Monday, Oct 12, 2009

Keynote Presentations

Infrastructures Use, Requirements and Prospects in ICT for Health Domain

Karin Johansson
European Commission, Brussels, Belgium

Progress in Integrating Networks with Service Oriented Architectures / Grids: ESnet's Guaranteed Bandwidth Service

William E. Johnston
Senior Scientist, Energy Sciences Network, Lawrence Berkeley National Laboratory, USA

The past several years have seen Grid software mature to the point where it is now at the heart of some of the largest science data analysis systems - notably the CMS and Atlas experiments at the LHC. Systems like these, with their integrated, distributed data management and work flow management routinely treat computing and storage resources as 'services'. That is, resources that that can be discovered, queried as to present and future state, and that can be scheduled with guaranteed capacity. In Grid based systems the network provides the communication among these service-based resources, yet historically the network is a 'best effort' resource offering no guarantees and little state transparency. Recent work in the R&E network community, that is associated with the science community, has made progress toward developing network capabilities that provide service-like characteristics: Guaranteed capacity can be scheduled in advance and transparency for the state of the network from end-to-end. These services have grown out of initial work in the Global Grid Forum's Grid Performance Working Group. The services are defined by standard interfaces and data formats, but may have very different implementations in different networks. An ad hoc international working group has been implementing, testing, and refining these services in order to ensure interoperability among the many network domains involved in science collaborations. This talk will describe these services, their evolution, and their current state.

Distributed Computing at the Petascale

Peter Coveney
University College London, Department of Chemistry, UK

It now seems likely that PRACE will provide a unified Europe wide distributed environment for production level high end computing. To be maximally effective, this infrastructure will need to be well balanced, supporting not only such massive computing resources but facilitating data movement across fast networks to and from end-users in a secure manner, in conjunction with smart middleware and policies which support various requirements including advance reservations and urgent computing.

These many requirements have contributed to the poor levels of usability of grids, an issue that has concerned us for a number of years. Our own approach uses application virtualization to address grid-

based HPC usability. We describe a middleware tool that handles the virtualization of grid applications, called the Application Hosting Environment (AHE). We describe various features of AHE, which provides access to computational grid resources in standard and non-standard ways; it also federates collections of such grids thanks to its interoperability features. New and forthcoming features include support for urgent computing, audited credential delegation for single sign-on based only on local user credentials, and brokering.

We then discuss a few examples of our computational science research that uses 'petascale' resources to achieve scientific results at unprecedented scales and resolution. The applications span a range of domains from investigation of fundamental problems in turbulence through nanomaterials science research to biomedical applications at the forefront of HIV/AIDS and cancer research, as well as cerebrovascular haemodynamics. Much of this work has been performed on TeraGrid's petascale resource, "Ranger", located at Texas Advanced Computing Centre, and on "Intrepid", the IBM Blue Gene/P system at the Argonne Leadership Computing Facility.

Cloudifying Grids and Gridifying Clouds: A Plea for Co-Operating Clouds and Grids

Michael Schiffers
Ludwig-Maximilians-Universität, Germany

HPC, Grids and Clouds: Synergies and Challenges

Frank Baetke
Global HPC Technology Program Manager Hewlett-Packard / Scalable Computing Infrastructure Organization

The TOP500 list and other metrics clearly show that Blade-based architectures are now fully established as the new standard architecture for HPC-Systems. Those systems provide vast opportunities in terms of system efficiency and density but will also lead to new challenges. New node types will provide additional benefits in terms of scalability and usability.

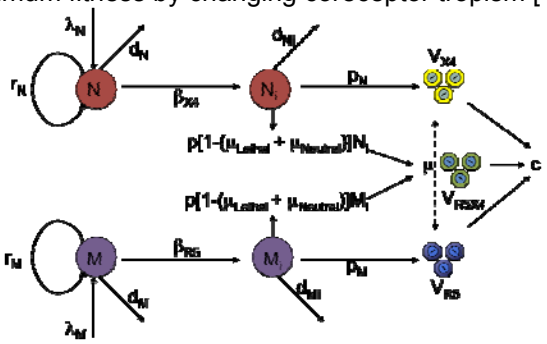
At the software and middleware level Grids have been established successfully in various dimensions and from there a new trend "Cloud Computing" is emerging. Clouds are enabled through automation and virtualization technologies that make it possible to abstract away much of the complexity of accessing vast amounts of computing and storage. Cloud providers can and will pass on the economies of scale of huge data centres through a pay per use model. Current trends, challenges and examples will be discussed and compared to classical Grid environments.

Poster Presentations

1. Transmission & Mutation Model of HIV-1 Coreceptor Tropism: Meta-Population Perspective

Gökhan Ertaylan (1), Peter Sloot (1)
(1) Computational Science, The University of Amsterdam, The Netherlands

Human immunodeficiency virus (HIV) requires a secondary receptor for successful entry to the cell. The most important secondary receptors in-vivo are chemokine receptors CCR5 (R5) and CXCR4 (X4).
Early infection with HIV is characterized by predominance of R5-tropic virus. However, over the course of infection X4-tropic virus appears in the later stage of infection in approximately 50% of infected individuals and usually precedes an accelerated CD4+ T cell depletion with rapid disease progression.
We propose an individual based model of HIV target cell tropism to examine the population dynamics between virus strains within the infected host. We have designed our model with emphasis on spatial interactions between target cells and R5, R5X4 and X4-tropic HIV-1 strains. We have imposed a coreceptor change scheme, based on stepwise multiple point mutations allowing for biologically natural trade-off between viral mutation rate and the overall reproduction capacity of the viral quasispecies. Given the transmission with a R5-tropic virus, we evaluate the mutation rate carrying the virus quasispecies to an optimum fitness by changing coreceptor tropism [1].



Transmission & Mutation model schema. N and M represent Naïve and Memory T cells. VX4, VR5X4 and VR5 represent X4, R5X4 and R5-tropic HIV respectively. “r” is the reproduction rate, “λ” is born rate, “d” is lifespan, “β” is infection rate and “p” is burst size. Generic parameters are mutation rate μ and HIV lifespan c.

We have constructed a two dimensional Cellular Automata (CA) where two types of agents are defined: T cells and HIV. One time tick in the simulation corresponds to six hours. T cells are defined as static entities of size one, residing on continuous Cartesian coordinates where the shape of the grid is a closed torus. The size of the grid is ~2.104 and approximately 104 T cells (half of them are TN and the other half TM) reside in the equilibrium without infection in the standard model. They proliferate with rate rN, M and born with rate λN, M. The coreceptor designation of each T cell is defined as an internal parameter and the coreceptors on the T cells are not modeled separately. HIV virions are defined as mobile point particles that are released from infected and dying T cells (from TNi or from the TMi). Their motion is Brownian with step size one. They can be mobile until they are bind to a healthy T cell or until the end of their lifespan. They have a tropism parameter, which is set when they are budding from the infected T cells.
We conclude that there is an optimum range of mutation rate for HIV coreceptor tropism and this coincides with the experimental range. The time of the co-receptor switch was dependent on the fluctuations on the HIV population size. The bottlenecks in the viral population size increased the chance of occurrence of the co-receptor switch therefore reducing the time required for the co-receptor switch. R5-tropic variants benefit from the emergence of X4 tropic variants and outcompete dual-tropic variants together leading to sympatric speciation of HIV-1 *in-vivo*. This implies a strong link to in-vivo compartmentalization of the virus while adapting to its host.

Acknowledgements. This work was supported by EU project ViroLab IST-027446

References

1. P.M.A. Sloot; P.V. Coveney; G. Ertaylan; V. Müller; C.A.B. Boucher and M.T. Bubak: HIV Decision Support: From Molecule to Man, Phil . Trans. R. Soc. A, vol. 367, nr 1898 pp. 2691-2703. 2009.

2. Development of the Model for Molecular Dynamics Simulation of Membrane Proteins to be Incorporated to ViroLab Project

Veronica Zobnina (1), Irena Roterman (1)
(1) Department of Bioinformatics and telemedicine, Collegium Medicum – Jagiellonian University Krakow, Poland

The simulation of molecular dynamics for membrane proteins is critical for analysis of the proteins, their interaction with membrane and for computer aided drug design modifying their activity. The molecular dynamics simulation (MDS) is the most time-consuming computation in bioinformatics due to high standards for period of time covered by simulation (no less than 2 ns). The large number of atoms present in protein molecule, membrane system and water environment makes the computation of the large scale character. The design of drugs modifying the membrane protein is highly important due to membrane receptors role in signal transduction participating in many regulation processes. The simulations of the processes of ligand-protein interaction particularly for transmembrane proteins are highly desired for pharmacology and drug design.

The standard programs available for MDS localize the protein molecule in water environment. The Gromax [1-4] program is able to localize the protein molecule in membrane.
The number of atoms in such system is of the range of tens of thousands making the calculations highly time consuming.

The model simulating the environment in form of external force field of hydrophobic character expressed by the three-dimensional Gauss function mimicking the distribution of hydrophobicity density according to the model of hydrophobic core [5]. The traditional “oil-drop” model got extended to the form of “fuzzy oil drop” model with high concentration of hydrophobicity in the center of the protein body with hydrophilic residues exposed on the surface. Modification of the magnitude of the force in the Newton equation acting on the atom by the coefficient proportional to the value of the hydrophobicity density for the point in space of the localization of particular atom introduces the controlled constraints assumed to mimic the influence of the environment.

The simulation of molecular dynamics simulation of the transmembrane protein – rodopsine (opsine) was performed in three forms: 1 – in vaculo (only as reference test), 2 – according to traditional model including protein, membrane and water molecule and 3 - in the presence of external force field of hydrophobicity character. The comparison of the final results revealed high accordance between model 2 and 3 with the time consumption comparable to model 1. It makes the model 3 very profitable due to significant lowering of time consumption with simultaneous high quality of final results (comparable with golden standard which is the all-atom model – model 3) [6].

The application of the model 3 is also of the fundamental character. It may be concluded that the controlled distribution of constraints is accordant to the force distribution in the real environment in which the protein is acting.

The model is planned to be implemented in ViroLab service [7] as the tool for MDS applied for any forms of proteins particularly the membrane proteins.

Acknowledgements. This work was supported by Collegium Medicum Jagiellonian University grant K/ZDS/000483.

References

1. Nagle, J. F., and Tristram-Nagle S.. Structure of lipid bilayers. *Biochim Biophys Acta*1469:159-95, 2000.
2. White S.H. and Wimley W.C.. Membrane protein folding and stability: Physical Principles. *Annu. Rev. Biophys. Biomol. Struct.* 28:319–65, 1999.

3. Walker J.E. and M. Saraste M. Membrane proteins. Membrane protein structure. *Curr Opin in Struct Biol* 6:457-459, 1996
4. Nielsen N.C., Malmendal A. and Vosegaard T. Techniques and applications of NMR to membrane proteins. *Mol Membr Biol 21:129-141, 2004*
5. Konieczny L, Bryliński M, Roterman I. Gauss-Function-Based Model of Hydrophobicity Density in Proteins *In Silico Biology* 6, 15-22, (2006).
6. Zobnina V, Roterman I. Application of the fuzzy-oil-drop model to membrane protein simulation. *Proteins: Structure, Function and Bioinformatics (*2009) – in press
7. Peter M.A. Sloot, Ilkay Altintas, Marian Bubak, Charles A. Boucher: From Molecule to Man: Decision Support in Individualized E-Health; IEEE Computer Society, vol 39, no.11, pp. 40-46, Nov., 2006.

3. Bioinformatics Applications in the ViroLab Virtual Laboratory

Tomasz Jadczyk (2), Maciej Malawski (1), Marian Bubak (2,3)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland
(3) University of Amsterdam, The Netherlands

Bioinformatics is the field of science in which biology, computer science, and information technology merge to form a single discipline. The ViroLab virtual laboratory [5, 6] is an environment where the applications are available as reusable components (gems) and are exploited within (in silico) experiments on different infrastructures [7]. The subject of this work is to prepare a set of applications suited to solve common problems in the field of bioinformatics and to integrate them into the virtual laboratory.

As a result of the analysis of available bioinformatics applications the main bioinformatics research areas to be supported were selected: protein sequence and structure comparison [1], ligand binding site prediction and microarray analysis (see Fig. 1). The applications responsible for data gathering, preliminary data analysis and visualization of the results are also included in the prepared set. Two classifications of applications have been developed: by scope of usage and by technology. On the basis of the first one, the complete set of applications that were integrated into the virtual laboratory has been chosen.

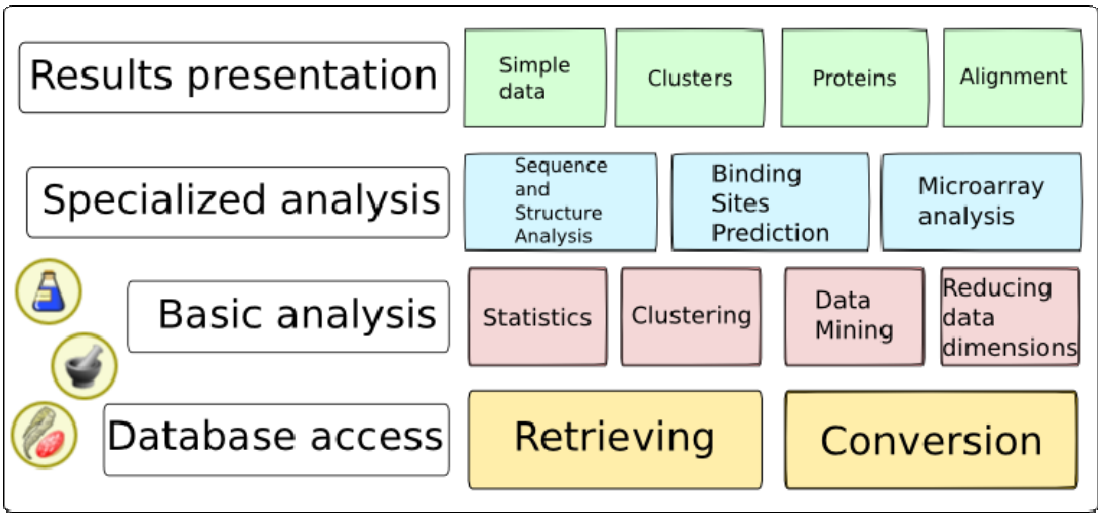


Figure 1 Layers of the gem scope of usage. Four main layers are recognized: Bioinformatics Database access, for retrieving data and converting file formats to the one accepted by appropriate software; Basic analysis, used in a variety of experiments, where statistical analysis, data clustering, data mining and data dimensions reducing gems are available; Specialized analysis used to solve specific type of problem and Results presentation for visualizing data

The set of bioinformatics gems consists of 42 gems, from which 5 gems are responsible for data preparing (4 gems for database access and one that converts data into appropriate format), 11 gems

create basic analysis layer (5 data mining gems, 3 gems for data clustering, 2 reduce data dimensionality and one is suited to statistical computations). 9 gems designed to protein comparison with 10 gems for comparing services for predicting ligand binding sites and 2 gems for microarray analysis are placed in specialized analysis layer. Additionally, 5 gems are provided for visualizing data and results.

In every experiment, the gems responsible for performing main computation parts may be substituted by another application (algorithm) that solves the same problem. Applications are integrated using existing technologies as well as two new mechanisms: task queuing system and binary program wrapper.

All experiments are divided in groups with well defined responsibilities. The most advanced experiments, developed and used in the virtual laboratory are:

- Protein sequence and structure comparison with following setctions: protein sequence and structure information gathering (DbFetch [2], EarlyFolding); sequences and structures alignment (ClustalW [2], Mammoth), *W* score computation (ClustalWUtils) and results visualization (Gnuplot, Jmol).
- Comparison of services for predicting ligand binding site: predicting binding sites in various services (CastP [4], ConSurf, PocketFinder, QSiteFinder, WebFeature, Pass, Fod); results conversion to common format (ResultsConverter) and visualization (Jmol).
- Microarray data analysis: microarray data preparing (GeoDataProvider, FormatConverter), data analyzing (WekaClustering, Cluto) and results visualization (JtreeView)
- Data mining in Weka [3]: data preparing (WekaURLGem, WekaFilter), data analyzing WekaClassifier

The experience gathered during the development of bioinformatics gems and experiments allows concluding that the ViroLab virtual laboratory is a powerful and flexible environment for building and running complex scientific applications. Moreover, we demonstrated that bioinformatics problems well fit the model of applications for which this virtual laboratory was designed.

Acknowledgments: The authors express their gratitute to prof. Irena Roterman-Konieczna, dr. Monika Piwowar and Katarzyna Prymula for the fruitful collaboration. This work has been partly supported by the European Commission ViroLab Project Grant 027446 [8], Polish SPUB-M grant, the AGH grant 11.11.120.777, and ACC CYFRONET-AGH grant 500-08.

References

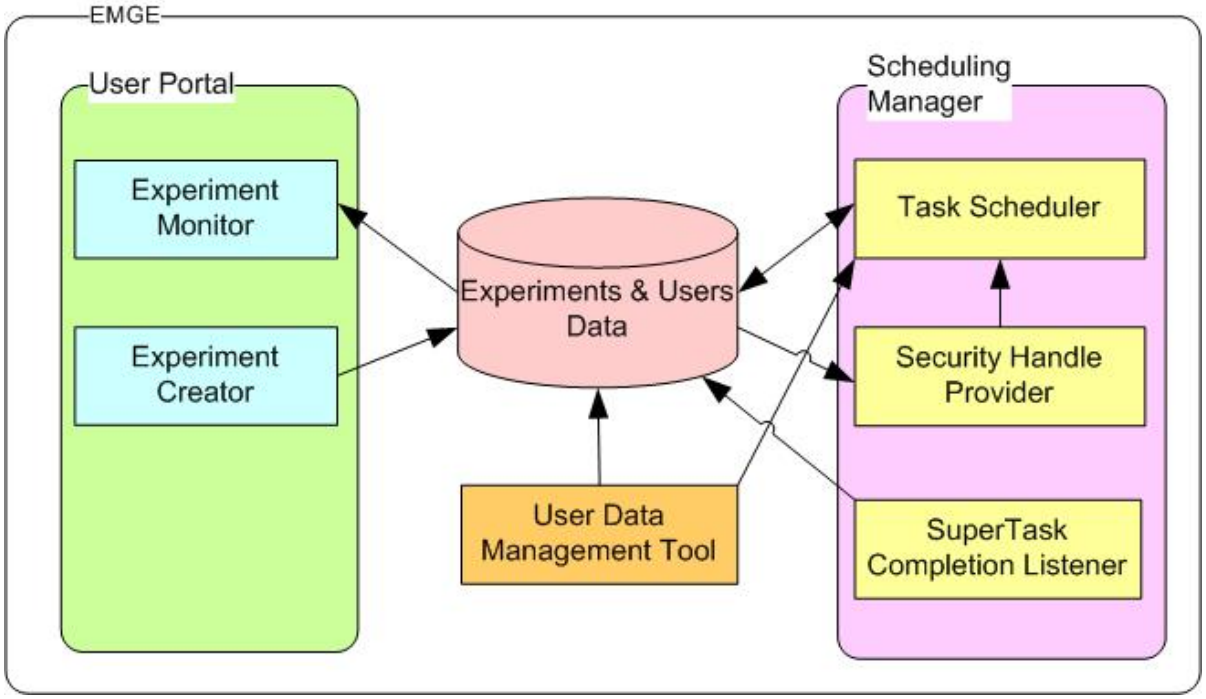
1. M. Brylinski, L. Konieczny, P. Czerwonko, W. Jurkowski, and I. Roterman. *Early-stage folding in proteins (In Silico) sequence-to-structure relation.* Journal of Biomedicine and Biotechnology, 2 (2):65–79, 2005.
2. A. Labarga, F. Valentin, M. Anderson, and R. Lopez. *Web Services at the European Bioinformatics Institute*. Nucl. Acids Res., 35(Web Server issue), June 2007.
3. H. Witten and E. Frank. *Data mining : practical machine learning tools and techniques*. Morgan Kaufmann series in data management systems. Morgan Kaufman, June 2005
4. Dundas, Z. Ouyang, J. Tseng, A. Binkowski, Y. Turpaz, and J. Liang. CASTp*: computed atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues.* Nucl. Acids Res., 34:W116–118, July 2006
5. M. Bubak, M. Malawski, T. Gubala, M. Kasztelnik, P. Nowakowski, D. Harezlak, T. Bartynski, J. Kocot, E. Ciepiela, W. Funika, D. Krol, B. Balis, M. Assel, and A. Tirado-Ramos. *Virtual Laboratory for Development and Execution of Biomedical Collaborative Applications*. In: M.Cannataro (Ed.) Handbook of Research on Computational Grid Technologies for Life Science, Biomedicine and Healthcare. Information Science Reference, 2009, IGI Global
6. ViroLab Team at CYFRONET: *ViroLab Virtual Laboratory Homepage*: http://virolab.cyfronet.pl
7. M. Malawski, T. Bartynski, and M. Bubak, *Invocation of operations from script-based grid applications*, Future Generation Computer Systems, vol. In Press, Accepted Manuscript, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.future.2009.05.012
8. ViroLab Project Homepage: http://virolab.org

4. Environment for Management of Grid Experiments

Pawel Charkowski, Maciej Malawski (1), Marian Bubak (1,2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

E-science is a vital part of all modern scientific researches. Since in-silico experiments often have very complex structure, computing them as a sequence of experiments would drastically prolong their execution time. That is why experiments are divided into separate, independent tasks that can be executed in parallel, minimizing their sequential part. Manual management of such experiments is a very complex process and time consuming for users. Therefore fully functional, user friendly experiment management environment is a relevant part of each modern virtual laboratory. There are many existing experiment management environments available for the Grid and cluster computing, such as DIANE [1], Nimrod/G [2] and ZENTURIO [3] environments. They are intended to assist users with automated experiment jobs management and scheduling based on information provided by the user. However, as they require their users to posses basic programmatic knowledge and learn new languages, they are not targeted for non-informaticians, who are majority of ViroLab [4] virtual laboratory users. In addition some of them support only parameter study experiments.

An experiment in ViroLab virtual laboratory is composed of several tasks represented by scripts written in Ruby programming language. Each such task is divided into jobs which execute the same script, but operating on different input data. To relieve ViroLab users from manual scheduling, management and monitoring of non-trivial experiments we introduce an Environment for Management of Grid Experiments (EMGE) – an environment that will automatically manage experiment execution according to experiment plan provided by its submitter. EMGE uses GridSpace Engine [5] (GSEngine), combination of an interpreter of the JRuby language, in which experiments are written, with a runtime library that provides all the specific functionalities of the ViroLab virtual laboratory, for submission of task execution to Grid. EMGE has been implemented in Java programming. A central point of EMGE



is a database which stores all information regarding experiments managed by EMGE and users authorized to use EMGE. Database access is managed by Database Access Component, which uses Hibernate for ORM mapping. Above the Database Access Component reside two modules independent from each other: User Portal and Scheduling Manager. User Portal is a web application (implemented using Google Web Toolkit), that provides user interface for means of communication between EMGE system and its users. It allows submission of new experiments to the system and presents the current status of scheduled experiments basing on information stored in database. The

Scheduling Manager is the entity responsible for managing tasks execution basing on experiment structure information stored in database, and keeps those information up to date.

EMGE has been tested on a protein folding experiment, objective of which was to perform creation of the early stage intermediate structure in protein folding for over a thousand different protein codes. Experiment has been divided into thousand tasks, each of them performing operations on a different protein code. EMGE successfully managed execution of scheduled test experiment's subtasks.

By providing a user-friendly web interface, EMGE allows non-informatician users to harness grid infrastructure for execution of complex workflow experiments, allowing them also to monitor their experiment current state. EMGE does not require its users to install any additional software, as all that they need to use it is a standard web browser.

This work has been partly supported by the European Commission ViroLab Project [4] Grant 027446, Polish SPUB-M grant, the AGH grant 11.11.120.777, and ACC CYFRONET-AGH grant 500-08.

References

1. J.T. Moscicki: DIANE - Distributed Analysis Environment for GRID-enabled Simulation and Analysis of Physics Data, NSS IEEE 2004, Portland, Oregon, USA, October 2003
2. Abramson, D., Giddy, J. and Kotler, L. "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?", International Parallel and Distributed Processing Symposium (IPDPS), pp 520-528, Cancun, Mexico, May 2000.
3. Radu Prodan, Thomas Fahringer. ZENTURIO: An Experiment Management System for Cluster and Grid Computing. In Proceedings of the 4th International Conference on Cluster Computing (CLUSTER 2002), Chicago, USA, September 2002. IEEE Computer Society Press.
4. Marian Bubak et al., Virtual Laboratory for Collaborative Applications, In: M. Cannataro (Ed.) Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine and Healthcare, Information Science Reference, 2009, IGI Global
5. M. Malawski, T. Bartynski, and M. Bubak, "Invocation of operations from script-based grid applications," Future Generation Computer Systems, vol. In Press, Accepted Manuscript, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.future.2009.05.012

5. Integrating EGEE Storage Services with the Virtual Laboratory

Marek Pomocka (1), Piotr Nowakowski (2), Marian Bubak (3,4)
(1) Faculty of Physics and Applied Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland
(3) Institute of Computer Science AGH, Krakow, Poland
(4) Informatics Institute, University of Amsterdam, The Netherlands

The advent of Grid technologies has enabled research at a pace not achievable using earlier methods, which facilitates easier access to high-end computing and data resources. However, employing Grids in scientific work is still a domain of highly skilled researchers, able to tackle the complexity of the Grid environment. Although there have been successful endeavors that strive to provide a mature scientific environment [1, 2, 3, 4] for scientific disciplines not normally related to computer science, fundamental obstacles still prevent scientific communities from adopting Grids. These include the complexity of Grid security solutions, such as Grid Security Infrastructure (GSI) and intricate access to core Grid services, e.g. data catalogues and storage resources.

