POSTER PRESETATIONS
(ABSTRACTS)

Aula, 1st floor

1. Chemomentum Virtual Organization Services

Krzysztof Benedyczak, Piotr Bala
Faculty of Math. and Computer Science, N. Copernicus University and ICM Warsaw University, Poland

Within the Chemomentum project we have designed and implemented novel Virtual Organizations service (VO service). The VO features which in general include user and their privileges management are used for for authorization purposes but can be also used with other applications as support of authentication in WWW environments.

The Chemomentum VO service is founded on well established attribute based authorization pattern. This pattern states that entity must posses set of attributes in order to get positive authorization decision. In this case VO service acts as Attribute Information Point (AIP). It allows also for identity mapping supporting different types (or formats) of identity tokens, describing the same entity.

Both features can be easily consumed by authorization/authentication points, assuming usage of well established and standard protocol. In particular, Chemomentum VO Services uses SAML 2.0 as querying interface. Additional SAML profiles are used to achieve high interoperability level.

Different identity types are supported. Those include email, distinguish name, and full blown X509 certificates (more can come if needed).

VO service uses group hierarchy to help management of users. VO contents managers are able to assign permissions and attributes by using groups and related concepts like inheritance.

Arbitrary attributes can be used without any limitations. All attributes can be assigned to entities globally or only in the scope of some group.

Chemomentum VO Service holds it's detailed history. It allows to get snapshot of previous state of the any VO maintained contents, what can be useful for tracking/reverting changes.

Because of modular design and standard interface Chemomentum VO is an easy to install and use replacement of other VO systems.

This work is supported by the EU IST grant Chemomentum.

2. ViroLab Virtual Laboratory

Tomasz Gubala (3,2), Bartosz Balis (3), Maciej Malawski (1), Marek Kasztelnik (3), Piotr Nowakowski (3), Matthias Assel (4), Daniel Harezlak (3), Tomasz Bartynski (3), Joanna Kocot (3), Eryk Ciepiela (3), Dariusz Krol (3), Jakub Wach (3), Michal Pelczar (3), Wlodzimierz Funika (1), Marian Bubak (1,3)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) Informatics Institute, University of Amsterdam, The Netherlands
(3) ACC CYFRONET AGH, Krakow, Poland
(4) HLRS High Performance Computing Center Stuttgart, University of Stuttgart, Germany

Today scientific research frequently appears as a joint effort of multiple scientific teams and institutions. The integration of those parties usually takes place in a form of scientific projects and the process of collaboration takes place in different channels, both direct and indirect, virtual. In order to reach their scientific goal there se archers also use computational models and simulations – in fact, more and more of their tools, documents, results and contacts are performed, stored and shared in computers. The ViroLab Virtual Laboratory [1] presented in this work brings a platform of cooperation for scientists of multiple expertise and locations on common scientific goals.

The main objective is to deliver an environment that could combine efforts of computer scientists, virology and epidemiology experts and experienced physicians to support future advances in HIV-related research [2]. We partially apply ideas and methods developed and proven in successful environments like Kepler [3] or MyGrid's Taverna [4] and we enrich them with new unique features.

The main strength of the virtual laboratory comes from the support for different types of users and mechanisms to integrate their tasks together (like in reality different workers are required for a laboratory to function). The presented system provides means for the scientific programmer to plan in-silico experiments and for the scientists to use these experiment plans to perform complex research. Therefore a set of dedicated tools need to be provided (as different types of users need specific, targeted approach) in a tightly integrated fashion (as they all still work together in the same research endeavor). The two main subject of cooperative efforts in the ViroLab Virtual Laboratory are experiments and their results. The experiment developers use their technical skills and the knowledge of the virology domain to plan new experiments while the researchers use the planned experiments to obtain scientific results.

For a successful virtual experiment three main factors are needed: the input data that is being analyzed, the analytic modules that perform some task of that data (simulation, transformation, inference, etc.) and the "glue" logic that combines these elements into a well-defined process of specific experiment. The laboratory provides the Data Access Service to obtain important HIV-related data from remote databases (e.g. hospitals) and the Grid Operation Invoker to call remote computing elements that perform particular tasks on the data. Since many sophisticated tools (simulators) require significant computing power the laboratory provides access to both powerful computational testbeds using the Grid technology (like the EGEE project [5] testbed) and stand-alone, specialized computation servers (using MOCCA [6] or WebServices technologies). Moreover, a specialized, built-in optimization module (GridSpace Application Optimizer) takes care of optimal load-balancing on smaller computation servers and machines. As the notion of experiment repeatability and the provenance of results is very important for scientists the provenance tracking, recording and publishing system PROToS is also provided in the presented platform. Users are able to find past experiments, browse archived data and track origins of certain experiment results.

Since the system is fairly complex there are dedicated tools to help the users to perform their tasks. The Experiment Planning Environment is an Eclipse-based development tool that aims specifically at supporting quick yet not restricted process of experiment plan development: it uses the Ruby scripting language as the main development platform for the future experiments. On the other hand, the scientists use their web browsers to access the web-based Experiment Management Interface where one may load an experiment, execute it and collect results for off-line analysis. This interface is accompanied by a QUaTRO query tool that allows the use of the provenance system to ask for past experiments and their results. A set of web-based collaboration tools that help scientists exchange such results, observations and opinions is also being prepared to enrich the environment functionality.

In order to achieve the sufficient level of integration of all these components, several additional modules are in place. The Experiment Repository stores experiment plans prepared by developers and published for scientists. The GridSpace engine provides the experiment execution capabilities and the Grid Resources Registry holds information on what computing elements are available and where. The planned Laboratory Data Base will hold the results obtained by scientists so they are not forced to save them locally on their own in order to use them later.

The development of the virtual laboratory is quite advanced and the first versions of the core modules and the end-user tools are already released [1]. The developers are closely cooperating with virology scientists and scientific programmers to steadily improve the quality and usability of the platform. The current prototype was already used to plan and execute important virological experiments that perform various types of analysis on an HIV virus genotype (including interpretation of resistance values to HIV drugs).

New functionality in future versions of the virtual laboratory, among others, will include the mechanisms for resources and experiment monitoring to gather better status and provenance information, an experiment result management system that helps scientists to store, revoke, share and comment on experiment results and wider support for high-performance computing platforms like EGEE or DEISA. As the ideas and design concepts that are the basis for this laboratory are more general, the future applications in other fields of eScience are possible and planned.

Acknowledgements. This work was supported by EU project Virolab IST-027446 and the Foundation for Polish Science.

References

1. ViroLab Vitual Laboratory; virolab.cyfronet.pl
2. ViroLab – EU IST STREP Project 027446; www.virolab.org
3. B. Ludäscher, I. Altintas, et. al., Scientific Workflow Management and the Kepler System, Concurrency and Computation: Practice & Experience, 18(10), pp. 1039-1065

4. R. Stevens, et. al. Exploring Williams-Beuren Syndrome Using myGrid Bioinformatics 20 Suppl. 1 2004, i303-i310
5. Enabling Grids for E-sciencE, http://www.eu-egee.org/
6. MOCCA Metacomputing-oriented CCA Framewor, http://www.icsr.agh.edu.pl/mambo/mocca

3. Development and Execution of Collaborative Application on the ViroLab Virtual Laboratory

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

Modern practices of science in such area as investigation of HIV virus drug resistance, requires collaborative sharing, processing and analyzing virological, immunological, clinical and experimental data, as well as advanced tools for (bio) statistical analysis, visualization, modelling and simulation [1]. A process integrating these computational tools and data, which leads to obtaining results relevant to the application domain, is called in-silico experiment. An experiment in virtual laboratory combines data and activities which are available on the distributed Web- and Grid-based infrastructure and it needs to orchestrate them in possibly complex scenarios.

As existing solutions to the problem of experiment orchestration, there are many scientific workflow systems available for the Grid, such as Pegasus [2], Triana [3] and K-WfGrid [4] systems. They are intended to assist non-programmer users in developing applications, however in the case of workflows with many components and complex interactions, they can also become difficult to use and lack expressiveness.

To overcome the limitations of workflow systems, we decided to define an experiment plan notation based on a high-level scripting language, namely Ruby. An experiment plan is a Ruby script, which features the concise and clear syntax combined with a full set of control structures, allowing expressing experiments from very simple to complex ones. In order to relieve the experiment developer from the sophisticated details of the underlying grid infrastructure, we introduce a high-level object-oriented API, which allows requesting "which" computational functionality is required, without a need to specify "how" to access it with available middleware. On the other hand, if a developer needs to retain a full control over the experiment plan, it is possible to specify all the technical details on the lower level of abstraction.

In order to support development and execution in this high-level scripting paradigm, a set of tools in the virtual laboratory has been developed. An experiment developer uses an Experiment Planning Environment (EPE)based on Eclipse platform, which offers user friendly editor for writing scripts, integrated with a set of tools. The developer can use the semantic-web based Domain Ontology Store graphical browser to discover the available data and computational services, coupled with Grid Resources Registry which provides the available operations which can be invoked directly from a script. A script can be executed from the EPE, which is integrated with the GridSpace Engine (GSEngine) being a core of the runtime system. GSEngine includes the Grid Operation Invoker [2], which translates high-level operations specified in the script into concrete invocations on computational resources using appropriate technologies. It is possible by using the information stored in the registry and with the assistance of Grid Application optimizer, which automates the process of optimal resource selection.

The process of experiment planning and execution is collaborative in the sense that the virtual laboratory supports cooperation of multiple experiment developers and users. When a developer prepares an experiment script, it can be published in the experiment repository and thus made available to others. Then, a scientist who does not intend to get into the details of scripting can access the virtual laboratory through a portal, and execute the published experiments using Web browser, providing only input data when necessary. The main idea is that the experiments in the repository can be shared and reused, which is the important means to promote collaboration between scientists. Provenance data on the experiment is also recorded and available for queries, making the results more reliable, reproducible and scientifically relevant.

