1. Employing WS-BPEL Design Patterns for Grid Service Orchestration using a Standard WS-BPEL Engine and a Grid Middleware

André Brinkmann (3), Stefan Gudenkauf (1), Wilhelm Hasselbring (1), André Höing (2),
Holger Karl (3), Odej Kao (2), Holger Nitsche (3), Guido Scherp (1)

(1) OFFIS Institute for Information Technology, Oldenburg, Germany
(2) Technische Universität Berlin, Berlin, Germany
(3) Universität Paderborn – Paderborn Center for Parallel Computing, Paderborn, Germany

In BIS-Grid, a BMBF-funded project in the context of the German D-Grid initiative (http://www.d-grid.de), we focus on employing Grid technologies for information systems integration. The goal is to enable small and medium enterprises (SMEs) to integrate heterogeneous business information systems and to use external Grid resources and services with reasonable effort. To achieve this goal, we develop a Grid workflow middleware – the BIS-Grid engine – that is capable to orchestrate WSRF-based Grid Services. This engine is based upon service extensions to the UNICORE 6 Grid middleware, using an arbitrary WS-BPEL workflow engine. WS-BPEL is an XML format for workflow description and the industry de-facto standard language for service orchestration. One main design decision in BIS-Grid is to leave the WS-BPEL language unmodified to avoid incompatibility with commercial WS-BPEL engines, nor to modify a WS-BPEL engine to ensure its exchangeability and hereby the sustainability of the BIS-Grid engine [1]. Thus, our UNICORE 6 service extensions fill the gap between WS-BPEL and the Grid world. This is in contrast to other approaches where WS-BPEL for Grid Service orchestration is utilised. For example, in [2] a solution is presented that is based on extending BPEL4WS, the predecessor of the WS-BPEL language, also requiring extensions to the workflow engine. This leads to a proprietary solution in which the workflow engine is not exchangeable.

Our design decisions led to several technical challenges. Some are solved by applying so-called *WS-BPEL design patterns*. One category of patterns is called *Grid utilisation patterns*, which describe how to invoke stateful WSRF-based Grid Services with standard WS-BPEL. The development of these patterns is partly based on work by Ezenwoye et al. [3]. Since we regard the application of Grid utilisation patterns as workflow designer tasks, we plan to provide a workflow design tool that supports their application. Another identified design pattern category is called *implementation-specific patterns*. Patterns of this category address architecture specific details that arise within BIS-Grid [1]. For each active workflow, there exist two instances, a workflow service instance in the UNICORE 6 service container, and a workflow instance in the WS-BPEL engine. Implementation-specific patterns describe how these instances are mapped to each other. Since implementation-specific patterns do not represent functional logic of the workflow, they have to be concealed from workflow designers. Thus, they are inserted automatically within the UNICORE 6 extensions.

This paper presents and discusses the identified WS-BPEL design patterns for Grid Service orchestration. Thereby, we focus on Grid utilisation patterns and implementation-specific patterns. The approach is illustrated by the example of the BIS-Grid engine, which is developed by the BIS-Grid project, using ActiveBPEL as WS-BPEL workflow engine and UNICORE 6 as Grid Service container. However, our patterns are intended to address WS-BPEL-based Grid Service orchestration in general. This means that any WSRF-compliant middleware such as Globus Toolkit 4 can be used as Grid Service container. Accordingly, we emphasise the technical issues, present the general solution approach and provide examples on pattern application. Beforehand, we discuss similar approaches and related work. We also describe the implementation of automatic Grid pattern injection and conclude the paper with an overview on our future work in this area.

Acknowledgements. This work is supported by the German Federal Ministry of Education and Research (BMBF) under grant No. 01IG07005 as part of the D-Grid initiative.

References

1. Stefan Gudenkauf, Wilhelm Hasselbring, Felix Heine, André Höing, Odej Kao, Guido Scherp: BIS-Grid: Business Workflows for the Grid; The 7th Cracow Grid Workshop, Academic Computer Center CYFRONET AGH, 2007
2. Tim Dörnemann, Thomas Friese, Sergej Herdt, Ernst Juhnke, Bernd Freisleben: Grid Workflow Modelling Using Grid-Specific BPEL Extensions. German e-Sience Conference 2007, May 2007.
3. Onyeka Ezenwoye, S. Masoud Sadjadi, Ariel Cary, Michael Robinson: Orchestrating WSRF-based Grid Services. Technical report, School of Computing and Information Sciences, Florida International University, April 2007.

2. Towards Workflow Sharing and Reuse in the ASKALON Grid Environment

Jun Qin, Thomas Fahringer

Institute of Computer Science, University of Innsbruck, Austria

Scientific workflow systems are increasingly being used by scientists for the construction of complex experiments executing on distributed Grid resources. Much work in Grid workflows has focused on improving performance by optimizing the use of Grid resources through schedulers and resource managers. As the gain in productivity obtained by workflows is more noticeable, workflow sharing and reuse becomes important, because (i) building a Grid workflow application from scratch is still a challenging task for domain scientists, and (ii) reusing of established and validated workflows not only reduces workflow authoring time but also improves the quality of workflows. Existing work such as [1,2,3] only focuses on workflow component reuse. Goderis et al. [4] identified seven bottlenecks to scalable workflow reuse and repurposing in e-Science, however, mainly focused on how reasoning over ontologies could help in discovering and ranking workflows fragments. ^{my}Experiment [5] identified multiple level of workflow reuse. However, no sub-workflow reuse is discussed.

In this paper, we present a Grid workflow sharing and reuse framework and its implementation as part of the ASKALON Grid environment. The framework introduces the ASKALON Workflow Hosting Environment (AWHE) for workflow sharing among different research groups. The AWHE uses a backend database to store and retrieve Grid workflows, as well as their corresponding metadata (e.g., domain, author, version). The AWHE provides functionalities for users to: (1) publish workflows directly from the workflow composition tool; (2) associate workflows with metadata, especially, workflow versions can be generated automatically based on the latest version in the database; (3) search workflows by their metadata; (4) show views of workflow graphs, including graphs of its sub-workflows; (5) run workflows with a single click; (6) mark a workflow as deprecated for discouraging their use because a better alternative exists. The framework also extends the semantics of the type attribute of atomic activities in the Abstract Grid Workflow Language (AGWL) for simple reuse of Grid workflow components, sub-workflows, and workflows. The extended type attribute can refer to activity types, sub-workflows and workflows. If an activity has a type referring to a sub-workflow or a workflow, executing this activity invokes the corresponding sub-workflow or workflow. In case of sub-workflow reuse, recursive sub-workflow invocation is enabled. To prevent infinite recursion, we develop an algorithm to detect incorrect recursive sub-workflow invocations by traversing the workflow models while building workflows and before submitting workflows for execution on Grid resources. In case of workflow reuse, two kinds of AGWL code generation mechanisms are provided: (1) embedding the AGWL code of the reused workflows into the new workflow so that the workflow engine can execute the new workflow directly without contacting with the AWHE, and (2) using the references of the reused workflows in the new workflow without any embedded AGWL code. The latter is the so-called late binding mechanism. We demonstrate the applicability of our framework using two real world Grid workflows from different scientific domains, i.e., meteorology and material science, respectively.

The presented Grid workflow sharing and reuse framework provides workflow sharing among research groups, and a simple and consistent way for reusing workflows. By using our approach, the workflow representations are simplified in terms of the number of control flows and data flows. The user efforts to compose Grid workflows are also significantly reduced.

Acknowledgements. This work is partially funded by the European Union through the IST-034601 edutain@grid and INFSO-RI-222667 EGEE-III projects.