Our work aims to minimize the learning curve for access to Grid data services, specifically to LCG File Catalogue (LFC) storage elements and GSI, concealing most technical details. The API we have devised creates an abstraction of working with local files with no intervening GSI, i.e. with no Grid certificate-related operations, although the user works with files stored on the Grid with all GSI mechanisms in place. As regards other projects that deal with comparable issues, the *Credential Mapping Service* [5] allows mapping one security system onto another, e.g. Kerberos authentication tokens onto GSI certificates. Similarly, in our solution, Shibboleth handles are automatically mapped to GSI certificates, relieving users from the burden of managing their own credentials. Furthermore,

Yaodong et al. [6] have developed GFISH (Grid File Sharing system), which includes a server providing a web service API for the LFC catalogue and a related Java client with Grid user credentials retrieved from a MyProxy server. They implemented the server using gSOAP, while utilizing Axis on the client side, thus introducing significant transmission overhead. Our approach is also service-oriented, however we relied on RMI-based protocols and libraries, namely the Cajo library for overall communication and RMIIO for streaming. To provide secure transmission, our solution employs SSH tunneling; thus we avoid the need to generate server certificates and to manage keystores (which is an inherent feature of Transport Layer Security). Our development effort did not commence from scratch. Instead, we build on previous work, such as ChemPo [1] LFC command wrappers and the data access infrastructure prepared for the ViroLab [3, 4] project, specifically DAC2 [7] and Data Source Registry (DSR). We have extended DSR so that it is able to store Grid user credentials and information on new data source types, prepared a server that acts as a gateway between DAC2 and EGEE/WLCG, developed a client library that communicates with this server and, finally, developed a new DAC2 GScript [8] interface which makes use of the aforementioned components.

The result of our work is a new convenient API for managing and accessing files on the Grid, which automates certificate management and mimics local file access and directory operations, e.g. the user requesting a file from the Grid is handed a Ruby IO reference that points to a remote input or output stream. Last but not least, the client API is independent of the gLite software, which makes it more accessible to end users and does not impose additional dependencies on the GridSpace Engine [8] – the Virtual Laboratory [3, 4] runtime. Future work might include providing fine-grained security. In addition, further tailoring of the API to specific scientific scenarios may prove very valuable.

Acknowledgements
This work has been partly supported by the European Commission ViroLab Project [43] Grant 027446, Polish SPUB-M grant, the AGH grant 11.11.120.777, and ACC CYFRONET-AGH grant 500-08, as well as the Polish national PL-Grid project.

References

1. Mariusz Sterzel, Tomasz Szepieniec, and Daniel Harężlak. Grid Web Portal for Chemists. In EGEE User Forum, Catania, Italy, March 2009. Presentation slides.
2. Mariusz Sterzel and Tomasz Szepieniec. Enabling Commercial Chemical Software on EGEE Grid – Gaussian VO. In Marian Bubak, Michał Turała, and Kazimierz Wiatr, editors, *Proceedings of Cracow Grid Workshop - CGW'06, October 2006*, Krakow, Poland, 2007. ACC-Cyfronet AGH.
3. PMA Sloot, P.V. Coveney, G. Ertaylan, V. Müller, CA Boucher, and M. Bubak. HIV decision support: from molecule to man. Philosophical Transactions A, 367(1898):2691, 2009.
4. Marian Bubak et al., Virtual Laboratory for Collaborative Applications, In: M. Cannataro (Ed.) Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine and Healthcare, Information Science Reference, 2009, IGI Global
5. Mehran Ahsant, Esteban Talavera Gonzalez, Jim Basney, "Security Credential Mapping in Grids," ares, pp.481-486, 2009 International Conference on Availability, Reliability and Security, 2009
6. C. Yaodong, W. Lu, L. Aigui, and C. Gang. Sharing LCG files across different platforms. In Journal of Physics: Conference Series, volume 119, page 062024. Institute of Physics Publishing, 2008.
7. Matthias Assel, David van de Vijver, Pieter Libin, Kristof Theys, Daniel Harezlak, Breanndann O Nuallain, Piotr Nowakowski, Marian Bubak, Anne-Mieke Vandamme, Stijn Imbrechts, Raphael Sangeda, Tao Jiang, Dineke Frentz, and Peter Sloot: A Collaborative Environment Allowing Clinical Investigations on Integrated Biomedical Databases. In Tony Solomonides, Martin Hofmann-Apitius, Mathias Fredigmann, Sebastian Caludius Semler, Yannick Legre, and Mary Kratz: Healthgrid Research, Innovation and Business Case; Proceedings of HealthGrid 2009, Studies in Health Technology and Informatics, vol 147, IOS Press, ISSN 0926-9630, pp 51 -61
8. M. Malawski, T. Bartynski, and M. Bubak, "Invocation of operations from script-based grid applications," Future Generation Computer Systems, vol. In Press, Accepted Manuscript, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.future.2009.05.012

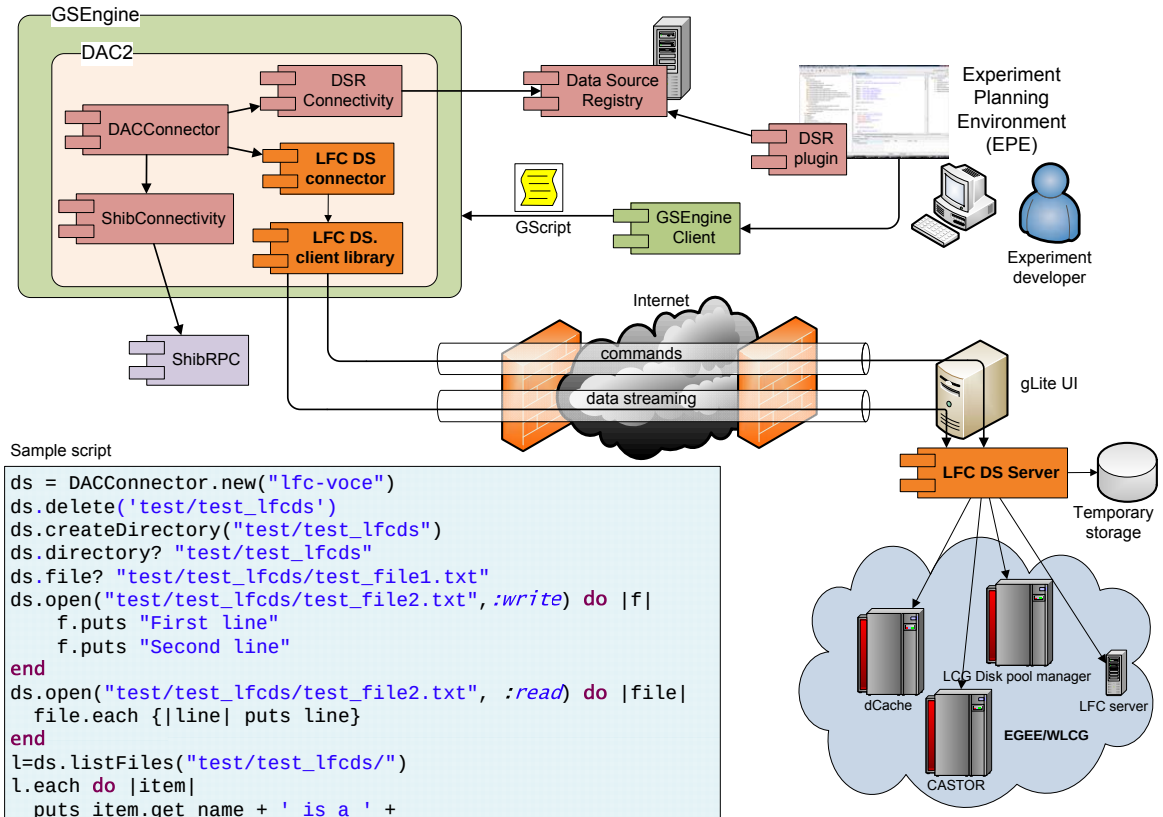


Figure 1: Conceptual view of our solution together with a sample script accessing data and operating on LFC catalogue

6. Benchmarking the 3G-bridge in the EDGeS Project Infrastructure

Naghmeh Ramezani Ivaki (1), Diogo Ferreira (1), Filipe Araujo (1)
(1) CISUC, Dept. of Informatics Engineering University of Coimbra, Portugal

Many different grids have emerged in the last decade. Two of these are EGEE, which is quite used among the European scientific community, and BOINC[1], which has a huge worldwide popularity. Since they follow different paradigms, it is difficult, if not impossible, to send jobs from one of these grids to the other. This results from their fundamentally different architecture. For instance, while EGEE is a Service Grid (SG) that receives jobs submitted by the users (push approach), BOINC sends jobs to users that request them (pull approach). To overcome this separation, the Enabling Desktop Grids for e-Science project (EDGeS) [2], aims to let jobs cross grid boundaries, thus building larger infrastructures with improved utilization of resources.

At the heart of the EDGeS project lays the 3G-Bridge, which connects DGs, like BOINC and XtremWeb [3] to EGEE and vice-versa [4]. The fundamental component of the 3G-Bridge is a core that handles jobs from one grid to the other and brings results back. This core can be extended to connect many different types of grids, whenever the appropriate grid adapters exist. The main role of the 3G-Bridge is to connect SGs and DGs, thus having two different directions of operation: one from DGs to SGs, the other from SGs to DGs.

Unfortunately, these flows of jobs and results impose undesired overheads. These overheads can be of two types: latency, i.e., delays imposed on job completion, which are not the responsibility of the grids themselves, but also throughput overheads, as the bridge can only handle a certain number of jobs per unit of time. Here, we present our effort to benchmark EDGeS. By observing performance of the entire EDGeS infrastructure, with particular attention to the 3G-Bridge, we were able to detect and

identify system's bottlenecks. As a result, system developers can do a better investment with their limited time to modify the bridge.
After careful inspection, made independently from the developers, we found several critical points in the source code of the bridge. We call "critical" to these points, because they have very sensible effects on the latency and throughput of the bridge. These consist mainly on timeouts that control internal operation of the bridge. In other words, the bridge runs using an internal poll-based approach.
The evaluation of the EDGeS infrastructure that followed consisted of three main parts: i) evaluation of the bridge disconnected from any infrastructure; ii) evaluation of the DG→SG direction; and iii) evaluation of the SG→DG direction. We defined several benchmark tests to verify all components in the 3G-Bridge and the relationship among them. To benchmark the core operation of the bridge, without any grid attached, we added a component called jobwrapper that submits jobs directly to the bridge, and a special plugin, called loopback, which fakes the role of a grid. In our tests, we used a batch of jobs with fixed size and submitted successive batches one after the other.
The main conclusion taken from the work is that an appropriate tuning of the bridge timeouts is quite dependent on the scenarios where the bridge runs. We observed that their default values make the bridge respond somewhat poorly to a couple of tests. Considering the single case of the 3G-Bridge throughput, we could double it, just by reducing internal timeouts. However, in day-to-day operation, the bridge actually needs more conservative timeouts, or otherwise it would consume too many computing resources, and could even drag down overall performance by running into active waits. This serves to say that we cannot simply wipeout the timeouts for the sake of performance. So, if any recommendation results from this work, such recommendation is that future versions of the bridge should consider the internal use of event-based communication instead of polling-based communication, whenever possible. However, we think that we reached a second, yet more important conclusion: the EDGeS project produced a bridge that is efficient enough to run and that is already connecting together service and desktop grids.

Acknowledgements. The EDGeS (Enabling Desktop Grids for e-Science) project receives Community funding from the European Commission within Research Infrastructures initiative of FP7 (grant agreement Number 211727).

References

1. http://boinc.berkeley.edu/
2. http://edges-grid.eu/
3. http://www.xtremweb.net/
4. JRA1.1: Prototypes of Bridge from Desktop Grids to Service Grids. Deliverable DJRA1.1 of the EDGeS Project, June 2008.

7. BalticGrid-II on the Way to Interoperability

Olgerts Belmanis (1), Ake Edlund (2), Robert Palak (3), Ilmars Slaidins (1)
(1) Riga Technical University, Latvia
(2) Royal Institute of Technology in Stockholm, Sweden
(3) Poznan Supercomputing and Networking Center, Poland

Fast and non-coordinated development of regional ICT based e-Infrastructures creates challenges of interoperability. Now new projects and initiatives have started to make e-Infrastructures standards based, user friendly and interoperable. In this paper the BalticGrid II project is presented and the project efforts in the promotion of standards and to maintain interoperability of grid systems.

The Baltic region is a relatively small geographic area where three major middlewares' user communities overlap. There are gLite, ARC and Unicore. gLite is the dominant middleware used in the region due to the official status of it in the BalticGrid and BalticGrid-II projects.

The key services of Grid are Job Submission, Information Services, Storage Management, Accounting, Job Monitoring, Database Access and Virtual Organization Management. BalticGrid-II project is promoted interoperation of the gLite-based infrastructure with UNICORE and ARC. Activity

SA1 “Grid Operation” include the separate task to develop interoperation with UNICORE and ARC resources within the region. There are two solutions: interoperability and interoperation. Interoperability between these middlewares can be achieved organizationally and technically.

Now BalticGrid-II use gLite and therefore is interoperable with EGEE. To provide progress MoU with KnowARC project and Unicore has been signed. The first important solution is Migrating Desktop. Developed by PSCN within Joint Research activity Migrating Desktop is advanced user friendly interface which provide interoperation between several production grids, based on various middlewares (e.g. gLite,,Unicore, ARC, i2g, GT4). Project partners are actively participating in e-IRG, OGF, EUGridPMA and EU Technical Groups of Grid and implementing accepted recommendations.

8. UNICORE Access to PL-GRID Resources

Krzysztof Benedyczak (1,2), Piotr Bała (1,2)
(1) ICM, Warsaw University, Poland
(2) Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland

UNICORE is well established middleware to provide access to computational resources. The biggest infrastructure is run under DEISA project and provides access to the main supercomputer center across Europe. UNICORE has been also selected as middleware supported by the European Grid Initiative.
In the PL-Grid project, the users will be allowed to access resources in the various ways including DEISA compatible one. In this paper we present architrecture of the UNICORE grid in PL-Grid. We present design of the UNICORE Operational Center which coordinates access to the distributed resource through UNICORE and provides central services such as global registry, informational services and Unicore Virtual Organization Services. The whole infrastructure is based on the recent release of the UNICORE 6.2 release.
We present procedure established to add computational resource to the UNICORE grid with the low technical and administrative barrier.

9. Initial Deployment of Distributed Java Programs in Clusters of JVMs Through Extremal Optimization Approach

E. Laskowski (1), M. Tudruj(1,3), I. De Falco (2), U. Scafuri (2), E. Tarantino (2), R. Olejnik (4)
(1) Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
(2) Institute of High Performance Computing and Networking, ICAR-CNR, Naples, Italy
(3) Polish-Japanese Institute of Information Technology, Warsaw, Poland
(4) Computer Science Laboratory of Lille, University of Science and Technology of Lille, France

The motivation for the research reported in the paper is optimized execution of distributed Java programs in clusters of Java Virtual Machines (JVM) implemented on clusters of workstations and desktop Grids. Optimization of the execution time of distributed object programs has been always a challenging task due to specific execution paradigm of object programs and particular features of the architecture of the JVM. Efficient load balancing on Grid platforms requires adequate computational and communication load metrics. Java program balancing strategies should account for optimization of initial distribution of components of a Java application (initial object deployment) and dynamic load balancing (objects migration) at run-time.

Execution of distributed Java programs in clusters and Grids is usually done using an execution management support. In this work, we have selected the ProActive Java-based framework for cluster and Grid computing. This framework provides a distributed Java API and a set of tools for program management in different environments such as desktop, SMP, LAN, clusters and Grid. The application model is based on Remote Mobile Objects, Group Communications and Asynchronous Execution with

synchronization (Futures mechanism), OO SPMD, task migration, Web Services and Grid support. All this is adapted to various protocols such as rmi, ssh, LSF, Globus.

The paper is concerned with initial deployment of Java applications optimization where we can identify the following steps:1. Measuring properties of the environment: CPU power and network resources availability,
2. Execution of programs for some representative data to create a Method Call Graph (MCG) with the use of method dependency graph and measured data.
4. Finding the optimal mapping of the MCG graph in a given executive system.
5. Deployment of the application inside the ProActive framework.

Environment monitoring (system observation) predicts CPU and network services availability based on current CPU load and network utilization**.** The applied principle is based on the observation that an average idle time that threads report to the OS is directly related to the CPU load. It also accounts for the maximal number of RMI calls per second. Object behaviour monitoring (application observation) determines the intensity of communication between active objects. The principle is based on measuring the number of method calls between ProActive active (global) objects and the volume of serialized data.

Finding the optimal mapping of application tasks onto the nodes in heterogeneous environment is NP–hard, so, we use an Extremal Optimization algorithm for mapping of tasks to nodes. Extremal Optimization (EO) is a very fast co-evolutionary algorithm proposed by Boettcher and Percus [1]. EO works with a single solution S made of a given number of components s_i, each of which is a variable of the problem. There are two fitness functions used, one for the components s_i and one for a global solution. In an EO algorithm, after an initial random solution S is generated, the fitness value is computed for each of the components s_i. The worst variable is randomly updated, so that the solution is transformed into another solution S' belonging to its neighborhood. To avoid stucking in local optima, a probabilistic version of EO /τ-EO/ is used [2], in which the variables are chosen for updating using a value-controlled selection inside the fitness functions order of all variables.
wo application program models are used: an undirected weighted Task Interaction Graph (the TIG model) and a directed weighted acyclic graph (the DAG model).

References

1. Boettcher, S., Percus A.G.: Extremal optimization: methods derived from co-evolution. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99), San Francisco, (1999), 825–832.
2. Boettcher, S., Percus, A.G.: Extremal optimization: an evolutionary local–search algorithm. Computational Modeling and Problem Solving in the Networked World, eds. H. M. Bhargava and N. Ye. Kluver, Boston (2003).

10. Connecting Community Grids on the Meta Scheduling Level: The DGSI Project

Georg Birkenheuer (1), Arthur Carlson (2), Mikael Högqvist (3), Alexander Papaspyrou (4), Bernhard Schott (5), and Wolfgang Ziegler (6)
(1) Paderborn Center for Parallel Computing, University of Paderborn, Germany
(2) Leibniz-Rechenzentrum, München, Germany
(3) Zuse Institute Berlin, Germany
(4) TU Dortmund University, Germany
(5) Platform Computing GmbH, Ratingen, Germany
(6) Fraunhofer Institute SCAI, Sankt Augustin, Germany

Most Service Grids share the ability to efficiently distribute user workload to the resources available. This issue, usually generalized under the term 'Grid Scheduling' or Meta Scheduling,' is already very diverse within a community: both submitted jobs and available resources differ considerably, to the extent that coordination has to handle specialized knowledge about usage scenarios and infrastructure. This leads to very different, community-specific approaches for the development of Grid scheduling services.

The resulting incompatibility on the meta-scheduling level, however, proved to be a major hurdle for the coordinated cooperation of different Service Grids especially when focusing on the overall goal of better resource utilization. Moreover, cooperation on a scientific, cross-disciplinary level, are being impaired as well. As such, two major use cases for temporarily including alien resources into the own community arise: first, the need to cover peak demand, and second, the usage of specialized resources (vector-based HPC/HTC systems, CAVE environments, etc.).

The „D-Grid Scheduler Interoperability (DGSI)“ is targeting these use cases with the conception and development of a standards-based interoperability layer for Grid level scheduling in service Grids. By allowing the users of a community to distribute the workload among resources within the management domain of another community while keeping the individual, specialized scheduling solutions being run by the communities, it offers new perspectives for community collaboration, resource sharing, and efficient utilization.

The two scenarios foreseen within the framework of the project are thus the delegation of activities and the delegation of resources. In resource delegation, a local scheduler can place a resource available within its own domain under the exclusive planning authority of another scheduler outside of the community, for a previously negotiated time window. This approach is particularly suited where the partners participating in the negotiation and delegation are using the same or similar base middleware systems, and when the schedulers involved exhibit differences in the area of activity description or management functionality. In activity delegation, a Grid scheduler hands over an activity and the management of its execution to the domain of the scheduler of another community. In this way, an incompatibility of the basis middleware of two communities can be bridged orthogonally to the first approach introduced. This approach also requires negotiation between both Grid scheduling services.

In this paper, we discuss the technical aspects of these two approaches in the context of protocol design, incorporation of Service Level Agreements, and modeling of activity and resource requirements in the information model. Furthermore, we show for a number of popular Grid scheduling systems the integration into a federated environment using the DGSI protocols. Finally, we discuss the notion of mediation over third parties in the context of delegating activities and resources.

Acknowledgements. This work is supported by German Ministry of Education and Research under project grant #01IG09009.

References

1. S. Freitag, C. Grimme, A. Papaspyrou, and L. Schley, **On the Applicability of OGSA- BES to D-Grid Community Scheduling Systems**, In *Proceedings of the German E-Science Conference (GES)*, Baden-Baden, Germany, May 2007.
2. U. Schwiegelshohn und R. Yahyapour, **Attributes for communication between Grid scheduling instances,** GFD-I.6, *Scheduling Working Group, Global Grid Forum (GGF)*, Dezember 2001.
3. A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke und M. Xu, **WS-Agreement - Web Services Agreement Specification**, *Grid Forum Document,* GFD.107, Open Grid Forum, Mai, 2007.

11. Processing and Negotiation of Natural Language Based Contracts for Virtual Organizations

Mikołaj Pastuszko (1), Bartosz Kryza (2), Renata Słota (1), Jacek Kitowski (1,2)
(1) Institute of Computer Science AGH-UST, Krakow, Poland
(2) ACK Cyfronet AGH, Krakow, Poland

Nowadays practices of sharing both hardware and software resources between companies and institutions are often based on the idea of Virtual Organizations (VO). In order to define responsibility areas for participants of VOs the contracts are prepared. The most important part of the contract from the perspective of IT infrastructure are Service Level Agreement (SLA) and security assertions as they enable technical administrators of the VO to configure middleware components so that they meet negotiated contract requirements. However, the configuration process – often tedious and error prone, could be performed by a software application instead of a human, which is a goal

targeted by the Framework for intelligent Virtual Organizations (FiVO) [1][2]. For representation of contract content FiVO proposes the use of OWL ontologies [2]. Most contracts however are written in natural language. This raises a need for a system that would process the contracts, analyze them and translate into the contract ontology. The most significant application that uses information extraction techniques for ontology learning is Text2Onto [3]. What makes it inapplicable in this case is that it focuses on learning entirely new ontologies rather than creating individuals based on the already existing ontologies. Moreover, Text2Onto uses Probabilistic Ontology Model for knowledge representing purpose while the contract statements representation requires very high level of certainty, correctness and exactness. Other two useful approaches are presented in [4] but neither of them covers the ontology as output requirement.

Our research focuses on processing of contracts in order to create their ontology-based representation. The system behaves as a batch processor. The input consists of a contract document written in natural language, an ontology describing VO resources (e.g. services shared among VO) and a contract dictionary (also in natural language) that associates ontology entities with their natural language representations (e.g. full names or phrases). The desired output is an ontology that reflects the statements defined in the contract (especially its SLA part). The contract ontology makes use of the VO resources ontology delivered on the input so that the former ontology creates individuals that are instances of entities defined in the latter ontology. The process is divided into stages so the user may control and verify transitional artifacts produced by each stage. The first stage analyses a contract dictionary and generates transitional files (written with use of formal syntax) that are later used by the contract analyser during the second stage. This phase relies mainly on regular expressions. The second stage is based on NLP processing components organized and controlled by the General Architecture for Text Engineering (GATE) platform. The process includes tokenization, sentence splitting, Named Entities recognition, Part-of-Speech tagging (including Polish), resource ontology associations recognition as well as contract statements recognition and analysis. As a result of this phase an annotated contract document is produced. The third stage uses annotations created during the previous step in order to create an output ontology. The ontology may be then used by the other components of FiVO system to automatically configure VO resources.

The prototype implementation have been tested with simple contract documents similar to the actual VO contracts. The desired statements describing organizations' obligations was properly extracted and included in the output ontology. Future development will focus on extending support for more complex sentences (in a sense of natural language syntax).

In this paper an architecture of the prototype contract processor is introduced. It covers both high-level architecture (system as a set of modules) and low-level architecture (description and internal structure of individual modules). A module responsible for the contract text analysis is described in detail as a sequence of GATE processing resources. Finally, an example contract processing as well as the results gathered during tests with the prototype application are presented and summarized.

Acknowledgements. This research is supported by the AGH grant 11.11.120.777 and partly by the POIG.01.03.0100-008/08 Project "IT-SOA".

References

1. B. Kryza, L. Dutka, R. Slota and J. Kitowski, Security Focused Dynamic Virtual Organizations in the Grid based on Contracts, in: P. Cunningham and M. Cunningham (Eds.), Collaboration and the Knowledge Economy, Issuses, Applications, Case Studies, vol.5, part II, IOS Press 2008, pp. 1153-1160
2. B. Kryza, L. Dutka, R. Slota, and J. Kitowski, Dynamic VO Establishment in Distributed Heterogeneous Business Environment, in: G. Allen, et al. (Eds.), ICCS 2009 – 9th Intern. Conf. Baton Rouge, LA, USA, May 25-27, 2009, LNCS 5545, Springer 2009, pp. 709-718
3. P. Cimiano, J. Völker, Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery, in: A. Montoyo, et al. (Eds.), Proc.10th Intern.Conf. on Applications of Natural Language to Information Systems (NLDB), Alicante, LNCS, 3513, Springer 2005, pp. 227-238
4. Z. Milosevic, R. Geoff Dromey, On Expressing and Monitoring Behaviour in Contracts, Proc.6th Intern.Conf. Enterprise Distributed Object Computing Conference, September 17-20, 2002, p.3-14

12. Semantic-Based SLA-oriented Performance Monitoring in the ProActive Environment

Dariusz Król, Włodzimierz Funika
Institute of Computer Science AGH, Krakow, Poland

Nowadays, when the most of the commercial companies are aided with some kind of computer systems, the effort needed to manage them is still increasing. Each of these companies wants to provide its clients with a service whose quality is as high as possible. One way to guarantee the service quality is signing an agreement (called *Service Level Agreement - SLA*) between the service provider and the service consumer that defines the terms of exploiting the service. However, maintaining the SLA requires even more effort, e.g. a dedicated administrator has to watch out for any potential violations of the contract. It means that the *Total Cost of Ownership* (*TCO*) of such a service is very high and may be cost-ineffective from the company point of view.