The unique features of the virtual laboratory developed for ViroLab are that by using a scripting language it is possible to define possible to define even complex experiments easily, still remaining on a high-level of abstraction and concealing the details of underlying grid middleware. By providing a set of user friendly tools, both advanced experiment developers and domain scientists can productively collaborate and conduct their research in modern higly distributed environment.

Acknowledgements. This work was supported by EU project Virolab IST-027446 with the related Polish grant SPUB-M and by the Foundation for Polish Science.

References

1. 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

2. Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, and Miron Livny. Pegasus: Mapping scientific workflows onto the grid. In Grid Computing: Second European Across Grids Conference, AxGrids, volume 3165 of Lecture Notes in Computer Science, pages 11-20. Springer, 2004.
3. an Taylor, Matthew Shields, Ian Wang, and Andrew Harrison. Visual grid workflow in Triana. Journal of Grid Computing, 3(3-4):153-169, September 2005.
4. Tomasz Gubala and Andreas Hoheisel. Highly dynamic workflow orchestration for scientific applications. In CoreGRID Intergation Workshop 2006 (CIW06), pages 309-320. ACC CYFRONET AGH, 2006.
5. Tomasz Bartynski, Marian Bubak, Tomasz Gubala, Maciej Malawski: Universal Grid Client: Grid Operation Invoker; Proceedings of International Conference of Parallel Processing and Applied Mathematics (PPAM'07), Gdansk, September 2007, LNCS (to appear).

4. Developer and User Interfaces to the Virolab Virtual Laboratory

Wlodzimierz Funika (1), Daniel Harezlak (2), Dariusz Krol (2), Piotr Pegiel (2), Marian Bubak (1,2)
(1) Institute of Computer Science AGH, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

In complex software infrastructures like virtual laboratories, such as Virolab [1], specialists from different fields of science need on one side a user-friendly interface specific to their domain, on the other side a convenient interface for sharing experience and results within a domain and between domains. Experiments developers need a sort of an experiment development environment that facilitates the development process (planning) of experiments. On the other hand, experiment end users need an easily accessible interface for experiment management – to find and execute predesigned experiment plans, manage the results obtained, and send their comments to the developers. The presented work addresses the above issues.

The first group of users of the ViroLab Virtual Laboratory are experiment developers, who combine their domain knowledge and technical skills to plan and develop new experiments. However, they do not need to fully comprehend all the data and results of a specific experiment, provided there is appropriate expert (scientist) assistance available to evaluate the developer's work. ViroLab Experiment Planning Environment (EPE) is dedicated for experiments developers, being based on the Rich Client Platform which is part of the Eclipse project[2]. The latter feature ensures that EPE can be extended with new functionality, in form of pluggable component scalled plugins. The main purpose of EPE is to provide a powerful GScript editor, aimed to make easier the experiment development process: EPE provides syntactical auto completion and enables developers to retrieve and modify existing experiments from within a repository.

The second group of users of the virtual laboratory comprises the experiment end users who exploit the Experiment Management Interface (EMI) to manipulate the queue of run experiments. These users consist mainly of clinicians and doctors of specific medical domains, who are not specialists in scientific computing. Therefore a challenge is to develop a generic interface for experiment management. The analysis of user requirements allowed to distinguish four main tasks of the experiment management interface: The first task is to locate a desired experiment via browsing through experiment repositories and then through experiment versions. The next task is to manage the experiment execution queue via access to the currently running experiments and their status. The third task is to gather, store, and share results. Finally, it is also required to be able to submit feedback about the quality of an experiment to its developers. A major requirement in addition to the aforementioned one is to deliver the interface in the most user-friendly way, in terms of accessibility and usability. To accomplish this goal, the interface is implemented as a lightweight web application. The concept of thin client enables users to manage their experiments even through mobile devices like PDA, usually lacking large computational power.

Since the described interfaces, EPE and EMI, are decoupled, it is necessary to identify a common point of integration. This role is played by the experiment repository which is used, on one side, by EPE to store and manage versions of developed experiments and, on the other side, by EMI to provide details on available experiments. The presented approach allows to access more than one repository. Thus repositories constitute external dependencies for both interfaces.

The prototypes of both Experiment Planning Environment and Experiment Management Interface have been implemented and integrated with other components of the ViroLab Virtual Laboratory [3] and are being improved according to the feedback from end users. Both groups of users successfully use and test the interfaces, thus cooperate on new scientific experiments.

Acknowledgments. This work is supported by the EU IST project Virolab 027446.

References

1. 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
2. W. Funika and P. Pegiel, GScript Editor as Part of the ViroLab Presentation Layer, in: M. Bubak, M. Turala, and K. Wiatr (Eds.). Proceedings of Cracow'06 Grid Workshop, Oct 15-18, 2006, Cracow, Poland, ACC Cyfronet AGH, 2007, pp. 320-327.
3. ViroLab Virtual Laboratory, http://virolab.cyfronet.pl

5. Virtual Laboratory Engine – GridSpace Engine

Eryk Ciepiela (1), Joanna Kocot (1), Tomasz Gubala (1,3), Maciej Malawski (1), Marek Kasztelnik (1), Marian Bubak (1,2)
(1) ACC CYFRONET AGH, Krakow, Poland
(2) Institute of Computer Science AGH, Krakow, Poland
(3) Informatics Institute, University of Amsterdam, The Netherlands

The major challenge that the ViroLab Virtual Laboratory undertakes is to bring into action the capabilities of Grid environment for virological experiments. When enabling Grid fabric its complexity and heterogeneity have to be covered by a convenient and uniform way of experiment launching. The idea behind the GridSpace Engine is to decouple logically a client that requests an execution of an experiment from an engine that actually carries out the experiment. Aside experiment execution service, GridSpace Engine is considered as well as a façade for all the users who would like to access specialized Virtual Laboratory services, such as Data Access Service.

GridSpace Engine as a shared, reliable, efficient and high-available service may be also physically separated from the experiment owner workstation. Such an approach delivers the services as a facility for dispersed groups of (possibly mobile) users who employ computation, memory and time-consuming experiments in their research. For them GridSpace Engine constitutes an entry point to the ViroLab Virtual Laboratory.

As the spectrum of users is wide, ranging from scientific programmer to clinical virologists, the GridSpace Engine could be interfaced with different, task-oriented tools. Since GridSpace Engine itself provides solely a Java API along with a basic command-line tool for technically-advanced users (like programmers), the elaborated client applications are built upon it such as integrated Experiment Planning Environment nased on Eclipse platform or Experiment Management Interface portlet in the Virtual Laboratory Portal.

GridSpace Engine is a runtime environment for Virtual Laboratory experiments, managing the session of the experiment execution. It provides the scope of the user context (including security credentials) as well as experiment execution context (indispensable e.g. for the monitoring and provenance events correlation) and monitors the course of the experiment along with involved actions. Moreover, such an environment enables remote computation capability and an access to shared data since in a scope of the same experiment process. That functionality is provided by Grid Operation Invoker and Data Access Client, respectively, which are integrated internally as runtime libraries.

We plan to deploy the engine as a stateful service that is accessed in an asynchronous and not-blocking manner. It will support bidirectional data streaming with clients and should introduce a mechanism for interactive data input. However, it should support the offline mode as well – while client application is temporarily not available, the execution of experiment remains uninterrupted and the data streams are buffered and flushed as client reappears. This capabilities are supported by a dedicated access protocol with a specification that implies interoperability with a wide range of client application technologies. The protocol will be connection-oriented and will support multiple conversation modes (querying, streaming, notification) along with multiplexing.

The core building block of GridSpace Engine is an experiment evaluator based on a Java-based implementation of a Ruby language interpreter, namely JRuby. Is extended with the variety of experiment plan extractors to retrieve experiment script from the repository. The engine incorporates also the Virtual Laboratory runtime libraries supporting such functionality as distributed data access and innovation of grid operations on computational resources.

The ability to serve in both online and offline mode with support for various, dedicated client applications makes the GridSpace Engine a centerpiece of the collaborative virtual laboratory of ViroLab. The goal of open specification of the access protocol is to significantly improve its interoperability and to allows for future extensions and refinement.

Acknowledgements. This work was supported by EU project Virolab IST-027446 with the related Polish SPUB-M grant and by the Foundation for Polish Science.

References

1. ViroLab Vitual Laboratory; virolab.cyfronet.pl
2. ViroLab – EU IST STREP Project 027446; www.virolab.org

6. Invocation of Grid Operations in the ViroLab Virtual Laboratory

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

Diversity of middleware technologies allowing access to computational resources in Web and Grid based infrastructures [1] results in the fact that utilizing them in a coordinated way becomes a challenge. Such a problem is faced by the virtual laboratory environment developed for the scientists involved in HIV drug resistance research in the scope of ViroLab [2] project. The virtual laboratory is centered around the idea of programming experiment scenarios in a scripting language (Ruby) on a high level of abstraction. Therefore there is a need to provide means to uniformly interface existing middleware technologies using a high-level object oriented API.

We have identified four major types of technologies, which can be used as the means to provide access to computation. The first one is the Web Services technology, based on SOAP and WSDL, and provides stateless remote invocation and messaging semantics. As the second one, there is WSRF, which is based on extensions of Web Services providing standardized access to stateful entities (resources). Third one is the component technology, which is in the research interest of European CoreGRID project and implemented in such technologies as MOCCA and ProActive. Last, but not least, having access to computational resources in the infrastructures such as EGEE and DEISA requires interfacing with their middleware, which is based on a model of batch processing of jobs.

As a solution to the problem of interfacing diverse technologies, we introduce grid object abstraction level hierarchy. Each grid object class is an abstract entity which defines the operations which can be invoked from the script. Each class may have multiple implementations, representing the same functionality, but different technologies. Next, each of the implementations may have multiple instances, possibly running on different resources, thus differing in performance. Grid objects may have different properties, such as stateless or stateful interaction mode and being private or shared between experiments runs and users.