References

1. Altintas, I., Birnbaum, A., Baldridge, K.K., Sudholt,W., Miller, M., Amoreira, C., Potier, Y., Ludaescher, B.: A Framework for the Design and Reuse of GridWorkflows. In: Proceedings of Scientific Applications of Grid Computing, 2005.
2. von Laszewski, G., Kodeboyina, D.: A Repository Service for Grid Workflow Components. In: International Conference on Autonomic and Autonomous Systems International Conference on Networking and Services, Papeete, Tahiti, French Polynesia, IEEE, 2005
3. Cao, J., Mou, Y., Wang, J., Zhang, S., Li, M.: A Dynamic Grid Workflow Model Based On Workflow Component Reuse. In: Proceedings of Grid and Cooperative Computing (GCC), 2005.
4. Goderis, A., Sattler, U., Lord, P., Goble, C.: Seven bottlenecks to workflow reuse and repurposing. In: Fourth International Semantic Web Conference (ISWC). Galway, Ireland, 2005.
5. Roure, D.D., Goble, C., Stevens, R.: Designing the ^{my}experiment virtual research environment for the social sharing of workflows. In: E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing, Washington, DC, USA, IEEE, 2007

3. e-Science Infrastructure (Tier-2 & Tier-3) for High Energy Physics Data Analysis

Santiago González de la Hoz (1), Gabriel, Amorós (1), Álvaro Fernández (1), Mohamed Kaci (1), Alejandro Lamas (1), Luis March (2), Elena Oliver (1), José Salt (1), Javier Sánchez (1),
Miguel Villaplana (1), Roger Vives (1)

(1) Instituto de Física Corpuscular, Centro mixto Universitat de València – CSIC, Valencia, Spain
(2) Universidad Autónoma de Madrid, Madrid, Spain

The ATLAS computing model [1] describes a hierarchical distributed computing facility consisting of Tier-1 and Tier-2 computing centres, having certain specific memorandum of understanding (MOU) agreed roles and capacities to be used for the benefit of ATLAS as a whole. ATLAS research program decides how these MOU pledged resources are used. In this model primary functions of the Tier-1 are to host and provide long term storage for, access to and re-reconstruction of a subset of the ATLAS RAW data, provide access to the event summary data (ESD) [2], analysis object data (AOD) [2] and TAG data sets and support the analysis of these data sets. The primary functions of the Tier-2's are simulation (they provide the bulk of simulation for ATLAS), calibration, chaotic analysis for subset of analysis groups and hosting of AOD, TAG and some physics group samples.

Tier-3 sites are institution-level non-ATLAS funded or controlled centres/clusters which wish to participate in ATLAS computing, presumably most frequently in support of the particular interests of local physicists (physicists at the local Tier-3 decide how these resources are used). These are clusters of computers which can vary widely in size. An ATLAS Tier-3 task force at CERN has been created to help to document requirements to facilitate setting up Tier-3 for ATLAS use.

The main goal of the Tier-3 task force is to develop a model for Tier-3 and analysis facility sites in ATLAS (including CERN Analysis Facility). Within the ATLAS model such sites will be used mostly for interactive and/or batch analysis of the so called derived physics data (DPD) [2] data sets, which have been produced from AOD data using distributed analysis tools. The definition of different possible DPD formats is been discussed in the analysis model group and will be physics working group or even analysis specific.

It is up to the Tier-3 task force to propose possible Tier-3 configurations and software setups that match the requirements according to DPD analysis needs, as formulated by the analysis model group. The goal of the task force is to provide:

- A set of physics analysis examples to motivate various sizes of ATLAS Tier-3s,
- A set of recommendations and documentation on how to setup a typical ATLAS Tier-3 centre at a university in order to provide a guideline for institutes joining ATLAS and/or starting to set up their own ATLAS computing cluster, and finally,
- A worked out proposal for a software infrastructure to operate such a compute and disk farm for interactive and batch analysis according to the needs of the proposed analysis model.

According to the ATLAS analysis computing model, the analysis is divided into “group” and “on-demand” types. This analysis will be performed by physics groups on Tier-2 resources. This means that users from universities and institutes need some extra computing resources to perform their own work and then contribute with their studies and algorithms to the group effort. Tier-3 centres refer to local computing resources, beyond Tier-1 and Tier-2,that are required to support physics analysis by researchers at universities and institutes. IFIC (Instituto de Física Corpuscular de Valencia), as many other centres, institutes and universities, has a Tier-3 prototype with particular goals and steps.

In this document we present the prototype of the computing infrastructure for data analysis sets up at IFIC (Tier-3), within the framework of the ATLAS experiment [3].The model of this IFIC Tier-3 is based upon the idea that the analysis of the AOD and DPD data needs, in one side, local interactive pre-analysis treatment for analysis software debugging and final DPD analysis, and, on the other side, massive AOD samples treatment running on the grid in order to get large data statistics for precise analysis. Therefore, the computing resources of the IFIC Tier-3 infrastructure will consist on both: A local non-grid farm and resources coupled to the Tier-2 infrastructure sited at IFIC. In that way, the IFIC Tier-3 infrastructure will provide to the user physicists optimized and flexible computing resources for analysis.

Acknowledgements. This work was supported by the Spanish National Research Council (CSIC)

References

1. D. Adams et al., The ATLAS Computing Model, ATL-SOFT-2004-007, CERN, 15 Dec 2004
2. ATLAS Eventa Data Model, https://twiki.cern.ch/twiki/bin/view/Atlas/EventDataModel.
3. S. González de la Hoz et al., Analysis facility infrastructure (Tier-3) for ATLAS experiment, Eur. Phys. J. C 54, 691-607 (2008).

4. Approaching Fine-grain Access Control for Distributed Biomedical Databases within Virtual Environments

Matthias Assel, Onur Kalyoncu, Yi Pan

High Performance Computer Center of the University Stuttgart,
Intelligent Service Infrastructures, Stuttgart, Germany

Currently, doctors still have to make use of traditional and "local" knowledge or information - i.e. own expertise, colleagues at the hospital, libraries, literature etc. as far as available - for making diagnoses and finding the best treatment possible for their patients. With more rare and critical diseases, and in particular under time constraints, such information is typically too difficult to access, if accessible, data might be easily outdated. One of the major difficulties to overcome in order to improve medical diagnosis and treatment consists hence in making knowledge, information, and data accessible for medical experts in a fast, simple and particularly secure way.

The concept of Virtual Organisations (VO) approaches similar issues from an eBusiness perspective, where resources and products (as opposed to information and data) are integrated and consumed dynamically on-demand [1]. By treating (bio)medical data as resources of information, and by applying similar processes to their integration and usage, the capabilities of local resources, too, can be greatly enhanced. Considering the particular circumstances and the high sensitivity of the data involved with respect to privacy issues, careful access control and security play the dominant role in these forms of (virtual) collaborations [2].

In order to ensure that only foreseen users can access certain data resources and, in particularly, view just a snapshot of the entire database content, today's database management systems already provide capabilities to specify fine-grain access rights for pre-defined groups or even single users. In fact, these systems are generally used within one specific environment limited to a particular organisation. While dealing with distributed and heterogeneous data resources for example within virtual collaborations, the local data management approach cannot longer be directly applied. However, lots of work has already been carried out in the field of virtual organisations with respect to fine-grain access control for certain resources typically computational ones [3]. Furthermore, such resources are usually equal to services offered by corresponding service providers such as interfaces to submit computational jobs or storing a huge amount of data. Unfortunately, database systems have not been explicitly the real focus of current research so far.

In this paper, we will describe our approach how fine-grain access control can also be realised for dispersed data resources within such virtual organisations. We will present that our existing solution [4], which is based on access control policies being defined using the Extensible Access Control Markup Language (XACML), and which has already been validated on the data resource level, can be easily extended to master the next step towards fine-grain access control policies for individual, heterogeneous database technologies. We will focus on the definition of corresponding access policies and clearly show how the mapping of certain access rules onto specific data views is implemented in order to limit the user's access accordingly.

Acknowledgements. The work presented in this paper is partially funded by the European Commission through the support of the ViroLab Project Grant 027446.

References

1. L. Schubert, S. Wesner and T. Dimitrakos. Secure and Dynamic Virtual Organizations for Business. Paul Cunningham and Miriam Cunningham [Eds]: Innovation and the Knowledge Economy - Issues, Applications, Case Studies, Volume 2, 201–1208, 2005.
2. M. Assel and A. Kipp. Data Management and Integration within Collaborative Working Environments. In *Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS 2008),* pp. 258-263, Barcelona, Spain, June 2008.
3. A. Butt, S. Adabala, N. H. Kapadia, R. Figueiredo and J. A. B. Fortes. Fine-Grain Access Control for Securing Shared Resources in Computational Grids. In *Proceedings of the 16th international Parallel and Distributed Processing Symposium,* pp. 159, IEEE Computer Society, Washington, DC, April 2002.
4. M. Assel and O. Kalyoncu. Dynamic Access Control Management for Distributed Biomedical Data Resources. *Proceedings of the eChallenges e-2008 Conference*, Stockholm, Sweden, October 2008 (In print).

5. Adopting GLUE 2.0 for Interoperation of Grid Monitoring Systems

Timo Baur (1), Rebecca Breu (2), Tobias Lindinger (3), Anne Milbert (4), Gevorg Poghosyan (5), Mathilde Romberg (2)

(1) Leibniz Supercomputer Centre, Garching, Germany
(2) Forschungszentrum Jülich GmbH, Jülich, Germany
(3) Ludwig-Maximilians-University Munich, Germany
(4) Max Planck Institute for Gravitational Physics, Golm, Germany
(5) Forschungszentrum Karlsruhe GmbH, Karlsruhe, Germany

Problem: Interoperable Grid monitoring has been a topic since many years [1] and standardization efforts have been made by e.g. the Global Grid Forum with respect to monitoring architecture [2] and data schema [3]. Nevertheless, in practice, today's large Grid Infrastructures like EGEE or DEISA are based on a single middleware with a middleware-specific, often proprietary, monitoring system and, independently, additional monitoring components for specific purposes. The German Grid Initiative's infrastructure even extends the diversity by providing access to computing resources through multiple middlewares at the same time [4]. For monitoring such an infrastructure a homogeneous grid-wide solution is necessary. Current existing solutions do e.g. monitoring for special purposes [5] or monitoring level [6], or they offer user-centric monitoring [7]. Homogeneous grid-wide services as well as VO-centric monitoring are missing.