This paper is intended to present a novel solution to the problem of enforcement of SLA by distributed applications in an automatic way which was proposed in the [1]. The main idea is to develop a self-adapting system (called *Autonomic Manager*) that will optimize the application deployment at runtime (i.e. usage of available machines) according to the current environment status. Information about the environment is retrieved from a monitoring system which is able to provide the values of various kinds of metrics measured on the running application. To integrate the managed application-Autonomic Manager-monitoring system triple, semantic web technologies, e.g. RDF/OWL, are exploited.

To explain the Autonomic Manager concept let us consider a typical use case that describes an SLA violation. An SLA specifies a metric and a value that has to be preserved during system runtime. A violation of the SLA means that at some time the actual metric value is higher (or lower) than the defined one. To avoid the SLA violation by detecting dangerous situations, the monitoring system and the Autonomic Manager are configured with different values of metrics (i.e. a little bit higher or lower than these specified within the SLA). When the actual value of metric during the system runtime exceeds a configured threshold, the monitoring system detects it and alerts the Autonomic Manager that something wrong may happen. The manager can change the configuration of the running application based on the semantic description of the environment in order to optimize its performance and to prevent possible SLA violations in the future.

The considered system is based on the self-adaptation and self-optimization. These features of computer systems are still new and fresh concepts in computer science. Their foundations were defined by IBM in the Autonomic computing initiative [2]. The main goal of the initiative is to reduce interactions between the computer system and its administrator by building more intelligent and self-manageable systems that will be able to handle some administrator's tasks by themselves. Others are also exploiting this new area by developing different kinds of autonomic systems in various areas, e.g. [3]. On the other side business companies are increasingly adding support of SLA in their products, e.g. [4]. However, to our best knowledge there is no system that would bring these two concepts together.

During the prototype implementation of the presented system, the ProActive suite [5] was used to develop distributed applications. To monitor the applications the SemMon monitoring system [6] was exploited. Each component of the mentioned ones communicates with others with the standard Java RMI mechanism.

Various tests performed with the implemented prototype prove that autonomic systems can be successfully adapted to enforce SLA in distributed applications. However, there is still a lot of work to be done. One of such a task is to enable the SemMon to monitor other types of environments, e.g. Enterprise Service Bus or to enhance methods of optimizing the monitored environment.

References

1. Krol, D: Semantic-based SLA-oriented proactive performance monitoring, Msc. Thesis, AGH, Krakow, 2009.
2. Autonomic computing web site IBM, 2001, http://www.research.ibm.com/autonomic
3. Autonomic network project web site, http://www.ana-project.org/
4. Alexander Keller, Heiko Ludwig. Dening and Monitoring Service Level Agreements for dynamic e-Business. Proceedings of the 16th System Administration Conference – 2002.
5. Service Level Monitoring with Cisco IOS Service Assurance Agent, White paper, Cisco, http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/slvsa wp.pdf

6. Godowski, P., Pegiel, P.: Adaptive monitoring of distributed Java applications, Msc. Thesis, AGH, Krakow, 2007.

13. Virtual Organization Security Layer Deployment in the FiVO Framework

Jakub Fibinger (1), Bartłomiej Puzoń(1), Bartosz Kryza (2), Renata Słota (1), Jacek Kitowski (1,2)
(1) Institute of Computer Science AGH-UST, Krakow, Poland
(2) ACK Cyfronet AGH, Krakow, Poland

Nowadays, as the IT infrastructures become the core backbones of the enterprises, it becomes mandatory for these organizations to be able to collaborate with other organizations by means of integrated IT systems. Two basic problems arise however when trying to apply this in practice: the lack of means for defining and controlling such collaborations and the high heterogeneity of the software used in different organizations. To address this problem, the FiVO framework [1] enables multiple institutions to create Virtual Organizations (VO) by providing tools needed to negotiate the contract which defines the rules of their cooperation in terms of security and QoS as well as support the configuration of their middleware in order to enable proper sharing of their resources. The FiVO's SECE component aims to ensure that the authentication and authorization layers of the VO members cooperate seamlessly, based on the negotiated contract. This paper will present the ontology allowing for unified definition of role based security assertions, architecture of the system, as well as early attempts to its application in configuration of security infrastructure in a Virtual Organization.

As the members of a VO used to be, and indeed still are, individual institutions, they use their own authentication and authorization layers to perform security to its users. Frequently, the security infrastructures of the collaborating parties are provided by different vendors, build with different assumptions and respond to different standards, e.g. X509 certificates [2] and Shibboleth [3]. SECE tries to solve these incompatibilities by providing tools that can be used by system administrators to reach the level of interoperability required by the VO to function. SECE enables the collaborating parties to include certain information about the security layers configuration in the FiVO contract. Contract is formed using Web Ontology Language (OWL). This information has to be sufficient in order to configure systems in existing institutions correctly. Having read the security information conveyed in the contract, SECE processes it and determines which configuration changes need to be performed in order for the Virtual Organization to work – that is, to enable the users to authenticate themselves at the shared resources and to execute their rights towards them (e.g. look up a nearest hotel). Afterwards, SECE performs three steps. First it applies configuration changes, including: database updates, LDAP updates, communication with security daemons, adding entries to configuration files. Next, it provides system administrators with scripted tools that will perform the needed configuration changes, if executed with certain administrative permissions that the FiVO framework have not been granted. Finally, it provides system administrators with information that need to be considered in order for the virtual organization to function seamlessly.

As configuring the security system of multiple institutions wishing to collaborate is a tedious and error-prone process, an automated tool for system administrators should encourage parties to create collaborative environments, thus being able to deliver to the customers services of better quality, responsiveness and value.

Acknowledgements. This research is supported by the POIG.01.03.0100-008/08 Project "IT-SOA".

References

1. B. Kryza, L. Dutka, R. Slota and J. Kitowski, Security Focused Dynamic Virtual Organizations in the Grid based on Contracts, in: P. Cunningham and M. Cunningham (Eds.), Collaboration and the Knowledge Economy, Issuses, Applications, Case Studies, volume 5, part II, IOS Press 2008, pp. 1153-1160
2. S. Chokhani, W. Ford, R. V. Sabett, C. R. Merrill and S. S. Wu, 'Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework' , Internet RFC 3647, 2003 .
3. R. O. Sinnott, J. Jiang, J. P. Watt and O. Ajayi, Shibboleth-based Access to and Usage of Grid Resources., *in* 'GRID' , IEEE, pp. 136-143, 2006

14. Securing Software Licenses in a Location Independent Manner

Björn Hagemeier (1), Daniel Mallmann (1), Wolfgang Ziegler (2)
(1) Research Centre Juelich, Juelich Supercomputing Centre, Juelich, Germany
(2) Fraunhofer Institute SCAI, Schloss Birlinghoven, St. Augustin, Germany

Grid and Cloud Computing have become very popular over the years. Cloud computing received a real boost through several commercial offers by big companies initially selling their spare CPU cycles to the general public and now making Cloud business with dedicated resources. Now that this scheme gets adopted more and more, there is a strong demand to not only run home-grown software in the Grid or Cloud, but also licensed, off-the-shelf software created by Independent Software Vendors. However, current licensing methodologies have several aspects that severely limit the usability of licensed software in these environments.

First of all, licensing models do not always allow running licensed software outside the administrative domain stated in the license agreement. There are a number of common contractual measures in license agreements to protect the ISV's ownership of the code and stop users from using the code in an unlawful manner. Other restrictions include geographical ones that allow for the execution of code within a certain geographical area, e.g. in a 30km radius around a customer's headquarters.

In the traditional licensing scenario, it is technically quite easy to bind the execution of software to a certain environment or hardware. This is used in node-locked licenses or per-user licenses. Using license servers, one can identify hosts from the local area network by their IP addresses and issue licenses to them based on this information. These solutions are not viable in distributed environments, where the actual execution hosts may not even be known in advance, nor be located in the same local network. It can also not be expected that licenses are bought in advance for all hosts or environments where software may potentially be executed. This would multiply license cost and not be affordable in most cases. Therefore, in order to allow ISVs to offer more flexible license management, which is not necessarily tied to hardware or a local administrative domain, ways need to be found to overcome these limitations and still provide strong protection against misuse of licensed software.

A small number of projects have already tried to tackle the problem. Some of them technically bent the existing licensing solution and opened them up to the public world, e.g. by setting up authenticated proxies and pretending that external requests to the license server come from the internal network [2][3].

In the SmartLM project [1], we developed a solution, which allows license servers to issue tokens of limited lifetime for the execution of a job. This job usually has a defined input, which can be bound to the token using digital signatures. The difference between this and previously existing solutions is that there is no need for a permanent connection to the license server. This enables us to move compute jobs wherever they need to go. A problem with this is that a license token used in such a manner, could be copied and thus exploited for additional compute jobs. In order to avoid that the user has any added value from using the token multiple times, the token is bound to exactly one set of input data.

Vendors have to trust the license server to issue license tokens on their behalf. This needs to be verified by the policy decision point (PDP) of the application for licenses, before the policy enforcement point then grants or denies the usage of the application based on the PDP's evaluation of the token. The trust is conveyed in a signed XML document, which binds the issuance of license tokens to a particular server. This document can be seen as an authorization. When a license token is issued, it has to contain the authorization. The token is only accepted, if the license server issuing it is the one mentioned as the subject of the authorization. As authorizations have lifetimes, they have to be renewed regularly and it is up to the ISV, for how long they trust the owner of a license server to issue tokens on their behalf.

SmartLM also tackles the legal and business aspects, by including ISVs and business experts in the development of new licensing models. The technical measures described in this work incorporate their requirements.

SmartLM is a project funded by the European Commission under contract number 216759.

References

1. SmartLM project web site: http://www.smartlm.eu/

2. Dalheimer, M.; Pfreundt, F.-J., "GenLM: License Management for Grid and Cloud Computing Environments," Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on , vol., no., pp.132-139, 18-21 May 2009
3. Raekow, Y.; Simmendinger, C.; Krämer-Fuhrmann, O.: License Management in Grid & High Performance Computing. Computer Science, Research + Development 23 (2009), 3-4,P. 275-281

15. Integration of the OCM-G Monitoring System into the MonALISA Infrastructure

Włodzimierz Funika, Bartosz Jakubowski, Jakub Jaroszewski
Institute of Computer Science AGH-UST, al. Mickiewicza 30, 30-059 Kraków, Poland

In this paper we present the way to integrate the grid monitoring system OCM-G [1] into the monitoring agent-based architecture of MonALISA [2]. OCM-G is designed to monitor distributed, parallel and grid applications and provide on-line information about their execution status as well as parameters of nodes they are running on. Our aim is to provide extensible properties of performance visualisation of grid applications with using powerful capabilities of MonALISA services. They make data human readable by plotting graphs from oncoming parameters values and make them available in Internet through common protocols. Since grid applications feature a high extent of dynamics, there is a need in special techniques of visualization, lacking in the conventional performance tools. When speaking about grid monitoring there is another important benefit of MonALISA - services and repositories are registered in a Lookup Service. It makes all the data available for every registered agent which is then capable of making decisions upon them. The interface between two tools is based on the OMIS specification [3]. We take advance of what is best in both tools: access to various abstraction levels of performance monitoring data in OCM-G and scalable modes of execution visualization in MonALISA to provide a tool that makes possible extensible fine-grained monitoring of grid nodes and convenient presentation of results. In the full paper we describe more technical details and show a sample configuration and performance visualization session.

References

1. OCM-G web site, http://grid.cyfronet.pl/ocmg
2. Newman, H.B., Legrand, I.C., Galvez, P., Voicu, R., Cirstoiu, C.:MonALISA : A Distributed Monitoring Service Architecture. In Computing Research Repository, vol. cs.DC/0306096, 2003
3. Ludwig, T., Wismüller, R.: OMIS 2.0 — a universal interface for monitoring systems. In Recent Advances in Parallel Virtual Machine and Message Passing Interface, pp. 267-276, Springer, 1997
4. Chalakornkosol, N., Uthayopas, P.: Monitoring the Dynamics of Grid Environment using Grid Observer, IEEE CCGRID2003, Toshi Center, Tokyo, May 12-15, 2003.
5. Massie, M. L., Chun, B. N., Culler, D. E.:The ganglia distributed monitoring system: design, implementation, and experience. In: Parallel Computing, Volume 30, Issue 7, pp. 817-840, 2004

16. Grid Added Value for Running Ordinary Applications from Developing Countries

Leandro N. Ciuffo (1), Bruce Becker (2), Giuseppe La Rocca (1), Roberto Barbera (1,3)
(1) Istituto Nazionale di Fisica Nucleare, Catania, Italy
(2) Meraka Institute, Pretoria, South Africa
(3) Università di Catania, Italy

Grid computing is a powerful paradigm for running ever-larger workloads and services. Such a technology, which was initially driven by the requirements of efficient processing of the huge amounts of data generated by High Energy Physics experiments, has proven to be a very effective way of tackling intensive computing needs such as those found in climate simulations and drug discovery. A

typical Grid-demanding application requires several CPU-hours to run, lasting for even days when running on a local computing cluster.

However, recent investments to promote Grid computing within new scientific communities in several regions of the world [1] have been attracting new research groups interested in investigating the potential benefits that Grid computing can bring for their academic pursuits.

Especially in the developing countries - where the majority of research labs are small (a dozen people or so), focus their research on some narrow topic and cannot afford to have a cutting-edge computing system -, the affiliation of such groups with a large Grid project also allows to overcome the digital divide either by enabling them to speed up the retrieval and analysis of their experiments or by just granting access to extra computing resources.

Based on our experience supporting more than 60 applications in the framework of both EELA-2 [2] and the former EUMEDGRID [3] projects, we have noticed that many Latin-American and North-African applications that have been ported on these projects' Grid Infrastructures are not quite computational-intensive, requiring only a few CPU-hours for running. In addition, most of these applications are submitted to run on an occasional basis. This evidences that many researchers are still going through a "proof-of-concept phase", trying to understand the actual gain that the Grid paradigm represents.

The work presented in this paper is organized in two distinct but complementary parts: (i) Firstly we analyzed the profile of 20 applications developed by different Latin-American institutions and that have been recently ported to the EELA-2 Grid infrastructure. We also included in this study 6 applications from the EUMEDGRID project developed by North-African and Western-Asian countries. (ii) Based on the analysis made and considering the average application profile, we built a test-application that requires less than 6 CPU-hours to run on a personal computer and empirically measured its total job delay while running on the EELA-2 Grid infrastructure. Altogether 200 jobs were submitted from August 2nd to September 10th. Our aim is to discuss the pros and cons of using Grids for running ordinary applications.

EELA-2 adopts the gLite middleware [4] - an European solution developed by the CERN jointly with the EGEE project [5].

Since cloud computing is currently seen as a prominent cost-effective way for organizations to run scalable applications, we also executed our test-application using two different standard instance types of the Amazon Elastic Compute Cloud (EC2) service [6].

Acknowledgements. This work makes use of results produced by the EELA-2 project (http://www.eu-eela.eu), co-funded by the European Commission within its Seventh Framework Programme.

References

1. BIRD, I., JONES, R. & KEE, K.F. The Organization and Management of Grid Infrastructures, Computer, vol. 42, no. 1, pp. 36-46, Jan. 2009, doi:10.1109/MC.2009.28.
2. EELA-2. (2008). E-science grid facility for Europe and Latin America. Retrieved February 7, 2009, from: http://www.eu-eela.eu
3. EUMed-Grid - Empowering eScience across the Mediterranean (2006). Retrieved February 8, 2009, from: http://www.eumedgrid.org
4. gLite (2008). Documentation. Retrieved February 7, 2009, from: http://glite.web.cern.ch/glite/documentation/
5. EGEE. (2008). Enabling Grids for E-sciencE. Retrieved February 8, 2009, from: http://www.eu-egee.org/
6. Amazon EC2 Instance Types. Retrieved August 29, 2009, from : http://aws.amazon.com/ec2/instance-types/

17. Investigating Dynamics of Mammalian Cortical Hypercolumn in Parallel PCSIM Simulations

Grzegorz M. Wojcik (1)
(1) Institute of Computer Science UMCS, Lublin, Poland

The amplitude and pitch fluctuations of natural soundscapes often exhibit "1/f" spectra which means that large, subnormal changes in pitch or loudness occur statistically less frequently that gentle

sound fluctuations. It was proved on ferrets in laboratories that neurons in auditory cortex is indeed tuned to "1/f" dynamics. Psychological experiments conducted on human beings show that listeners indeed prefer 1/f distributed melodies to melodies with faster or slower dynamics [1].

Such frequency tuning found in biological systems lets us predict Self-Organizing Criticality behavior manifesting in mammalian brain cortex [2]. In this paper we investigate the dynamics of cortical hypercolumn consisting of about 130 thousands of simulated neurons. The set of 128 Liquid State Machines [3], each consisting 1024 neurons, was simulated on a simple cluster built of two double quad-core machines (16 cores).

PCSIM is a tool for simulating artificial biological-like neural networks composed of different models of neurons and different types of synapses. The simulator is written in C++ with a primary interface designed for the Python programming language. According to its authors it is intended to simulate networks containing up to millions of neurons and on the order of billions of synapses. This is achieved by distributing the network over different nodes of a computing cluster by using MPI [4].

In this report we will discuss the results obtained for different models of neurons used for the construction of the hypercolumn model, different time of simulations and varying density of inter-column connections. The method of parallelisation and benchmarking results for using the PCSIM on the cluster and predictions for grid computing will be presented to some extent. Research presented herein makes a good starting point for the simulations of very large parts of mammalian brain cortex and in some way leading to better understanding of the functionality of human brain.

Acknowledgement: Support for this research was provided by the Polish State Committee for Scientific Research (grant N N519 403734).

References

1. Jose A. Garcia-Lazaro, Bashir Ahmed, J.W.H. Schnupp: Tuning to Natural Stimulus Dynamics in primary Visual Cortex; Current Biology 16, 264-271, 2006.
2. P. Bak: Self Organizing Criticality; Physical Review A, 38, 364-374, 1988.
3. W. Maass, T. Natschlaeger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation, 14(11):2531-2560, 2002.
4. PCSIM: A Parallel neural Circuit SIMulator: http://www.lsm.tugraz.at/pcsim/

18. Scaling-up MATLAB® Application in Desktop Grid for High-Performance Distributed Computing — Example of Image and Video Processing

Olexandra Baskova, Olexander Gatsenko, and Yuri Gordienko,
G.V.Kurdyumov Institute for Metal Physics of the National Academy of Sciences of Ukraine, Kiev, Ukraine

Recently, the distributed computing model becomes very popular due to feasibility to use donated computing resources of idle PCs by means of the BOINC software platform [1] and availability of simple and intuitive Distributed Computing Application Programming Interface (DC-API) [2]. Usually, a sequential application by slight modifications in its code could to be ported to the parallel version for worker nodes of a distributed Desktop Grid (DG). It is possible for an application with independent processing of big volume of data in a client-server model, for example, for batch image and video processing. The current common tendency is to use the specialized software with many standard functions in an interactive manner by means of a high-level language. For example, MATLAB® software includes Parallel Computing Toolbox™ which allows users to take advantage of parallel processing and MATLAB® Distributed Computing Server software allows users to run as many MATLAB® workers on a remote cluster of computers as their licensing allows. Unfortunately, by many reasons (technical, economic, legal, privacy, etc.), it is not possible to scale up this approach significantly without additional investments.

The main aim of the work was to develop and test the applicability of integration of MATLAB® objects and codes in a DG for high-performance distributed computing on the example of image and video processing in solid state physics and microscopy [3]. Usually, the experimental data, namely, separate

snapshots of video file (typically 10^2-10^5 frames), can be considered as independent ones from each other, divided in independent chunks (tasks), which could be spread among multiple "workers" with very high efficiency. This application potentially could have a very wide range of users, because modern laboratories has commercial microscopes with digital video output connection to PC and perform everyday tasks of complex morphology analysis: in biology, geology, chemistry, physics, materials science, etc. For this purpose our MATLAB®-code for intensive image and video processing was compiled to the standalone sequential applications for Windows and Linux platforms, wrappers for the compiled M-code were provided, and the external calls in C++ functions generated by MATLAB® Compiler. Then ported parallel version of the sequential application was developed and tested on the worker nodes of a DG installed on the basis of the BOINC software platform and DC-API.
The experiments shown that performance of the DG-enabled application is much higher than for sequential version and closely proportional to the number of worker nodes involved in the computation. Of course, it was far from theoretical values, because of DG management overheads, heterogeneity of computing infrastructure, human factor, etc. The potential problems were brought to light that were related with video format and conversion issues, import/export of data structures, resource limitations on application in heterogeneous distributed computing infrastructure, etc. The speedup of the DG-enabled version of the application was measured and compared to the original sequential version.
The main achievement is the practical possibility to port, distribute, and use DG-enabled version of a MATLAB®-code application to workers that do not have MATLAB® software on their systems (except for the free MATLAB® MCR-software needed to support the wrappers). The further improvements are related to the special measures to get more uniform computing infrastructure (and avoid legacy PCs that create bottlenecks), more efficient usage of RAM and hard drive memory, etc. In addition, the big reserve of improvement is related with development of CUDA-enabled versions of client part, scaling-up local DG, and delivering the application to EDGeS DG->EGEE Bridge.

Acknowledgements. The work presented here was funded by the FP7 EDGeS project. The EDGeS (Enabling Desktop Grids for e-Science) project receives Community funding from the European Commission within Research Infrastructures initiative of FP7 (grant agreement Number 211727) (www.edges-grid.eu).

References

1. BOINC — Berkeley Open Infrastructure for Network Computing (http://boinc.berkeley.edu/).
2. SZTAKI Desktop Grid set up by the Computer and Automation Research Institute (SZTAKI) of the Hungarian Academy of Sciences (MTA) (www.desktopgrid.hu/).
3. Yuri Gordienko, Elena Zasimchuk, Rimma Gontareva: Heterogeneous Pattern Formation at Aluminum Surface — Fourier, Morphology and Fractal Analysis; Materials Science Forum, vol 567-568, pp.169-172, 2007.

19. Desktop Grid Computing in Materials Science Lab — Example of Development and Execution of Application for Defect Aggregation Simulations

Olexander Gatsenko, Olexandra Baskova, and Yuri Gordienko,
G.V.Kurdyumov Institute for Metal Physics of the National Academy of Sciences of Ukraine, Kiev, Ukraine

The processes of monomer aggregation in clusters are investigated in many branches of science: defect aggregation in materials science, population dynamics in biology, city growth and evolution in sociology, etc [1-3]. The typical simulation of a cluster aggregation scenario takes several days on a single modern processor, depending on the number of Monte Carlo steps (MCS). However, thousands of scenarios have to be simulated with different (a) input parameters (physical parameter sweeping), (b) sizes of simulated systems (to investigate a scaling effect), (c) initial configuration files with the same input parameters (sweeping the statistical realizations). The parallel execution environment can reduce the waiting time and scale up the simulated system to the desirable realistic values.

The main aim of the work was to test the applicability and everyday efficiency of the Desktop Grid computing technology in an ordinary materials science lab with heterogeneous distribution of desktop computing resources. For this purpose the ported parallel version of the sequential application was developed and tested on the worker nodes of a Desktop Grid (DG) installed on the basis of the BOINC software platform [4] and Distributed Computing Application Programming Interface (DC-API) [5]. The only difference is that instead of running thousands of runs on the same computer sequentially, DG-enabled parallel version distributes the runs to the DG PCs. As these experiments are independent from each other, a master-worker type of parallelisation (embarrassing parallelism) is used that is ideally suited to our distributed computing infrastructure.
The experiments shown that performance of the DG-enabled application is much higher than our expectations even in the low-scale local DG testbed. It is especially related with a huge economy on personnel working time due to the better usability and more efficient housekeeping than for running thousands runs of sequential applications. The quantitative analysis was performed and potential pitfalls were brought to light that were related with random blind distribution of workunits among workers, dynamical memory allocation, resource limitations in heterogeneous distributed computing infrastructure, correct checkpoint, and calculation of fraction of work done. The speedup of the DG-enabled version of the application was measured and compared to the original sequential version. Theoretically, the performance gain of the DG-enabled version compared to the sequential one should be close to the number of worker nodes involved in the computation. But the DG-produced overheads, heterogeneity of computing infrastructure, human factor, of course, reduce the actual speedup.
The main achievement is that BOINC-DG-DC-API ideology was proved to be effectively used by non-professional developers in an ordinary materials science lab with significant increase of productivity. The further feasible improvements are proposed on the basis of special measures as to get more uniform computing infrastructure (and avoid legacy PCs that create bottlenecks), more efficient usage of memory, more correct calculation of fraction done by the application and related scheduling, etc. In addition, the big reserve of improvement is related with development of 64-bit and CUDA-enabled versions of client part. The absence of significant performance difference of the Windows and Linux versions of the application shows that if our private local DG will be scaled up, or the application will be delivered to EDGeS DG->EGEE Bridge, than the performance will increase also.

