compute jobs to a set of different queues and instead of the human operator the scheduling system is picking the jobs from the different queues depending on defined policies aiming for an optimized load of the system partially reaching 99% utilization. The major shortcoming of this approach is that the optimization

strategy defined by the queues and the scheduling system policies is oriented towards a global optimization rather than an individual service offer.

If a user needs a special service (e.g. guaranteed start time of a job during a demonstration, interactive visualization or exhibition) beyond regular job submission the negotiation is typically done directly with the system operator and the performed steps are mostly done manually.

The availability of multi-core CPUs will lead to compute nodes with 32 cores and more in the near future, the rise of GPU-based computing with several hundred "cores" per card allows a reasonable number of applications to run on a single node. This is particularly true if the application is not targeting for a high-end simulation e.g. in the area of Computational Fluid Dynamics (CFD) domain with a very fine-grained mesh but more on exploring the problem space. Other examples are cases where the full simulation has been done before and now only small changes in geometry are done interactively demanding much less intensive computing to reach a stable state again as it is based on the previously achieved results.

Driven by the availability of cloud service providers and emerging products such as the Amazon Cluster Compute Instances also high-end computing service providers change their offers to be more *elastic* and realize a more *dynamically changing* infrastructure having certain queues available only during specific time periods or realizing a dynamic allocation of resources to logical partitions depending on the load situations or specific time bound agreements.

The pre-dominant use of high-end computing services will continue to be highly scalable technical simulations demanding a large number of compute nodes for exclusive use. However additional use cases have emerged driven by changes on the hardware level and competition with cloud service providers in particular for small scale simulations. The exclusive access for a user to one single node might even for compute intensive applications become a relic of the past. This substantially more complex management model for HPC service providers that cannot rely anymore on a quite homogeneous user behavior and long running jobs demands for a more complex management solution for operating their resources. The challenge is to integrate the demands of policies from different levels such as business policies (e.g. users with highly scalable and long running jobs should experience a preferred treatment) with more short term policies reacting on the current load situation (e.g. reducing prices or accepting more small jobs to fill gaps in the current schedule) and the demand of the users on a per-job basis.

The following sections aim to cover in examples the three major use cases driving the need for an SLA-guaranteed HPC service provision. Abstracting from concrete cases three different cases can be identified:

A. Interactive Validation

In many areas simulations have already replaced real experiments or physical prototypes during the development process. However at certain control points in the process simulation results have to be verified using physical prototypes. Within the IRMOS project augmented reality techniques are used to overlay real experimental data like a smoke train in the wind channel with a visualization of trace lines from the corresponding simulation. This "hybrid" prototype allows experts to directly compare the behavior of the real prototype with the results of the simulation. Such a design review session typically involving several people of a development team spread around the globe demands a fixed availability of the wind-tunnel, the computing resources, the visualization resources, the corresponding network resources and all involved experts, for example via video-conferencing.

In such a scenario simulation data will be generated continuously by a simulation running on a compute resource that is directly connected to the visualization resources. The current configuration of the wind channel like the air speed will be communicated as boundary condition for the simulation, thus the same parameters for both will be used while the experiment is running. This requires a coordinated and automated provision of the resources involved in the overall setting.

Such a scenario cannot rely on batch queue-based access as the computing and simulation part is just one piece in the overall setting. The demand for a co-ordinated availability also opens questions on how penalties are applied if one of the pieces in the overall setting is failing. For example if the compute resources are not provided as promised in time and the wind tunnel cannot be used the costs for it still accumulate. This applies also the other way around if the wind tunnel is not available or fails to communicate the boundary conditions for the simulations or the network connection is not delivering sufficient bandwidth.

As the resources needed for the full scenario are provided by different organizational entities the different quality levels needed by each individual contributor need to be put in a formal SLA, covering the terms of service as well as the agreed penalties in case of failures.

B. Guaranteed Environment

As outlined in [12] beside quality constraints there is also a demand to ensure a certain environment or other procedural constraints such as data handling, security policies or environmental properties (version of the operating system, available Independent Software Vendor (ISV) applications, etc.).