Solution: To ease homogeneous access to resource data for grid-wide services and users, we designed a translating service which acts as a proxy. This service realizes an interoperable and standards-based monitoring between the different monitoring services which are used in the major middlewares. In contrast to just superseding older standards by new ones by exchanging the existing older implementations (which would be a rather costly task), the translating service integrates multiple monitoring systems and thus acts as a bridge. It uses the newly standardized GLUE 2.0 [3] model to exchange the data. The chosen solution has at its core a distributed relational database implementing the GLUE 2.0 schema. For each specific underlying monitoring service (BDII, MDS4, CIS), the translation service is realized as an Extract-Transform-Load (ETL) process. It gathers the data and transforms it by the use of XSLT-scripts into the GLUE 2.0 schema before uploading it to the database. An OGSA-DAI client and Gridsphere portlets then provide access to the integrated data according to VO membership of the querying users and services.

Results: The current system comprises the database, XSLTs, Gridsphere Portlets and an OGSA-DAI service and client. It is installed in the D-Grid Infrastructure and contains actual data from this environment. So far, the data modeled includes administrative and user domains, computing services, endpoints and managers.

The system allows an interoperable, standardized and grid-wide provisioning of the monitoring data provided by different monitoring systems. Furthermore, it allows the identification of inconsistencies in the data of underlying monitoring services caused by their diverging configurations. Thereby, GLUE 2.0 provides an ideal basis for an integration of monitoring data from multiple Grid monitoring systems, especially as it supports the notion of virtual organizations (GLUE 2.0 user domain).

Conclusion: The presentation and the full conference contribution provides a detailed report on the use of the new GLUE 2.0 draft standard for an interoperable and grid-wide resource monitoring realized by a translating service. It also gives a first overview on the experiences gained with the integrated data and the subsequent implications on Grid operations.

Acknowledgements. This work was supported by the German Ministry of Education and Research within the D-Grid Initiative under contract FZK-01IG07010A.

References

1. Serafeim Zanikolas, Rizos Sakellariou: A taxonomy of grid monitoring systems; Future Generation Computer Systems 21 (2005) 163-188
2. B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, R. Wolski: A Grid Monitoring Architecture; Open Grid Forum Document GFD.7, 2002, http://www.ogf.org/documents/GFD.7.pdf
3. Sergio Adreozzi (Ed.), S. Burke, F. Ehm, L. Field, G. Galang, B. Konya, M. Litmaath, P. Millar, J.P. Navarro: GLUE 2.0 Specification V2.0 (draft 33); Open Grid Forum Specification, 2008
4. M. Alef, T. Fieseler, S. Freitag, A. Garcia, C. Grimm, W. Gurich, H. Mehammed, L. Schley, O. Schneider, G.L. Volpato: Integration of multiple middlewares on a single computing resource; Future Generation Computer Systems, Available online 20 May 2008
5. Matthew L. Massie, Brent N. Chun, and David E. Culler: The Ganglia Distributed Monitoring System: Design, Implementation, and Experience; Parallel Computing, Vol. 30, Issue 7, July 2004.
6. INCA http://inca.sdsc.edu/drupal
7. D. Lorenz, S. Borovac, P. Buchholz, H. Eichenhardt, T. Harenberg, P. Mattig, M. Mechtel, R. Muller-Pfefferkorn, R. Neumann, K. Reeves, Ch. Uebing, W. Walkowiak, Th. William, R. Wismueller: Job monitoring and steering in D-Grid's High Energy Physics Community Grid; Future Generation Computer Systems, Available online 28 May 2008.

1. A Need for Data-centric Semantics-Based Infrastructure for e-Science

Marian Bubak (1,2), Tomasz Gubala (2,3), Maciej Malawski (1), Piotr Nowakowski (2), and Tomasz Szepieniec (2)

(1) Institute of Computer Science, AGH University of Science and Technology, Poland
(2) ACC Cyfronet AGH, Krakow, Poland
(3) Informatics Institute, Faculty of Science, Universiteit van Amsterdam, The Netherlands

Contemporary means of sharing scientific results fall short of the requirements of modern science [1,2]. While modern virtual laboratory and grid systems enable their users to store and access large quantities of scientific data, they typically do not permit cross-system sharing and community access to such data.

In order to shift focus from secondary sources to the actual data and algorithms used in scientific research, a unified, collaborative environment is required. There is a need for an operational, semantic framework integrating various sources and forms of data, ensuring trust among community members and facilitating retrieval and reuse of actionable knowledge [3,4].

We propose to develop a broad, multidisciplinary, production data infrastructure, basing on the European achievements of network and grid infrastructures, enabling e-Science researchers to extract meaning from masses of data stored in institutional, national and community repositories. This infrastructure will support major data repositories along with processes involved in the research cycle: authoring, publishing, managing, sharing, accessing, analyzing, reusing and annotating data.

To achieve this, we present how to develop, deploy, and support a unified data sharing service platform for all major domains of e-Science; as well as we describe a set of common services which permit exposure of scientific data (including algorithms) to wide communities while respecting the policies of each individual user community and data provider.

It is necessary to push for integration and federation of existing system-level science technologies and to involve representatives of major European grid projects (EGEE, DEISA, and national grid initiatives) and virtual laboratory developers (Taverna, myExperiment, VL-e, GridSpace)

The beneficiaries of the project include collaborating individuals and groups of users applying the system-level science research methodology.

Acknowledgements. The Authors are grateful to Peter M.A. Sloot, Carole Goble, David De Roure, Sean Bechhofer, Matthias Assel, Mario Cannataro, Adam Belloum, Zhiming Zhao, Rob Belleman, Ad Emmen, Daniel Harezlak, Tomasz Bartynski and Eryk Ciepiela for many fruitful discussions.

References

1. Foster, I. and Kesselman, C.: Scaling system-level science: Scientific exploration and IT implications. IEEE Computer, 39(11):31-39, 2006.
2. Sloot, P.M.A., Altintas, I., Bubak, M.T., Tirado-Ramos, A., Boucher, C.: From molecule to man: the system science of decision support in individualized e-Health, IEEE Computer, 39(11):40-46, 2006.
3. M. Bubak, T. Gubala, M. Malawski, B. Balis, W. Funika, T. Bartynski, E. Ciepiela, D. Harezlak, M. Kasztelnik, J. Kocot, D. Krol, P. Nowakowski, M. Pelczar, J. Wach, M. Assel, A. Tirado-Ramos: Virtual laboratory for development and execution of biomedical collaborative applications. In: S. Puuronen, M. Pechenizkiy, A. Tsymbal, D-J. Lee (eds) Proc. 21st IEEE International Symposium on Computer-Based Medical Systems, June 17-19, 2008, Jyvaskyla, Finland, pp. 373 - 378, DOI 10.1109/CBMS.2008.47
4. DeRoure, D., Goble, C., Stevens, R., Designing the myExperiment virtual research environment for the social sharing of workflows. e-Science 2007: Third IEEE International Conference on e-Science and Grid Computing, 2007, Bangalore, India; 10-13 December 2007; pp. 603-610.

2. Knowledge Supported Data Access in Distributed Environment

Renata Słota (1), Darin Nikolow (1), Jacek Kitowski (1,2)

(1) Institute of Computer Science, AGH, Kraków, Poland
(2) Academic Computer Center CYFRONET AGH, Kraków, Poland

Modern eScience approach requires usage of many advanced computer science technologies. The number of cooperating scientific teams constantly grows. Software utilities for storing and sharing the gathered knowledge in a global manner are necessary for a successful world-wide cooperation. Scientific experiments produce huge amounts of data. Relevant methods and software are necessary to access, integrate and process these data in any distributed computational environment.

As the data processing capabilities grow a demand for more efficient systems with bigger storage capacities rises up. Proper and efficient management of data storage systems, and especially HSM systems, is essential for many processes occurring in distributed computational environments such as: replica selection, new replica creation, specifying SLA parameters, guarantying the required quality of service or required level of data protection and availability. Currently, the computational environments, are often supplied with methods enabling to use knowledge represented by ontological descriptions of system components. In this way it is possible to use the distributed resources more efficiently and to create and integrate new applications easier using existing services semantically described.