In order to enable such a high-level programming, the virtual laboratory engine includes as its core a Grid Operation Invoker (GOI) [3] module. Its goal is to instantiate grid object representatives (proxies) and to handle remote operation invocations using appropriate technologies. GOI handles different technologies by pluggable adapters implemented in JRuby, which facilitates integration with Java client-side libraries. In the case of job-based middleware, it is possible to create grid objects which expose the functionality through application-specific method invocation in object-oriented style.

Grid Operation Invoker has been implemented and integrated with virtual laboratory engine [4], a registry providing technology descriptions and an optimizer which assists in choosing the best instances of grid objects. Currently available adapters are for Web Services, MOCCA [5] components and LCG jobs. The first experiments in the ViroLab virtual laboratory demonstrated successfully the possibility of interoperability between these technologies. Now, adapters for WSRF, AHE and Unicore middleware are under development.

Acknowledgements. This work was supported by EU project Virolab IST-027446 with the related Polish grant SPUB-M and by the Foundation for Polish Science.

References

1. Gentzsch, W., "Major grid projects around the world", Parallel and Distributed Processing Symposium, 2006.IPDPS 2006. 20th International Conference, p.1, 25-29 April 2006
2. ViroLab – EU IST STREP Project 027446; www.virolab.org
3. Tomasz Bartynski, Marian Bubak, Tomasz Gubala, Maciej Malawski: Universal Grid Client: Grid Operation Invoker; Proceedings of International Conference of Parallel Processing and Applied Mathematics (PPAM'07), Gdansk, September 2007, LNCS (to appear).
4. ViroLab Virtual Laboratory, http://virolab.cyfronet.pl
5. MOCCA – A CCA component framework for metacomputing, http://www.icsr.agh.edu.pl/mambo/mocca

7. Optimization of Application Execution on the ViroLab Virtual Laboratory

Maciej Malawski (1), Joanna Kocot (2), Eryk Ciepiela (2), Iwona Ryszka (2), Marian Bubak (1,2)
(1) Institute of Computer Science AGH, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

The objective of presented work is to build an optimization engine for the ViroLab Virtual Laboratory Runtime [1]. Its specific model – invoking operations on special objects which reside on Grid resources – imposes a new approach to optimization. Such an approach is presented in the shape of the GridSpace Application Optimizer (GrAppO) – an engine for optimizing Grid applications execution, designed especially for the purposes of the ViroLab Runtime.

The problem of optimizing application execution and resource utilization is a subject of wide research on Grid scheduling and load-balancing [3]. The challenges are e.g. dynamic nature of the environment, distributed sources of information, difficulty in defining suitable criteria. GridSpace Application Optimizer has to face the challenges imposed not only by the Grid environment but also those specific only to ViroLab [2] and related to the specific programming model with multiple levels of abstraction.

According to the abstractions introduced in GridSpace, GrAppO is designed to answer the question which grid object instance should be chosen for invocation of operation on a specified grid object class. There may be many instances corresponding to one grid object class, with the same functionality, but of different performance. The decision produced by GrAppO is based on the information retrieved from the registry and monitoring components of virtual laboratory, but also depends on implemented policy. Optimization modes offered by GrAppO include a short-, medium- and far-sighted ones. In the first one an optimum solution is chosen only for one Grid Object Class at a time, the second mode finds solutions for a group of Grid Object Classes without reordering tasks or using queues. The last one would require the whole application to be analyzed and ordering the Grid Object Classes taking into account dependencies between them.

As the experiments with GrAppO demonstrate, the important source of information is the monitoring system. Here, we report on the research in the area of monitoring of infrastructure and applications in the virtual laboratory. It is intended to provide not only up-to-date information on the resources for the GrAppO optimizer, but also to assist the application developer and user with the process of monitoring the application execution. Therefore a challenge for the monitoring system is to be interoperable with the variety of grid object technologies supported, such as Web Services, LCG jobs, MOCCA components as well WSRF and AHE in the near future. Such a system has to preserve high level of abstraction of grid object notion, but also has to enable to get low-level and technology-specific information about the monitored entities when needed.

In addition to GrAppO, the monitoring information is used also by a provenance tracking system. Its responsibility is to collect and store significant data about the course of the applications. The last, but not least party that employs monitoring system are application users themselves. They have to be equipped with the tools enabling multilevel insight into application execution. In order to achieve it the proper means of instrumentation, monitoring data exposure, and monitoring data accessing are provided by the dedicated agiLE MONitoring ADherence Environment (LEMONADE), which applies the Aspect-Oriented Programming (AOP) approach.

The prototype of GrAppO was implemented and integrated with the Virolab virtual laboratory. The results of experiments on simulated data show the usability of the proposed optimization modes. The future work includes integration of the monitoring system with the optimizer and the research on the far-sighted optimization mode. The monitoring system is planned to be extended to support broader range of technologies available in ViroLab

Acknowledgments. This work was supported by EU project Virolab IST-027446 with related Polish grant SPUB-M and the Foundation for Polish Science.

References

1. ViroLab Virtual Laboratory, http://virolab.cyfronet.pl
2. ViroLab - EU IST STREP Project 027446; www.virolab.org
3. Nabrzyski J., Schopf J.M., and Weglarz J. (eds): Grid Resource Management. State of the Art and Future Trends, Kluwer Academic Publishers, 2003.

8. Provenance Tracking and End-User Oriented Query Construction

Bartosz Balis (1,2), Marian Bubak (1,2), Jakub Wach (2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

Users of modern e-Science systems requires high level services, developed on top of Grid environments, to conduct advanced experiments and explore large data sets. Provenance, defined as the process which led to the creation of a piece of data, is gaining more and more attention from developers. Tracking provenance of experiment results is essential for contemporary systems, which allows to conduct in silico experiments. Scientists, involved in HIV related research in the scope of ViroLab [2] can find relevant information by exploring and querying repositories of provenance (containing for example meta data describing how results of experiments were obtained). Modern technical solutions, used by provenance tracking solutions are quite complicated, especially when expressing advanced provenance queries. For example, main query language used by our solution (PROToS), XQuery, has very user friendly syntax, but still strict and not suitable for non-IT specialists who need access to provenance data repositories. Most users of the ViroLab Virtual Laboratory will not be familiar with those details in either technical or conceptual sense. Hence, there is a great need for supporting tools, enabling easy construction of complex queries over data and provenance, understandable for non-IT specialists.

Before designing and implementing our system, we have carefully studied other solutions and grid systems with provenance tracking, like myGrid [5] and Triana [6]. What is more, we also put into consideration models of provenance proposed by other authors, such as VDM [7] and p-assertions [8]. This research allowed us to tailor best solution for the ViroLab and avoid common provenance-related pitfalls.

For exceptional requirements of the ViroLab users, we have built system for tracking, gathering and retrieval of provenance information, called PROToS (PROvenance Tracking System). Our solution is based on the semantic grid concept as an infrastructure for e-Science (defined in [1]). Provenance data is represented as XML/RDF and modeled using ontologies described in OWL. From the technical point of view, PROToS is a knowledge-based, lightweight, and fully distributed set of components responsible for gathering provenance data, exposing it for later mining and building advanced provenance queries. Whole system was designed with storage space and query processing speed in mind. It features many modern solutions, such as ontology processing logic and XML storage and XQuery support. PROToS architecture and general overview of the ViroLab provenance infrastructure could be found in [3].

Following needs of powerful yet simple solution for provenance mining, we have taken an end-user oriented approach to querying repositories of data and provenance in the ViroLab environment. Our solution is based onontology models describing all domains – in silico experiments, data, and biological applications running in the Virtual Laboratory. Data-domain ontologies, contains mappings to underlying physical data models, allowing queries about repositories of data and provenance in a unified way, or even combine provenance and data aspects in one query. On top of these ontology models, we have placed so-called QUaTRO (QUery TRanslation tOols), which allow to construct complex queries over both data and provenance repositories, expressed in the terms of the domain familiar to end users.

From the technical point of view, QUaTRO is a set of advanced components, defining PROToS user interface. Each tool is equipped with carefully designed Graphical User Interface in form of portlet that will be integrated in the ViroLab portal. Design of the QUaTRO GUI aims at ease of use without sacrificing ability to create complicated queries. Currently we have managed to implement first QUaTRO tool. It allows to construct queries in a wizard-based manner, starting from an ontology concept and following its object and data properties, building query in form of a tree. User can set attributes values in relation to pieces of data, loaded automatically by using ViroLab Data Access Client (DAC). This allows user to choose existing parameters for new queries, that can be satisfied. Great details of this tool, with example queries and application screenshots are to be found in [4].

Future works concerning core PROToS system concentrate on better performance with heavy workloads and additional ontology-processing features, like reasoning and support for ontology languages. Internal provenance repository will be extended into a multi-node, hierarchical storage. This is a major task, because enabling PROToS for new VL applications will generate tremendous amounts of data that will certainly exceed capability of one node storage. On the QUaTRO front we are firstly going to extend current tool with new features, suggested by target users during series of demonstrations. This includes additional operators, aggregation functions, and ability to define relation between attributes in different query branches. Long term plans involve development of next tools supporting query construction in other ways, for example using a natural language and text mining to identify ontology concepts to be used.

Acknowledgements. This work was supported by EU project Virolab IST-027446 with the related Polish grant SPUB-M and by the Foundation for Polish Science.

References

1. D. de Roure, N.R. Jennings, N. Shadbolt, The semantic grid: a future e-science infrastructure, Grid Computing-Making the Global Infrastructure a Reality, Wiley, 2003, pp. 437-470.
2. ViroLab Virtual Laboratory, http://virolab.cyfronet.pl
3. B. Balis, M. Bubak and J. Wach. Provenance Tracking in the ViroLab Virtual Laboratory. In Proc. PPAM 2007 Conference. Springer.
4. B. Balis, M. Bubak and J. Wach, User-Oriented Querying over Repositories of Data and Provenane. e-Science 2007.
5. myGrid, A. Robinson, R.D. Stevens, and C.A. Goble. myGrid: Personalised Bioinformatics on the Information Grid. In Proc. 11th International Conference on Intelligent Systems in Molecular Biology, June 2003.
6. Triana I.Taylor, M. Shields, I. Wang, and A. Harrison. The Triana Workflow Environment: Architecture and Applications. In I. Taylor, E. Deelman, D. Gannon, and M. Shields, editors, Workflows for e-Science, pages 320-339. Springer, New York, Secaucus, NJ, USA, 2007.
7. Y. Zhao, M. Wilde and I.T. Foster. Applying the Virtual Data Provenance Model. In L. Moreau and I.T. Foster, editors, Proc. International Provenance and Annotation Workshop 2006 (IPAW2006), volume 4145 of Lecture Notes in Computer Science, pages 148-161. Springer, 2006.
8. P. Groth, S. Jiang, S. Miles, S. Munroe, V. Tan, S. Tsasakou, and L. Moreau. An Architecture for Provenance Systems. Technical report, University of Southampton, 2006.