This is especially necessary for simulations performed as part of an overall design cycle for a complex product such as a car or airplane. A software environment is freezed for a full development cycle in order to ensure reproducible simulation results. This fixed environment is typically ranging from operating system over certain versions of numerical libraries up to application codes. A typical approach to address this requirement is to have beside a paper-based SLA agreed for a design cycle period a dedicated computing resource with the requested environment.

Advances in virtualization technologies as well as the possibility to apply different boot images in diskless cluster environments allow a more flexible treatment. Using such

technologies a potentially unlimited number of pre-defined images, or even user-defined images, might be provided. As not all environments can be provided on all compute resources there must be a negotiation process between the user and the provider where a certain environment is demanded (e.g. expressed in a certain SLA bundle such as "Silver") and a corresponding reply about the conditions for the different options from the provider side is delivered.

The increased flexibility would allow to offer customized environments not only to large customers asking for resources for a long time period but also for users looking to meet their peak demands with outsourcing avoiding tedious customization activities of the environment reducing the entry gap.

C. Real-time Constraint Simulations

With the increasing role of simulations in design processes for complex products the demand to have a time-boxed simulation where results need to be delivered in time have emerged. This might be a set of simulations exploring a parameter space as input for a meeting of engineers the other day deciding on the focus for the future (long running simulation jobs). Another possibility is if the results of one single simulation (or a set of simultaneously running simulations) is the input to support an expert in taking a decision.

One important application area demanding for such an operation model is individualized patient treatment. For example in [13] a scenario for using simulations to validate different options to perform a bone implant for a specific patients is presented. In such cases the expert that needs to make a treatment decision has to ensure in advance of starting the simulation at a specific compute service provider that the results will be available in time before the treatment must be executed.

In such a scenario a negotiation with several providers would be started in parallel in order to make a case-by-case decision to which provider the job will be finally submitted. Such a loose binding to a specific provider would also require similarly to the scenario in the previous section a guaranteed or user-provided environment making the different providers interchangeable.

D. SLA Service Provision Benefits

From all the scenarios above it becomes clear that a much higher diversity of the offered services must be expected in the future. The requirements of the different scenarios on the provider's infrastructure are quite diverging. Additionally the consumer requirements are contradictory to the goal of the providers reaching a very high level of utilization of the provided resources.

As a result service providers will need to

- offer a mix of different services in order to combine the benefit of best-effort services (high utilization) with the benefit of special services (high value and price),
- offer a framework allowing consumers and providers to agree on the specific conditions for the service and
- actively manage their resources in a way that agreed SLAs are met, resources are most effectively used and any failures and incidents on the resource level are managed to avoid any impact on the agreed service levels.

The underpinning assumption presented in this section is that the provision of SLA controlled services is beneficial for consumers *and* providers. Consumer can negotiate guarantees and specific properties of the provided services as needed enabling new use models for high-end computing resources as outlined above. The provider perspective is clearly driven by business benefits to deliver as a part of the differentiation strategy specific products rather than aiming for a cost leadership approach. Consequently there is a clear need from the consumer side as well as a clear motivation from the provider side to deliver also in the HPC domain SLA based services. In other words the current model where the user needs to fully adapt to the provided environment and access model is changed to a model where the provider is offering certain possibilities or a kind of toolbox where the consumer can arrange the service offer according to their needs. Realistically this space of options needs to be discrete and limited allowing a management of the service offer from the provider side.

IV. Using long-term service level agreements for job control

Service level agreements, when they are used for the scheduling of compute jobs, are normally assumed to be on a per-job basis. That means that an individual SLA only contains terms for one specific job and a new SLA needs to be established for each job (see for example [14], [7] and [8]). This may be ideal to investigate the influence of SLAs and parameters specified therein on the scheduling of jobs in an isolated environment but does not correspond with the reality of how contracts are handled at HPC providers. At HPC providers, users usually agree to a contract that specifies charges for computational times and storage for available machines [15]. Jobs are then submitted in accordance with the acknowledged charges which therefore can be thought of as a long-term contract. This contract is, however, missing a specification of service levels. There may be some service level-related parameters specified - for example the availability of different machines to users and their characteristics, for example CPUs per compute node and memory size per node, but these are only specified in order to compute the amount finally billed to the user. By adding service levels to this contract, a long-term service level agreement is formed.