In this paper we present new vision of methodology concerning organization of data access in knowledge supported distributed computational environment with respect to various types of storage systems. Since the semantic approach [1] offers more flexible and more general representation of data and resources our solution is based on applying semantic technologies, using ontologies, to model storage systems and to organize data access. As part of this research a unified storage system model (SSM) for describing the state of any storage system including HSM systems has been proposed [2]. Relevant SSM ontology and corresponding services are being developed. Two use cases will also be presented in order to show how the SSM, its ontology and the proposed services can be applied. The first one concerns optimization of data access for replicated data sets. A special optimization service will be implemented for choosing the replica location of a newly created replica and for selecting the best replica for data read access. The second one concerns using the proposed services for controlling the quality of service in virtual organization by monitoring the appropriate parameters specified in SLA.

Acknowledgements. This research is supported by the MNiSW grant nr N516 405535.

References

1. NGG3 Group, *Future for European Grids: GRIDs and Service Oriented Knowledge Utilities*, January 2006, ftp://ftp.cordis.lu/pub/ist/docs/grids/ngg3_eg_final.pdf
2. D. Nikolow, R. Slota, and J. Kitowski, *Grid Services for HSM Systems Monitoring*, in: R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski (Eds.), Proceedings of 7-th International Conference, PPAM 2007, Gdansk, Poland, September 2007, LNCS 4967, Springer 2008, pp.321-330.

3. A File System based eScience Workbench

Roger Menday

Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, D-52425 Jülich, Germany

Grid middleware provides access to distributed resources and does so by normalising the interface to them. A request for some action at one resource can, with relative ease, be re-targeted to another. Acting seamlessly across multiple resources in the collective layer is arguably the distinguishing functionality offered by the Grid and leads to the coordinated usage of multiple resources. Examples include workflow and also searching for data across multiple storage systems and managing backup/replication strategies. In this paper we describe an environment, called Pages, a workflow system which provides tools to perform such collective layer functionality.

Most commonly workflow systems for the Grid involve parsing a workflow description format into a form which can be executed by a workflow engine. The result is quite a ‘hands-off’ experience for the user during execution, as it is difficult to interact with the running workflow. There are many reasons why this is the case, but one is that the source of the workflow is lost during the parsing stage. However the eScience process is often not ‘fire-and-forget’. Intermediate results are often as important as the final result, and 'tinkering' (trying things out) is an important part of the process too.

We first recognise that user-driven workflow is a valid usage model. Such ‘casual workflow’ starts before knowing where it will finish, i.e. it is an process with the user essentially orchestrating the various parts, normally often using command line tools or scripts to initiate and manage Grid activities. Our system supports this mode of operation, and then adds degrees of automation. The central strategy to achieve this was to eliminate the parsing stage by choosing a workflow representation which supports the co-existence of users interacting with the workflow throughout the entire lifecycle with the run-time automation. We use the filesystem as the basis of the environment and follow conventions on files, their arrangement in the file system and the current working directory. Through doing so we are able to integrate independent activities initiated through command line tooling into a larger workflow of activities.

The user’s environment consists of a set of workbooks. A workbook contains a number of pages, where each page groups sets of actions which are governed through implied dependencies. Explicit dependencies can be established between separate pages in a workbook. Workbooks can be packaged and re-distributed simply as a file archive. Command line and API tooling uses the current working directory to locate its context in the graph, for example, when adding new dependencies or when initiating a new job somewhere on the Grid. Furthermore, scripts are invoked as life-cycle events are reached in workflow which can then be used to dynamically manipulate the workflow further. Finally, publishing and remote access to workflows is achieved through publishing HTML representations of the workbooks and their contents.

The result is a problem solving environment for eScience which places the user as an active participant. It is deeply integrated into the working environment of the user, can be extended through scripting, allows for the integration of local processing, encourages dynamic interaction, run-time manipulation and finally, automatic dissemination and sharing of the resulting workbooks into the Web.

4. DFSgc: Distributed File System for Multipurpose Grid Applications and Cloud Computing

Carlos de Alfonso, Miguel Caballer, José V. Carrión, Vicente Hernández

Grid and High Performance Computing Research Group, Universidad Politécnica de Valencia, Valencia, España

Grid Computing has evolved and currently needs the support for managing huge quantities of storage. Most of Grid deployments only provide local storage support and the applications may guess where the data files are effectively stored and transfer them to where they are needed. There are some approaches of virtualization of the data storage such as EGEE's Replica Service [1], LCG File Catalog [2], Replica Location System from Globus and DataGrid [3], Hadoop Distributed File System [4] or Microsoft's Distributed File System [5]. None of them has been consolidated by the community, because they omit essential features such as decentralization of the file catalog, seamless bringing the data closer to where it is needed or automating an intelligent replication.

This paper introduces the design of a distributed file system whose aim is to provide a virtual storage for multipurpose applications. This file system provides tools for automatic synchronous and asynchronous replication. The replication system introduces heuristic techniques for bringing files to physical storages which are nearer in the network to the applications which are using them.

The virtual storage is understood as a whole memory made up of caches (storage nodes), and applies cache coherence techniques to ensure the integrity of the copies in the system, and replication for maintaining the whole accessibility of files.

Many of other approaches introduce some kind of central catalog or file access. In our case, the system introduces a virtual namespace which can be used by the applications in order to reference the files. Also the virtual namespace is not joined to any physical storage. On the other hand, the namespace is maintained by a distributed catalog, which is spread among the distinct storage nodes. The catalog includes metadata in order to create a robust, fault tolerant system which is easy to recover in case of failure of a storage node, and also it is easy to scale by introducing new storage nodes.

The idea of this distributed file system is to specify an upper level layer which virtualizes the storage and allows users and applications to read and write files in the system as if they were working with a local file system.

In the lower layer the system integrates with several storage endpoints such as FTP, HTTP, GridFTP servers or simple access to local file system, by means of a plug-in-like architecture. Also the transference of files is carried out by standard transference protocols such as FTP, HTTP or GridFTP.

This file system is mainly addressed to Grid applications to transparently access to files which are anywhere in the physical storage nodes in the deployment. The goal of the system is to boost the applications by bringing the data near to them.

Also the system is suitable for current cloud deployments as it fully virtualizes the access to the storage and does not introduce any physical addressing or dependence for applications to access to the files.

References

1. Jean-Philippe Baud, James Casey, Sophie Lemaitre, Caitriana Nicholson, Graeme Stewart. LCG Data Management: From EDG to EGEE. CERN, European Organisation for Nuclear Research, 1211 Geneva, Switzerland.
2. LCG File Catalog administrators' guide. https://twiki.cern.ch/twiki/bin/view/LCG/LfcAdminGuide
3. Chervenak, et. al, Giggle: A Framework for Constructing Scalable Replica Location Services, Proc. of SC2002 Conf., Baltimore, MD, 2002.
4. The Apache Hadoop project. http://hadoop.apache.org/core/
5. Microsoft Distributed File System. http://technet.microsoft.com/en-us/library/cc738688.aspx

5. Scalable Services for Digital Preservation

Rainer Schmidt, Christian Sadilek, Ross King

Austrian Research Centers GmbH - ARC, Vienna, Austria

Due to rapid changes in information technology, digital data, documents, and records are doomed to become un-interpretable bit-streams within short time periods. Digital Preservation deals with long-term storage, access, and maintenance of digital media objects. In order to prevent a loss of information, digital libraries and archives are increasingly faced with the need to electronically preserve vast amounts of data by holding limited computational resources in-house. However, due to the potentially immense data sets and computationally intensive tasks involved, preservation systems have become a recognized challenge for e-science and e-infrastructures [1]. In this paper, we argue that Grid and Cloud technology can provide the crucial technology for building scalable preservation systems. We introduce a strategy for utilizing cloud infrastructures that is based on platform virtualization (e.g. based on Xen [2]) as a scaling infrastructure for the execution of preservation workflows. We present recent developments on a Job Submission Service (JSS) that is informed by Grid standards (e.g. JSDL[3], HPCBP[4]) and capable of utilizing large clusters of virtual machines. Moreover, we present initial experiments that have been conducted using the Amazon EC2 and S3 cloud services (AWS) [3].

The EU project Planets aims to provide a service-based solution to ensure long-term access to the growing collections of digital scientific and cultural assets. Within this project, the Interoperability Framework (IF) provides the technical environment for integrating preservation services, meta-data, and archival storage elements. Components that perform *preservation actions* often rely on preinstalled tools (e.g. a file format converter) that are wrapped by a service interface on the lowest-layer. The Planets workflow engine implements a component-oriented enactor that governs life-cycle operation of the various preservations components, such as instantiation, communication, and data provenance. Distributed preservation workflows are conducted from high-level components that abstract the underlying protocol layers. A crucial aspect of the preservation system is the establishment of a distributed, reliable, and scalable computational tier. A typical preservation workflow may consist of a set of components for data conversions, storage, quality assurance, and data model manipulations and may be applied to tens of thousands of digital objects. In principle, these workflows could be easily parallelized and run in a massively parallel environment. However, the fact that preservation tools typically rely on closed source, 3rd party libraries and applications that often require a platform-dependent and non-trivial installation procedure prevents the utilization of standard HPC facilities. In order to efficiently execute a preservation plan, a varying set of preservation tools would need to be available on a scalable number of computational nodes. The solution proposed in this paper tackles this problem by incorporating hardware virtualization, allowing us to instantiate sets of transient system images on demand, which are federated as a virtualized cluster.