9. Data Access and Virtualization within ViroLab

Assel Matthias, Krammer Bettina, and Loehden Aenne
HLRS – High Performance Computing Center of University Stuttgart, Germany

One of the major difficulties to overcome in order to improve medical diagnoses and to find the best available treatment for a patient consists in making any relevant information accessible for medical experts in a fast, secure, and easy way, and, especially, in allowing to share updated information as soon as new data become available.

From an eBusiness perspective, the concept of a Virtual Organization (VO) is widely used to approach similar issues, namely to make (data) resources available dynamically, securely, and on-demand [2]. The main purpose of such a concept consists in enabling dynamic collaborations with easy access to different resources, respectively a secure sharing of relevant data/knowledge, tools/services, and workflows [3]. To achieve this, a set of virtualization services that guarantees access to resources in a consistent, resource-independent, and, at the same time, efficient way shall be provided to facilitate a smooth integration of distributed and heterogeneous resources, thus enabling collaborative research or workflow execution.

Within the EU research project ViroLab [1], a virtual laboratory for HIV research and medication support shall be developed that allows different experts in this field to interactively share their expertise and results while working together on the same data and information sets, which are widely dispersed over Europe and currently with outcross-national collaboration. In order to meet the specific requirements for exchanging confidential biomedical information within such a virtual environment [4], the solution introduced in ViroLab is built on existing Grid technologies – Globus Toolkit [5] and OGSA-DAI [6] – which provide the basis or core for our own designed services, called Data Access Services (DAS). These services implement standard user interfaces to support various user groups such as researchers, medical doctors, etc. for accessing the distributed data in a user-friendly way but they also allow the integration of different data resource types to be smoothly exposed within the virtual infrastructure. With only one central entry point acting as the only "visible" and accessible system, users are unaware that they are dealing with a federation of different data resources rather than a single one. Thus, when answering requests for data, the services need to transform and translate heterogeneous data, access heterogeneous technologies, consolidate data gained from several resources, and assure the availability of new /current data while observing data confidentiality and ownership.

This paper will describe the general approach developed and already used within ViroLab to manage the integrated data resources via a single point of access, our DAS. We will explain how existing concepts like OGSA-DAI or Shibboleth [7] can be basically applied but also how they need to be further enhanced in order to guarantee a certain level of flexibility, reliability, sustainability, and last but not least security and trustworthiness. We will furthermore depict the functionalities of the services and show how one can submit distributed queries to multiple databases concurrently. Finally, we will indicate current limitations and improvements needed to make such operations more robust and scalable.

Acknowledgements. The results presented in this paper are partially funded by the European Commission under contract IST-027446 through the project ViroLab. The authors want to thank all members of the consortium who contributed to this paper.

References

1. ViroLab. EU IST Project (IST-027446). http://www.virolab.org
2. Schubert L., Wesner S., Dimitrakos T., Secure and Dynamic Virtual Organizations for Business. Paul Cunningham & Miriam Cunningham, ed., Innovation and the Knowledge Economy: Issues, Applications, CaseStudies, IOS Press Amsterdam, 2005, pp. 1201-1208.
3. Assel M., Kipp A. A Secure Infrastructure for Dynamic Collaborative Working Environments. Proceedings of the International Conference on Grid Computing and Applications 2007, Las Vegas, USA, June 2007.
4. Assel M., Krammer B., Loehden A. Management and Access of Biomedical Data in a Grid Environment. Proceedings of the 6th Cracow Grid Workshop 2006, Cracow, Poland, October 2006.
5. The Globus Toolkit Homepage. http://www.globus.org/toolkit
6. The OGSA-DAI Project. http://www.ogsadai.org.uk
7. The Shibboleth Project http://shibboleth.internet2.edu/

10. Semantic Analysis of Grid Workflows

Martin Seleng, Zoltan Balogh, Emil Gatial, Michal Laclavik, Ladislav Hluchy
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

Performance estimations about existing Grid Services are needed in many projects dealing with grid workflows. The performance estimations are also needed by few components in all information systems which deal with workflows: schedulers, workflow composition tools, and others. Schedulers and workflow composition tools need to know which deployment of a service is the best to select. This selection is dependent not only on the response time and execution duration of services but also on other factors such as reliability of results (delivered results from different services can have different reliability), price (services which delivered identical results can have different prices for their delivery) and usability (usability of the results for the users). These four metrics (performance, cost, reliability and usability) are common metrics for QoS (Quality of Service). Many QoS approaches deal only with performance-related quality criteria, like response time and execution duration. Only a few approaches consider advanced criteria such as security, payment model or execution price. QoS can be viewed (perceptive) from the client (Requestor of the service) side as a QoE (Quality of Experience) and from the server (Provider of the service) side as a QoBiz (Quality of Bussines). These QoE and QoBiz metrics quantify the user experience and the business return, respectively. Using semantics, QoE and QoBiz can be mapped to QoS. Other estimators that can be used for suggesting grid or web services to schedulers and workflow composition tools can be the input parameters(entered by the users). If users want to run the workflow in a grid environment, the monitoring tool can recommend older results which were reached by another workflow with the same input parameters. This must be done very carefully, and such recommendation cannot be easily extended to all grid or web workflows.

In this contribution we present semantic analysis of workflows in grid environment. The contribution describes aWXA tool (Workflow Xml Analyzer, which is part of KAA-Knowledge Assimilation Agent used in K-WFGrid project – www.kwfgrid.eu and extended in national project SEMCO-WS – semco-ws.ui.sav.sk) that enables semantic analysis of workflows (not only in grid applications). The tool is based on analysis of xml files that represent activities in grid workflows. WXA is programmed as a generic tool. It uses Xpath technique, and queries are read from property file which can be edited by a user. As a result, there is no problem of adjusting this tool to any workflow in which activities are represented by xml files. In our contribution we present semantic analysis of grid workflows from technical point of view.

The WXA tool is now working in a way suggested in previous part. WXA runs in the background as a service and works in the following cycle: WXA retrieves and analyses data from past workflows stored in the XML Workflow database (we are using native XML database eXist – exist.sourceforge.net) generated by execution service (in SEMCO-WS we use the BPEL4WS – www.ibm.com/developerworks/library/specification/ws-bpel/). Based on the context received, semantic annotation tool (EMBET – ups.savba.sk/kwfgrid/uaa/) retrieves all the relevant information from the organizational memory and shows text notes on the portal. The text is shown to all users with the same context as the context of the text note. If identical workflow is found, the result of such workflow is offered to the current user for potential reuse through portal solution. As we have mentioned above, the Workflow XML Analyzer is based on XPath language and all XPath queries are stored in configuration file. WXA can be reused in all applications which are based on XML files parsing. When changes are made in XML Schema only queries in configuration files need to be changed. This means that WXA is a generic tool reusable for parsing an XML file. There are also other approaches of parsing XML files, e.g. DOM, SAX, Xerces or Castor. If DOM or was used, the tool would be too robust and the search expression would have to be placed inside the code (the tool wouldn't be generic). Using Xerces or Castor, when an XML schema is changed then entire project must be rebuild. In SEMCO-WS project we want to extend the WXA tool in a following way. We want to use QoS ontology which will include the QoE and QoBiz metrics. Then we want to calculate the execution duration of workflow (which consists of more services) based on the input parameters of the individual services. This calculation will be based on historical and statistical data. Finally, we want to include user view of the received results to the ontology.

11. Knowledge-based Approach to QoS Estimation of Services for Grid Workflows

Zoltan Balogh, Emil Gatial, Martin Seleng and Ladislav Hluchý
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

In service-oriented Grid individual services are combined into grid workflows thus creating complex composite applications. The first stage of grid workflow construction is based on composition of abstract service classes. Service class defines a service with specific functionality, inputs and outputs on an abstract level. A service class incorporated into a grid workflow is considered to represent a workflow activity. There are usually several implementations of a service class called service deployments. Service deployments provide the functionality necessary for a workflow activity execution. During execution of a grid workflow a single candidate must be selected out of several service deployment alternatives for each workflow activity execution. A component called Scheduler is responsible for selecting the best service deployment alternative. Performance is most often used as the main criteria of the Scheduler's utility function. Since Schedulers deal exclusively with workflow schedules, they need to rely on an Estimator service concerning service performance estimations. Such Estimator should enable means for integration of information from multiple sources in different formats and with unambiguous meaning, thus enabling integration of and reasoning about available information feasible.

An approach for performance estimation of services for Grid Workflows is already available in [1], where ontology of basic concepts and relations among them for the domain of performance service estimation was designed. This ontology is called Knowledge Assimilation Schema (KAS) and provides the basis for integration of information relevant to service performances from several sources as well as for the estimation of future service performance. The approach designed in [1] is based on Instance-Based Learning (IBL) techniques to application performance prediction. The approach is build using a knowledge layer (provided by KAS), and uses the power of machine-processable semantics to provide more flexible and more accurate performance estimations of services.

Unfortunately the approach provided in [1] deals mainly with run-time estimations while other quality of service criteria should be also considered as service performance measures. Nevertheless KAS ontology is designed in away which enables extension of service invocation results by any other service performance measures. In this article we are extending KAS ontology by further QoS (Quality of Service) concepts in order to enable estimation of more advances service performance characteristics. We investigate integration of KAS with several existing QoS ontology initiatives as well as with custom-build ontology settings. Evaluation and comparison in terms of implementation efficiency is given for both approaches – proprietary QoS ontology vs. custom-build QoS ontology integration with KAS.