Long-term SLAs add the missing specification of service levels but keep the familiar contract behavior used by HPC providers intact. A simple specification of priorities, for example, might be realized through the following service levels:

Bronze Computational time is cheap, but there is no assurance on the scheduling of a job. This corresponds to the best-effort services provided today at HPC centers.

Silver Moderate prizing for computing time due to prioritized scheduling. Silver jobs can have timing

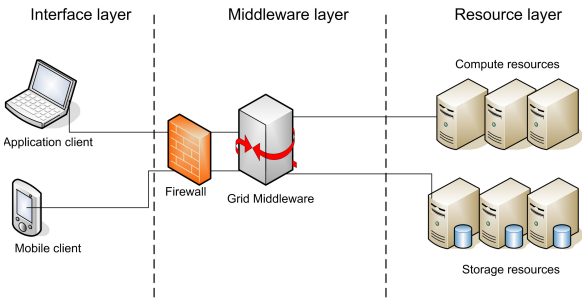

Fig. 1. Layered architecture

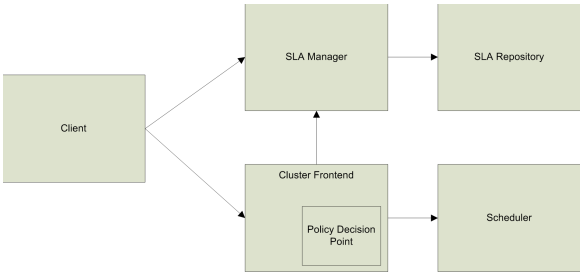

Fig. 2. High-level SLA management components

guarantees and might preempt best-effort jobs. The increased prize is justified since guarantees on the job's scheduling are given.

Gold High-prized jobs that are only rarely used, for example for urgent computing when computations need to be started immediately.

In contrast to the current situation the possibility of providing different service levels allows users to potentially have multiple contracts in place in parallel. On job submission time, a user decides which contract to reference in the submission depending on the current requirements and conditions such as urgency of the simulation result, load situation of the provider(s) etc. This can be seen as a using the middle way between using SLAs on dynamic, per-job basis and solely having singular long-term contrasts. In contrast to dynamic, per-job SLAs, this approach reduces the amount of negotiation as only few contracts are in place. Additionally, it avoids the problem that an urgent job cannot be submitted due to a failed negotiation. In contrast to having only singular contracts, this approach is more flexible and allows users to choose necessary priorities depending on the prize they like to pay.

V. An Integrated Approach to Service Level Management

The basic service level approach given above can be realized with current techniques, for example the usage of specialized priority queues; however, providing more complex service levels cannot be realized that easily but requires the integration of service level management techniques across interface, middleware and resource layer.

Figure 1 shows the typical three-layered setup that is used by HPC providers. The left-hand sides shows clients of the HPC provider, either static or mobile[1]. The middleware layer is positioned between the client and the low-level resources and serves as a central entry point to the HPC provider's system and provides access through grid middlewares, for example the Globus Toolkit. The Grid middleware takes jobs submitted by the client and passes them on to low-level resources by means of a resource manager which employs a job scheduler in order to determine which jobs are placed on which resources.

[1]Mobile in this context should not be mixed with cellular phones and is understood as a nomadic user that is connecting from different locations without pre-defined IP addresses

A. Service level selection by the client

Enhancing the client with the ability to select service levels is very straightforward. This can be either done by changing the job submission client, adding SLA information to the message sent to the middleware or by integrating SLA functionalities into the application, if job submission is performed directly out of it.

B. Enhancing the middleware

SLA management on a middleware level has been investigated by various research projects and therefore different components and solutions already exist [16] [17] [18] [19] [20]. As these solutions often have the drawback of being very complex, a simpler solution is preferable, as it eases the amount of work necessary for installation, integration and maintenance.

Figure 2 is a diagram depicting how easily SLA management can be implemented on a middleware level and the underlying resource layer. The client thereby can either communicate with the SLA Manager, a central component on the HPC provider side responsible for SLA management, or submit jobs to the cluster front-end in the manner already explained.

The SLA Manager provides data regarding the long-term SLA contracts, for example contract information, accounting pertaining to contracts etc. It uses an internal SLA Repository for storing the contracts and other relevant information and is the central point that is queried by other components regarding SLAs. The cluster front-end, for example, on submission of a job, can query the SLA Manager for the validity of SLAs and can, after job completion, send accounting data to the SLA Manager.