We present a Job Submission Service that is informed by standard Grid technology and protocols, and utilized as the computational tier of a digital preservation system. Jobs are capable of executing data-intensive preservation workflows by utilizing a Map/Reduce implementation that is instantiated within the cloud infrastructure. The presented system relies on the Planets Interoperability Framework, Apache Hadoop [5], and a JSS prototype providing a Grid middleware layer on top of the AWS cloud infrastructure. The paper gives an overview of the Planets Interoperability framework, it's extensions towards utilizing Grid and Cloud technology, and initial results that have been conducted on the Amazon cloud infrastructure.

Acknowledgements. Work presented in this paper is partially supported by European Community under the Information Society Technologies (IST) 6th Framework Programme for RTD – Project IST-033789.

References

1. Digital Curation: digital archives, libraries, and e-science. http://www.dpconline.org/graphics/events/digitalarchives.html [Sept. 2008]
2. Xen Project Homepage: http://xen.org/ [Sept. 2008]
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, 2008.
4. Dillaway, B., Humphrey, M., Smith, C., Theimer, M. and Wasson, G. HPC Basic Profile, v. 1.0. GFD-R-P.114. Available at http://www.ogf.org/documents/GFD.114.pdf.
5. Amazon Web Services. http://aws.amazon.com [Sept. 2008]
6. Apache Hadoop Project. http://hadoop.apache.org/ [Sept. 2008]

6. METACenter Virtual Networks

David Antoš, Jiří Sitera, Petr Holub, Luděk Matyska

CESNET, Zikova 4, Prague, Czech Republic

Advances of the METACenter – the Czech national grid and supercomputing infrastructure--are tightly connected with virtualization concepts and technologies during last years. Important part of METACenter computation resources is currently virtualized, managed by a service we designed and implemented [1], gaining flexibility for both end users and resource owners.

The METACenter project traditionally builds on advanced services of the Czech National Research and Education Network (CESNET2+). In the era of virtualization, our cooperation is mainly related to network services for virtual private L2 networks [2]. Virtualization of the network is one of the basic building blocks in providing dynamic virtual clusters – sets of virtual machines interconnected by its own private virtual network encapsulating the clusters, protecting them, and managing access.

Technologies like Virtual Private LAN Service, QinQ (IEEE 802.1ad), and Cisco Xponders can be deployed in a high-performance state-wide network to transport virtualized traffic over the backbone without degrading performance. Those technologies are available in the CESNET core among all METACenter sites (Brno, Prague, Pilsen). Each virtual network can be built and torn down on demand without intervening of network core administrators and without significant resource usage. In particular, thus we have full range of VLAN IDs as defined by 802.1q available dynamically for METACenter needs under full control of grid middleware.

The common Layer 2 network allows us to build geographically distributed clouds of virtual machines on a single logical network segment with the possibility of transparent migration without any noticeable borders previously represented by separate METACenter sites. With the possibility to dynamically create many virtual networks we can also make new borders totally independent on the physical ones. Each user, group or specific application can have its resources connected to its own virtual network. The VLAN simply becomes a new class of resource available to the users, accessible in a way similar to other resources (via scheduler and under coverage of common authorization service).

Each virtual cluster can contain not only set of virtual machines but also other resources/services (a filesystem/sandbox server, batch system, special devices, etc.) and have its own level of network based isolation. Each VLAN/cluster can be directly or indirectly connected to the Internet, possibly hiding details (local IP addresses, DNS names, insecure ports, etc.) or isolated (accessible only by its user by a VPN-like tunnel service, even allowing the user to publish the cluster under his/her own address space).

The concept of virtualized cluster network is also an effective way how to run "insecure" virtual machines. It need not be only machines running user's own OS image, but also machines with applications containing components not proved to be secure or with potential side effects to other machines (typically nonstandard communication libraries or daemons).

The paper will describe design and prototype implementation of virtual network project including detailed use-cases, experiences and measurements of real-world CESNET L2 services behavior.

References

1. Miroslav Ruda – Jiří Denemark, Luděk Matyska: Scheduling Virtual Grids: the Magrathea System. In Second International Workshop on Virtualization Technology in Distributed Computing, USA, ACM digital library, 2007. p. 1-7. 2007, Reno, USA.
2. Václav Novák, Pavel Šmrha, Josef Verich: Deployment of CESNET2+ E2E Services in 2007, CESNET technical report 18/2007, http://www.cesnet.cz/doc/techzpravy/2007/cesnet-e2e-services/

7. Analysis of Overhead and Waiting Times in the EGEE Production Grid

Max Berger, Thomas Zangerl, Thomas Fahringer

Leopold-Franzens-Universität, Innsbruck, Austria

“Enabling Grids for E-sciencE (EGEE) is the largest multi-disciplinary grid infrastructure in the world, which brings together more than 140 institutions to produce a reliable and scalable computing resource available to the European and global research community.” [1] The EGEE project is now in its third phase, where the focus has switched from Grid research to that of an infrastructure platform. We analyze this infrastructure to verify if it fulfills the promised claims.

Glatard et. al [2,3,4] have tried to model the latency in EGEE jobs with a standard mathematical model. They found the time between submitting a job and its execution to average 393 seconds with a standard deviation of 792 seconds. Oikonomakos et. al [5] have analysed the distribution and waiting-time of jobs on one particular Grid site. This work, however, left room for improvement: First, the measurements were taken two years ago; the gLite middleware has significantly improved in both speed and reliability. Second, the measurements were not related to the site they where run on, or where taken on only one site. And last, the measurements were taken on the reported time, and not the actual time; the EGEE software has significant delays in notifications due to the design of its information service.

In our measurements we discovered that the speed of the gLite middleware has significantly improved in the last two years, averaging about 280 seconds for job execution delay. We have taken measurements for every step of the jobs lifecycle, from submission to WAITING, SCHEDULED, RUNNING, DONE, and CLEARED state. We have also measured the actual time when a job started and finished its execution through a callback-mechanism. We present current results which show that the information service introduces an overhead of about 220 seconds between the time a job actually finishes and the notification of the user.

We also analyze the “weekend-effect”. While a “folk theorem” claims that Grid scheduling times are shorter on the weekend, there were no previous measurements to support or deny this fact. We discovered that the weekend-effect exists – but not on Saturday and Sunday, but on Sunday and Monday.

The next effect we analyze is the reliability in relation to the actual grid site a job was scheduled to. We discovered that the job latency and reliability are directly dependent on the site a job is actually scheduled to, and that some sites proof to be much faster and more reliable than others.

We conclude that a simple analysis of scheduling time in the EGEE network or on one single site does not provide sufficient results – the factors weekday and site play a large role in the actual scheduling latency. We outline some of the possible decision changes that could be made to improve scheduling in the EGEE infrastructure. We also set the overhead time in relation with job run-time, to estimate the types of jobs suitable for EGEE Grid execution.

Acknowledgements. This work was partially supported by EU project EGEE-III, INFSO-RI-222667

References

1. Enabling Grids for E-Science, http://www.eu-egee.org/, retrieved 9.9.2008
2. Tristan Glatard, Johan Montagnat, Xavier Pennec: Optimizing jobs timeouts on clusters and production grids, Proceedings of International Symposium on Cluster Computing and the Grid (CCGrid). Rio de Janeiro, 2007, 100-107
3. Diane Lingrand, Johan Montagnat, Tristan Glatard: Modeling the Latency on Production Grids with Respect to the Execution Context, Proceedings of 8th IEEE International Symposium on (CCGrid), 2008, 753-758
4. Diane Lingrand, Johan Montagnat, Tristan Glatard: Estimation of latency on production grid over several weeks, Proceedings of the ICT4Health, Oncomedia, Manila, Philippines, 2008
5. Michael Oikonomakos, Kostas Christodoulopoulos, Emmanouel Varvarigos: Profiling Computation Jobs in Grid Systems, Proceedings of International Symposium on Cluster Computing and the Grid (CCGrid). Rio de Janeiro, 2007, 197-204

8. AgroGrid: Composition and Monitoring of Dynamic Supply-Chains

Eugen Volk

High Performance Computing Center Stuttgart, Stuttgart, Germany

Today, enterprises in the agricultural sector collaborate in fixed partner structures with long term contract relations. Short term appearing peaks in supply and demand of capacities cannot be levelled out by using appropriate supply chain management systems involving all market partners. Thus, today's capacities cannot be exploited in an economic efficient manner. So, the motivation of AgroGrid – Business Experiment within EU project BEinGRID – is to create grid enabled market and to enable companies to deploy their capacities extensively and, simultaneously, to ensure food safety via efficient tracking and tracing of goods and monitoring mechanisms. Therefore, concepts from the Grid-community will be employed to enable composition and monitoring of dynamic supply-chains in agriculture food industries [1].