Keywords: Service, QoS, Instance-Based Learning, Ontology, Performance Estimation.

References

1. Zoltan Balogh: Knowledge-based Approach to Performance Estimation of Services for Grid Workflows, PhDThesis, January 2007

12. Knowledge-based Negotiation of Service Level Agreement

Emil Gatial, Zoltan Balogh, Martin Seleng and Ladislav Hluchý
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

This paper deals with knowledge-based mechanism for Service Level Agreement (SLA) negotiation between Service Provider and Service Customer. The evolution of contemporary Grid technologies has been aimed to engage the Service Oriented Architecture resulting into wide scale architecture. SLA is a formal agreement between a Service Provider and a Service Requestor to provide a service with well-defined service levels, accompanied by possible penalty clauses if the SLA is not met. SLA is signed between two parties for satisfying clients, managing expectations, regulating resources and controlling costs. SLA management involves the procedure of signing SLA thus creating binding contracts, monitoring their compliance and taking control actions to enable compliance. SLA includes a description of the provided services and determines the specifications for the performance level and the results that the user of the service expects to receive from the Provider of the service. Many Quality of Service (QoS) approaches deal only with performance-related quality criteria, like response time and execution duration. Only a few approaches consider advanced criteria such as security, payment model or execution price. QoS can be viewed from the service requestor's perspective as a Quality of Experience (QoE) and from the service provider's perspective as a Quality of Business (QoBiz). Such metrics quantify the user experience and the business profit, respectively. Using of semantics can be QoE and QoBiz somehow mapped to QoS. Most recent efforts on QoS support in Web Services are for example ASG framework, WSLA framework, WSML supported by HP or WS-Agreement.

In our work we focus on SLA negotiation process that starts by building the QoE and the QoBiz metrics along with the knowledge captured by using specific services that leads into more precise specification of QoS and thus appropriate service discovery. Knowledge is produced using CBR (Case Base Reasoning) applied to the measured QoS data using Proxy service. The reasoning is applied in the domain of service performance according to the input parameters. SLA defines the formal deal between the service customer and provider about provided service. Such agreements specify the various metrics of quality levels that should be carried out during the service enactment. The exceptional cases are usually specified in order to handle violation of SLA. The proposed brokering platform will allow specify various kinds of SLA penalties that will result in the better required service quality. Such penalties will be specified by SLA exceptions that can make contract with the third party service provider automatically if the service provider violates SLA (under specific conditions). The expenses of service are moved to service provider violating the SLA. Another motivation for service providers to fulfill QoS is to apply the service rating altered automatically by impartial Proxy service when the agreed QoS is violated. The negotiation platform is focused to act on the behalf of both parties the service customers and providers. The service provider can specify quality of business (preferences of business) measures altering the negotiation to maximize the profit from the provided service. The service customer can specify its preferences to make the best possible deal along with the guarantee of specific service level conditions.

Keywords: Service, Service Level Agreement, SOA, QoS, Negotiation, Penalty, Contract.

13. A Semantic Framework for Grid-based Service Registries

Marian Babik, Ladislav Hluchy
Department of Parallel and Distributed Computing, Institute of Informatics, Slovak Academy of Sciences, Slovakia

The Semantic Grid is a recent initiative to expose semantically rich information associated with Grid resources to build more intelligent Grid services. Recently, several projects have embraced this vision and there are several successful applications that combine the strengths of the Grid and of semantic technologies. These applications require support for the dynamic and complex scientific workflows, which are based on processing and sharing of large amounts of heterogeneous data. Such workflows can be based on the composition and interoperability of the grid and web services. Thus they often require support for discovery, description, composition and executions of the grid and web services. One of the major obstacles for such systems is the lack of scalability of the existing semantic repositories, which are based on the Semantic Web standards such as RDF/RDFS or OWL.

In this paper we present a novel storage and inference system for large-scale OWL ontologies on top of relational databases. It aims to meet the requirements of the real Grid applications and provide practical reasoning capability, which is often needed in the process of service matchmaking and service composition. The method is based on a novel approach with the following characteristics. Our method combines the existing OWL reasoners for computing taxonomies (TBoxes), with rule-based reasoners for the reasoning with large number of instances (Aboxes). Based on the proposed combination we can re-use the existing optimizations (i.e. classification and satisfiability techniques) of the Description Logics reasoners to perform fast classifications of the complex schemas. Further, we can exploit the optimizations of the rule-based systems (i.e., join-order and magic sets) to perform queries over ontologies with large number of instances. Since deductive databases are designed to perform the queries over existing relational databases, it is possible to integrate our system with existing RDBMS-based Grid systems.

The overall architecture of the system is based on extending the existing tableau reasoner Pellet [1] with the optimizations for the conjunctive query answering and a database backend [2]. The core of the system is composed of two reasoners, a tableau-based reasoner and a disjunctive datalog engine. The aim of the tableau reasoner is to check the consistency of the TBox and to compute its classification. The disjunctive datalog engine is used for the answering of the instance retrieval queries and to check the consistency of the knowledgebase. The effective integration of the ontology inference and storage is expected to improve reasoning efficiency, while instance retrieval querying on top of the disjunctive datalog engine guarantees satisfactory response time. Extensive experiments on multiple ontological benchmarks shows the high efficiency and scalability of our system.

References

1. E. Sirin, B. Parsia: Pellet: An OWL DL Reasoner. Volker Haarslev, Ralf Möller (Eds.): Proceedings of the 2004 International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8, 2004. CEUR Workshop Proceedings 104 CEUR-WS.org 2004
2. Y. Kazakov, B. Motik. A Resolution-Based Decision Procedure for SHOIQ. In Ulrich Furbach, John Harrison, and Natarajan Shankar, editors, Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR 2006), volume 4130 of LNAI, pages 662-667, Seattle, WA, USA, August 17-20 2006. Springer.

14. Enhancing Grid Computing with Virtual Grid Concept Model

Jacek Kosinski, Krzysztof Zielinski
Department of Computer Science, University of Science and Technology, Cracow, Poland

Virtualization techniques enables the creation of illusion of real computer built of shared resources. These methods allow to create from the limited amount of physical devices a pool of virtual computers, achieving hence simplification of distributed computing through lessening of abstraction level – unification of types and parameters of particular devices – simplification of network topologies and simplification of communication patterns.

The usage of virtualization allows full control over 'physical' parameters of emulated machines as number and processors speed, parameters of operating memory, exclusive access to physical devices as defined in configuration. These parameters can be modified on the fly without stopping the virtual computer. Additionally it is possible to influence the configuration and layout of network topology.

A virtual grid can be implied as a form of requirements definition or resource description delivered by an application in order to create an environment enabling its proper and efficient operation. This specification describes required resources, their configuration and other elements that could be influenced through the application because of virtualization. These elements are crucial to achieve optimal resource consumption.

This paper proposes a model of a virtual grid as a tool for more effective grid resource usage. It also includes validation tests of using light virtualization techniques as a method of creating computing nodes. Also the results of research proving the level of degradation of computing power of grid built of virtual components.

In order to create virtual grid environments with this approach, it has to be effective concerning the minimalization of possible overhead introduced by virtualization. Light virtualization a.k.a para-virtualization was proposed as an alternative to emulation of complete set of devices. This approach moves the operating system to a special architecture and in a physical node provides a supervisor responsible for separation of shared hardware resources.

Grid features as variable availability, geographical distribution, components belonged to and managed by different companies, different requirements, security and management policies, and communication through heterogeneous network technologies would be guaranteed through the concept of virtual grid. Building the grid infrastructure based on virtual components would allow achievement of a grid vision as a flexible, adaptable on demand computing tool for a wide variety of applications.

15. Hybrid method for on-line testing of grid software using Abstract State Machines

Daniel Lamch, Roman Wyrzykowski
Institute of Computer and Information Sciences, Czestochowa University of Technology, Poland
[draco, roman]@icis.pcz.pl

Abstract State Machines (ASM) are mathematically defined environment for high-level system design, verification and analysis. In our previous work [1], a hybrid approach to specification, analysis and testing of grid stateful services has been presented. This approach is based on using ASM, and allows for an easy integration of created specifications of developed middleware with existing components of grid systems.

In this paper we show how this approach can be utilized for testing grid software, in order to verify the conformance between behaviours of specification and implementation. An important advantage of the proposed method is possibility to carrying out the on-line mode of testing, when tests are generated simultaneously with their execution. The use of this mode allows for reducing the total time of verification.

References

1. D. Lamch, R. Wyrzykowski: Specification, Analysis and Testing of Statefull Grid Services Using Abstract State Machines. Proc. CGW 2006, Cracow 2007, pp.417-424.

16. GREDIA Project – Grid Enabled Access to Rich Media Content

GREDIA Consortium

Grid technology has achieved significant advances in the past few years with a plethora of prestigious organisations contributing to middleware that opens the horizons for new exploitation opportunities. However, this potential exploitation has not yet been seen to materialise in emerging applications. The use of Grid technology is still confined mainly within scientific applications, developed by scientific organisations, being experts in Grid principles. The majority of IT application developing organisations is still afraid to delve into the use of Grid technologies, as these still sound new and remote to them.

GREDIA will address this problem by providing a Grid application development platform, with high level support for the implementation of Grid business applications through a flexible graphical user interface. This platform will be generic in order to combine both existing and arising Grid middleware, and facilitate the provision of business services, which mainly demand access and sharing of large quantities of distributed annotated numerical and multimedia content.

Furthermore, GREDIA will facilitate for mobile devices to exploit Grid technologies in a seamless way by enabling mobile access to distributed annotated numerical and multimedia content. The project will define and analyse the appropriate interfaces to support the protection of data and transactions of Business Grid applications at all levels through a dedicated security framework.