The SLA functionality for the cluster front-end in the grid middleware can be realized in a non-intrusive way, for example through a policy decision point (PDP) that checks incoming requests and their SLA specification for validity. Incoming requests that do not contain SLA specifications can be mapped internally to a default SLA specifying a best-effort style service level, thereby realizing complete SLA-functionality and being backwards-compatible to clients.

C. Acknowledgement of SLAs

Honoring service levels of submitted jobs depends on the software used on this low level. Very simple schedulers, like the default scheduler supplied with the TORQUE resource manager, cannot honor service levels and need to be replaced by an SLA-enabled scheduler. This can be implemented by the provider itself, but this is a time-consuming and error-prone job. Rather, an SLA-enabled scheduler, for example the Moab Cluster Suite, should be used. It allows the formulation of quality of service levels for resource access, priority and accounting.

The providers main task is then to express the high-level SLAs offered to customers in such a way that the scheduler can implement them on the resource layer. Additionally, the incoming job requests have to be mapped to the corresponding service levels.

VI. From high-performance to cloud computing

In the previous sections we have elaborated a concept to enhance "classical" high-performance computing with service levels through the use of long-term service level agreements. Cloud computing, in many terms similar to the previous scenarios, seems like a logical step for the provisioning of services and can be a sensible offer to provide for HPC providers besides their usual role. Even though the term cloud computing is not clearly defined, it can be seen as distributed computing with virtual machines. Virtualization allows for more flexibility, scalability and abstraction of the underlying resources. Accounting and billing are usage-dependent. [21]

Cloud computing brings benefits both for consumers and providers. Virtualization allows the provider to use free resources for the execution for virtual machines as the underlying hardware is mostly irrelevant, although requirements specified by the user of course still need to be met. The usage of virtual machines means that the provider can offer a multitude of different environments tailored to customers, which was previously infeasible. Users might even be allowed to provide their own virtual machines, therefore giving them control over the complete environment.

Cloud computing began on a best-effort basis and many solutions provided today don't offer any more service [22] [23]. Service level agreements for cloud computing are, however, provided by some service providers, but they provide only minimal service levels [24] [25].

It has been shown that both Infrastructure as a Service and Platform as a Service - two types of cloud computing where the first one offers the concept described above and the second one offers a scalable, flexible but pre-defined environment to users - can benefit from service level agreements as well [26].

VII. Conclusions

The preceding work has described how an integrated approach to using service level agreements for the control of compute jobs allows HPC providers to offer support for various quality of service levels. Due to the solution including both high-level SLA management and low-level resource management and job scheduling, service providers can take advantage of service level agreements through the complete infrastructure. This is even valid for the recently introduced cloud computing paradigm.

The concept described above has been partially realized for an HPC scenario where the SLA management layer has been implemented and a simple integration with the grid middleware and resource layer has been achieved. Following the trend to provide Infrastructure as a Service solutions, we have decided to adapt the general concept for the Gridway meta-scheduler which is compatible with the OpenNebula cloud toolkit. This will enable HPC providers to offer IaaS with distinct service levels, which is not possible at the moment.

The offering of service levels can be a distinctive advantage for HPC providers as current contracts normally do not foresee the provision of service levels. Customers gain flexibility by having the possibility to choose between different service levels when submitting jobs. This also allows providers the option of offering previously unsupported service models, for example for urgent computing, which can generate a new revenue stream.

Acknowledgments

This work has been supported by the IRMOS project (http://www.irmosproject.eu/) and has been partly funded by the European Commission's IST activity of the 7th Framework Program under contract number 214777. This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work.

References

[1] Argonne National Laboratories, "OpenPBS Public Home," http://www.mcs.anl.gov/research/projects/openpbs/.

[2] Cluster Resources Inc., "TORQUE Resource Manager," http://www.clusterresources.com/products/torque-resource-manager.php.

[3] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva, "Job submission description language (jsdl) specification, version 1.0," http://forge.gridforum.org/sf/go/doc12582?nav=1, [Online, accessed 8-March-2010].