AgroGrid - introduces Grid technology into the IT industry acting in the agricultural sector and enhances current IT capabilities by introducing Grid technology in order to enable companies delivering better, cheaper and faster services for their customers. In order to achieve these objectives, AgroGrid designs and implements services for the composition and monitoring of dynamic supply-chains in agriculture food industries using Grid technology making use of trust-building commercialization support mechanisms. The platform realized by AgroGrid consists of the AgroGrid portal and monitoring infrastructure. The AgroGrid portal provides a web-based user interface to AgroGrid services, like capacity publication, capacity query, Service Level Agreement negotiation and evaluation-reporting.

It integrates all AgroGrid services into a common interface with a unified look-and-feel, enabling, after successful user-authentication and authorization, a user-friendly interaction to all services.

The monitoring infrastructure of the AgroGrid platform is based upon GTNet® [2] functionality, which maintains tracking and tracing information of food-trade-units stored in the local databases of all supply-chain members. The monitoring infrastructure enables monitoring of Service Level Agreements (SLAs) between supply-chain members, based on query of monitoring information stored alongside supply-chain members' databases. A negotiated and contracted SLA between two parties in AgroGrid contains SLA-terms defining not only the amount, quality, and price of food products to be delivered, but also environmental conditions, under which they are stored and shipped. These SLA-terms define monitoring-metrics which are used to monitor the quality and especially environmental condition of incoming, stored or shipped food trade units. Every supply-chain partner publishes the monitoring data about incoming, stored or shipped food trade unit in his/her local-database, allowing access only to the buyer of the food trade unit, after receiving the trade-unit-id shipped with the food trade unit. The access to his/her local database is managed by each partner locally, based on the GTNet® access mechanisms. As a consequence of restricted access, the approach proposed in AgroGrid is based on the hierarchical SLA-Monitoring and SLA-Evaluation, deployed within each partner of the supply-chain separately. The SLA-Monitoring service within each partner query GTNet for the unique trade unit-id shipped with the food trade unit. As a result, the monitoring data from the database of the product provider or logistic company, which delivered the product, is returned. The monitored data are evaluated against the SLA-terms contracted in the SLA. If SLA violations are detected, the affected parties are informed immediately. The result of SLA-Evaluation is stored in the local database and is offered to hierarchically higher settled partner in the supply-chain – buyer of the buyer, after the shipment of the processed or transformed food trade unit.

The approach proposed in AgroGrid enables food industry to build and extend supply-chains (managed within AgroGrid as dynamic VO) by new partners, in flexible and dynamic manner, while ensuring monitoring of food-quality on each level and all sub-levels of the supply-chain hierarchy, determined by the order of the supply-chain partners. The quality management mechanism proposed in AgroGrid establishes trust-building commercialization support mechanism between all partners in the supply-chain.

Acknowledgements. The results presented in this paper are partially funded by the European Commission through the BEinGRID [3] project.

References

1. Eugen Volk, Ansger Jacob, Marcus Müller, Martin Waldburger, Peter Racz and Jon Petter Bjerke. D4.22.1: Design Specifications BE22 AgroGrid, 2008
2. Tracetracker, Technical Architecture Whitepaper, http://www.tracetracker.com/cgi/doc.cgi?id=25
3. BEinGRID EU IST Project, http://www.beingrid.eu
4. AgroGrid website: http://www.beingrid.eu/index.php?id=be22agrogrid

1. Application of Petri Nets to Evaluation of Grid Applications Efficiency

Wojciech Rząsa (1), Marian Bubak (2,3)

(1) Department of Computer *and Control Engineering, Rzeszow University of Technology, Poland*
(2) Institute of Computer Science AGH, Krakow, Poland
(3) ACC CYFRONET AGH, Krakow, Poland

Described research concerns efficiency of distributed applications as a function of communication overhead and delays caused by other distributed resources. Applications designed for the Grid environment and the Grid management systems are in the center of the research. The main goal of the work is to work out a method enabling estimation of efficiency of a distributed application depending on its topology and parameters of exploited resources. The method is designed to be convenient for developers of the applications and should make them able to evaluate their designs before laborious implementation.

Results of work concerning secure communication in Grid monitoring system were the first motivation for the research [1]. Experiments revealed significant overhead introduced by cryptography based secure communication. The overhead can significantly affect efficiency of whole distributed application [2,7]

Communication overhead for a single communication channel may be a result of various network link parameters, but may also be a consequence of application-level protocols exploited. The overall application efficiency depends on a combination of communication overheads for individual communication channels, delays being a result of exploitation of other resources, application topology and to some extent application logic that implies various exploitation of different resources. Estimation of the overall application efficiency on the basis of the partial overheads is the aim of this work. The first concept of the solution was presented in [6].

The first task to be solved is apparent contradiction between requirements concerning comfort of the application developers and the ones resulting from necessity to provide reliable results of the application analysis. On the one hand the developers should be able to create a model of their application on possibly high level of abstraction, providing possibly scant information to enable modeling of application in early development stages. On the other hand the application should be described using a formalism providing means to perform reliable analysis of the model. The formalism must necessarily be able to reflect and enable analysis of all activities of distributed, parallel applications. The solution for this problem is a high-level model of distributed application provided by the developers transformed automatically to a formalism providing reliable analysis methods.

Simulation was chosen as an analysis method for the application. The reasons for the choice were on one hand reliable results that can be obtained from the simulation and on the other hand limited restrictions for the model that can be analyzed using this method. Moreover simulation gives an opportunity to observe operation of the application being analyzed facilitating comprehension and removal of occurring problems.

The simulation of the application is performed using Timed Colored Petri Nets (TCPN) [3,4] that are a convenient modeling language designed for concurrent activities enabling step-by-step simulation of a model. Thus the high-level model of a distributed application is transformed to TCPN based executable model and then simulated. The transformation is performed automatically and thus it is transparent for the user that need not be aware of any details concerning the executable model in particular the use of Petri nets.

Working out the TCPN model of distributed application presents another research problem. The other issue not included, but connected with the executable model is that while simulation we must necessarily be able to estimate network transmission time for particular communication channels depending e.g. on available bandwidth, link capacity (being a result of queues implemented in network devices), delay-bandwidth product, two-way or one-way transmission and obviously network protocols used.

In research over the TCPN model existing tools are exploited: Petri Net Kernel and Petri Net Cube. Taking advantages of them and supplementing shortages prototype implementation of the simulator is being implemented to enable case study of the method. For the first experiments Linux based testbed and globus_xio based test application were prepared. Preliminary results of simulation will be presented together with conclusions concerning future work. The further research will include comparison of simulation results with real-world application. The Grid extension for ATLAS TDAQ filtering system deployed in CERN [5] will be used to verify results obtained from the simulator.

References

1. Baliś B., Bubak M., Rząsa W., Szepieniec T., Wismüller R.: Two Aspects of Security Solution for Distributed Systems in the Grid on the Example of the OCM-G. In proc. of CGW'03, Kraków 2004.

2. Baliś B., Bubak M., Rząsa W., Szepieniec T.: Efficiency of the GSI Secured Network Transmission}, Proc. of Internationnal Conference on Computational Science, Kraków, Poland, June 2004, LNCS~3036, pp 107-115, 2004.
3. Jensen K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Vol 1: Basic Concepts. EATCS Monographs on Theoretical Computer Science, Springer-Verlag 1994.
4. Jensen K.: Coloured Petri Nets with Time Stamps. Computer Science Department, Aarhus University, Denmark 1993.
5. Korcyl K., Szymocha T., Kitowski J., Zieliński K., Funika W., Slota R., Dutka L., Pieczykolan J., Balos K., Guzy K., Kryza T.: The Atlas Experiment on-line Monitoring and Filtering as an Example of Soft Real-time Application, Presented at The Conference of the High Performance Computers' Users, Zakopane, Poland, March 6-7, 2008. Submitted for publication in Computer Science AGH.
6. Rząsa W., Bubak M., Baliś B., Szepieniec T.: Simulation Method for Estimation of Security Overhead of Grid Applications, In proc. of CGW'05, Kraków 2006.
7. Rząsa W., Bubak M., Baliś B., Szepieniec T.: Overhead Verification for Cryptographically Secured Transmission in the Grid. Computing and Informatics, Vol. 26, 2007, 89-101.