The effectiveness and the reliability of the GREDIA Grid application development platform will be validated through the deployment of pilot applications in two operational domains:

- A media and journalism application which will allow journalists and photographers to make their work available to a trusted network of peers at the moment it is produced, either from desktop or mobile devices. This pilot will bring together the market orientation and pervasiveness of mobile communication technology with the promise of a dynamically concerted use of resources and services provided through Grid infrastructures.

- A banking application which will enable the exchange of private information between banking organisations and their potential customers for the assessment of their creditability and the risk associated with granting a specific loan according to the Basel II regulations. This pilot will utilize the advances in web services and other middleware developed in GREDIA to demonstrate how Grids can be accessed by many partners regardless of location, ensuring that the data sharing is consistent with the principles of the financial sector.

Acknowledgements. This project is partly supported by the European Commission through the FP6 IST Framework Programme.

References

1. www.gredia.eu

17. Supporting Management of Dynamic Virtual Organizations in the Grid Through Contracts

Bartosz Kryza (1), Lukasz Dutka (1), Renata Slota (2), Jacek Kitowski (1,2)
(1) ACK Cyfronet AGH, Krakow, Poland
(2) Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland

Existing trends in Grid computing are moving the Grid systems from laboratories to complex business settings such as for instance inter-banking systems. These environments pose several new requirements on these systems. Most importantly these include easier management of the infrastructure and more flexible security mechanism, especially in the context of Virtual Organizations which span several administrative domains of real world organizations participating in some VO.

We propose a framework, called FiVO (Framework for intelligent Virtual Organization) that will support creation and management of dynamic Virtual Organizations with special focus on authorization of access to resources. The main feature of FiVO is the contract negotiation and management component, which enables coordinated establishment of agreement among partners who want to create a new Virtual Organization. The contract, described by special ontology, provides all the information necessary for configuration of Virtual Organization in a Grid system, by translating proper contract statements to configuration options of such systems as for example VOMS (Virtual Organization Management System) or PERMIS. Contracts also allow for specification of non-functional parameters of the envisioned VO collaborations, especially SLA's (Service Level Agreements) in order to provide for the Grid monitoring layer necessary data for contract enforcement.

The framework is evaluated within the framework of EU-IST project Gredia, on two commercial applications. First one is related to inter-banking solution for automatic credit-scoring of bank users credit requests. The second one is a media application oriented on providing a collaborative environment for nomadic journalists.

This paper will present the architecture of the envisioned framework, relation to the Gredia platform, analysis of requirements for the contract management system in a Grid, ontology and negotiation model for contracts, analysis of possible integration with existing VO management and authorization systems in the Grid including VOMS, CAS, PERMIS and AKENTI. The paper will also include extensive state of the art on the subject of contracts in Virtual Organizations.

Acknowledgement. The work is supported by EU IST Gredia Project, FP6 34363.

18. A Grid Application Design Environment for Business Users

Marian Bubak (1,2), Daniel Harezlak (2), Marek Kasztelnik (2), Piotr Nowakowski (2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

Modern business applications are often constructed from disparate services and data repositories, which reside at various sites, connected by high-speed networks, and which can be combined to yield nontrivial functionalities, targeted for business users.

While the development and deployment of services for Grid infrastructures is a well researched and well founded area (see for instance [1]), the actual composition of applications which encapsulate business logic over these distributed services remains a problem [2]. We believe that a new solution should be presented, which takes the technical burden related to invoking and interacting with various types of services off the shoulders of application developers.

Appea (the GREDIA Application Development Platform [3]) is a system which enables users to construct application scenarios from available computational services and by making use of the services provided by actors (for instance journalists and editors in the media application, or banking assistants in the banking application). Appea provides means for developers to express application scenarios in a formalized manner (by using a specially-prepared notation based on the JRuby framework [4]) and for application users to execute predeveloped scenarios using their end-user interface. The aim of this paper is to describe the way in which such an environment can provide added value for end users within the specified application domains.

The idea behind Appea is to uniformize access to three types of entities present in the Grid infrastructure: services (encapsulating computational power), data sources (repositories of business information) and actors (the people involved in the processing and execution of use cases). We present how an application scenario -corresponding to a business process – is constructed and executed using the architecture provided by the GREDIA project.

While the preparation of application scenarios requires the scenario developer to possess knowledge regarding the way in which a given organization goes about its business, it does not require developers to deal with the low-level details of locating and invoking Grid services, nor does it require them to implement per-case interaction with Grid data sources (see [5] for comparison). By concealing the middleware layer of Grid systems, Appea enables developers to prepare business application in a straightforward and effective manner.

The Appea environment is currently entering its implementation phase and thus the paper focuses on the design and intended functionality of the system.

Acknowledgements. This work was supported by EU project GREDIA (FP6-34363) with the related Polish grant SPUB-M and by the Foundation for Polish Science.

References

1. The W3C Web Services Activity; http://www.w3.org/2002/ws
2. F. Baude, D. Caromel, and M. Morel, From Distributed Objects to Hierarchical Grid Components, in International Symposium on Distributed Objects and Applications (DOA), Catania, Sicily, Italy, 3-7 November, ser. LNCS, vol. 2888. Springer-Verlag, 2003, pp. 1226-1242.
3. The GREDIA project; http://www.gredia.eu
4. JRuby Project Home; http://jruby.codehaus.org/
5. Raul Sirvent, Josep M. Perez, Rosa M. Badia and Jesus Labarta, Automatic Grid workflow based on imperative programming languages. Concurrency and Computation: Practice and experience, Vol.18. Issue 10. Pages: 1169-1186., December 2005.

19. Gredia Middleware Architecture

Ioannis Konstantinou, Katerina Doka, Athanasia Asiki, Antonis Zissimos and Nectarios Koziris
National Technical University of Athens, School of Electrical and Computer Engineering, Computing Systems Laboratory, Zografou, Greece

The explosive evolution of networking infrastructures along with the growing adoption of grid computing initiatives in resource sharing create attractive perspectives for the deployment of large-scale, distributed computing and storage systems. On the other hand, as far as data management is concerned, peer-to-peer (P2P) algorithms have emerged as a promising solution, offering incontestable advantages in terms of scalability, fault-tolerance and the ability to adapt in dynamic node arrivals and departures [4]. A distributed facility incorporating practices from both areas has been envisioned by numerous related research initiatives.

The main purpose of the Middleware component of the generic platform being developed by the Gredia project [2], is to provide the prerequisite services for storage, search and retrieval of annotated, rich media content in a large-scale distributed environment. The users of the Gredia platform will be granted the ability to upload their files in the Gredia repository and to perform advanced searches upon the data items stored in the repository, based on the metadata descriptions stored in a Peer-to-Peer overlay. Every node connected to the platform will be able to act not only as a consumer of storage resources, by means of searching existent files, but also as a provider of storage capacity and data services.

The Gredia Middleware Architecture is a service-oriented approach consisting of WSRF compliant web services providing core system functionality. In particular, the services provided to facilitate data manipulation are the *Metadata Service* and the *Data Service*. Every file of the system is associated with a corresponding metadata file containing descriptive information about the data item. A general metadata structure has been defined taking into account the requirements posed by the users of the Media and Banking pilot applications and the applied standards for multimedia content, e.g. the MPEG-7 standard [5]. The metadata files stored in the *Metadata*

Overlay are indexed properly in order to be used during the search by the *Metadata Service*. Techniques originating from P2P computing and coinstantaneous indexing of multiple attributes, referred to as multidimensional indexing [9,11,13], are exploited to assure system scalability and to achieve minimization of response time and communication overhead implied by the performed searches, which may include point, categorical or range queries. Due to the distributed nature of the Metadata Overlay and the fact that nodes with different storage capacities, network connections and processing power participate in the overlay as long as they are capable of executing the Metadata service, it is necessary to avoid uneven distributions of load. For this reason, a load balancing technique of exchanging virtual servers among physical nodes according to their storage capacity and the available bandwidth is deployed ([12], [10]).

The *Data service* is the service responsible for manipulating data items. More precisely it undertakes the uploading and storing of multimedia files in the Gredia *Storage Overlay*, so as to make them available to other users of the platform, as well as the downloading of a requested file, by providing the search client with a data stream. The *Storage overlay* is the repository where all data items are stores, and in which remotely located, disjoint and diverse storage facilities are integrated. The returned datastream is used by the GridTorrent transfer mechanism [14], which enables the simultaneous downloading from all nodes that keep replicas of the requested files, either directly from established GridFTP [3] servers, or from other GridTorrent peers requesting the same data item. GridTorrent constitutes a modified BitTorrent protocol [1], tightly coupled with the Grid middleware components dealing effectively with large file transfers and flash crowd situations. The file replica locations are supplied to the Data service through its interaction with the *Rich Data Location Service (RDLS)*, a Replica Location Service [6] managing a global distributed catalogue with mappings between Logical File Names (LFNs) and Physical File Names (PFNs), which indicate the physical locations of replicas stored in the Storage overlay. The distributed catalogue is implemented as a structured Peer-to-Peer overlay in order for scalability and fault-tolerance to be ensured. In the design of the RDLS the feasibility of enabling mutable data storage with the use of timestamp indicators to handle update procedures has been forseen, a feature that is not supported in existent networks.

The poster depicts in detail the core service functionality of the Gredia Middleware Architecture along with its interaction and communication with external components of the Gredia platform. The main achievements include the specification of principles and services for a scalable, fault-tolerant and reliable large-scale distributed environment. Centralized structures subjected to failures, as catalogues and repositories, have been replaced by distributed ones, derived from structured DHT-based Peer-to-Peer overlays. However, all services have been deployed according to the standards imposed by the Open Grid Services Architecture (OGSA), as it was suggested by Foster et al. in [8] and was later enhanced in [7] and communicate with other services through well-defined interfaces. The design of the system leads to an extensible architecture favoring the integration with other systems.

Acknowledgements. The described work is partly supported by the European Commission through the FP6 IST Framework Programme.