[4] V. Yarmolenko and R. Sakellariou, "An evaluation of heuristics for sla based parallel job scheduling," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, April 2006, p. 8

[5] R. Sakellariou and V. Yarmolenko, *Job Scheduling on the Grid: Towards SLA-Based Scheduling*. IOS Press, 2008. [Online]. Available: http://www.cs.man.ac.uk/~rizos/papers/hpc08.pdf

[6] J. MacLaren, R. Sakellario, K. T. Krishnakumar, J. Garibaldi, and D. Ouelhadj, "Towards service level agreement based scheduling on the grid," in *Proceedings of the 2 nd European Across Grids Conference*, 2004, pp. 100–102.

[7] C. S. Yeo and R. Buyya, "Managing risk of inaccurate runtime estimates for deadline constrained job admission control in clusters," in *ICPP '06: Proceedings of the 2006 International Conference on Parallel Processing*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 451–458.

[8] K. Djemame, I. Gourlay, J. Padgett, G. Birkenheuer, M. Hovestadt, O. Kao, and K. Voß, "Introducing risk management into the grid," in *e-Science*. IEEE Computer Society, 2006, p. 28.

[9] C. L. Dumitrescu, I. Raicu, and I. Foster, "Usage sla-based scheduling in grids: Research articles," *Concurr. Comput. : Pract. Exper.*, vol. 19, no. 7, pp. 945–963, 2007.

[10] T. Sandholm, "Service level agreement requirements of an accounting-driven computational grid," Royal Institute of Technology, Stockholm, Sweden, Tech. Rep. TRITA-NA-0533, September 2005.

[11] J. Seidel, O. Wäldrich, P. Wieder, R. Yahyapour, and W. Ziegler, "Using sla for resource management and scheduling - a survey," in *Grid Middleware and Services - Challenges and Solutions*, ser. CoreGRID Series, D. Talia, R. Yahyapour, and W. Ziegler, Eds. Springer, 2008, also published as CoreGRID Technical Report TR-0096.

[12] S. Wesner, "Integrated management framework for dynamic virtual organisations," Dissertation, Universität Stuttgart, Stuttgart, Germany, 2008.

[13] R. Schneider, G. Faust, U. Hindenlang, and P. Helwig, "Inhomogeneous, orthotropic material model for the cortical structure of long bones modelled on the basis of clinical ct or density data," *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 27-29, pp. 2167 – 2174, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6V29-4VNH3RH-B/2/1be5c0dd92d2a3f8604519f9cad33a2e

[14] V. Yarmolenko and R. Sakellariou, "An evaluation of heuristics for sla based parallel job scheduling," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, April 2006, pp. 8.

[15] M. Resch, *Entgeltordnung fr die Nutzung der Rechenanlagen und peripheren Geräte des Höchstleistungsrechenzentrums Stuttgart (HLRS) an der Universität Stuttgart*, http://www.hlrs.de/fileadmin/_assets/organization/sos/puma/services/Entgeltordnungen/Entgeltordnung_16-09-2008.pdf, 2008.

[16] BEinGRID Consortium, "BEinGRID project home page," 2008, http://beingrid/.

[17] BREIN Consortium, "BREIN project home page," 2008, http://www.eu-brein.com/.

[18] IRMOS Consortium, "IRMOS project home page," 2008, http://irmos-project.eu/.

[19] NextGRID Consortium, "NextGRID project home page," 2008, http://nextgrid.org/.

[20] FinGrid Consortium, "FinGrid project home page," 2008, http://141.2.67.69/.

[21] C. Baun, M. Kunze, J. Nimis, and S. Tai, "Web-basierte dynamische it-services," 2009.

[22] E. Systems, "Eucalyputs - your environment. our industry leading cloud computing software." [Online; accessed 22-June-2010].

[23] O. P. Leads, "Opennebula: The open source toolkit for cloud computing," [Online; accessed 22-June-2010].

[24] A. W. S. LLC, "Amazon EC2 SLA," http://aws.amazon.com/ec2-sla/, [Online; accessed 2-March-2010].

[25] M. Corporation, "Download details: Windows Azure Compute SLA document," http://go.microsoft.com/fwlink/?LinkId=159704, 2010, [Online, accessed 2-March-2010].

[26] G. Gallizo, R. Kuebert, K. Oberle, A. Menychtas, and K. Konstanteli, "Service level agreements in virtualised service platforms," in *eChallenges 2009, Istanbul, Turkey*, 2009.