2. Automatic Verification of SLA for Firewall Configuration in Grid Environments

Gian Luca Volpato, Christian Grimm, Martin Janitschke

RRZN – Regional Computing Centre for Lower Saxony, Gottfried Wilhelm Leibniz Universität Hannover, Germany

Integration of new partners within existing Grid environments may always present unforeseen hitches, especially in the area of network configuration. The definition and setup of a correct firewall configuration are processes prone to mistakes and inaccuracy, also because site administrators are not eager to continuously adapt firewall configurations to the ever-changing needs of the Grid user communities [1]. Each misconfigured firewall represents a double threat for resource providers: on one side it may prevent legitimate communications to take place, on the other side it may allow malicious code to establish forbidden communication paths. For both reasons, verification of correct firewall configuration must be performed always before allowing a new partner to enter the collaboration and then constantly for the entire lifetime of the project.

Actually, in the German Grid Initiative (D-Grid) we envisage that the periodic certification of firewall setup becomes a fundamental part of the Service Level Agreement signed among the project partners. The specification of such an SLA should provide also for different levels of security, defined in terms of minimum network accessibility and corresponding firewall rules to be enforced at a site. In particular, this requirement applies to Grid infrastructures providing resources to communities with different security standards, e.g. aerospatial engineering compared to astrophysics.

In our work, we started by identifying up to four different security profiles [2], taking into account the type and number of Grid middleware components that can be deployed. Resource providers wishing to enter a scientific collaboration are requested to authorize incoming traffic directed to the ports listed in the selected profile.

The periodic verification of the SLA part concerning firewall setup could be done by means of some network and firewall analysis tool, already available either as open source project or as commercial product. None of the considered tools, however, is designed to be used as a central service responsible for the comprehensive and continuous validation of multiple sites participating to a common Grid project. Therefore we present here a newly-developed software that automatically checks at given intervals the firewall configuration of selected partners. As input it needs the list of sites to be tested and the security profile chosen by each site. The complete test suite for the given site and security profile is then automatically generated and executed. In its basic operation mode the tool verifies that all ports listed in a specific profile are not blocked and reachable from external hosts; optionally the tool can be instructed to verify that correct services are running at correct ports and that connections to all other ports are rejected. Site administrators will receive a warning notification whenever one or more tests fail.

Thanks to the definition of the security levels for the SLA and the tool for automatic firewall verification we are now able to provide a comprehensive solution for the easy integration of new partners into the D-Grid project.

References

1. T. Metsch et al.: Requirements on operating Grids in Firewalled Environments, February 2008, http://www.ggf.org/Public_Comment_Docs/Documents/2008-09/ogf-firg-firewall-existing-solutions-overview-v1.0.0.pdf
2. G.L. Volpato, C. Grimm: Definition of Security Levels for the D-Grid Infrastructure, June 2008, http://www.rrzn.uni-hannover.de/fileadmin/ful/mitarbeiter/volpato/FG3-3_MinSecurityLevel.pdf

3. Applying Risk Management to Support SLA Provisioning

Dominic Battré (1), Georg Birkenheuer (2), Matthias Hovestadt (1), Odej Kao (1), Kerstin Voss (2)

(1) Technische Universität Berlin, Germany
(2) Paderborn Center for Parallel Computing, Universität Paderborn, Germany

The attractiveness of using computational Grids from the perspective of small and medium enterprises (SMEs) is that they do not have to buy and maintain own compute resources. However, outsourcing job executions implies that users have no longer any control on job scheduling and management of failures. Service Level Agreements (SLAs) are powerful instruments for contractually defining guarantees and obligations [1] in a business relationship between a service consumer and the service provider in the Grid.

From the perspective of Grid providers, committing to an SLA inherits a risk since resources can fail which might lead to SLA violations. The analysis of Iosup et. al. about resource availability in Grid'5000 [2] shows that Grids are unreliable environments and resources fail often, i.e. the mean time between failure of a cluster node was observed to be about 45 hours. This unreliability demands for a suitable management on the provider site. On the one hand fault-tolerance mechanisms, such as checkpointing and migration [3], are essential; on the other hand decisions have to be made by considering probabilities of events and their consequences, i.e. risk. Standard risk management processes, such as FERMA [3] or the AS/NZS standard [4], are not suitable since these require human interaction. In order to integrate risk management into the automated processes within the provider's resource management system (RMS), a Grid risk management process was developed which needs no human interaction after the configuration phase.

This paper describes the automated risk management process which was derived from the FERMA standard and is applicable for the usage in Grids. The process is depicted; the steps are detailed and compared with the FERMA standard. Most of the FERMA tasks can be transferred into the Grid context; in particular, the Grid risk management steps can be defined more precisely due to the RMS is the specific field of application. Since the objectives of all providers conform to making the highest profit and fulfilling as many SLAs as possible, the tasks of the Grid risk management process are fixed and can be refined by individual policies of the provider. By performing the manual configuration phase, as described in the paper, the provider is able to adjust the risk management process to its individual environment.

A reference risk management implementation is realized in scope of the AssessGrid project. A detailed analysis of threats is performed which forms the basis to define a Grid risk management process applicable in various RMSs. Modules involved in the Grid fabric are the negotiation manager as well as the planning-based scheduler OpenCCS which includes the fault-tolerance manager. As shown in [6], applying risk-aware scheduling is more profitable than using fault-tolerance without risk considerations. In addition to supporting the resource and failure management, providers can decide to accept or reject an SLA request under the consideration of an objective measurement, the risk.

Introducing risk management into the processes of the Grid is highly beneficial for providers since they can improve their resource and failure management. Even though Grids are unreliable, an integrated risk management supports providers to fulfill the most profitable SLAs and to increase profit. Since the risk management process has to run completely automated, we adapted the FERMA risk management standard to a Grid specific solution.

Acknowledgements. This work was partially supported by EU project AssessGrid IST-031772.

References

1. A. Sahai, S. Graupner, V. Machiraju, and A. van Moorsel, Specifying and Monitoring Guarantees in Commercial Grids through SLA, Tech. Rep., 2003
2. A. Iosup, M. Jan, O. O. Sunmez, and D.H.J. Epema, On the Dynamic Resource Availability in Grids, Proceedings of the 8thIEEE/ACM International Conference on Grid Computing (GRID 2007), September 19-21, 2007, Austin, Texas, USA, pp. 26-33.
3. M. Hovestadt, Fault tolerance mechanisms for SLA-aware resource management, Proceedings Workshop on Reliability and Autonomic Management of Parallel and Distributes Systems (RAMPDS-05), 2005.
4. FERMA, A risk management standard, Federation of European Risk Management Association (FERMA), 2003.
5. AS/NZS, AS/NZS 4360: 1999 – Australian Standard Risk Management, Standards Australia, Standards New Zealand (AS/NZS), 1999.
6. D. Battré, M. Hovestadt, A. Keller, O. Kao, and K. Voss: Enhancing SLA Provisioning by Utilizing Profit-Oriented Fault Tolerance, Proceedings of PDCS'2008, Orlando, Florida, November 2008, IEEE (to appear).

4. Towards a Comprehensive Accounting Solution in the Multi-Middleware Environment of the D-Grid Initiative

Jan Wiebelitz, Wolfgang Müller, Michael Brenner and Gabriele von Voigt

Regional Computing Centre for Lower Saxony, Leibniz University Hannover, Germany

The German e-science community in 2004 jointly started to build a national Grid infrastructure funded by the German Ministry for Education and Research (BMBF). Down to the present day D-Grid is the only national Grid initiative, which claims to support the three middleware packages Globus Toolkit 4, UNICORE and LCG/gLite in a heterogeneous Grid environment.

To reach the objective of the D-Grid initiative and to guarantee the continuity, maintenance and future growth of this infrastructure it is necessary to become independent of limited public funding and achieve the therefore required revenues by chargeable customer services, user support and developments. To establish a sustainable and self-financing long-term operating e-science infrastructure all over Germany the development of business models is of great importance for D-Grid.

The realization of business models is based on the development of a comprehensive accounting service that enables a seamless recording of the usage of all resources as a precondition for necessary pricing and billing solutions demanded by Grid resource providers. Resources that have to be accounted are not only compute or storage resources but also for example services, data and software. The existing accounting units that were developed for the three middleware packages Globus Toolkit, LCG/gLite and UNICORE namely the Distributed Grid Accounting System (DGAS) [1] and the SweGrid Accounting System (SGAS) [2] are introduced and evaluated according to their capabilities to account the resource usage of different middleware packages.

The choice for the Distributed Grid Accounting System (DGAS) as accounting system in D-Grid is reasoned. The capabilities of DGAS to gather accounting information about the usage of compute resources in a heterogeneous Grid environment independent from the LCG/gLite middleware is presented and the necessary modifications to support the three mentioned middleware packages are described.

HLRmon [3] is presented as a web-based interface to visualize monitored accounting information provided by DGAS. The easy and intelligible way to access different suitable reports on resource usage provided by HLRmon are elucidated. Furthermore the role-based authorization to grant access to the different ranges and kinds of accounting information is depicted.