Acknowledgements. The work presented here was funded by the FP7 EDGeS project. The EDGeS (Enabling Desktop Grids for e-Science) project receives Community funding from the European Commission within Research Infrastructures initiative of FP7 (grant agreement Number 211727) (www.edges-grid.eu).

References

1. I. M. Lifshitz, V.V. Slyozov: The Kinetics of Precipitation from Supersaturated Solid Solution; J. Phys. Chem. Solids, 19, [1/2] , pp. 35-50, 1961.
2. S. Ispolatov, P.L. Krapivsky, S. Redner: Wealth Distributions in Models of Capital Exchange; Eur. Phys. J., B 2, pp. 267-276, (cond-mat/9708018), 1998.
3. F. Leyvraz , S. Redner: Scaling Theory for Migration-Driven Aggregate Growth; Phys. Rev. Lett., 88, 068301, (cond-mat/0108289), 2002.
4. BOINC — Berkeley Open Infrastructure for Network Computing (http://boinc.berkeley.edu/).
5. SZTAKI Desktop Grid set up by the Computer and Automation Research Institute (SZTAKI) of the Hungarian Academy of Sciences (MTA) (www.desktopgrid.hu/).

20. GriF: a new collaborative Grid Framework for SSCs

Carlo Manuali (1), Antonio Laganà (2)
(1) *Department of Mathematics and Informatics, University of Perugia, Perugia (IT)*
(2) *Department of Chemistry, University of Perugia, Perugia (IT)*

Grid empowered calculations have become nowadays indispensable for advances in many scientific areas. Despite this, access to the grid is still far from being user friendly for normal scientists and there is need for this purpose to develop new tools. In our laboratory we have spent several efforts to simplify and standardize this task for the computational applications of the COMPCHEM

Virtual Organization [1] of EGEE [2]. In particular we have used a Web Service approach to implement a Collaborative Grid Framework specialized in running on the Grid reactive scattering codes.

Reactive scattering quantum calculations are presently used routinely either to validate Potential Energy Surfaces or to carry out ab initio calculations of the reactive properties of atom-diatom systems. Even for the simple atom-diatom case, however, a full three dimensional treatment needs a massive computational campaign that we have made viable by developing for this purpose a new Grid based Framework (GriF) and structuring a program performing reactive scattering calculations as a Web Service distributed on the Grid.

GriF is a JAVA Service Oriented Architecture Framework that provides Web Services aimed at exploiting the articulation of scientific computational applications in sequential, concurrent or alternative paths on the Grid and is specialized to deal with molecular processes. The application can be built by composing one or more Web Services without knowing the detail of their underlying implementations. Along this line, an extensive use of the Simple Object Access Protocol (SOAP) is made. SOAP is the XML-based messaging format established as transmission framework for inter-service communication (via HTTP or HTTPS). Accordingly, the Universal Description, Definition, and Integration (UDDI) protocol is adopted for the service registry operations which acts as a directory listing. Applications inspect the registry for services and invoke the desired Web Services which are described using the Web Services Description Language (WSDL). This provides GriF Users with the additional advantage of not needing to bear Grid Certificates, to handle binary programs, to choose among the Grid resources the suitable CEs, SEs and UIs, and to know the specific commands of the operating system. GriF operations to be carried out are, in fact, mainly based on natural-like language.

As a case study the ABC program [3] was considered and applied to the quantum study of the $N + N_2$ reaction. Implications for Quality of Service and Quality of Users [4] evaluation as well as for the related Credit assignment algorithms will also be presented.

References

1. Laganà, A., Riganelli, A., and Gervasi, O.: Lecture Notes in Computer Science, 3980, 665-674 (2006)
2. The Enabling Grids for E-sciencE (EGEE), http://www.eu-egee.org/
3. Skouteris, D., Castillo, J. F., Manolopoulos, D. E.: ABC: A Quantum Reactive Scattering Program, Comp. Phys. Comm., 133, 128-135 (2000)
4. Manuali, C., Laganà, A., Gervasi, O., Costantini, A.: On the Structuring of a Molecular Simulator as a Grid Service, Chemistry and Material science applications on Grid infrastructure - ISBN 92-95003-42-X, School GRID (INFN), International Centre for Theoretical Physics (ICTP) - Trieste (IT), 24-28 November (2008)

21. Time and Memory Requirements in Natural Language Processing of Polish

Marcin Kuta, Jacek Kitowski
Institute of Computer Science AGH, Krakow, Poland

Part of speech (POS) tagging is a cornerstone of almost any text processing task including parsing, machine translation, information retrieval, word sense disambiguation, and many others. Polish is a highly inflective language and tagging such language is much more difficult than analytic languages like English. One of difficulties is that tagging algorithms are especially computationally time demanding when applied to languages described with large tagsets.

The document clustering is another important technique of Natural Language Processing (NLP), which finds applications in unsupervised organization of documents, returning web search results, creating lexico-semantic nets (wordnets), information retrieval and topic extraction. POS tagging methods help to select from a text the most important features for clustering algorithms.

Works on NLP algorithms usually focus on quality measures, expressed in terms specific to family of methods, (e.g. accuracy for POS tagging or purity for clustering algorithms). In this paper we present other aspect, i.e., time and memory requirements of some of POS tagging and clustering algorithms applied to the Polish language [1,2]. These requirements are important factors, which can affect the choice of respective algorithm or make useless an algorithm, which is the best according to

the main measure. This justifies application of grid architectures or cloud platforms to above NLP computations.

Some of POS tagging or clustering algorithms are time and memory consuming but tuning above algorithms to Polish requires consideration of many parameters and multiple, even more intensive computations. The grid architectures are well suited to such multivariant computations and satisfy well time and memory demands. The distributed nature of grid is useful for combination taggers, which base on several, independent baseline taggers.

For the POS algorithms the multiple computations were required due to such factors as:

- training taggers with n-fold validation [2],
- effect of training data size (analysis learning curves) [3],
- number of baseline taggers used in composition of combination taggers [4].

The second group of experiments, i.e. clustering experiments, were conducted on the set of 10,000 articles (the corpus) selected from one of the main Polish newspapers [5]. The following aspects of clustering algorithms were taken into account:

- different representations of the corpus (feature selection),
- selection of appropriate clustering algorithm [6] (e.g. agglomerative or repeated bisections),
- alternative between Vector Space Model (VSM) and Latent Semantic Analysis (LSA) space,
- investigation of optimal number of dimensions, in case of LSA model,
- several cluster criterion functions,
- different term weighting schemes,
- multiple execution of clustering algorithms depending on random initialisation.

The detailed results of time performance of selected algorithms will be depicted in the paper.

Acknowledgements. This work was supported by AGH-UST grant no. 10.10.120.802.
ACC CYFRONET AGH is acknowledged for the computing time.

References

1. Kuta, M., Chrząszcz, P., Kitowski, J.: Increasing quality of the Corpus of Frequency Dictionary of Contemporary Polish for morphosyntactic tagging of the Polish language. Computing and Informatics 28(3) (2009) 319–338
2. Kuta, M., Wrzeszcz, M., Chrząszcz, P., Kitowski, J.: Accuracy of baseline and complex methods applied to morphosyntactic tagging of Polish. In: Proc. of the 8th Int. Conf. on Computational Science (ICCS 2008). (2008) 903–912
3. Kuta, M., Chrząszcz, P., Kitowski, J.: A case study of algorithms for morphosyntactic tagging of Polish language. Computing and Informatics 26(6) (2007) 627–647
4. Kuta, M., Wójcik, W., Wrzeszcz, M., Kitowski, J.: Application of stacked methods to part-of-speech tagging of Polish. In: Proc. of the 8th Int. Conf. on Parallel Processing and Applied Mathematics (PPAM 2009). (2009)
5. Weiss, D.: The corpus of the Polish daily *Rzeczpospolita* (years 1993-2002). http://www.cs.put.poznan.pl/dweiss/rzeczpospolita
6. Zhao, Y., Karypis, G.: Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery 10(2) (2005) 141–168

22. Blended Metrics Comparative Algorithm: Initialization Phase

Marián Knězek (1)
(1) Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies, Slovak University of Technology, Bratislava, Slovakia

In more extensive networks, coexistence of several protocols is needed for routing at network layer. There are several reasons why multiprotocol environment is used: network politics, migration between routing protocols, vendor incompatibility, application and business requirements.

Protocols at network layer process of selecting appropriate paths to packets destinations [1, 4, 5]. The best path selection influence overall network efficiency. Although the best path selection methods are sophisticated within particular protocols, there are no warranty that router really chooses the best path which includes results among optimal routes particular protocols. On the present, the most

common technique for the shortest path selection among several routing protocols is based on assigning administrative distance (next AD). AD designates routing protocol reliability [4].

However, better route can be routed by less reliable routing protocol. When AD is used, router is not able to choose truly the best path among more routing protocols. Router only knows that routes with less constant (AD) will be preferred. Selecting suboptimal paths on routers can affect less overall network efficiency.

In existing work, there is solution based on redistribution with route maps. There is no guarantee that using route redistribution is optimal due to its basic disadvantages like routing loops, incompatible routing information and inconsistent convergence time [3].

There are also solutions based on changing metrics in route maps for specific cases or based on edge router [5]. These principles are exerting manually in specific cases. Our solution based on Blending Metrics (next BM solution) [1, 6] is intended to find automatic approach, which overcomes manual settings.

To overcome problem caused by AD, we published BM solution [1] based on finding the best route among particular routing protocols. Without our solution, routing protocols are not able to communicate with each other in case of sharing metrics results. Router sorts routes by AD. Our solution supports collaboration among routing protocols. We are developed solution based on comparison of different kind of metrics. Consequently, router is able to make more exact the best path decisions.

Whole process is covered by Blended Metrics Comparative Algorithm (next BMCA). It covers whole BM solution what can be directly implemented in router. In general, BMCA process routes with different kind of metrics. Result of BMCA is the best path selection for each destination. This paper is focused on core of BM solution - Initialization phase of BMCA.

RSim - new network simulator provides environment for testing and verification of BM solution. Based on processed tests, our network simulator shows quite similar outputs in comparison with real network topology built in Cisco Laboratory.

In RSim, we have analyzed contribution of BM solution in various network topologies. We have analyzed outputs with and without BM solution for various network topologies. Based on our actual tests, BM solution finds more optimal routing paths than AD solution. A detailed description of our solution is given as well as actual results produced by the RSim.

Routing intermediate data among grids influences total performance in virtual organization. The approach presented could be adapted to routing of computing elements as encapsulated packets in virtual organization operating arbitrary jobs in grid computing.

Acknowledgements. We are very grateful to Assoc. Prof. Margaréta Kotočová for valuable advices. This work was supported by the Grant No.1/0649/09 of the Slovak VEGA Grant Agency.

References

1. Knězek M.: Blending Routing Metrics for Optimalization the Best Path Selection in Multiple Protocol Environment, Cracow Grid Workshop'08 Proceedings, 2009. ISBN 978-83-61433-00-2
2. Orallo E., Carbo J.: Fast and efficient routing algorithms for delay-bounded and dependable channels, Comput. Commun. 2007
3. Lofgren S.: Route Optimalization (Available on 15.4.2009) http://www.mdh.se/netcenter/ct3390/vt-2006/forelasningar/Rout%20opt-vt06.ppt
4. Lammle T.: CCNA Study Guide, 5th Edition, Sybex, Alameda, California, USA, 2005, ISBN: 9780782143911
5. Timm C., Wade E.: CCNP: BSCI Study Guide, 2nd Edition, Sybex, Alameda CA, USA, 2003, ISBN: 978-0782142938
6. Knězek M: Optimalization of Routing Approaches in TCP/IP Networks (in Slovak) Available on 30.8.2009 http://knezek06.babyland.sk/pub04.pdf

23. DAG4DIANE - Enabling DAG-based Applications on DIANE Framework

Grzegorz Grzesło (1), Tomasz Szepieniec (1,2), and Marian Bubak (2,3)
(1) WSZIB, Kraków, Poland
(2) Institute of Computer Science AGH, Krakow, Poland
(3) ACC CYFRONET AGH, Krakow, Poland

Scientific applications often are expressed as workflows, which can be described as directed acyclic graph (DAG). Support of DAG-based application in a distributed e-infrastructure is a source of new requirements. This is related to the fact that a job should not be submitted until all dependencies are satisfied, and this leads to large overheads. We can find support of DAGs in several systems as Pegasus[1], or DAGMan extension for Condor[2]. Alternative solution for resources allocation is provided by DIANE framework[3], which enables collecting computational resources (so called: worker pool) available in different infrastructures. The worker pool can be dynamic, so it can be used for application-level scheduling. The fact that resources in the pool are available without overheads makes the framework suitable to build the DAG scheduling on them. Therefore, our work described in this paper aims at enabling support for DAG-based applications in DIANE framework including interfaces for static and dynamic application scheduling. The project was named DAG4DIANE.

The representation chosen for DAG4DIANE is Condor/DAGMan job description, as this specification for DAG is widely used and it enables assigning used-defined properties to the jobs. The latter is important for some of the application schedulers. DAG4DIANE provides programmer with the interfaces to define advance user-level scheduling algorithms that can be both static (prioritizations running before the application is spawned) and dynamic (done during the process). The dynamic scheduling algorithms are provided with monitoring data that can be useful for the algorithm. All the events that are related to jobs that are running in DIANE are transmitted to the scheduler that can update its status accordingly. This feature makes DAG4DIANE a well-suited tool for development and testing of scheduling algorithms.

Our goal was to minimize effort needed to run a DAG application described by DAGMan script. As a result the only thing the user must do to run workflows is to write simple Python script, based on a delivered template, and to set two fields corresponding to file with DAG/workflow specification, and scheduling algorithm which will be used. As a programming language for module implementation Python was chosen, because DIANE was also implemented in Python. For better usage, DAG4DIANE was developed as a separated DIANE module, which can be easily added after DIANE installation.

For more information please refer to: http://grid.cyfronet.pl/dag4diane

Acknowledgements: We would like to thank Jakub Mościcki from CERN, the DIANE project leader, for his invaluable support.

Refences:

1. E. *Deelman,* D. *Gannon, M. Shields, I. Taylor: Workflows and e-Science: An overview of workflow system features and capabilities, Future Generation Computer Systems, Volume 25, Issue 5, pp. 528-540, 2009.*
2. Condor team, DAGMan: A Directed Acyclic Graph Manager, July 2005. http://www.cs.wisc.edu/condor/dagman/
3. DIANE team, DIANE: Distributed Analysis Environment, http://cern.ch/diane/

24. Grid Resource Bazaar: Efficient SLA Management

Tomasz Szepieniec and Małgorzata Tomanek
ACC CYFRONET AGH, Krakow, Poland

In large grid environment like EGEE/EGI [1] and PL-Grid [2] there are hundreds of actors providing and using computational resources who need some principles in order to satisfy requirements and expectations of both sides. For that reason it seems indispensable to define quality of services, usually called SLAs, between a VO and a resource center. Due to a large number of interactions as well as many details that need to be negotiated for each such contract, a formalized, traceable process of resource allocation is crucial for successful operation. Efficiency of the process should be supported by a collaboration platform assisting resource management by both users and providers.

"Grid Resource Bazaar" is a web portal aiming at facilitating SLA management related to resource allocation in grids. The functionality allows to define and broadcast calls for resources, negotiate and

manage existing SLAs and communication with partners. The tool is maintained by PL-Grid and deployed for operation in Polish NGI, Central European ROC within EGEE Project, and in EGEE seed-resources activity.

For more information please refer to: http://grid.cyfronet.pl/bazaar

References

1. Enabling Grid for E-sciencE, http://www.eu-egee.org/
2. Polish National Grid Initiative – PL-Grid, http://www.plgrid.pl/en

Contributed Presentations

Status and Current Achievements of PL-Grid Project

Jacek Kitowski (1,2) and Lukasz Dutka (2)
(1) Institute of Computer Science AGH, Kraków Poland
(2) *ACC Cyfronet AGH, Kraków, Poland*

PL-Grid, a Polish national grid initiative for supporting computational science in the European research space, which involves scientists from five largest Polish supercomputing and networking centers, was started in 2009 [1-2]. The main goal of the project is to provide scientific communities in Poland with Grid services and environments to enable realization of the e-Science model of research in various scientific fields and making international collaboration easier and more profitable. Elements, like: collaborative research, system-level science going beyond individual phenomena, dynamic organization of scientific teams are addressed. The project is closely related to ongoing European activities coordinated by the EGI organization located in SARA Amsterdam to represent Polish Community as one of the National Grid Initiative organizations.

In the paper the status and current achievements will be outlined. They are related to both – infrastructure and research issues [3]. The focus will be given to the management and advancement of the Operational Center of the PL-Grid, problems of security of operation as well as to identification of application packages potentially available for the grid users. The special interest is given to the promotion of the grid potential within the scientific community and to the users' support in moving to the grid. The advancement of research topics and development of computing and storage infrastructure will be shown.

Acknowledgements. This work is supported by PO IG Project PL-Grid funded by the Polish Ministry of Science and Higher Education (POIG.02.03.00-00-007/08-00). The authors kindly acknowledge contributions of all Partners of the Project. Special thanks go to: M. Turala, K. Wiatr, M. Bubak, T. Szepieniec, M. Radecki, Z. Mosurska, M. Sterzel (ACK Cyfronet AGH), J. Janyszek, B. Balcerek (WCSS Wroclaw), P. Bala, M. Filocha (ICM Warsaw), N. Meyer, K. Kurowski (PCSS Poznan), R. Tylman and J. Rybicki (TASK Gdansk).

References

1. J. Kitowski, Structure and Status of National Grid Initiative in Poland, in: M. Bubak, M. Turala, K. Wiatr (Eds.), Proceedings of Cracow Grid Workshop - CGW'08, October 13-15 2008, ACC-Cyfronet AGH, 2009, Krakow, pp. 39-51.
2. J. Kitowski, PL-Grid: National Grid Initiative in Poland, Structure and Status, HP-Cast Conference, Madrid, May 10-13, 2009, invited talk, http://www.hp.com/techservers/hp-cast/hp_cast_12_Madrid /HP-CAST12_Monday_am/JacekKitowksi-CYFRONE.pdf
3. http://plgrid.pl

Operational Architecture of PL-Grid Project

Marcin Radecki, Tomasz Szepieniec, Łukasz Flis, Małgorzata Krakowian, Małgorzata Tomanek, and Wojciech Ziajka
ACC CYFRONET AGH, Krakow, Poland

The establishment of EGI initiative [3] aimed at providing sustainable grid infrastructure in Europe which will inherit resources from EGEE-III [4] is almost finalized. Dozens of National Grid Initiatives (NGI) will cooperate for sharing their resources and knowledge. However, the NGIs are given an autonomy which is also a chance for improvements in organizational and technical aspects. One of main goals of PL-Grid project [5] is to organize the operational model in more compact and integrated way. We consider three key points to achieve more stable and efficient infrastructure operations: well-

defined, manageable and traceable.

In this paper we assess an EGEE grid operational model and as an outcome we propose some improvements which are focused on attributing users with reliable infrastructure based on SLA with resource providers [1]. This can be achieved by strengthening links between management on policy level and grid fabric administration. Foreseen improvements in grid operations include in particular:

- introducing a resource allocation model supported by a specialized web platform which provides users with manageable process of applying for resources and gives resource providers direct contact to users, which are served in the traceable process according to electronically signed SLAs
- improving reliability of provided services by linking their availability with alerts from infrastructure monitoring
- reduce latency of grid information system [2], more straightforward management of resources being accessible as production and limiting data updated frequently
- more efficient technical support with use of well-defined procedures, knowledge base sharing and instant communication

Specialized tools and organizational changes are necessary to fit proposed improvements into overall operational architecture of PL-Grid. However, proposed solutions have the advantage of being transparent to the rest of the infrastructure and can be easily deployed in other NGIs.

References

1. T. Szepieniec, A. Pagacz: SLA Negotiation and Enabling Resources for Users in Grids; CGW'08, Proceedings of CGW'08, Kraków, March 2009
2. M. Radecki, W.Ziajka: Improvements of Grid Information Service in EGEE; CGW'08, Proceedings of CGW'08, Kraków, March 2009
3. EGI project web page: http://web.eu-egi.eu/
4. EGEE project web page: http://www.eu-egee.org/
5. PL-Grid project web page: www.plgrid.pl/en

Extension of the Social Grid Agents Functional Engine for Greater Information Model Interoperability

Gabriele Pierantoni (1), Keith Rochford(2), Brian Coghlan (1), Eamonn Kenny (1)
1) Trinity College Dublin, Dublin, Ireland
2) Dublin Institute for Advanced Studies, Dublin, Ireland

The increasing variety of computational resources available to research and industry present not only significant opportunities, but also a number of new challenges. These challenges include the interoperation of the underlying resources, the lack of common user interfaces and the difficulties in brokering access to the resources from diverse user communities linked by differing relationships.

Social Grid Agents are a socially-inspired solution designed to address the problem of resource allocation in Grid Computing. Thanks to ongoing research, Social Grid Agents have emerged as a viable solution to alleviating some of the problems associated with interoperability and utilization of diverse computational resources and to modeling the large variety of relationships among the different actors.

The Agents provide an abstraction layer between resource providers and consumers. This layer itself employs a two-layer architecture; the social layer where the Agents engage in social and economic relations and a production layer where Grid services are composed and performed.

At the core of the Social Grid Agent architecture is its own native language based on the functional language, ClassAd. This language is processed by a functional engine present in every agent and is

used to describe actions, objects, their status and the policies that govern their behavior. Significant characteristics of Social Grid Agents include their agnosticism of the grid middleware technologies and their lack of fixed economic precept.

Despite this level of abstraction from the underlying infrastructures, Social Grid Agents are very closely tied to the functional nature of the language that governs their operation. Further, this constraint extends to the very precise and unique information model utilized in which all information is conveyed and distributed through non-standard messages exchanged by agents.

We speculate that it would be profitable to extend the Social Grid Agents architecture giving it the same degree of agnosticism with regard to the differing information models. On the other hand, the very nature of Social Grid Agents is too tightly linked to their functional language to allow for a complete re-design of their core structure. Accordingly, we propose an extension to the functional engine of the Social Grid Agents that is capable of a certain degree of flexibility in interfacing with both producers and consumers of information.

This work presents an extension to the architecture of the Social Grid Agents that provides a unified event-based information system, based on the standard provided by the WS-Notifications specification. We present some preliminary results of this extension of the Social Grid Agents information system for the tracking of job submissions.

Quality Assurance with ETICS – Multi-Node Automated Tests

Marian Żurek (1,2), Andres Abad Rodriguez (1,2), Alberto Aimar (1,2), Alberto di Meglio(1,2), Lorenzo Dini (1,2)
(1) European Organization for Nuclear Research, Geneva, Switzerland
(2) ETICS Project, http://eticsproject.eu

Assuring the success and sustainability of short-living projects in the era of software development being done in multi-national and geographically spread teams requires a systematic approach and frequent testing not to mention a common collaborative platform. The ETICS Project in its first phase provided developers, integrators and release managers with a collaboration platform and appropriate toolkits to make their work easier. In its follow-up phase ETICS 2 provides integrators and especially tester teams with multi-node automated testing facilities. Automating the process of complex, multi-node deployment largely reduces time spent between software development and its deployment in production.

Testing activities can start from simpler one node test scenarios where tested software and test suite run on the same machine. More advanced cases involve dynamic automated deployment of some nodes and tests to run against statically deployed external resources. The final goal is to provide automated deployment and configuration of complete testing environments meeting the requirements of today's software operational infrastructures, enabling integrators, release managers and testers with the possibility of deploying self-contained grid infrastructures in an automated, controlled and deterministic way. ETICS facilitates the whole process providing an infrastructure for scheduled and on-demand building and testing of software. All produced artifacts like build and test reports, packages in various formats (tar.gz, RPM, deb, etc) and QA metrics are stored in the ETICS repository.

The validation and testing becomes a vital activity especially when interoperability of different software stacks becomes a requirement. The testing features offered by ETICS are an ideal solution as they offer the full automation and reproducibility of tests scenarios. Additionally, the quality model integrated in ETICS (A-QCM) makes ETICS a complete suite to support the development of high level quality software which is a must when aiming at a widespread adoption of grid and distributed software by increasing trust between software providers and users.

The results obtained by projects using ETICS, such as the DILIGENT and D4Science Projects [1] clearly demonstrate that the time invested into automation of the deployment and testing scenarios brings immediate benefits in terms of resource utilization and quality provided. It would be not fair to say that the automation comes at no cost. Effort is always needed to make sure that different stages

like software packaging, installation, configuration and finally testing can be implemented without human intervention and be fully automatable. However, such a continuous testing process enables developers, integrators and tester to quickly identify newly introduced bugs, backwards compatibility violations, and interoperability issues before they become expensive to fix, improving the whole quality and cost-effectiveness of the software.

Acknowledgements: this work was supported by EU project ETICS Phase 2, INFSO-RI-223782.

References

1. Diligent - A Digital Library Infrastracture on Grid ENabled Technology, http://diligentproject.org
2. Di Meglio, Alberto; et al. (2008). "ETICS: the International Software Engineering Service for the Grid". Journal of Physics: Conference Series 119. doi:10.1088/1742-6596/119/4/042010.
3. Di Meglio, Alberto; et al. (2004). "A Pattern-Based Continuous Integration Framework for distributed egee grid middleware developent". Computing in High Energy and Nuclear Physics (CHEP'04): 579.
4. Bégin, Marc-Elian; et al. (2007). "Build, Configuration, Integration and Testing Tools for Large Software Projects: ETICS". Lecture Notes in Computer Science (Berlin/Heidelberg: Springer) 4401: 81-97. doi:10.1007/978-3-540-71876-5_6.