References

1. Bittorrent.org. http://www.bittorrent.org/index.html.
2. Gredia.eu. http://www.gredia.eu/?Page=home.
3. W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link. The globus striped gridftp framework and server. *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, 2005.
4. H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking up data in p2p systems. *Communications of the ACM*, 46:43–48, 2003.
5. S.F. Chang, T. Sikora, and A. Purl. Overview of the mpeg-7 standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11:688–695, 2001.
6. Antony Chazapis, Antonis Zissimos, and Nectarios Koziris. A peer-to-peer replica management service for high-throughput grids. In *2005 International Conference on Parallel Processing (ICPP05)*, Oslo, Norway, 2005.
7. K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, and S. Tuecke. From open grid services infrastructure to ws-resource framework: Refactoring & evolution. *Global Grid Forum Draft Recommendation, May*, 2004.
8. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration. *Open Grid Service Infrastructure WG, Global Grid Forum, June*, 22:2002, 2002.
9. P. Ganesan, B. Yang, and H. Garcia-Molina. One torus to rule them all: multi-dimensional queries in p2p systems. *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004*, pages 19–24, 2004.
10. B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica. Load balancing in dynamic structured p2p systems. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 4, 2004.
11. J.K. Lawder and P.J.H. King. Querying multi-dimensional data indexed using the hilbert space-filling curve. *ACM SIGMOD Record*, 30:19–24, 2001.

12. A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica. Load balancing in structured p2p systems. *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS?03)*, 2003.
13. C. Schmidt and M. Parashar. Enabling flexible queries with guarantees in p2p systems. *Internet Computing, IEEE*, 8:19–26, 2004.
14. Antonis Zissimos, Katerina Doka, Antony Chazapis, and Nectarios Koziris. Gridtorrent: Optimizing data transfers in the grid with collaborative sharing. In *11th Panhellenic Conference on Informatics (PCI2007)*, Patras, Greece, May 2007.

20. Adaptation of Grid Execution Management Systems for Mobile and Ubiquitous Grid Services

Kleopatra Konstanteli, Antonios Litke, Magdalini Kardara, Konstantinos Lalos and Theodora Varvarigou
Dept .of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

Mobile Grid, in relevance to both Grid and Mobile Computing, is a full inheritor of Grid with the additional feature of supporting mobile users and resources in a seamless, transparent, secure and efficient way. It has the ability to deploy underlying mobile networks and provide a self-configuring Grid system of mobile resources (hosts and users) connected also by wireless links and forming logically arbitrary and unpredictable topologies. Mobile Grid is a network aware Grid that offers to the users aggregated services, not ignorant of the attributes of the underlying network, the Grid infrastructure and the applications logic. For this reason Grids and Mobile Grids can form the basis and the enabling technology for pervasive and utility computing due to their ability of being open, highly heterogeneous and scalable.

In a mobile Grid changes are numerous and highly variable. In order to provide commercially viable infrastructures, the provided Grid services should be able to support mobility in an efficient and seamless manner, which should avoid complexity and be transparent to the end user. The Akogrimo ("Access to Knowledge through the Grid in a Mobile World") project deploys the Session Initiation Protocol (SIP) in order to address mobility support, session management and information regarding the context and the presence of the user. In this approach, applications use SIP for signaling, i.e. to manage session and mobility information, locate resources and get additional details about their availability and status. Once the session is established and both applications (client and server) have all the data needed to proceed with the service execution, they use Simple Object Access Protocol (SOAP) to exchange service requests and responses, introducing thus the "SIP with SOAP" approach to Grid services. This approach necessitates an advanced management of the Grid services through an appropriate component which will coordinate the execution of the services, allowing a SIP/SOAP collaboration.

The Execution Management Service (EMS) described in this paper addresses the issues mentioned above. Developed within the Akogrimo Project, EMS builds heavily on the Open Grid Services Architecture (OGSA) specifications and can guarantee end-to-end Quality of Service (QoS) while at the same time achieves "mobile awareness" by establishing communication with the mobile network middleware layer of the Akogrimo Infrastructure. The latter builds heavily on the SIP, used for creating, modifying, and terminating sessions with mobile machines, such as laptops, where Grid services are running. Although the management of mobile Grid services is different to that of ubiquitous ones due to their inherent differences, EMS is able to achieve it in a seamless fashion making them transparent to the user.

By exchanging information with the SIP Infrastructure, the EMS is able to keep track of the current location of the mobile Grid services. In particular, EMS becomes aware of changes regarding the availability of the mobile Grid services by establishing a notification mechanism with the SIP Infrastructure. When there is a change in the availability of the mobile Grid service, a notification message including information about the availability/unavailability of the mobile service is generated by the SIP Infrastructure and sent to the EMS. Upon receiving such a notification, the EMS is responsible for updating the registry it uses to store info for mobile Grid services and take corrective actions such as reallocation of the execution, in case the shift takes place during the actual execution of the mobile Grid service.

The EMS has been implemented using the Globus Toolkit 4 (GT4) platform and builds heavily on the capabilities offered by the WSRF and WS-Notifications specifications. In order to create a powerful and flexible EMS, high-level services that are included in GT4 and take advantage of the potentials that the toolkit offers to the fullest have been leveraged into EMS.

Achieving correct and efficacious execution management while providing end-to-end quality of service across a mobile Grid environment where the set of resources that are available for use can change location frequently is a significant undertaking. In the full paper we present an execution management architecture, EMS, that addresses this problem. We give an overview of its design and architecture and present in detail the interactions and internal processes that take place. In addition, a prototype EMS implementation has been constructed and preliminary tests have been conducted that quantify the efficacy of our techniques.

21. Management of Virtual Organisation for Demanding Applications in the Grid Environment

L. Skital (2), R. Slota (1), M. Janusz (1), J. Kitowski (1,2)
(1) Institute of Computer Science, AGH University of Science and Technology, Cracow, Poland
(2) ACK Cyfronet AGH, Cracow, Poland

Grid environments evolve from being 'better batch systems' into full-featured systems supporting variety of application types. Int.eu.grid [1] project focuses on creating grid environment for interactive or real-time applications. Due to it's strict requirements, real-time applications belongs to the most demanding group, which is very difficult to support in distributed infrastructure. To support one of this difficult applications we have created a specialized virtual organisation (HEP VO). Management of this kind of VO is not trivial. The VO should provide stable application run-time environment basing on resources available from many grid sites. To be up to this task, VO is based on contracts strengthened by Service Level Agreements (SLA) [2] with sites. However, management of the environment and monitoring of contracts fulfilment require some additional effort, which can be minimized with adequate tools.

In the paper we present HEP VO Management System which supports VO management process. The system consists of two web-services and two user interfaces. The first web-service, Grid job submission service, is responsible for application run-time environment. It keeps number of grid jobs on level desired by the application. The service takes advantage of dynamic resource allocation offered by the grid. The second web-service is responsible for site management and SLA monitoring. It is used to keep track of site operations, including site certification, monitoring and failure recovery. The service gathers SLA metrics values from different information sources (like JIMS, O-CMG, SAM or BDII) distributed over the grid environment and checks, if SLAs are fulfilled. It can react on any SLA infringement by changing rules for resource allocation (eg. exclude failing resources) and sending warnings or alarms.

HEP VO Management offers portal-based interface, which can be used by VO Manager, site administrators and application supervisors. This interface provides, among others, quick overview of application's run-time environment condition basing on SLAs fulfilment. The system is also integrated with Migrating Desktop (MD),offering interface (plug-in) for application supervisor. MD plug-in allows to adjust grid job submission to current experiment needs.

The system is in implementation phase with first prototype working.

References

1. The int.eu.grid project http://www.interactive-grid.eu/
2. Service Level Agreement metrics for real-time application on the Grid; Skital L., Janusz M., Slota R., Kitowski J., presented on PPAM 2007

22. Grid Accounting Concept Based on the Model of Public Utility Pricing

Csongor Somogyi, Imre Szeberenyi
BME Department of Control Engineering and Information Technology

Accounting counts as key component for commercial application of the Grid. To create an accounting solution many initiatives have set up their own models. Some of them are derivations of existing process accounting practice introduced in Unix operating systems. Other concepts in their models may take into account the characteristics of the Grid, for example, by compensating the resource user for unexpected queue waiting or fining the resource user for exceeding the expected deadline of job execution. In this article the authors introduce a concept based on the worldwide applied pricing model of public utility services. The authors state not only that this concept is far more convenient for utility providers and consumers, but also, that this approach is more generic than those applied in earlier models.

The article shows that during the design of the model authors considered those definitions and characteristics which determine the Grid. The most important definitions include the generalization of Grid 'resource', 'virtual organizations' and 'job virtualization'.

The model currently covers the process from resource usage metering to issuing the invoice. However, the authors point out that the model can and will be extended by creating the missing building blocks which are necessary to commercialize the Grid.

Applying the model, the authors have created a framework called Grid Accounting and Charging System (GACS). GACS has an own XML based protocol which has XSD based definition. It is built upon Java Servlet technology. All of its services are stateless, thus its applier does not have to deal with transactions which makes the application very robust in its distributed context. It is fully Grid Security Infrastructure (GSI) enabled. A graphical user interface is built upon GridSphere portlet technology.

GACS is decomposed mainly into five modules: metering, accounting, pricing, charging and billing. These modules can be distributed over the accounting sites independently and each can handle the accounting data defined on an arbitrary subset of resources and their belonging accounted parameters, thus allowing scaling the accounting system.

Currently, GACS is integrated with the Hungarian ClusterGrid and integration with the EGEE related APEL is being developed. With these integrations an accountant can produce invoices of resource usage on ClusterGrid and/or certain Hungarian sites of the EGEE infrastructure either on per job or on monthly/weekly basis.