A comprehensive accounting [4] which comprises accounting information from all kinds of resource usage is a necessary precondition for economic considerations in a Grid. As todays Grid accounting systems only allows the accounting of compute resources necessary steps to build a comprehensive accounting system are sketched.

Acknowledgements. The German Federal Ministry of Education and Research (BMBF) within the D-Grid Integration Project (01IG07014F) supports this work.

References

1. Piro, M., A.Guaresi, A. Werbrouck: An Economy-based Accounting Infrastructure for the DataGrid. In Proc. 4th Int. Workshop on Grid Computing (GRID2003), Phoenix, AZ., November 2003, URL: http://www.to.infn.it/grid/accounting
2. Elmroth, E., P. Gardfjäll, O. Mulmo, and T. Sandholm: An OGSA-Based Bank Service for Grid Accounting Systems, June 2004. URL: http://delivery.acm.org/10.1145/1040000/1035207/p279-sandholm.pdf
3. Dal Pra, S., E. Fattibene, L. Gaido, G. Misurelli, F. Pescarmona: HLRMON: A Role-based Grid Accounting Report WebTool. in Proceedings of Int, Conference in High-Energy and Nuclear Physik 2007, CHEP2007 URL: http://www.iop.org/EJ/abstract/1742-6596/119/5/052012
4. Pettipher, M.A., A. Khan, T.W. Robinson, X.Chan: Review of Accounting and Usage Monitoring Final Report, July 2007, URL: http://www.jisc.ac.uk/publications/publications/accountingusagereport.aspx

5. SZTAKI Desktop Grid: Security Enhancements for BOINC

Attila Csaba Marosi, Gábor Gombás, Péter Kacsuk

MTA SZTAKI Laboratory of Parallel and Distributed Systems, Budapest, Hungary

The original vision of the Grid was that anyone could donate and claim resources according to their needs. This twofold aim lead to two different trends in Grid computing. Followers of the first are creating *service oriented Grids*, which is accessible by many users, but requires the deployment and management of a complex middleware, meaning that individuals cannot easily offer resources. The second aim lead to Desktop Grid Computing (commonly reffered as Internet-based Distributed Computing, Public-Resource Computing or Volunteer Computing). Unlike traditional Grids, which are based on complex architectures, volunteer computing has demonstrated a great ability to integrate dispersed, heterogeneous computing resources with ease, scavenging cycles from idle desktop computers. In DG systems anyone can bring resources into the Grid system, offering them for the common goal of that Grid. Installation and maintenance of the Grid resource middleware is extremely simple, requiring no special expertise. Therefore, large number of donors can contribute to the pool of shared resources. On the other hand, only a very limited user community (or target applications) can use those resources for computation.

The common architecture of Desktop Grids typically consists of one or more central servers and a large number of clients. The central server provides the applications and their input data. Clients join the Desktop Grid voluntarily, offering to download and run tasks of an application with a set of input data. When the task (*"work unit"*) has finished, the client uploads the results to the server where the application assembles the final output from the results returned by clients. A major advantage of Desktop Grids over traditional Grid systems is that they are able to utilize non-dedicated machines. Besides, the requirements for providing resources to a Desktop Grid are very low compared to traditional Grid systems using a complex middleware. Desktop Grids may be deployed (and gathering) institutional computing resources (Local Desktop Grid - LDG) or volunteer public resources (Public Desktop Grid - PDG). The most popular Desktop Grid platform is BOINC[1] which aims to provide an open infrastructure for Public Desktop Grids. On the other hand SZTAKI Desktop Grid, which is based on BOINC aims to provide enhancements to fulfill the needs of Local Desktop Grids. In this paper we will discuss these extensions.

The most important factor in desktop grid computing is the trust between the clients and the project providing the application. Allowing foreign code to run on a computer always has a risk of either accidental or intended misbehavior. BOINC mitigates this risk by only allowing to run code that has been digitally signed by the project the client is connected to. Clients trust the operators of any DG BOINC project not to offer malicious code, and digitally signing the application by the project provides technical means to ensure this trust relation. The problem with this approach is that the Client has either to trust all applications originating from the project or none, there is no information about the origin of the application (most of the time operators of a DG and the application developers are separate roles), also there is no way to ensure that the key infact is belonging to the project and to no mallicious third party sent the binary (signed by his valid key). While these issues might be ignored by volunteers, but in an insitutional environment are minimum requirements. SZTAKI LDG provides a new X.509 certificate based authentication system, separating DG Projects, Application Certifiers and Clients. The Application Certifiers role is to sign application to be deployed at any Project, while the Project can add its additional signatures. Clients choose what Application Developer and Project signatures to accept, and also allows to build a “chain of trust” to ensure no mallicious third party binary (but with valid signature) has been supplied. Projects can choose which Clients they allow to request work.

Another important aspect of security is to isolate the running application from the rest of the system. Desktop Grid systems (especially BOINC) simply use *fork()* to execute any application. SZTAKI DG provides two extensions for BOINC to support sandboxing: first is the capability to run Java applications. Java uses its own Virtual Machine which can be restricted according to the set security requirements. For non-Java BOINC and legacy applications SZTAKI DG provides a Virtual Machine based sandbox using QEMU[4]. While there are some related works [5,6,7] on providing sandbox based execution environment for Desktop Grids, they are all limited either by beeing tied to a specific operating system or require special preparations by either the project administrators or at the deployed client. SZTAKI DG provides an immutable, transparent for the users and admintrators VM based sandbox for Windows/ Linux and Mac OS X. The sandbox is immutable because no mallicous application may render it unusable, transparent since it needs no modifications on the DG Project and does not constrain limitations on the Cliens and runs on all platforms supported by SZTAKI DG/ BOINC.

The sandbox is not tied to any task, meaning that no VM images need to be transferred allowing to keep network utilization low. Also the sandbox is not specially tied to SZTAKI DG/ BOINC, meaning it can be adapted to other DG systems too.

References

1. D. P. Anderson. Boinc: A system for public-resource computing and storage. In Rajkumar Buyya, editor, Fifth IEEE/ACM International Workshop on Grid Computing, pages 4–10, 2004.

2. Zoltán Balaton, Gábor Gombás, Péter Kacsuk, Ádám Kornafeld, Attila Csaba Marosi, Gábor Vida, Norbert Podhorszki, and Tamás Kiss. Sztaki desktop grid: a modular and scalable way of building large computing grids. In Workshop on Large-Scale and Volatile Desktop Grids, PCGrid 2007, 2007.
3. Attila Csaba Marosi, Gábor Gombás, Zoltán Balaton, Péter Kacsuk, and Tamás Kiss. Sztaki desktop grid: Building a scalable, secure platform for desktop grid computing. In Making Grids Work, pages 363–374. Springer Publishing Company, Incorporated, July 2008
4. Fabrice Bellard. Qemu, a fast and portable dynamic translator. In ATEC'05: Proceedings of the USENIX Annual Technical Conference 2005 on USENIX Annual Technical Conference, pages 41–41, Berkeley, CA, USA, 2005. USENIX Association.
5. D. Lombraña González, F. Fernández de Vega, L. Trujillo, G. Olague, M. Cárdenas, L. Araujo, P. Castillo, K. Sharman, and A. Silva. Interpreted applications within boinc infrastructure, 2008.
6. D. Lombraña González, F. Fernández de Vega, G. Galeano Gil, and B. Segal. Centralized boinc resources manager for institutional networks. In IPDPS 2008. IEEE International Symposium on Parallel and Distributed Processing, 2008., pages 1–8, April 2008.
7. Franck Cappello, Samir Djilali, Gilles Fedak, Thomas Herault, Frederic Magniette, Vincent Neri, and Oleg Lodygensky. Computing on large-scale distributed systems: Xtremweb architecture, programming models, security, tests and convergence with grid. Future Generation Computer Systems, 21(3):417–437, 2005.

6. Description in ClassAd Language of Complex Policies for Resource Allocation in Grid Computing

Gabriele Pierantoni, Brian Coghlan, Eamonn Kenny
Trinity College Dublin, Dublin, Ireland

Social Grid Agents are a proposed solution for the problem of resource allocation in Grid Computing that is inspired by social behaviours. It is based on a two layers: the social layer where Social Grid Agents engage in social and economic relations and a production layer where Production Grid Agents compose Grid services.

As they encompass different middlewares and engage in relationships of different nature, Social Grid Agents need a native language to describe their status, to communicate with each other and, finally to communicate with the existing middlewares they are connected to.

This paper illustrates how ClassAd, a functional language, is used as a native language for the Social Grid Agents to describe status and behaviour and how messages based on this native language describe exchanges, actions and policies that encompass different aspects of Grid Computing such as security and execution parameters.