Extensions for the ETICS Client for Multi-Platform Support in NGIs Driven by User Requirements

Eamonn Kenny, Peter Lavin, Brian Coghlan, John Walsh
Trinity College Dublin, Ireland

In the transition towards the European Grid Initiative (EGI) infrastructure throughout Europe, the need for local sites and consortia to be able to locally build multiple middlewares on multiple non-reference platforms is perceived to be of great importance. This work must be driven by the needs of the user community across many fields and disciplines. In addition, the need for interoperability between multiple middlewares requires that such work be successful, but again must be driven by specific user requirements. Ideally all such works should be fed back to the EGI for integration and future sustainability.

Two years ago, in the EU FP7 EGEE-II project, Trinity College Dublin produced a primitive Python/XML wrapping[1] for the ETICS client allowing local sites to patch checkouts of the gLite middleware source code, producing builds that remain well ahead of the EGEE centrally managed ETICS infrastructure. The wrapping is now in a more modular, advanced state and is therefore worthy of a more detailed description.

The ETICS client as of August 2009, is functioning very well on most Linux platforms (RedHat/SL/CentOS, Debian/Ubuntu, openSUSE), Mac OS X (10.4/10.5), IBM AIX 5.3 and the PlayStation 3 (PPC32), making it the client of choice for building many Grid middlewares. However, because most build system infrastructures focus primarily on a homogeneous development cycle, it is difficult for a group of National Grid Initiatives (NGIs) or non-EGEE consortia to sustainably build their applications for even one EGI grid middleware on even one single new platform.

Trinity College Dublin has developed many modular wrappings for changing and analyzing the grid middleware development on new platforms, greatly speeding up the effort in producing sustainable builds. These consist of a Python/XML/GNU-patch system, a Python wrapper for the ETICS client to allow it to work on new platforms, an object-oriented build infrastructure written in Python that allows cron scheduled builds on each new platform, a locally managed webserver of artifacts and build reports used to compare results with the centralized build results, an intermediate monitoring tool to assess the progress of a local build while its still taking place, a HTML/XML results page auto-generated by Python scripts on the webserver and auto-generated timestamped patches for easy upload to a centralised bug tracking system.

This local infrastructure has proven itself to be highly effective. We present as a case study the build of a compute node (glite-WN) for the PlayStation 3 (PS3). This work was driven by the need to provide a mechanism to run the electronic high-throughput drug screening software (eHiTS) of Simbiosys Inc.[2] across the Grid. eHiTS allows users to fully utilise the Cell Broadband engine of the PS3, resulting in large computational speedups. Building a glite-WN on the PS3 allows applications previously not available to the Grid to be integrated, broadening the user community. Many application areas are currently left untouched because the Grid middleware has not been ported to a sufficiently wide range of architectures. By using the local build infrastructure described here, NGIs or consortia will have the ability to build for the middleware and platform of choice, and contribute the results back to EGI, increasing the user community and extending the EGI focus well beyond the current set of applications.

References

1. Kenny, Eamonn and Coghlan, Brian and Walsh, John and Childs, Stephen and O'Callaghan, David and Quigley, Geoff, Extensions to the ETICS build system client to allow porting to multiple platforms at local sites, CGW 2007 proceedings, Crackow Grid Workshop, 2007, Krakow, Poland, pp437 – 444., 2008.
2. Kerwin, S. M.: Computer Software Review of: eHiTS 5.1.6, SimBioSys Inc.; J. Am. Chem. Soc, June 2005.

Large-Scale Cross-Cluster Parallel Simulation Using QosCosGrid

Tomasz Piontek (1), Krzysztof Kurowski (1), Mariusz Mamoński (1), Piotr Kopta (1)
(1) Poznan Supercomputing and Networking Center, Poznań, Poland

The main aim of the “Quasi Opportunistic Supercomputing for Complex Systems in Grid” [1] project was to bring supercomputer-like performance and structure to cross-cluster computations. Tools and services developed in the QosCosGrid project were designed to build multi-layer infrastructure being capable of dealing with computationally intensive simulations of Complex Systems, defined as systems with many interdependent parts which give rise to non-linear and emergent properties determining the high-level functioning and behavior of such systems. The QosCosGrid middleware enables computing clusters in different administrative domains to be welded (virtually) into a single powerful compute resource that can be treated as a quasi-opportunistic supercomputer. The core of QosCosGrid middleware stack consists of two kinds of services: the grid level meta-scheduler called GRMS with pluggable scheduling algorithms and a set of administrative domain level SMOA services, providing consistent, remote, multi-user access to various queuing systems such as Platform LSF, SGE or PBSPro and Maui. Using standardized OGF DRMAA and HPC-Profile interfaces extended with Advance Reservation capabilities from advanced queuing systems users are able to create on-demand cross-cluster parallel execution environments using well know programming tools, namely OpenMPI and ProActive [2]. To the best of our knowledge, QosCosGrid offers the most efficient and powerful multi-user access to job management and co-scheduling features comparing to other existing grid middleware services. The QosCosGrid middleware and tools have been successfully deployed in some productive HPC environments (e.g. INRIA, UPF, Dortmund University, PSNC, etc.) extending functionalities of infrastructures services based on UNICORE or gLite. Currently the QosCosGrid middleware is being enhanced and tested under the national Polish Grid Infrastructure (PL-Grid) project.

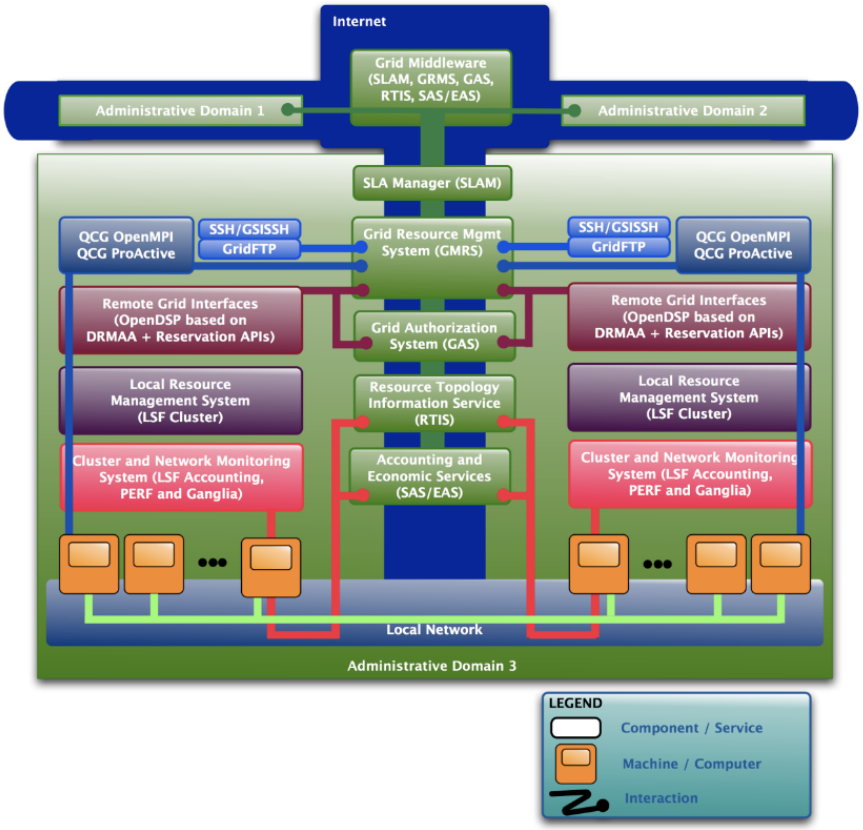


Figure 1. QosCosGrid Middleware architecture

References

1. Kravtsov, V., Schuster, A., Carmeli, D., Kurowski, K. Dubitzky, W. (2008). Grid-enabling complex system applications with QosCosGrid: An architectural perspective, in Proc. of The Int'l Conference on Grid Computing and Applications (GCA'08), Las-Vegas, USA.
2. Kurowski, K., Back, W., Dubitzky, W., Gulyas, L, Kampis, G., Mamonski, M., Szemes, G., Swain, M. (2009). Complex System Simulations with QosCosGrid, In Proc of 9th International Conference on Computational Science (ICCS 2009), pp. 387-396, Baton Rouge, USA

Tuesday, Oct 13, 2009

ViroLab Overview and Demonstrations

Virolab HIV Decision Support: From Molecule to Man

P.M.A. Sloot
University of Amsterdam, The Netherlands

The Virolab Virtual Laboratory

M. Bubak, T. Gubala, M. Malawski, B. Balis, K. Prymula

VPE Virtual Patient Experiment Demonstration

Andrea De Luca
Catholic University, Rome, Italy

Contributed Presentations

VL-e eScience Farmework: from Design to Sharing

Adam Belloum, Zhiming Zhao, Vladimir Korkhov, and Marian Bubak
SCS, Informatics Institute, Universiteit van Amsterdam, The Netherlands

Modeling and simulation is an important approach to understand the none-linear dynamics in the complex problems such as social phenomena, changes in stock markets and biological morphology of infection diseases. Such kind of experiments computing intensive and require integration between different technologies, because 1) refining parameters of a complex model require massive computing tasks to compute different value sets, 2) improvement in the model resolution or inclusion of new parameters result in dramatic increasing of computing tasks, 3) analyzing simulation outputs require data processing and visualization tools, which should be able to handle large volume data sets, and 4) crafting a simulation engine into real world application such as for decision support and design validation requires further support for human interaction and for integration data with other information sources. Computing technologies are playing a crucial role.

The Dutch Virtual laboratory for e-Science (www.vl-e.nl) project aims at developing new methods and framework to support highly complex scientific experiments in various scientific domains. This framework will cover all the phases of a complex scientific starting from design, and ending with the publication and sharing of the outcome of these experiments. The design is driven by a number of real application use cases, with different characteristics covering both CPU and data intensive application, such as the ADDA Distributed Light Scattering [1], the Chromosome location in the genome [2], or Affymetrix Permutation-based Probe Level Estimation [3]. The framework allows seamless access to geographically distributed resources to fulfill the computational and storage requirements and also the sharing of the knowledge and expertise among the scientists as well as a gradual design of these complex experiments by allowing reuse of generic tools developed within the scientific community.

Grid Technology brings the power of many computers and storage systems geographically distributed to scientists. However, the development of Grid-enabled applications requires knowledge about Grid infrastructure and low-level API to Grid services. A high-level environment for rapid prototyping of experimental computing systems is thus needed to fill the gap between the required complex infrastructure and the various scientific domains. Coupling Grid and a high level Problem-solving environment is important for the scientific community: it makes the power of the Grid easily available to the end user. The VL-e framework aims at generic functionalities that support a wide class of specific complex scientific application environments and set up an experimental infrastructure for the evaluation of the ideas. A set of tools is composing this framework, and together they provide full support for the design and execution of scientific experiment over a complex and distributed infrastructure.

Scientists have access to the knowledge and expertise developed within the scientific community which is a valuable assets that can rarely be thrown away, and often contain within them great value to the design of new complex application. This is obtained by an environment in which scientist interactively access resources of various types to manipulate data (upload, download, search, annotate, and view), start applications (prepare and execute experiments) and monitor resources (status, control, notification). This support is done via tools that support **Templates and components manage,** this include tools a mapping the template description to an underlying data presentation, tools for semantically annotating software components, tools for searching and discovering results and components shared within the Virtual Organization, and tools for browsing the available resources**.**

The Runtime support is the next step required by scientists, this include the staging of the components composing the experiment, the efficient execution on the available computing resources, and the monitoring of the progress of the experiments. Often Interactive control of the execution is required to allow scientist to act at runtime and modify some parameters. Workflow tools, allow coordinating the execution of distributed simulations and scientific experiments as well as monitoring and interactive runtime control of the application workflows, and task farming, parameter sweep, access to Web services and Workflow aggregation using the recent achievements in Grid and web technologies.

Following is the list of tools composing the VL-e framework: for modeling and managing workflow templates, for browsing resources stored a workflow tool for composing workflows and for managing components, and a framework for coupling VL-e workflows with other legacy workflows.

Templates and components manage tools [4] include Olingo, a mapping tool between the workflow template description and underlying data presentation, CLAMP, an annotation for VL-e components, and Hammer, a storing and query tools for components.

Virtual resource browser: VBrowser [5] offers scientists an environment in which they interactively access resources of various types to manipulate data (upload, download, search, annotate, and view), start applications (prepare and execute experiments) and monitor resources (status, control, notification).

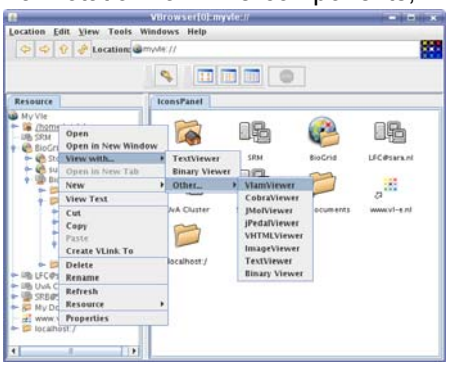


Figure 1: Virtual browser Graphical interface

The VL-e workflow system: WS-VLAM system [6] has a set of client-side applications that allow scientists to design and monitor the execution of the workflows with intuitive interfaces, and provides also server-side applications, including a `workflow engine' that schedules and executes the workflow on the Grid.

Interactive parameter sweep: FRIPS [7] aims to support interactive execution of applications that require parameter sweep. It allows scientists to monitor the experiment execution, view intermediate results, and give interactive feedback on a running experiment.

Workflow aggregation: VLE-WFBus [8] provides tools to recognize different workflow descriptions stored in the system and interface to wrap and integrate legacy scientific workflows.

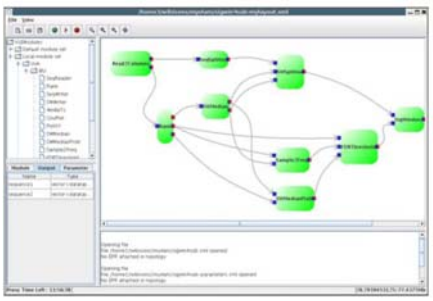


Figure 2: worflow composer allow intuive workflow composition using a simple drag-and-drop mechanism

Via the tool, a user can execute a workflow via the workflow bus, and integrate it with the other workflows via different connectors.

References

1. M. A. Yurkin, K. A. Semyanov, V. P. Maltsev, and A. G. Hoekstra. "Discrimination of granulocyte subtypes from light scattering: theoretical analysis using a granulated sphere model", Opt. Express 15, 16561-16580 (2007).
2. M.A Inda, M.F van Batenburg, M. Roos, A.S.Z Belloum, D. Vasunin, A. Wibisono, A.H.C van Kampen and T.M Breit, SigWin-detector: a Grid-enabled workflow for discovering enriched windows of genomic features related to DNA sequences BMC Research Notes 2008, 1:63
3. WimCde Leeuw, HanRauwerda, MartijsJJonker and TimoMBreit, Salvaging Affymetrix probes after probe-level re-annotation BMC Research Notes 2008, 1:66 doi:10.1186/1756-0500-1-66
4. Víctor Guevara-Masís, Konstantinos Krommydas, A. Belloum, L.O. Hertzberger, Semantic Workflow Discovery in VL-e, Proceedings of KnowledgeGrid 2006, IST 2006, Strategies for Leadership, Helsinki, Finland
5. T. Glatard, K. Boulebiar, P. de Boer, S. Olabarriaga, fMRI analysis on EGEE with the VBrowser and MOTEUR, 3rd EGEE User Forum, Feb 2008
6. A. Wibisono, Z. Zhao, A. Belloum, M. Bubak: A Framework for Interactive Parameter Sweep Applications. CCGRID 2008: 703
7. A. Wibisono, V. Korkhov, D. Vasunin, V. Guevara-Masis, A. Belloum, C. de Laat, P. Adriaans and L.O. Hertzberger, WS-VLAM: Towards a scalable workflow system on the Grid, Proceeding of the 16th IEEE International Symposium on High Performance Distributed Computing, June 27-29, 2007, Monterey Bay, California, USA
8. Z. Zhao, S. Booms, A. Belloum, C. de Laat and L.O. Hertzberger, VLE-WFBus: A Scientific Workflow Bus for Multi e-Science Domains, 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam, 2006, Proceedings, p. 11

Globus Plug-in for WINGS Wokflow Engine

Carlos de Alfonso (1), Miguel Caballer (1), Vicente Hernandez (1), Elizabeth Martí (1).
(1) Grid and High Performance Computing Research Group, Instituto ITACA - Universidad Politécnica de Valencia

In a previous work [1] we have described WINGS, which is a workflow engine focused on multi-grid capabilities and easy extensibility. It is based on four main concepts: **(1) Data Sources** act so as sources as sinks for data. They are used as points for interchange of data among the different executions in the workflow. **(2) Activities** are the abstractions of tasks to be run on the Grid. They describe the functionality of the tasks that will finally be executed by the run-time system. **(3) Executions** are specific instances of an activity. They represent the task which would be actually launched and executed in the Grid. **(4) Operations** which are simple executions that will be carried out by the workflow runtime, in order to pre or post process the information used in the executions.
WINGS engine has been developed using a modular architecture. Every element is implemented as a plug-in and so new elements can be added without the need of modifying the other parts of the system.
In the first version of the workflow engine, the Fura [4] middleware plug-in was developed to test the system functionality. In the second version, a Globus Toolkit plug-in has been developed to enable multi-grid tests. Globus has been implemented due to the great number of current infrastructures that use it as the underlying grid middleware (EGEE, EELA, etc.), thus enabling to access an important amount of resources. In this paper the Globus Toolkit (pre-ws) plug-in is described, focusing on the generalisation of its concepts to be interoperable in the WINGs multi-grid architecture.
The core engine of the workflow system performs the logic and control operations in order to prepare and select the tasks ready to be launched and the data to use in each execution. The plug-ins are in charge of effectively perform the file transferences and all the needed operations to complete the execution. This paper summarizes the most important operations made by the plug-in to execute the grid tasks.

The first step is to prepare the activities defined in the XML file, which will be later executed in the Workflow: (1) Obtaining the user credentials, and (2) Creating working and executing directories, to store all the input data and the executable files needed for the execution, respectively.
Once the activity is prepared, the tasks are ready to manage the input data for the execution from the proper data sources. The plug-in must copy the input files in the executing resource and create all the possible combinations of the input data. Each combination of inputs involves a task execution, which is called "microtask". In the former plug-in, the Fura middleware was able to iterate over the input data to create the combinations of inputs and the microtasks by itself, but Globus is not provided with such tools and the plug-in is responsible of making the input data combinations.
At this point, we have all the needed information for the execution. The next step is to set the attributes for each job to run, and control the execution. At the end of the executions, the plug-in has to collect the output data information. For each microtask, we have an output directory where the results are stored after the execution.
In order to demonstrate the functionality of the system, a use case is shown, using a combined testbed with Fura and Globus servers. The test case has been used to show the performance and the multi-grid capabilities of the workflow system using a mixed Globus and Fura testbed.
Finally, some conclusions are described summarizing the exposed work and the results obtained in the performed tests. Also many future works are being studied as adding new plug-ins to other execution environments as PBS or SGE and grid meta-schedulers as GridWay [2], GMarte [5] or gLite WMS [3].

References

1. Carlos de Alfonso, Miguel Caballer and Vicente Hernández: WINGS: A Multigrid Wokflow Engine. Proceedings of CGW'08, Cracow, October 2008. pp. 129 – 136.
2. Eduardo Huedo, Rubén S. Montero and Ignacio M. Llorente. "The GridWay Framework for Adaptive Scheduling and Execution on Grids". Scalable Computing - Practice and Experience 6 (3): 1-8, 2005
3. EGEE, gLite Lightweight Middleware for Grid Computing. http://www.glite.org, 2006.
4. GridSystems, 2006. Fura. http://fura.sourceforge.net/
5. J. M. Alonso, V. Hernández, G. Moltó. GMarte: Grid middleware to abstract remote task execution. Concurrency and Computation: Practice and Experience. Volume 18 Issue 15, Pages 2021 – 2036

Secure Grid Workflow Repository for ASKALON

Malik Muhamad Junaid, Maximilian Berger, Thomas Fahringer
Institute of Computer Science, University of Innsbruck, Austria

Workflows play a pivotal role in distributed computing environments. Scientists are creating more sophisticated workflows for complex scientific applications everyday. The increase in complexity of the applications being run on the distributed computing environments like Grids is resulting in more complex workflows.
Designing a workflow from the scratch is a hard job, it demands a lot of time a effort from the domain scientists. Therefore it is a good idea to reuse the existing workflows to save time and effort.
Most of the grid development environments being used today mostly focus on the creation and execution of workflows. The storage of workflows in such a manner that they can be easily shared among different domain experts is mostly overlooked. In general workflow storage is performed using standard XML based document and are stored on the filesystem.
Scientists working with these workflows manually keep a version record of the workflow and also there is no specific mechanism to protect the workflows against unauthorized access or modifications.
Addressing the workflow security and reuse problem a workflow hosting environments [1] is presented.

We extend the idea of workflow hosting environment presented in [1] into a complete workflow hosting and management system. In contrast to the existing systems the current implementation of the workflow repository (SWFR) provides a completely secure mechanism of workflow retrieval and storage. The system uses extended role based access control (E-RBAC) for authorizing users and authenticating their access to different workflows. Also the system provides automatic version

management system which monitors the workflow design sessions and updates the workflow version automatically or based on user requests.
There are three main modules of the SWFR, the workflow storage and retrieval module, the authentication and authorization module and the version management module. The three components are implemented in service oriented architecture and together serve as a complete and secure workflow repository.
To ensure the security of the workflows in the repository, and for a systematic workflow access users and roles are created. Users can have normal or administrator access rights. New users can register to be able to use the workflows. The repository administrator allows or denies the registration requests, and sets access rights for the user. Once a user is registered he can access the repository to store the workflows he creates, or, to access other workflows based on her access rights. The access mechanism in SWFR is role based as specified in [2], for the RBAC system roles are defined with certain predefined rights.
We implemented an extended RBAC version for access control, where a user can have one or more roles and also exceptional rights, exceptional rights are granted by owners of specific accessible entities. For example a workflow owner can grant the right to modify his workflows to a user who has a readonly access role for all other workflows. Some other systems for restricting access to workflows have also been implemented for example using the policy based access control [3], but RBAC system is faster and more efficient as compared to the policy driven access control system, which is further enhanced by implementing the extended RBAC with exceptional rights matrix for the access control.

The SWFR is implemented for ASKALON [4], ASKALON workflows are represented using AGWL[5], an XML based workflow description language. The workflow storage component of the SWFR is implemented as an RDBMS database, where workflows as stored along with their versioning information. Modifications to the workflows are stored in incremental fashion, and only the changes occurring to specific workflows are stored. This incremental storage strategy keeps the database from getting redundantly populated and makes it more efficient to store and retrieve required data. The system has been implemented and integrated into the ASKALON.
We have performed various comparison experiments for testing the performance of the system. As already discussed the whole SWFR system is implemented in a service oriented architecture, so it is easy to deploy the system in a real grid environment and invoke its different functions via service calls. We tested the system in the Austrian Grid[6] by deploying the repository service in five different grid sites available in Austria. The overall performance comparison with that of the filesystem based storage have seen a very minor degrade in the efficiency which is almost negligible. In ideal network situation and with moderate site load the system gave 80% performance as compared to the filesystem based approach. We assume this much throughput to be reasonably acceptable because of the added advantage of security, version management and possible of distributed deployment of the service. The system can further be optimized by creating replica storage locations on busier grid sites to serve the requests more efficiently and in the real-time.
In the future we plan to further extend the system by providing distributed storage repositories and automatic synchronization of the replica databases. Moreover the workflow repository will be integrated to the provenance recording system to enable the design time provenance system to take benefit of the different versioning information of the workflows and associated users.

References

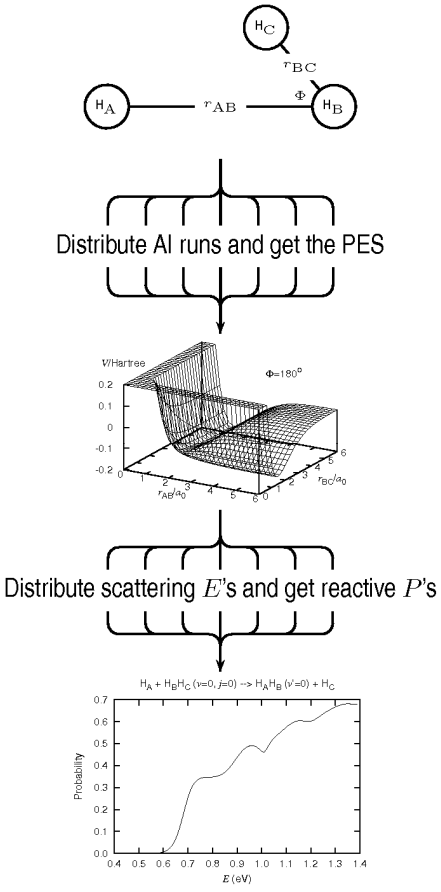
1. Jun Qin, Thomas Fahringer, and Maximilian Berger, Towards Workflow Sharing and Reuse in the ASKALON Grid Environment. In Proceedings of the *Cracow Grid Workshop 2008*, Octover 13 - 15, 2008, Cracow, Poland, ISBN: 978-83-61433-00-2, Page 111-119.
2. Hui Zhao; Zhiyi Fang; Peng Xu; Lianyu Zhao; Jin Liu; Tianyang Wang An Improved Role-Based Workflow Access Control Model, Information Technology: New Generations, 2008. ITNG 2008. Volume , Issue , 7-9 April 2008 Page(s):551 – 556
3. Long, D.L.; Baker, J.; Fung, F. A prototype secure workflow server Computer Security Applications Conference, 1999. (ACSAC apos;99) Proceedings. 15th Annual Volume , Issue , 1999 Page(s):129 – 133
4. Fahringer, T., Jugravu, A., Pllana, S., Prodan, R., Seragiotto, C., and Truong, H. 2005. ASKALON: a tool set for cluster and Grid computing: Research Articles. Concurr. Comput. : Pract. Exper. 17, 2-4 (Feb. 2005), 143-169. DOI= http://dx.doi.org/10.1002/cpe.v17:2/4

5. Thomas Fahringer, Jun Qin and Stefan Hainzer, Specification of Grid Workflow Applications with AGWL: An Abstract Grid Workflow Language, IEEE International Symposium on Cluster Computing and the Grid 2005 (CCGrid 2005), Cardiff, UK, May 9-12, 2005.
6. http://www.austriangrid.at/

A Priori Modeling of Chemical Reactions on a Grid-Based Virtual Laboratory

Sergio Rampino[1], Antonio Monari[2], Stefano Evangelisti[3], Elda Rossi[4], Gian Luigi Bendazzoli[2], Antonio Laganà[1]
(1) Dipartimento di Chimica, Università di Perugia, Italia
(2) Dipartimento di Chimica Fisica e Inorganica, Università di Bologna, Italia
(3) Laboratoire de Chemie et de Physique Quantiques, Université Paul Sabatier Toulouse III, France
(4) CINECA, Bologna, Italia

Progress in the capability of simulating chemical processes on a molecular basis is an important component of the advance in modeling natural phenomena, designing new materials and products, mastering new technologies and carrying out innovative experiments. Such progress typically requires the assembling of various pieces of software, the convergence of the competences of different experts, the concurrence of the elaboration on several processors. The difficulty of gathering in the same place all the necessary hardware, software and human resources, makes computational grid platforms the ideal environment for the exploitation of collaborative computing and interoperability.



The past few years have seen significant progress in the grid-porting of scientific applications related to different chemical scenarios by the COMPCHEM Virtual Organization [1] on the EGEE grid [2]. Moreover, the problem of interoperability among a large class of codes pertaining to the Quantum Chemistry (QC) domain singled out the need for the definition of common data formats, such as the Q5Cost [3].
Moving from these advances, we have assembled on the COMPCHEM segment of EGEE the core of the Grid Empowered Molecular Simulator (GEMS) queueing in a common workflow a suite of codes for the a priori modeling of elementary processes. As shown in the figure for the H + H_2 prototype system, the Bologna FCI code [4] integrated with a QC code (DALTON [5], in our case) via the Q5Cost format evaluates the interaction of the three nuclei at several geometries concurrently and a fitting routine (GFIT3C [6]) provides the Quantum Dynamics (QD) code ABC [7] with a potential energy surface (PES) over which the dynamics are run at several scattering energies and the reactive properties extracted.
Such computational machinery - whose structure is here presented paying particular care to the issue of the data interchange - is the core engine of the nextcoming global simulator GEMS [8], meant to deal with much more complex processes thanks to the modern paradigm of high throughput computing – the Grid.

Acknowledgments. Financial support from the project EGEE III is acknowledged. The present work has been carried out as a joint activity of the working groups DeciQ and QDYN of the COST CMST European Cooperative Project CHEMGRID (Action D37).

References

1. A Laganà, A Riganelli, O Gervasi, On the structuring of the computational chemistry Virtual Organization COMPCHEM, Lect Notes Comp Sci 3980, 665-674 (2006)
2. Enabling Grids for E-Science in Europe, http://eu-egee.org
3. C Angeli, G L Bendazzoli, S Borini, R Cimiraglia, A Emeerson, S Evangelisti, D Maynau, A Monari, E Rossi, J Sanchez-Marin, P G Szalay, A Tajti, The problem of interoperability: a common data format for Quantum Chemistry codes, Int J Quantum Chem 107, 2082-2091 (2007)
4. G L Bendazzoli, S Evangelisti, A vector and parallel full configuration interaction algorithm, J Chem Phys 98, 3141-3150 (1993)
5. DALTON, a molecular electronic structure program, Release 2.0 (2005), see http://www.kjemi.uio.no/software/dalton/dalton.html
6. A Aguado, C Tablero, M Paniagua, Global fit of ab initio potential energy surfaces I. Triatomic systems, Comp Phys Comm 108, 259-266 (1998)
7. D Skouteris, J F Castillo, D E Manolopoulos, ABC: a quantum reactive scattering program, Comp Phys Comm 133, 128-135 (2000)
8. A Laganà, Towards a grid based universal molecular simulator, in A Laganà, G Lendvay Eds "Theory of the dynamics of elementary chemical reactions", 363-380 (Kluwer, 2004)

Molecular Distributed Computing on the Grid: Quantum Reactive Scattering and Visualization Tools

Alessandro Costantini (1), Osvaldo Gervasi (2), Antonio Laganà (1)
(1) Dept. of chemistry, University of Perugia, Perugia, Italy
(2) Dept. of Math. And Comp. Science, University of Perugia, Perugia,Italy

The increasing availability of computer power on Grid platforms has prompted the implementation of complex computational chemistry suites of codes on distributed systems and, at the same time, the development of appropriate graphical tools for the visual rendering of the outcomes of the calculations. This is indeed the goal of the virtual organization (VO) COMPCHEM [1] that has been assembled by a group of molecular and material sciences laboratories committed to implement their computer codes on the section of the production EGEE Grid [2] infrastructure available to the VO. Moreover the implementation of grid empowered versions of quantum reactive scattering codes dealing with atom-diatom systems and of advanced visualization tools and virtual world instruments devoted to the study of the behavior of complex molecular systems is the joint task of the QDYN [3] and ELAMS [4] working groups of the COST Action D37.

To this end the quantum mechanical atom-diatom reactive scattering program called ABC [5] (that carries out accurate calculations of the quantum S matrix elements to evaluate reaction probabilities as well as state-to-state integral and differential cross sections) has been ported to the EGEE Grid environment using P-GRADE Grid Portal and some visualization tools have been developed [6].

PGRADE (release 2.7) [7], that has been implemented as a COMPCHEM facility, provides graphical tools and services supporting grid application developers in porting legacy programs onto grids without reengineering or modifying the code.

The ABC code used for porting has significant CPU demand. Depending on the input parameters, one fixed energy execution of the ABC code can take about 10 hours on a single PC. Moreover, the ABC code is rarely used for a single set of parameters. In a typical use case the ABC program must be executed several times for different sets of input parameter, consuming a large amount of hours of CPU time. The visualization tool targeted to analyze the ABC output files that has been implemented on the P-GRADE system like a web portal consists in a set of java based graphical tools implemented as standard pluggable user interface components (Portlets). The Portlet developed for ABC is made of three stand alone applications which enable the final user to visualize, in different formats, the output produced by the calculations.

The applications having the form of collapsible windows enable the user to:

- visualize in a text fashion the selected job displayed in a TextAreaBox;
- visualize the 2D-Graph render of the Cumulative Reaction Probabilities;
- visualize the 2D-Graph render of the State To State Reaction Probabilities.

The final user can perform in a user friendly way the implementation of a distributed concurrent version of the code, analyze and compare the output carried out for a selected atom-diatom reaction (with no need to download all the output files which remain on the server) and evaluate the possible strategies for a new calculation.

Acknowledgements. Acknowledgements are due to the financial support by COST CMST Action D37 GRIDCHEM, through the activities of QDYN and ELAMS working groups, and the EGEE III EU Project, an FP7 project funded by the European Commission under contract IST-2003-508833.

References

1. COMPCHEM website: http://compchem.unipg.it
2. EGEE (Enabling Grids for E-Science in Europe): http://public.eu-egee.org
3. QDYN is the working group n. 2 of the CMST COST ActionD37, called GridChem: http://www.cost.esf.org/index.php?id=189&actionnumber=D37
4. ELAMS is the working group n. 3 of the CMST COST Action D37, called GridChem: http://www.cost.esf.org/index.php?id=189&actionnumber=D37
5. Dimitrios Skouteris, J.F. Castillo, D.E. Manolopulos: ABC: a quantum reactive scattering program; Comp. Phys. Comm., 133, pp. 128-135, 2000
6. D. Skouteris, A. Costantini, A. Laganà, G. Sipos, A. Balasko, and P. Kacsuk: Implementation of the ABC Quantum Mechanical Reactive Scattering Program on the EGEE Grid Platform. Lecture Notes in Computer Science, 5072, pp. 1108-1120, 2008
7. P-GRADE Grid Portal: http://portal.p-grade.hu

A Test Infrastructure for Inspecting the Availability of Grid Resources

Jie Tao and Mathias Stümpert
Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Germany

The main goal of a computing grid is to provide resources and associated services. Therefore, the number of CPUs (or cores) and the amount of storage are commonly included in an SLA. However, quantity is only one of the criteria that evaluate a resource centre. Availability of the resources, for example, is another metric and this metric is even more important for the user community.

The monitoring approach has been widely applied in the grid world to inspect the resource and job execution [1, 2, 3]. This approach can be extended to monitor the availability of resources and services. Nevertheless, testing is a better means for this work. An example is the EGEE project which deploys the SAM [4] system to execute sample tests every hour on resource centres.

Similarly, we also developed a test framework for verifying the functionality of grid resource centres. Besides the simple “ping” tests commonly used to examine whether a machine is reachable, we perform real testing which models potential actions of grid users. For example, the functionality of a computing element is validated by first submitting jobs to the CE, then tracing the job status, and finally downloading the execution results. Storage elements are tested by both conducting file operations, like copy, delete, modify, and rename, on a single SE as well as by moving data across the storage elements. Depending on the functionality, services are tested using different ways. An SRM service, for instance, is verified with data operations, while the information system is regarded as available if it can deliver correct information about a resource centre.

The test infrastructure is implemented using the g-Eclipse [5] APIs. Actually, g-Eclipse aims at providing an intuitive, middleware-independent graphical interface for users to access the grid infrastructure. It is developed on top of Eclipse, in the form of plug-ins. However, the implemented Java classes have been applied by programmers to issue grid operations within a program code. In this work, we use the g-Eclipse implementation for authorization/authentication, data management, and job management etc. to perform real grid accesses.

The tests are written in the form of a unit test. These tests can be manually started on the Eclipse platform and in this case the results are depicted on-line on the same platform. Additionally, we also integrated the test infrastructure in the g-Eclipse nightly build system, where all tests are automatically executed at every night and the results are stored and can be retrieved using a web browser. The

testing covers all resources and services of a computing grid. Here, the site information is acquired using the information system.

Overall, we have built a test infrastructure which can be used by grid administrators to verify the functionality of resources and services of grid centres and thereby to remove the problems before they bring the user any burden. A specific contribution of the work is that the testing uses real actions and grid operation workflows to answer the question “does the tested object function?” rather than simply examining “whether it is available”. Therefore, the testing goes a step further than traditional availability monitoring. In addition, the test infrastructure is generic and middleware independent. This feature is inherited from g-Eclipse.

In the next step, we will develop mechanisms for detecting the reason of the problems. With this extension we can even answer the question “why it does not work”.

References

1. A. W. Cooke et al.: The Relational Grid Monitoring Architecture: Mediating Information about the Grid. Journal of Grid Computing, vol. 2, no.4, pp. 323-339, 2005.
2. B. Balis, M. Bubak, J. Dziwisz, and K. Rozkwitalski: {Towards a Generic P2P Monitoring Intrastructure for Grid Resources and Applications. Proceedings of the second CoreGRID Workshop on GRID and Peer to Peer Systems Architecture, January 2006.
3. R. Mueller-Pfefferkorn et al.: Monitoring Jobs and their Execution for the LHC Computing, Proceedings of CGW’06, pp. 224-231, 2006.
4. A. Duarte and P. Nyczyk and A. Retico and D. Vicinanza: Monitoring the EGEE/WLCG Grid Services. Proceedings of International Conference on Computing in High Energy and Nuclear Physics, September 2007.
5. H. Kornmayer, M. Stümpert, M. Knauer, and P. Wolniewicz: g-Eclipse - an Integrated Workbench Tool for Grid application users, Grid operators and Grid application developers, Cracow Grid Workshop'06, Cracow, Poland, October 2006.

Simon 6 - Monitoring UNICORE 6 Resources

Michael Rambadt (1), Mathilde Romberg (1)
(1) Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany

Nowadays more and more scientists require a lot of high performance computing power to run their complex parallel applications. Even if the available systems get increasingly powerful, this often is not enough. As a consequence many applications are designed to run not only on one supercomputer but on several in parallel. Grid middleware is an essential offer to the scientists to hide the complexity and the heterogeneousness of the underlying systems and architectures. UNICORE 6 [1] is one of the leading Grid middleware systems. It provides a seamless interface for preparing and submitting jobs to a wide variety of heterogeneous distributed computing resources and data storages. Users are supported to generate scientific and engineering applications, to submit them and to monitor the results without having to know about the respective target system settings.

Running Grid middleware on distributed systems always means running a quite complex infrastructure. The more systems are connected in the Grid the more complex the installation will get. Each middleware provider will have a massive interest that the respective software really is running and working as expected. In a production environment, a 24 hours a day and 7 days a week availability of the underlying Grid infrastructure is essential and expected by the users. This is an ideal scenario of course because the health status of the Grid is dependent on various factors. All, broken hardware, defective processes, or network problems are risks that might disturb the Grid middleware functionality and so might danger the jobs submitted by the scientists. Even if it will not be possible to reach the 24/7 availability of all components, administrators are anxious to minimize the downtime of the provided services. The persons in charge have to be aware of problems before the user does so that one can react and fix the failure preferably in time. Therefore, it is essential to have automatic procedures to check the health status of the Grid middleware. In case of failures and problems the administrator should be notified directly and can initiate then the required steps.

This paper focuses on monitoring of Grid services at the example of UNICORE 6. It describes the development of SIMON 6 to monitor the health status of the UNICORE 6 components. Compared

to other Grid monitoring tools that are available already (for instance the Monitoring and Discovery Service (MDS) [2]), Simon 6 follows a more flexible approach. Even if it has been developed to monitor the UNICORE 6 Grid primarily, other Grid middleware could be integrated easily as well. SIMON 6 is the follow up development of SIMON 5 [3] which is available to monitor UNICORE 5 resources. In contrast to UNICORE 5, UNICORE 6 now is fully based on standard Grid services. Therefore a complete redesign of SIMON 5 to version 6 was necessary.

Besides easy installation and configuration, a central design requirement for SIMON 6 was to check not only the availability of the UNICORE 6 services but also their functionality. A service which is available in the process list does not necessarily deliver the expected functionality as well. To be able to check both availability and functionality at the same time, SIMON 6 bases on the UNICORE 6 internal job description. To check the functionality of the components it is essential to submit “real” jobs into the Grid. UNICORE 6 provides a comfortable command line interface for job submission which is used by SIMON 6 to monitor the components isochronously and automatically. All UNICORE 6 components (gateway registry, workflow engine, unicorex, XUUDB and TSI) are monitored one after the other, and afterwards the results come back to SIMON 6 again. If any problem with the UNICORE 6 components occur, the administrator will be informed by email or a short text message to his/her mobile phone.

SIMON 6 can be integrated easily into existing monitoring frameworks. So for instance reporters for INCA [4] and Nagios [5] are available.

References

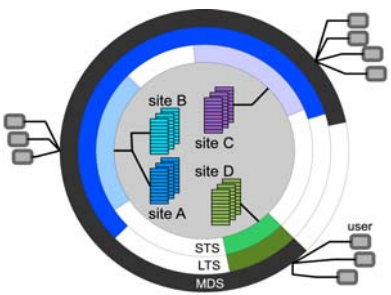
1. Streit, S. Bergmann, R. Breu, J. Daivandy, A. Giesler, B. Hagemeier, S. Holl, V. Huber, D. Mallmann, A.S. Memon, M.S. Memon, R. Menday, M. Rambadt, M. Riedel, M. Romberg, B. Schuller, Th. Lippert, UNICORE 6 - A European Grid Technology. In Trends in High Performance and Large Scale Computing, L. Grandinetti, G. Joubert, W. Gentzsch (Eds.), IOS Press, to be published
2. MDS: http://www.globus.org/grid_software/monitoring/index-service.php
3. M. Rambadt, R. Breu, L. Clementi, T. Fieseler, A. Giesler, W. Gürich, P. Malfetti, R. Menday, J. Reetz, and A. Streit, Experiences with Using UNICORE in Production Grid Infrastructures DEISA and D-Grid. International Symposium on Grid Computing (ISGC 2007), In Springer, ISBN: 978-0-387-78416-8, pp. 121 – 131 2007
4. http://INCA.sdsc.edu
5. http://www.nagios.org

A Scalable Infrastructure for Job-Centric Monitoring Data from Distributed Systems

Marcus Hilbrich (1), Dr. Ralph Müller-Pfefferkorn (1)
(1) Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden

A Grid infrastructure intends to provide a huge amount of globally distributed computing resources as well as services to access these resources. Since the end-users very often are not computer science experts, the processes of allocating resources or managing data are usually hidden in underlying Grid middleware. Nevertheless, the user wants to be informed about the jobs he is running on the Grid, e.g. their resource usage or problems in execution. Thus, an effective job monitoring system is needed.

Of particular importance for a monitoring system is that there should be almost no effort for the users to record and view the monitoring data. Hence, the gLite based monitoring system AMon was developed. The possibility of data recording from other sources is also required in order to address all needs of a wide and heterogeneous user-community. These needs can be other monitoring systems, file observers or tracing systems which give detailed information about the internal state of an application. Another important aspect is the perspicuous presentation of the monitored data and its easy access. Therefore, it should be avoided that users need to install additional programs or care about monitoring data storage. AMon achieves this by providing a web browser interface that presentes the monitoring data graphically.



We introduce the missing link between recording data and presenting them to the user by an infrastructure, which is able to deal with a lot of requests from simultaneous running jobs on distributed resources. It handles different kinds of data so that it can be used by different user-groups. Moreover, it avoids bottlenecks, like the limitation of network-bandwidth that can not be increased easily.

To meet these requirements the resulting monitoring infrastructure is widely scalable and can be increased with the user-base and the number of systems to monitor. It consists of components which are based on the Globus Toolkit 4: The Short-Time-Storage (STS) is a site specific component which temporarily stores monitoring data and manages the communication to components outside the site. The Long-Time-Storage (LTS) is used as second level monitoring data storage. The needed storage capacity and network bandwidth can be distributed by using multiple instances of the LTS at different physical locations and distinct network connections. The Meta-Data-Server (MDS) assures an easy way to access monitoring data without knowing where the data is stored. Again, several instances of it can be installed to distribute the network traffic. As shown in the drawing, the monitoring infrastructure can be sketched as a layered system. The outer layer is represented by the MDS which serves as interface to the presentation layer of AMon. The inner layers are built by LTS and STS. The components of the inner layers can be operated by a single site (e.g. site D) or can be used collaboratively. In the drawing sites A and B share the same STS instance and sites A, B and C use the same LTS. It is also illustrated that components of a layer hide the physical infrastructure of adjacent layers.

Integrated into AMon the system is able to analyze the monitoring data of Grid-jobs and find the problematic ones. It provides an interactive graphical interface and is capable of handling the load of a huge distributed system like current and future Grid installations.

Acknowledgements. This work is supported by Bundesministerium für Bildung und Forschung.

References

1. R. Müller-Pfefferkorn , R. Neumann, A. Hammad, T. Harenberg, M. S. Hüsken, P. Mättig, M. Mechtel, D. Meder-Marouelli , St. Borovac, P. Ueberholz: Monitoring of Jobs and their Execution for the LHC Computing Grid, in Proceedings of the Cracow Grid Workshop (CGW 06), Cracow, Poland, October 15-18, pages 224-231, 2006
2. R. Müller-Pfefferkorn, R. Neumann, Th. William: AMon - a User-Friendly Job Monitoring for the Grid, In "Towards Next Generation Grids: Proceedings of the CoreGRID Symposium 2007", pp. 185-192, ISBN: 978-0-387-72497-3, New York, Springer, 2007
3. Henrik Eichenhardt, Ralph Müller-Pfefferkorn, Reinhard Neumann, Thomas William: "User- and Job-Centric Monitoring: Analysing and Presenting Large Amounts of Monitoring Data"; In Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing; Sep 29th - Oct 1st, 2008; Tsukuba/Japan, ISBN 978-1-4244-2579-2; pages 225-232; 2008
4. gLite – lightweight middleware for Grid Computing, www.glite.org
5. Globus Toolkit Homepage – http://www.globus.org/toolkit/

Architecture for the Propagation of Changes of VO Membership Data to a Heterogeneous Application Landscape

Oliver Strauss, Anette Weisbecker
Fraunhofer-Institute for Industrial Engineering IAO, Stuttgart, Germany

In a Grid environment many different systems like the grid middleware, batch systems, web portals etc. play together to provide services and functionality to the Grid users. Managing user accounts in all those systems is a difficult task because of the heterogeneity of the systems and their distribution over many resources. A common administrative task is to periodically check the directory service of a virtual organization (VO) (e.g. VOMRS) for information about the users that belong to a given VO, store this information in a grid-mapfile and create the appropriate Linux/Unix user accounts (see e.g. [1]). For applications tightly integrated in the Grid middleware this approach is proven and can be automated with tools like Cfengine [2].
In the project PartnerGrid [3] we have a system architecture that includes less tightly integrated components like a Liferay portal as web frontend, WebDAV through an apache web server to transfer

data to and from the server and the Remote Component Environment (RCE) [4] to manage simple workflows and to provide a gateway to the Grid. Changes in VO membership should lead to creation, update or revocation of user rights and accounts in the aforementioned systems. Since this is a common problem usually dealt with by ad hoc shell scripting that quickly become complex and hard to maintain, we tried to come up with a more systematic and modular solution to deal with these administrative tasks that leaves existing systems untouched.
Or approach is based on three logical components: source adapters that detect changes in some data source such as a grid-mapfile or a VO-directory and emit creation, update or deletion events that contain all information about the change that can be gathered from the system. These events are consumed by the change manager component which uses a simple rule engine to enrich and normalize the data received in the change events and to route the events to a set of destination adapters. These adapters are responsible for the execution of the change events in the target system. This architecture cleanly separates the responsibilities of change detection, data enrichment and normalization, change scoping and change execution.
In addition to this separation of concerns the approach promises other advantages. New systems can be added by developing a Source- or destination adapter. The logic that determines how to interpret and normalize data from the various systems and which changes affect which systems is encapsulated in the rules of the change manager. It is also easy to add statistics or an audit trail to the system by routing all changes to a statistics or logging adapter. Since change events can be transferred to remote locations, it is possible to centralize the handling of changes in a configuration of heterogeneous distributed systems. It is also possible to implement pull and push strategies since the source adapters can be triggered either by some application specific event like e.g. a ModelChangeListener in the Liferay portal (push-model) or by the change manager itself (pull model).
The architecture sketched above is still in development and we are currently in the process of implementing a prototypical implementation using the dynamic language Ruby for evaluation in the scenario in PartnerGrid. First experiments look promising but further work on the manager component and new adapters has to be done.

Acknowledgements: This work is supported by the project PartnerGrid (01G07009A-D) funded by the German Federal Ministry of Education and Research (BMBF).

References

1. H. Enke: AGD Grid Account Management, http://www.dgrid.de/fileadmin/user_upload/documents/DGI-FG1-10/AstroGrid-GrAMa-VOWSII.pdf, last accessed on 09/11/2009, 2006
2. M. Burgess: A Tiny Overview of Cfengine: Convergent Maintenance Agent, Proceedings of the 1st International Workshop on Multi-Agent and Robotic Systems, MARS/ICINCO 2005.
3. A. Weisbecker: PartnerGrid - Cooperative Grid Solutions for Industrial Applications; in „German Grid Initiative“, pp. 54-56, Universitätsverlag Göttingen, 2007
4. D. Seider, O. Krämer-Fuhrmann: RCE – A Distributed Platform for the Integration of Applications; Product Flyer, http://www.scai.fraunhofer.de/uploads/media/RCE-Flyer.pdf, last accessed on 09/11/2009

Online Retrieval, Evaluation and Visualization of Messages in the UNICORE 6 Workflow System

Sandra Bergmann (1), Bastian Demuth (1)
(1) Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany

UNICORE 6 is a Grid middleware which combines heterogeneous resources to a uniform system. It provides graphical and command line clients for accessing the available resources via UNICORE Web services [1]. The UNICORE workflow system is an extension of UNICORE that allows for carrying out complex workflows involving numerous jobs to be executed in a predefined order [2]. The services of the workflow system communicate with each other by means of XML messages, transported via SOAP [3]. All messages are also sent to the tracing service which stores them in a relational database. Each message has the following attributes: type and id of the source and target services, time stamp, message identifier, name of the requested operation, and full content of the message in XML.

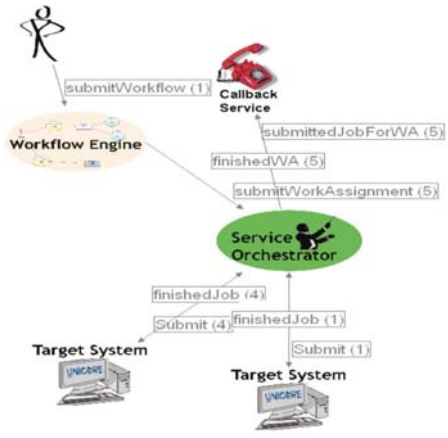
The tracing service provides an interface for retrieving recorded messages by using a query language that supports pattern matching against message attributes and timestamps, as well as Boolean combinations of such queries. The result set contains full messages or selected attributes. Similar to Ostermann et al. [4] who have analyzed workflow traces statistically, we employ message tracing for identifying communication bottlenecks and improving the performance of the workflow system. Moreover, tracing helps us elevate the user experience. Users can pose queries for investigating which services were involved in executing a workflow and how long it took for certain workflow parts to be executed.

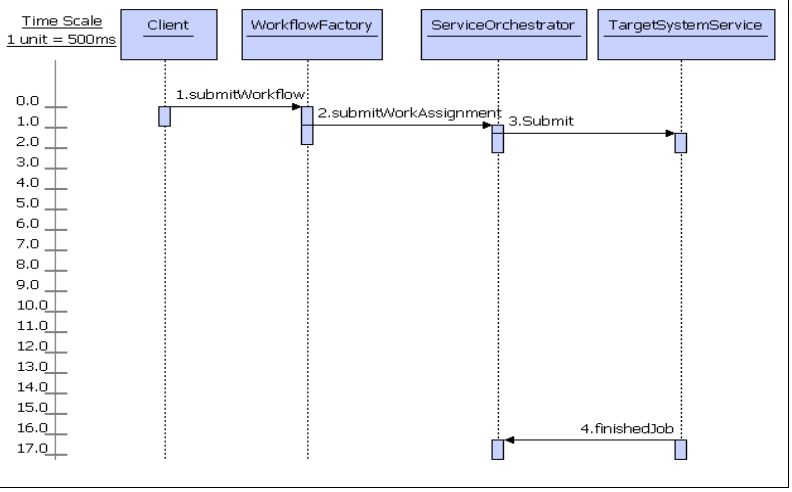
The graphical UNICORE client offers two different types of diagrams for visualizing message flows, both providing an intelligible overview of the communication but focusing on complementary aspects. Diagrams of the first type highlight the structure of services that were involved in the communication by arranging them in a hierarchical graph (Fig. 1). Service instances form the nodes of the graph, messages are depicted as directed edges between them. Message labels contain the operation type and the number of similar messages that were sent. The second diagram type is based on the AmaterasUML project [5], resembles a sequence diagram (Fig. 2), and is used for analyzing the timing of a traced communication. Services are drawn as coloured rectangles and messages are represented by arrows between the services' dotted life lines. An adjustable time scale helps to `zoom' into specific parts of the diagram. The content of message labels can be changed and tooltips are used in both diagram types for displaying additional information.

As stated above, tracing was used for monitoring the execution of workflows and analyzing the workflow system's performance in stress conditions, leading to certain adjustments in the service implementations.

Additional statistical analyses of real production workflows will help improve the system further. Future activities could also identify other visualization tools and meaningful queries, e.g. pie charts and histograms could be used for investigating which Grid services and applications were used most frequently and which workflow parts hold the highest speedup potential.

Fig.1 Diagram of a submitted workflow	Fig.2 Sequence diagram





References

1. Achim Streit: UNICORE – A European Grid Technology; http://www.unicore.eu; 2008
2. Krzysztof Benedyczak, Jason Milad Divandy, Bastian Demuth and Bernd Schuller: Chemomentum – WP 1 Architecture and Software framework. Deliverable D1.6; 2009
3. Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar and Yves Lafon: SOAP Version 1.2; http://www.w3.org/TR/soap12-part1/; 2007
4. S.Ostermann, R. Prodan, T. Fahringer, A. Iosup, D. Epema: On the Characteristics of GridWorkflows; http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0132.pdf; 2008
5. Takezoe and Rinoa: Amateras; http://amateras.sourceforge.jp; 2009

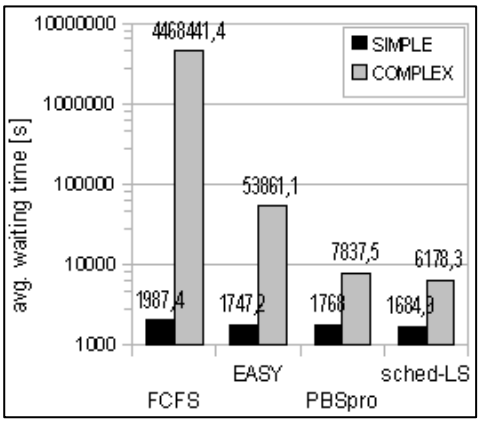
Complex Real-Life Data Sets in Grid Simulations

Dalibor Klusáček, Hana Rudová
Faculty of Informatics, Masaryk University, Brno, Czech Republic

Widely used workload sources such as Parallel Workloads Archive (PWA) [4] or Grid Workloads Archive (GWA) [5] often provide data sets that are still unrealistic. Typically, very limited information is available about the grid/cluster parameters such as architecture, speed, RAM size or resource specific policies. Moreover, no information concerning background load, resource failures, or specific user's requests are available. In heterogeneous environments, users often specify a small subset of machines that are suitable to perform the jobs. This subset is usually defined either by resource owner's policy (user is allowed to use such cluster) or by user's requirements (user specifies some property offered by some clusters only) or by both factors. When one tries to create a good scheduling algorithm and compare it with current approaches such as PBSpro [3], all such information and constraints are crucial, since they make the algorithm design much more complex.

So far, we have been able to collect complex real-life data set from the Czech national Grid infrastructure MetaCentrum [6] that covers many previously mentioned issues, representing 5 months of execution, involving 103 620 jobs completed on 14 heterogeneous clusters having 806 CPUs. This data set includes exact machine parameters involving both hardware setup (speed, architecture, RAM size) and supported properties (e.g., hardware/institution/queue based restrictions). For each job, a queue where the job was originally submitted is known, as well as the maximal runtime after which the job would be killed. Also, the numbers, types, time limits and priorities of all queues are known. Moreover, machine failures and restarts are known together with the list of temporary dedicated, thus unavailable, machines. In contrast to the GWA or PWA sources, our data set allows to perform far more realistic simulations involving all precedent features.

We have studied behavior of several objective functions in our solution that cover typical requirements such as average job slowdown, response time, waiting time and the algorithm's runtime. We use schedule-based algorithms involving Local Search (sched-LS) which we have been developing for couple of years [1], as well as widely used queue-based solutions such as FCFS, EASY Backfilling [2] or PBSpro algorithms [3] representing multiple-queues priority based backfilling. In our experiments, we have focused on two scenarios. SIMPLE scenario does not simulate dedicated resources or failures. Moreover, all jobs can be executed on any cluster (if enough CPUs are available), thus SIMPLE represents the typical amount of information available in the GWA or PWA data sets. On the other hand, COMPLEX scenario uses every additional information available in our data set such as queue priorities, machine failures, as well as additional machine and job properties that define the job-to-machine suitability. As observed during the experiments, the difference between SIMPLE and COMPLEX setup is dramatic as shown in the figure depicting the average waiting time (in log. scale). In case of SIMPLE, the differences between algorithms are quite small while the COMPLEX data set introduces huge differences among algorithms, causing that previously acceptable FCFS or EASY backfilling now degrade dramatically. Similar behavior was visible for all above mentioned objective functions.



It is clear that complex and "rich" data set influences the algorithms' performance and causes significant differences in the values of objective functions. We suggest that—beside the PWA and the GWA—complex data sets should be also used to evaluate existing and newly proposed algorithms under harder conditions. Therefore, at the time of the conference our data set will be publicly available for further open research.

Acknowledgements. We appreciate the support of the Ministry of Education, Youth and Sports of the Czech Republic under the research intent No. 0021622419 and the support of the Grant Agency of the Czech Republic with grant No. 201/07/0205.

References

1. Klusáček, D.; Rudová, H.; Baraglia, R.; Pasquali, M.; Capannini, G. *Comparison of Multi-Criteria Scheduling Techniques.* In Grid Computing Achievements and Prospects, Springer, 2008.
2. Skovira, J.; Chan, W.; Zhou, H.; and Lifka, D. A. *The EASY - LoadLeveler API Project*. In IPPS '96: Proceedings of the JSSPP Workshop. Springer. 1996.
3. Jones, J. P. 2005. *PBS Professional 7, administrator guide*. Altair. 2005.
4. Parallel Workloads Archive: http://www.cs.huji.ac.il/labs/parallel/workload/
5. Grid Workloads Archive: http://gwa.ewi.tudelft.nl
6. MetaCentrum: http://meta.cesnet.cz

Grid Software Detection

Max Berger (1), David Gstir (1), Thomas Fahringer (1)
(1) Leopold-Franzens-Universität, Innsbruck, Austria

Distributed computational environments such as the Grid or the Cloud provide heterogeneous environments, which are under different administrative domains. As such, each execution component may have a different configuration and installed software. Information systems describing the software installation are mostly manually administrated, and as such are often incomplete or out of date.

In this paper we describe an alternate approach to maintaining software databases through software detection.

Heterogeneity in distributed environments provides a challenge for application developers. A simple mismatch in an installed library can cause the execution of an application to fail. To handle this challenge, different approaches are used: Homogenization, virtualization, and publishing resource descriptions. Publishing resource descriptions provides the most flexibility and support for specific features. However, currently resource descriptions have to be managed manually.

We propose and automated software installation tool system to detect the actual installed software. Small detection scripts are run on a software site. They execute arbitrary commands which can check for availability, usability, or for the version of a program. The detection script produces a result file. In a second step, the result file is read by a parser component. The parser interprets the result file and extracts key-value data. This data can be stored either in a separate database, or along with the manually managed resource description of the system. This information can then be used to filter out sites based on given resource requirements, similar to what is done in current systems.

We also propose a plug-in system which can be used by ordinary users. This allows every Grid application developer to add arbitrary software tests. The plug-ins are executed under in the security context of the user who submitted the plug-in, as to avoid security issues. Plug-ins will also create key-value-data, which can be entered into the information system.

Software tests can be automatically re-run to produce updated data. This is of importance, as software updates frequently break existing software installations, and therefore invalidate previously collected data.

We conclude with the a verification case for the proposed system. We have implemented a prototype which can be used to detect installed software. The prototype includes detectors for the installation of the Java runtime environment. It checks for availability, version, and correctness of the installation. We tested the tool with the VOCE and CompChem VOs in the EGEE Grid, and will present this survey.

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

Selected References

1. Enabling grids for E-sciencE (EGEE). http://www.eu-egee.org/.
2. S. Andreozzi, S. Burke, L. Field, S. Fisher, B. Konya, M. Mambelli, JM Schopf, M. Viljoen, and A. Wilson. Glue schema specification version 1.2. OGF GLUE-WG.
3. Mario Antonioletti et al. The design and implementation of Grid database services in OGSA-DAI. Concurrency and Computation: Practice and Experience, 17(2-4):357–376, Feb 2005.
4. Andrew W. Cooke et al. The relational grid monitoring architecture: Mediating information about

the grid. J. Grid Comput., 2(4):323–339, 2004.
5. Ian Foster and Carl Kesselman. The Grid 2: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers, second edition, 2004.
6. Globus Alliance. GT information services: Monitoring & discovery system (MDS). http://www.globus.org/toolkit/mds/, retrieved 02/2007.
7. Claudio Grandi. The gLite middleware. In EGEE'07, 2007.
8. Rajesh Kalmady, Digamber Sonvane, Phool Chand, Kislay Bhatt, Kumar Vaibhav, Piotr Nyczyk, and Zdenek Sekera. Monitoring the availability of grid services using SAM and gridview. In Grid Computing – International Symposium on Grid Computing ISGC 2007, pages 163–168. Springer, 2009.
9. Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: Towards a cloud definition. Technical report, January 2009. http://ccr. sigcomm.org/online/?q=node/439.

Providing Scientific Software as a Service in Consideration of Service Level Agreements

A Survey for Classification of Scientific Software

Oliver Niehörster (1), André Brinkmann (1), Georg Birkenheuer (1), Sonja Herres-Pawlis (2), Julia Niehörster (3), Jens Krüger (2), Brigitta Elsässer (2), *Lars Packschies (2)*
(1) Paderborn Center for Parallel Computing, Universität Paderborn, Germany
(2) Department Chemie, Universität Paderborn, Germany
(3) Department Agricultural Sciences, Universität Hohenheim, Germany

Utility Computing is a new trend in computer science meaning the on-demand consumption of computing resources. This is similar to the electric power received from an energy provider. There are many levels of how compute capacity might be provided. It ranges from the offering of running entire virtual systems in the providers datacenter to the ability to consume applications.

The Cloud Computing principle, also called Infrastructure as a Service, is located at the bottom of the abstraction stack. A Cloud provider offers the possibility to run virtual guest systems in his datacenter. The user provisions, releases and removes his virtual machines as needed. On the client side there are no administration and management expenses. On top of the Cloud there are the IaaS users and Software as a Service (SaaS) providers. An SaaS provider offers complete applications. The user only needs a thin client to control the application running mainly in the Cloud. IaaS users benefit from the virtualization used in Clouds. They are able to build virtual environments independently of physical hardware. This differs from Grid computing and offers the possibility to run these environment on different clouds.

The provider does not need an own datacenter. Maybe he himself is a service consumer of an IaaS provider. Today SaaS providers concentrate on business and office software. There are offerings for online word processors, Content-Management, Customer-Relationship-Management or Human-Resourc-Management applications.

In our work we focus on scientific software. In a proof of concept we build up a private Cloud and evaluate the possibility to be a scientific SaaS (SSaaS) provider. We found out that there are many differences between business and scientific software. Scientific applications are normally HPC programs solving complex computation problems. The are batch-oriented and at runtime they consume as much resources as available.

This is very different from business applications that are mostly durable jobs with only sporadic peak loads. Another difficulty that occurs in scientific applications is the halting problem. In many experiments it is not possible to decide whether a the program will finish or run forever.

Yet another property is the facility of measuring the application performance. As an SSaaS provider we want to guarantee a specific service level, for example we ensure that a computation ends at a certain date. To find out the horizontal and vertical scaling of our virtual infrastructures we have to measure the application on different configurations. To scale the used resources during the

computation it is important to have some kind of progress data. It is interesting that only a few applications deliver this information very accurate and reliable. Another important property is the checkpointing to stop an resume the computation. Some applications does a period snapshot itself, for example after a time-consuming computation step, and others offers the possibility to do it manually.

In this paper we present the properties of different scientific applications with the focus on the mentioned aspects. We have started a survey at different scientific institutes. The scientists have to categorize and describe the applications they use in their research.

ValueGrids: Using Grids in Dynamic Service Value Networks

Frank Schulz (1), Dennis Westermann (1), Benjamin Blau (2), Wibke Michalk (2), Daniel Rolli (3), Omid Khodaverdi Afaghi (3), Markus Hedwig (4), Andrea Schmidt (5)
(1) SAP Research, Karlsruhe, Germany
(2) IISM, University of Karlsruhe, Germany
(3) Conemis, Karlsruhe, Germany
(4) Information Systems Research, University of Freiburg, Germany
(5) IBM Research & Development, Böblingen, Germany

In recent years, Grid infrastructures have proven to be invaluable in the area of scientific computing and high-performance computing (HPC). However, the use of Grid infrastructures in complex business scenarios remains an open research topic. A major goal of the ValueGrids project, funded by the German D-Grid Initiative, is the development of a concept to demonstrate how Grid infrastructures can substantially support complex business scenarios.
Nowadays, business processes need to be extremely flexible due to a growing innovation rate, highly dynamic markets, and more and more complex but individual customer expectations. Thus, companies often engage in partnerships in order to meet these requirements. Typically, Software-as-a-Service (SaaS) providers compose services from different software, infrastructure, and service providers to jointly create an integrated solution while forming networked economies – so called Service Value Networks (SVNs) [1]. To express the conditions of service delivery between the stakeholders in an SVN, Service Level Agreements (SLAs) have become a common means with the ultimate goal to treat services as tradable goods by establishing a legal and contractual basis. Thus, SaaS providers have to manage SLAs between multiple layers and stakeholders in a Service-Oriented Architecture (SOA).
SLA Management is an active research area (e.g. [2]), but a concept for the holistic management of SLAs by SaaS providers is still an open topic. Especially, the integration of Grid infrastructures and the identification of the requirements a Grid infrastructure provider has to fulfill are challenging unresolved tasks. Recent work on integrating Grids in business scenarios is done by research projects such as Biz2Grid [3] and BIS-Grid [4].
One aim of the ValueGrids project is the development a concept for the holistic management of SLAs in SVNs that seamlessly integrates grid-based services in complex business processes. On the one hand, this includes the translation of SLAs along the vertical IT/service stack as well as the aggregation of SLAs within one layer. On the other hand, it covers the runtime monitoring of instantiated SLAs by the SaaS provider. The concept outlined in this position paper comprises the derivation of rules for the aggregation and translation of SLAs as well as the design and reference implementation of an SLA Management framework. The framework consists of three main components:

- Service Repository: Includes service descriptions and corresponding SLA templates.
- Dependency Analyzer: Extends the Service Repository with analysis functionalities such as detection of service dependencies, detection of SLA dependencies, and determination of risk indicators.
- Management Cockpit: Visualizes the current state of the SaaS provider's SLA landscape based on Web 2.0 technologies.

In addition, ValueGrids provides a systematic approach for testing service compositions with respect to Quality of Service (QoS) properties. This is an essential precondition for the management of SLAs since it is the only way to identify, describe, measure, and predict the relevant non-functional parameters.

The project results yield two main benefits: (i) An SaaS provider can conduct a holistic management of his SLA landscape. (ii) Grid infrastructures can be integrated into complex and dynamic enterprise service value networks. Moreover, the results serve as basis for further research on the integration of Grid infrastructures in business scenarios.

Acknowledgements. This work is partially supported by the German Federal Ministry of Education and Research (BMBF) under grant No.01|G09004 as part of the D-Grid Initiative.

References

1. B. Blau, J.Kraemer, T. Conte, and C. van Dinther: Service Value Networks; Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing, July, 2009.
2. M.P. Papazoglou, W.J. van den Heuvel: Web Services Management: A Survey; IEEE Internet Computing 9, 58–64, 2005
3. J. Stößer, T. Meinl: Biz2Grid: A Framework for Market-Based Grid Scheduling, Proceedings of the Cracow Grid Workshop ’08, Oct., 2008.
4. S. Gudenkauf, W. Hasselbring, F. Heine, A. Hoeing, O. Kao, G. Scherp: Bis-Grid: Business Workflows for the Grid; Proceedings of the Cracow Grid Workshop ’07, Oct., 2007.

Virtual Clusters as a New Service of MetaCentrum, the Czech NGI

Miroslav Ruda, Zdeněk Šustr, Jiří Sitera, David Antoš, Lukáš Hejtmánek, Petr Holub
CESNET, Prague, Czech Republic

MetaCentrum, the Czech NGI, already started to virtualize the infrastructure several years ago. Major part of MetaCenter computation resources is currently virtualised. The virtual nature of the resources is mostly hidden to the end users due to integration with the resource management system. Currently we are introducing a new public service "virtual cluster" which turns the virtualized infrastructure directly into end user service. Virtual cluster service provides an illusion of totally dedicated clusters under complete user control running on a shared infrastructure, including administrator access and user specified application environment. Virtual machines and clusters are handled in a way similar to ordinary computation jobs, planned for batch or interactive processing. We developed an extension to job scheduler PBSPro and new management tools to smoothly integrate virtual cluster service into NGI environment. Networking is also a vital part of the service, where Czech NREN CESNET2 technology allows managing virtual cluster network without perceivable overhead. Virtual network becomes a new resource, that can be attached to user's home network. Benefits of this fully integrated virtualized infrastructure will be demonstrated through series of usecases.

Acknowledgements. This work was done with the support of Ministry of Education of the Czech Republic, research programs MSM0021622419 and MSM6383917201.

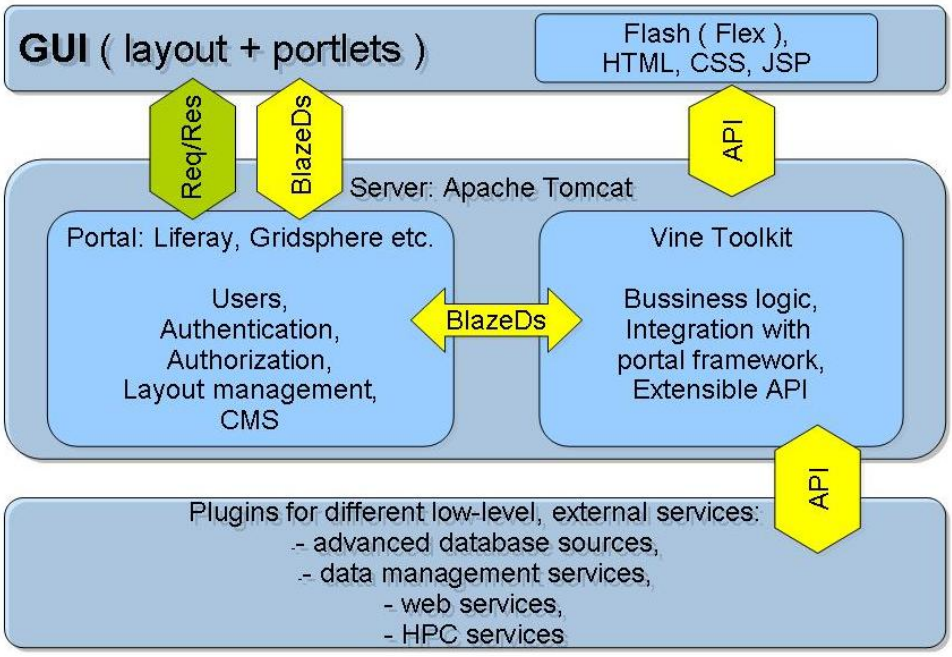
References

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. ANTOŠ, David - MATYSKA, Luděk - HOLUB, Petr - SITERA, Jiří. VirtCloud: Virtualising Network for Grid Environments-First Experiences. In The 23rd IEEE International Conference on Advanced Information Networking and Applications AINA 2009. Bradford, UK : IEEE Comp. Soc., 2009. od s. 876-883, 8 s. ISBN 978-0-7695-3638-5
3. Antoš D., Sitera J., Matyska L., Holub P.: MetaCenter Virtual Networks, proceedings of Cracow Grid Workshop 2008, Cyfronet AGH, 2009, str.: 86-93 , ISBN: 978-83-61433-00-2
4. MetaCentrum – http://meta.cesnet.cz/

Vine Toolkit – a Uniform Access and Portal Solution to Existing Grid Middleware Services

Piotr Dziubecki (1), Krzysztof Kurowski (1), Dawid Szejnfeld (1), Dominik Tarnawczyk (1), Małgorzata Wolniewicz (1)
(1) Poznań Supercomputing and Networking Center, Poland

Vine Toolkit is a modular, extensible Java library that offers developers an easy-to-use, high-level Application Programmer Interface (API) for Grid-enabling applications [1]. Vine can be deployed for use in desktop, Java Web Start, Java Servlet 2.3 and Java Portlet 1.0 environments with ease. Additionally, Vine Toolkit supports a wide array of middleware and third-party services, including various gLite, Unicore, GT4 and recently QosCosGrid. Using the Vine Toolkit, one composes applications as collections of resources and services for utilizing those resources. The Vine Toolkit makes it possible to organize resources into a heirarchy of domains to represent one or more virtual organizations (VOs). Vine offers security mechanisms for authenticating end-users and authorizing their use of resources within a given domain. Other core features include an extensible model for executing tasks and transparent support for persisting information about resources and tasks with in-memory or external relational databases.
In large scale production environments, the information sets to perform calculations on come from various sources. In particular, some computations may require the information obtained as a result of previous computations. Workflow description offers an attractive approach to formally deal with such complex processes. Vine Toolkit solution addresses some major challenges here such as the synchronization of distributed workflows, establishing a community driven Grid environment for the seamless results sharing and collaboration. In order to accomplish these goals Vine Toolkit offers integration on different layers starting from rich user interface web components, integration with workflow engine and Grid security and ending up with a built-in meta-scheduling mechanisms, that allow IT administrators to perform load balancing automatically among computing clusters and data centers to meet peak demands. As a result of this Vine Toolkit has been successfully integrated with Kepler workflow engine and it enables complex workflow executions fully controlled in the portal as well as provides an easy and user friendly graphical interfaces for workflow management [2]. Finally, it also provides a repository of workflow nodes and definitions for knowledge sharing and collaboration [3].



Vine Toolkit - architecture

References

[1] Vine Toolkit http://vinetoolkit.org/
[2] BeinGRID BE23 wow2green experiment http://www.beingrid.eu/wow2green.html
[3] Flowify/wow2green solution http://www.gridipedia.eu/wow2green_sol.html

Disk Array Performance Estimation

Darin Nikolow (1), Renata Słota (1), Jacek Marmuszewski (1), Jacek Kitowski (1,2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

While the computational power of modern computers is increasing exponentially the performance of hard disk drives is still about two orders of magnitude below that of the CPU. Disk arrays are deployed to achieve higher performance and level of data protection. Disk arrays are specialized computer systems for providing storage capacity at block level for multiple hosts. The performance of a disk array for a given data transfer is hard to predict because of the complexity of the array and because of the multiclient storage network environment being present at common array installations. Data storage performance estimation is essential for environments, where the performance for a given data access request need to be guaranteed.

In a storage environments with multiple physical storage systems where data might be replicated, a selection of storage system for the given I/O request needs to be made for optimal performance. The performance estimation, necessary for proper storage system selection, must be done in real time (much less than the time of serving the I/O request) which might be challenging taking into account that the system is distributed . For small size I/O transfers the estimation may not make any sense, but for large file transfers there might be significant performance improvement.

While the disk arrays and their performance have been widely studied [1,2,3] not much work can be found concerning pre-I/O-request performance estimation which motivated us to start this research.

In this paper we propose a method for disk array performance estimation. The method consists in earlier performance measurements to get some characteristic points of the performance graph of the given disk array for different load scenarios. We have observed during previous benchmark testing that we can define some static parameters specifying the array performance for different file sizes. The estimator can be automatically tuned by a benchmark test, gathering these static parameters. These parameters are used together with the data about the current load (and some other relevant dynamic parameters) of the array to estimate the performance for a future data transfer. Information about the disk array load is gathered indirectly by monitoring client hosts HBA utilization.

As part of our work monitoring agents running on clients hosts have been developed, as well as application to do estimation of future data accesses. The whole system has been tested and test results presented in the paper.

Acknowledgements. This work is supported by Polish MNiSW grant nr N N516 405535 AGH-UST grant is also acknowledged.

References

1. L. Dutka, J. Kitowski, Stochastic Approach for Secondary Storage Data Access Cost, in: P.M.A. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, M. Bubak (Eds.), Proc. of Advances in Grid Computing - EGC 2005 European Grid Conference Amsterdam, The Netherlands, February 14-16, 2005, Lecture Notes in Computer Science, no. 3470, Springer, 2005, pp. 796-804.
2. Thomasian, G. Fu, C. Han, Two-Disk Failure-Tolerant Disk Arrays, in IEEE Transactions on Computers, vol. 56, no. 6, June 2007, pp. 799-814.
3. E. Varki, A. Merchant, J. Xu, X. Qiu, Issues and Challenges in the Performance Analysis of Real Disk Arrays, in: IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 6, June 2004, pp. 559-574.

Wednesday, Oct 14, 2009

Keynote Presentations

Data Mining and Knowledge Discovery Services in Grids

Domenico Talia
Università della Calabria and ICAR-CNR, Italy

In the latest years Grids enlarged their horizon as they are going to run business applications supporting consumers and end users. To face those new challenges, Grid environments must support adaptive data management and data analysis applications by offering resources, services, and decentralized data access mechanisms. In particular, according to the service oriented architecture (SOA) model, data mining tasks and knowledge discovery processes can be delivered as services in Grid-based infrastructures. Through a service-based approach we can define integrated services for supporting distributed business intelligence tasks in Grids. Those services can address all the aspects that must be considered in data mining and in knowledge discovery processes such as data selection and transport, data analysis, knowledge models representation and visualization. We worked in this direction for providing Grid based architectures and services for distributed knowledge discovery such as the Knowledge Grid, the Weka4WS toolkit, and mobile Grid services for data mining. This lecture describes a strategy based on the use of Grid services for the design of distributed knowledge discovery services and discuss how Grid frameworks, such those mentioned above, can be developed as a collection of Grid services and how they can be used to develop distributed data analysis tasks and knowledge discovery processes using the SOA model.

HealthGrid: an International Initiative to Serve the Virtual Physiological Human and Bio-Medical Communities

Yannick Legre
International Relationship Director, MAAT and President of HealthGrid

System Software for Petascale and Beyond

Kamil Iskra
Argonne National Laboratory, USA

Contributed Presentations

Chelonia: A Self-healing Storage Cloud

Jon Kerr Nilsen (1,2), Salman Toor (3), Zsombor Nagy (4), Bjarte Mohn (5)
(1) University of Oslo, Dept. of Physics, Norway
(2) University of Oslo, Center for Information Technology, Norway
(3) Dept. Information Technology, Div. of Scientific Computing Uppsala University, Sweden
(4) Institute of National Information and Infrastructure Development NIIF/HUNGARNET, Hungary
(5) Dept. of Physics and Astronomy, Div. of Nuclear and Particle Physics, Uppsala University, Sweden

The task of developing a reliable, self-healing, resilient and consistent data management system at the web-scale is still an open challenge. An increasing number of applications demand not only increased CPU power, but also vast amounts of storage space [1]. The required storage space is not restricted to the duration in which the application runs. For certain applications the data needs to be available for years after its collection and production. Nowadays, we can easily find single Grid jobs which produce gigabytes or even terabytes of data, ramping up the requirements of storage systems to the petabyte-scale and beyond. In order to handle these challenges, we have identified two major areas. First, the high availability and security of the data and second, the consistency and reliability of the system itself. In the design of our storage system, Chelonia, these areas have been in focus.

Chelonia has been developed by the team of the Advanced Resource Connector (ARC) middleware [2] and is a user-friendly, grid-enabled storage system. Using the ARC library, Chelonia enables users to easily upload and download files and collections to the storage cloud. User authentication is managed by x509 certificates. Users can assign access policies to their files and collections to grant limited or full access to individual users and entire virtual organizations. Files and collections are managed in a global hierarchical namespace; similar to the concept of paths in a regular local file system, which allow users to access their files without dealing with the complex URLs. Together with a command-line-interface, Chelonia's users can also use the FUSE (Filesystem in Userspace) model which allows to handle the entire storage cloud as a local namepace. Chelonia has a lightweight gateway component to interact with other storage solutions such as dCache [4]. While keeping the namespace intact, it allows users to mount third party storage elements into the storage cloud managed by the system.

Chelonia has a distributed architecture that allows for flexible setup and the elimination of single points of failure. Based on SOAP, Chelonia consists of four web services residing within the ARC service container, the Hosting Environment Daemon (HED). The services are as follows: The Bartender provides the high-level interface to the user, and allows access to third-party storage systems. The Librarian handles the entire storage namespace, using the A-Hash as a metadatabase. The Shepherd is the frontend for the physical storage node. The replication of the services ensures consistency and high availability. Files in Chelonia are replicated and the system itself ensures that a file has the user specified number of replicas. The replicas are stored on different storage nodes. A storage node can easily be created from an empty directory on any machine by installing and configuring a small subset of Chelonia. Number of protocols is supported for data transfer. One time URL is generated by the Shepherd and the actual transfer is done directly between the client and the storage node. Being part of the ARC middleware bundle, Chelonia is also accessible through the ARC data client tools and through grid job descriptions. Chelonia is currently in a testing phase and is going through a series of performance and stability tests. This article will mainly focus on the design of the system. The flexibility of the architecture and our test results show that the proposed system is comparable with the other distributed storage solutions like dCache [4], Scalla [6] and Xtreemfs [5].

Acknowledgment. We like to thank UPPMAX, NIIF and USIT for providing resources for running the storage tests. The work has been supported by the European Commission through the KnowARC project (contract nr. 032691) and by the Nordunet3 programme through the NGIn project.

References

1. Hoschek,W., Jaen-Martinez, J., Samar, A., Stockinger, H., Stockinger, K.: Data management in an international data grid project, Springer-Verlag (2000) 77–90.

2. Ellert, M., et al.: Advanced Resource Connector middleware for lightweight computational Grids. Future Gener. Comput. Syst. 23(1) (2007) 219–240.
3. Nagy, Z., Nilsen, J.K., Toor, S.: Documentation of the ARC storage system. NorduGrid. NORDUGRID-TECH-17.
4. Behrmann, G., Fuhrmann, P., Grønager, M., Kleist, J.: A Distributed Storage System with dCache. Journal of Physics: Conference Series 119 (2008) 062014.
5. Hupfeld, F., Cortes, T., Kolbeck, B., Stender, J., Focht, E., Hess, M., Malo, J., Marti, J., Cesario, E.: The xtreemfs architecture—a case for object-based file.
6. Boeheim, C., Hanushevsky, A., Leith, D., Melen, R., Mount, R., Pulliam, T., Weeks, B.: Scalla: Scalable Cluster Architecture for Low Latency Access Using xrootd and olbd Servers. Technical report, Stanford Linear Accelerator Center (2006).

Bridging Grid and Cloud Storage Resources

Gian Luca Volpato, Yassene Mohammed
RRZN – Regional Computing Centre for Lower Saxony, Gottfried Wilhelm Leibniz Universität Hannover, Germany

The constant increase in the number of scientific applications requiring huge amount of storage space has led many application service providers and computing centers to question themselves whether they should outsource their own storage systems rather than enlarging them. Cloud storage represents a promising option for the extension of Grid storage resources with on-demand, dynamic, cost-efficient storage solutions. Usually these products consist of an online file system, with integrated user account management and access control management interfaces, accessible via standard-based web services.

In this regard, interoperability between Grid and Cloud storage resources becomes of paramount importance when we want these two similar in goal, yet different in technology, systems to coexist and collaborate in delivering the adequate storage capacity for the scientific challenges of the next decade [1]. Although several previous scientific works have already compared the functionality and/or the costs and benefits of using Cloud storage as an alternative to Grid storage [2], none of them considered the security aspects related to integrating Cloud systems into an existing Grid infrastructure.

In this paper we evaluate strengths and weaknesses of a small group of candidate Cloud storage providers (Amazon Simple Storage Service, Nirvanix Storage Delivery Network™, Rackspace Cloud Files™) in terms of authentication, authorization, confidentiality and integrity. Then we compare the technical details of their implementations to the corresponding technology deployed by well-established Grid storage systems like GridFTP and SRM (dCache). Our main goal is to verify whether the Cloud can deliver a solution providing a level of security that matches what granted by traditional Grid systems. Therefore we propose here two approaches to enable the concurrent usage of Grid and Cloud storage: a modification of the job workflow to take into consideration and perform different actions according to the kind of end storage system; the development of a generic interface (or API) that allows the transparent, seamless use of a Cloud storage system like if it was a traditional Grid storage system. Both solutions deliver an efficient way to enable users and resource providers to take advantage of the storage services offered by Cloud providers.

References

[1] Simson Garfinkel: *Commodity grid computing with Amazon's S3 and EC2*, February 2007, Usenix http://www.usenix.org/publications/login/2007-02/openpdfs/garfinkel.pdf
[2] EGEE Project: *An EGEE comparative study: Grids and Clouds – evolution or revolution?*, May 2008 https://edms.cern.ch/file/925013/4/EGEE-Grid-Cloud-v1_2.pdf

Issues of Processing Large Datasets in the Cloud

Kevin Glenny (1), Tomasz Mikolajczyk (1), Pawel Plaszczak (1)
(1) GridwiseTech, Krakow, Poland

The Cloud encompasses paradigms that were born out of Grid Computing, SOA, Web 2.0 and virtualization concepts. From these underlying concepts, usage models, such as Hardware-as-a-service, Platform-as-a-service and Software-as-a-service have been defined to simplify its acceptance within IT departments. When focusing on the processing of large datasets, the inherent challenges of the Grid still remain. Namely, how to create a seamless blend between efficient data processing, provisioning and storage. Industries, such as BioMedical, Manufacturing and Telecommunications process huge amounts of data, reaching into the petabytes. If this data is to be processed in the Cloud, it needs to get there first. How and where do the challenges lie? A real-world example will be presented that explores where the obstacles are when provisioning data within the Cloud.

One of the key things when discussing Cloud data processing is what software provisions the data. It is possible to use well known technologies ranging from database,s MySQL or Oracle; through to NoSQL community offerings, Google's BigTable [1] and Amazon's Dynamo [2]; or even less known data-grids such as Gigaspaces XAP. The challnages that are faced by all include: how to get the data into and out of the Cloud? Data transfer issues and QoS levels; data security and privacy issues; live migration benefits and issues; risk of down-time and preventative measures.

GigaSpaces XAP is a data-grid solution that provides high scalability and in-memory performance. Data is locally persisted with feeder applications uploading it to the Cloud enabled GigaSpaces XAP. Here the data is processed and a write-behind mechanism, implemented by a Mirror application, persists the data to an underlying storage. High Availability is intrinsic with backup replicas of data held on distinct nodes.

An example problem is illustrated with the use of a legacy infrastructure that uses two relational databases to perform end-of-day data aggregation of production line data. The company plans to increase production but are unable to process the data in time using the current infrastructure. This is because of database I/O bottlenecks and their inherent lack of scalability. A Cloud enabled GigaSpaces XAP is used to transfer the processing of data into the virtual space. A theoretically unlimited amount of data can be processed using the dynamic scalability of the Cloud. If more data is to be processed, then more GigaSpaces XAP instances are deployed.

Running GigaSpaces XAP in a virtualized Xen environment [3] was found to be slower compared to bare metal execution. However, the dynamic scaling linked to overall load provided optimal performance utilization. Additionally, this graceful scaling was found to allow the deployment to contract and expand according to production output. Data privacy and security considerations were encountered with the sensitivity of the data determining if it could be processed in the Cloud. The pay-per-use business model reduced CapEx and OpEx. Getting data into and out of the Cloud was found to be a bottleneck. As datasets grow in size it becomes optimal to physically mail the data rather than upload/download using limited bandwidth.

The Cloud provides an opportunity to gracefully scale out existing applications with minimal CapEx and OpEx outgoings. However, challenges remain that are still to be addressed, both political and technical. Political concerns are mainly focused on QoS, data security and privacy. Technical problems arise because of reduced performance as a result of virtualization and the difficulties in getting data into and out of the Cloud. Applications that constantly process new data reaching into the petabytes are currently unsuitable for the Cloud. Future work should be conducted that establishes a framework for Cloud data input and output.

References

1. Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber BigTable: A Distributed Storage System for Structured Data. OSDI 2006.
2. Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati,Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels. Amazon's Highly Available Key-value Store.
3. Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer†,

Ian Pratt, Andrew Warfield. Xen and the Art of Virtualization.

Design of a Platform of Virtual Service Containers for Service Oriented Cloud Computing

Carlos de Alfonso (1), Andrés García (1), Vicente Hernandez (1).
(1) Grid and High Performance Computing Research Group, Instituto ITACA - Universidad Politécnica de Valencia

In the last years Cloud Computing has become a topic of interest for the academic as well as enterprise environments, providing a growth of the research areas and the commercial applications in this field. The Cloud Computing concept has been defined several times [1], but currently it comprises three abstraction levels: (1) *Infrastructure as a Service* (provide bare metal machines), (2) *Platform as a Service* (provide a whole development and deployment platform) and (3) *Software as a Service* (provide a complete software suite ready to use), denoted as IaaS, PaaS and SaaS respectively. Those three levels share concepts as **virtualization**, **use on demand**, **scalability**, etc.
Out of these three fields, the one which has captured more attention is the IaaS due to the success of several commercial solutions [2] [3], arising new research projects and open platforms based on this concept [4] [5]. The other two levels have been paid less attention, driving to a lack of research projects in those fields. The current approach to deploy a SaaS solution is based on IaaS, by packaging a preinstalled and configured software product in a virtual machine image, and deploying it into an IaaS platform. This approach has some drawbacks such as the waste of resources. As an example, if we wish to deploy a simple web server in the cloud using this approach, we should deploy a virtual machine image with some software (web server, database system, etc.) preinstalled, and use a set of resources (CPU, memory, HDD) much higher than what we need, with the consequent waste of resources and the proportional economic cost.
In this paper we propose a higher abstraction level by using an IaaS style platform for virtualized services, instead of virtual machines, in the form of Virtual Containers instead of classic Hypervisors. The goal is to use virtual containers for deploying services, as IaaS is used to deploy virtual machine images. The objective is to save resources and to provide a better control over the services, suggesting a convergence of grid services and Cloud Computing. Using the Cloud Computing concepts, the system provides a PaaS platform where developers can deploy and manage their services, while the platform offers a transparent runtime environment allowing the services to be used in a *Software as a Service* basis.
This platform is designed around three major components, beside the basics such as Security, Storage and User Interface:

1. **Management and QoS**: This component enables the management of the deployed services, and ensures the Quality of Service of the platform, according a specific metrics. It allows service deployment, undeployment and live migration, as well as providing automatic load balancing, replication, fault-tolerance, etc. SLA integration is considered at this level, by allowing this component to manage user preferences and needs, specified by SLA agreements
2. **Catalog**: This component manages information about services and components, as well as system statistics and metrics. Virtual containers and system components report information about their state to the catalog, in order to keep it updated, while users make queries to look for services and get information about the system.
3. **Virtual Container**: Virtual containers are the base of the platform. They store, execute and give access to the virtualized services. A virtual container is a common service container, plus some extra functionality as service deployment, undeployment, live migration, statistics reports, etc. collocated services.

In this paper we present the design and specification of this platform and its components, giving a basic overview of the whole platform, followed by a more detailed view of each component. The paper concludes with the discussion about open problems (such as how services are integrated with the platform, or how it is carried out the live service migration) and outlines some other research lines to continue this work.

References

1. Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, Maik Lindner. A Break in the Clouds: Towards a Cloud Definition, ACM SIGCOMM Computer Communication Review, Volume 39, Issue 1, Pages 50-55, January 2009.
2. Amazon Elastic Compute Cloud (EC2), http://aws.amazon.com/ec2/, Retrieved 30/07/2009
3. GoGrid, http://www.gogrid.com/index.v4.php, Retrieved 30/07/2009
4. Eucalyptus, http://open.eucalyptus.com/, Retrieved 30/07/2009
5. OpenNebula, http://opennebula.org/doku.php, Retrieved 30/07/2009

From Grids to Clouds – Shift in Security Services Architecture

Syed Naqvi (1), Massimo Villari (2), Joseph Latanicki (3), Philippe Massonet (1)
(1) Centre d'Excellence en Technologies de l'Information et de la Communication, Belgium
(2) Università degli Studi di Messina, Facoltà di Ingegneria, Italy
(3) Thales European Research Centre for E-Gov & Secured Information Systems, France

Nowadays, the notion of Cloud computing is making significant headway in the computing landscape. Its emergence was initially seen as a complementary technology to address the major shortcoming of the Grid technology [1]. However, the business push to Cloud computing paradigm resulted in a more sophisticated concept of ***rent-a-technology*** that enabled a series of new networked business models. This emerging digital globalisation has not only raised the legal stakes and ownership concerns but also heralded the need of developing new security models to assure security and privacy of data in the peculiar characteristics of Cloud computing such as multi-tenancy, federation, workload mobility, etc. [2]

This paper presents an overview of the security requirements of the Cloud computing infrastructure. Some virtualisation challenges from the security point of view are also identified. A set of solutions is proposed to overcome these security challenges. These solutions are derived from a pragmatic threats analysis of a cloud infrastructure [3]. Cloud computing is the top of *virtualization-as-a-service* (VaaS) concept. However, virtualization itself introduces the threat posed by a new class of rootkit, called Virtual Machine based rootkit (VMBR) [4]. The results of this work show that the lifecycle of Cloud federation is quite different from the lifecycle of the virtual organisations used in the Grids [5] and therefore Cloud security cannot be assured only by the portability of security solutions developed for its ancestral distributed technologies. It is found that the Cloud deployments require more comprehensive access control and data confidentiality models. The risks associated with the deployment of Clouds infrastructure include privileged user access, regulatory compliance, data location, data segregation, recovery, investigative support, long-term viability. The bottom line of this work is to show that comprehensive security solutions do not compromise the performance and flexibility of the Cloud infrastructures.

Acknowledgements: The research leading to the results presented in this paper has received funding from the European Union's seventh framework programme (FP7 2007-2013) Project RESERVOIR under grant agreement number 215605.

References

1. Foster I., Zhao Y., Raicu I., Lu S., Cloud Computing and Grid Computing 360-Degree Compared, IEEE Grid Computing Environments (GCE08) 2008, co-located with IEEE/ACM Supercomputing 2008
2. Naqvi S., Massonet P., Latanicki J., 'Challenges of Deploying Scalable Virtual Infrastructures – A Security Perspective', CESNET Conference on Security, Middleware and Virtualisation, Prague, Czech Republic, September 25-26, 2008
3. Massonet P., Naqvi S., Tusa F., Villari M., and Latanicki J., Mitigating Security Threats to Large-scale Cross Border Virtualization Infrastructures, International Conference on Cloud Computing, Munich, Germany, 19-21 October 2009
4. King, S., Chen, P., SubVirt: implementing malware with virtual machines, IEEE Symposium on Security and Privacy 2006, Berkeley/Oakland, CA, USA, 21-24 May 2006
5. Naqvi S. Mori P., 'Security and Trust Management for Virtual Organisations: GridTrust Approach', IFIP Advances in Information and Communication Technology series IFIP-AICT-300, IFIP

International Conference on Trust Management 2009 (IFIPTM'09), West Lafayette, Indiana, USA, June 15-19, 2009. ISBN 978-3-642-02055-1

Virtualization - What Can We Learn from Commercial Datacenters?

Igor Wawrzyniak
Wroclaw Centre for Networking and Supercomputing, Wroclaw University of Technology, Poland

Business and scientific organizations solve the same task - accessing distributed, shared computer resources - in a different way. In commercial environment, it is common to employ virtualization platforms (hypervisors) that allow to run a complete operating system. Cloud computing platforms, running on top of clusters of virtualization-enabled servers, allow rapid provisioning of VMs and flexible infrastructure reconfiguration. Scientific communities use grid middleware that ditributes the tasks between nodes in the grid or cluster. This allows for high-performance computation, but severely limits available software and requires active support from grid operators.

In the recent years, several experimental "scientific clouds" were set up. An example is Eucalyptus platform [1], which started as a research project on University of California, Santa Barbara, Computer Science Department and was later commercialized. The same software was used in Baltic Cloud [2] – a joint programme of Estonia, Latvia, Lithuania and Belarus. Another software platform created with the scientific community in mind is Nimbus [3], created by the Globus Alliance, known for it's grid middleware.

Cloud computing approach can be used instead or in addition to existing grid platform. First way mimics commercial systems such as Amazon EC2 - a de facto standard for both commercial and academic clouds - and allows users to control the whole execution stack down to the level of operating system running on a virtual machine. The latter is to run a grid middleware on virtual instead of physical machines. While it wastes some processing power due to virtualization overhead, it greatly simplifies administration and reduces downtime. This leads to a better utilization of resources and allows to finish more tasks in the same time. Both of this approaches have been described in the scientific papers, eg. [4] and [5].

Acknowledgements. Research on virtualization in grid computing is conducted as a part of PI-Grid initiative.

References

1. http://open.eucalyptus.com/
2. http://cloud.balticgrid.eu/
3. http://workspace.globus.org/
4. I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, X. Zhang: Virtual Clusters for Grid Communities; In Proceedings of the Sixth International Symposium on Cluster Computing and Grid (CCGRID), IEEE Computer Society, 2006.
5. Peter A. Dinda, Renato J. Figueired, Jose´ A. B. Fortes: A Case For Grid Computing On Virtual Machines; Technical Report TR-ACIS-02-001; August 2002.

Introduction to the IBM BlueGene/P Installation at the Forschungszentrum Julich

Lukas Arnold

The IBM Blue Gene/P (BG/P) installation at the Forschungszentrum Jülich (FZJ) is the largest BG/P installation worldwide. It contains about 300k cores providing a peak performance of 1 PF/s. However, also the infrastructure around such a large installation is of significant importance. Therefore the FZJ's HPC infrastructure is introduced.

As a prelude for the BG/P workshop, this talk will briefly illustrate the main characteristics and setup of the BG/P architecture. The MPP system consists of rather weak compute nodes, running at 850 MHz with a peak performance of 3.4 GF/s, which are tightly coupled via a three dimensional torus network. Due to the low clock frequency this system is very balanced and power efficient. The design of this architecture focuses on extreme scaling. It provides no hardware or operating system barriers or resistance for application's scaling on the full petaflops machine.

NEC SX Installation at High Performance Computing Center Stuttgart and its Applications

Harald Klimach

This talk is a short introduction to the NEC SX9 vector system. The NEC SX9 is the latest incarnation of the NEC SX series and is the only remaining traditional large vector system for HPC on the market today. It provides an amazing system with important features. These features are briefly presented to provide a basic impression of this system. As an example installation the HLRS system Baku is described.

The main focus of this talk is on the powerful vector processor, with a theoretical floating point peak performance of 102.4 GFLOPs and the single node layout with its memory system. As this is a unique feature in this system and yields the largest differences to more common hardware architectures. The vector characteristics are also gaining importance on most other modern architectures which is elaborated in the talk.

Green Computing in Practice. Cons and Pros of Modern System Architectures

Radosław Januszewski
Poznań Supercomputing and Networking Center, Poznań, Poland

The continuous progress in many branches of science leads to almost unlimited demand for computational power.

For the last several years the dominant position in CPU architectures was taken by the Intel x86 architecture because it's relatively low cost, very decent performance and steady increase of the computational capabilities.
Because the main advantage of this approach was the price, the inexpensive servers featuring one or two CPUs became the most popular choice for capability computing. Constant optimizations introduced in the implementations of the x86 servers were mostly aimed at delivering maximum CPU performance in very small packaging. This approach resulted in power-hungry installations consuming unthinkable few years ago amounts of electrical power.
When aiming at future multi-petaflop computing environments, we have to consider a new approach to the way the price/performance ratio is defined / evaluated / calculated. In addition to common factors, the one related power consumption has to be taken into account as well. This consideration was the reason for the research on power-efficient solution activity in the PRACE project.
The work done within the confines of this activity has focused on two major approaches that were considered most rational.
The first one is to investigate the existing server architectures scrutinizing the power consumption profile, going down to the consumption of the individual sever components. This allowed us to identify the elements that are the hot spots of the machine, crucial to very large, power efficient installations. We have also tested several packaging solutions for the servers (i.e. several stand-alone servers, blade solutions) to identify whether the progress in power efficiency promised by the hardware vendors is a reality or myth.

The second approach was to assess the less traditional computing solutions, where instead of the general-purpose CPUs, be it x86, PowerPC or others, more specialized hardware is being used. We have performed a series of test on such platforms. The hardware set chosen for the tests encompasses both solutions created with the HPC in mind, such as ClearSpeed accelerators or SiCortex machine, and solutions derived from the areas where the raw computational power was crucial but only recently the hardware solutions matured in a sufficient degree that allowed for application in the HPC. An FireStream and Tesla solutions were selected as example for this class of hardware.

As the results of the research we are showing the potential and real power consumption related benefits that we were able to gain using different architectures, hardware and software components. The results are coupled together with the assessment of the usability of each solution understood as the effort required to get the benefits.

The work we have done does not exhaust the problem because the subset of solutions were limited to the hardware available at the moment. Nevertheless, because the subject will become increasingly important as number of machines with power requirements of several GWatts will grow, the impact on the natural environment and the operation costs will make the power efficiency factor one of the most important features of the computing solutions.