Keywords: Grid, accounting, accounting and charging, utility computing

23. DiProNN: Distributed Programmable Network Node Architecture

Tomás Rebok
Faculty of Informatics, Masaryk University, Brno, Czech Republic

Contemporary computer networks behave as a passive transport medium which delivers—or in case of the best-effort service tries to deliver—data from the sender to the receiver. The whole transmission is done without any modification of the passing user data by the internal network elements (not including firewalls, proxies, and similar elements, where an intervention is highly limited). However, especially for small or middle specialized groups (up to hundreds of people) using computer networks for specific purposes, the ability to perform a processing inside a network is sometimes highly desired. Multimedia application processing (e.g., video delivery and/or transcoding) or security services (data encryption over untrusted links, secure and reliable multicast, etc.)are a few of possible services which could be provided.

The principle called ‘’Active Networks’’ or ‘’Programmable Networks’’ is an attempt how to build an intelligent and flexible network using current networks serving as a communication underlay. Such a network allows processing of passing user data in a network, which is highly suitable especially for video streams processing. However, programming of complex stream processing applications for programmable nodes is not effortless since they usually do not provide sufficient flexibility (both programming flexibility and execution environment flexibility).Thus, depending on a programming language used there is an executable program created, which rely on specific execution environment (operating system, virtual machine (e.g., Java Virtual Machine), interpreter, etc.).

Usually, when an active program doing requested processing exists, there must not exist a programmable node capable of running it, and thus the original active program has to be revised. The usage of virtual machines principles can increase a lot the flexibility of programmable nodes' execution environment since they are able to run completely different execution environments simultaneously. Moreover, they can also bring other benefits (strong isolation, resource management, programming flexibility, etc.), and thus make the usage of programmable routers easier.

In our paper we present the programmable network node architecture named DiProNN (Distributed Programmable Network Node) that is able to accept and run user-supplied programs and/or virtual machines and process the mover passing data. All the DiProNN programs are described using novel modular programming model based on the workflow principles that takes advantages of DiProNN virtualization and makes programming of complex streaming applications more comfortable. Thanks to DiProNN's distributed architecture and possibilities of parallel processing, the DiProNN can also improve the robustness and scalability of such an active system with respect to number of active programs simultaneously running on the node and with respect to the bandwidth of each passing stream processed. As a possible application we show an implementation of simple MCU (Multipoint Control Unit) used for large videoconferences that profits from DiProNN properties, and finally we test the DiProNN prototype implementation behaviour.

24. Storage Throughput Performance Tuning in Grid Environment after Migration to Scientific Linux 4

Michał Dwużnik, Marek Magryś, Marek Pogoda, Grzegorz Sułkowski, Maciej Twardy
Academic Computer Centre CYFRONET AGH, Krakow, Poland

Many of the Grid nodes are still operating under Scientific Linux (SL) version 3 with kernel release 2.4. Grid administrators are facing with the necessity of migrating the operating system to SL version 4 with a more sophisticated kernel 2.6 release. This task is very complex for many reasons, not only because of the large number of the nodes to be upgraded but also compatibility issues of the grid middleware components, as well as VO proprietary software legacy issues. One of the aspects revealing the importance of proper handling of the new kernel's functionalities is storage I/O performance.

A simple upgrade of the OS in a storage environment tuned for the best throughput under SL 3 may unexpectedly cause significant performance degradation. This situation is closely related to introducing new I/O schedulers in the 2.6 release. Each of the schedulers has different properties dedicated to various workload scenarios. In addition Host Bus Adapters (HBAs) need different configuration depending on the type of the active I/O scheduler. Obtaining optimum performance, in this case, requires matching proper scheduler algorithm to the workload scenario and, what is more important, tuning the driver parameters of the Fibre Channel adapters.

The main topics presented/described in this article are:

- characterization of each type of I/O scheduler,
- description of the key configuration parameters affecting the throughput performance in the 2.6 kernel environment,
- search for optimum HBA and I/O scheduler configurations dedicated to Grid workload scenarios.

25. Knowledge Supported Data Access Optimization for Grid Environments

Darin Nikolow (1), Renata Slota (1), Jacek Kitowski (1,2)
(1) Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland
(2) Academic Computer Center CYFRONET AGH, Krakow, Poland

As the grid technologies get more matured and widely adopted demands for new functionalities appear. Clients are expecting that they will be served with the desired level of quality of service negotiated with the server. Efficient management of data storage resources in SLA (Service Level Agreement) respecting system is not a trivial task. Extended monitoring of the SLA fulfillment is needed in order to proactively prevent the lost of performance for the given client request. The monitoring of SLA fulfillment may not be possible without some low level storage monitoring services.

In this paper our idea of using services for monitoring and data access time estimation for data access optimization and the supporting of SLA in grid environment will be presented. These services could be used by the data management middleware and especially by the data replication module to effectively manage the storage resources fulfilling the performance and cost goals presented in the SLA.

There are many different storage systems available on the market today. Providing monitoring data for all kind of storage systems used in a given environment requires knowledge of details which are specific for the given storage system. The obvious approach is to have specialized storage monitoring services for every storage system used. From the other hand storage systems have some common or similar functionalities and characteristics which can be exploited by a more general service. The more intelligent approach proposed in the paper is creating a knowledge base addressing monitoring storage systems and then using this knowledge to create workflow which will gather and provide the necessary performance parameters of a given storage system.

Since the storage system monitoring service in production environment should be available at any time a functionality of non-disruptive updating of the low level services and the knowledge base should be present. Another problem which should be addressed in any monitoring system is the impact of the monitoring itself on the actual performance of the storage system. Special care should be taken to minimize the influence of the storage monitoring service on the performance of the storage system.

26. Key Keeper Service for Data Access Policies Enforcement in Grids

Marek Ciglan, Ondrej Habala, Jan Astalos, Ladislav Hluchy
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

Data access security in compute grids is of high concern for multiple scientific communities. Enabling the data sharing in a controlled manner, under fine grained data access policies is a must. Nowadays production grade grid middlewares provide highly secure methods for users authentication, however the access authorization methods are often insufficient, which is a major obstacle for certain communities to adopt the grid paradigm.

The requirements on the data access authorization might differ depending on the needs of the community or application. The systems exposing the data files to the grid infrastructure might not always comply with the requirement of a specific community. In addition, the widely used non-grid systems for exposure of the data files in distributed environment (such as http servers, webdav servers, or standard ftp servers) often can not be used in the grid infrastructure, due to the lack of integration with GSI security mechanisms.

In this paper, we present a key keeper service for data access policies enforcement to overcame this problem. The resulting software package comprise of a server component and client toolkit. The solution is based on the data encryption. A file it is encrypted when it is published to the grid infrastructure. We use multiple synchronous encryption keys to encrypt parts of the files. The meta-information on the published files and the encryption of the files are stored in a centralized key keeper service. The metadata is protected on the server and during the transfer using the GSI encryption and message level security. When a user or an application on behalf of a user require to access encrypted data, it transfers the encrypted data to the execution resource and contacts the key keeper service to obtain the information on encryption and the encryption keys. The authorization to the sensitive data files is thus done on the level of security metadata service, in addition to the authorization procedure at the data exposing resource. This enables us to provide strong authorization methods required by different communities without the need of changes of the existing infrastructure and services.

The obvious disadvantage of the centralized security service is the possibility of the service unavailability. If the key keeper service or a resource hosting this service fails, the encrypted data would be unusable in the whole grid infrastructure for the period of service downtime. We address this drawback by integration of the key keeper service with replica update synchronization service, called RUPAGATION. RUPAGATION was designed to address the synchronization of updatable database sources in the grid environment. It supports a large number of relational and XML databases and provides the basic functionality for update propagation and replica content synchronization. It provides plugin like mechanism for integration of consistency protocols implementations. As the key keeper service exploits a relational database back-end for data persistence, RUPAGATION was identified as a suitable solution to address the required fault tolerance for proposed security service. It was integrated with key keeper service and evaluated using two data consistency strategies – primary replica consistency model and two-phase commit consistency protocol.

The paper discuss the motivation for the key keeper service, presents the architecture and methods used in the solution. The properties of RUPAGATION service are summarized and the integration of the two services is described. The performance evaluation that summarize the data encryption/decryption overheads is presented in the results section.

27. Gather and Prepare Monitoring Data for Estimating Resource Stability

Georg Birkenheuer (1), Holger Nitsche (1), Kerstin Voss (1), Elmar Weber (1), Peter Majlender (2)
(1) Paderborn Center for Parallel Computing, University of Paderborn, Germany
(2) IAMSR, Abo Akademi University, Turku, Finland

The establishment of SLAs is essential in order to attract commercial end-users to use the Grid for their computations. However, providers may be unwilling to agree an SLA since they are aware of the possibility of resource failures, which might result in paying penalty fees. Since performing a job as negotiated in the associated SLA in a computational Grid is tightly coupled with the stability of used compute nodes, probabilities of a resource failure before the completion of running jobs are important criteria in the resource allocation decision process.

In order to provide the probability estimation process with adequate and well-formatted information, we have developed a data service called consultant service. It first prepares monitoring information by using data aggregation methods and afterwards generates statistics on their basis used by the estimation process. The statistics are well-formatted input for efficiently and realistically assessing the resource stability. The more detailed the input data of the estimation process, the more realistic is the assessed value.

In addition to probabilities of resource failures, probabilities of SLA violations should be assessable. As a consequence, the consultant service provides the estimation module with the SLA's service description terms and guarantee terms. The consultant service is responsible for gathering and preparing SLA information in a database in an efficient and scalable way.

The paper describes the aggregation of monitoring information in detail. We fix several time intervals as units of our time series. Then, for this fixed time interval, we measure the appropriate information in the nodes, and consider their arithmetic average.

We further highlight in the paper that the collected and prepared information of the consultant service is not only valuable for the estimation process. The administrator also benefits from the collected data when it generates SLA templates or searches for bottlenecks in the infrastructure. Therewith the consultant service is a data service, which enables on the one hand a measure for the resource stability and on the other hand supports the administrator in his tasks.

The paper will be organized as follows. After a motivation for using resource stability estimations in the resource allocation it will describe the concept of the data service as well as its interaction with other components in the Grid fabric. The monitoring framework will be also briefly presented as it is used to gather information for the consultant service. The last part of the paper will be the presentation of methods used for the aggregation process of historical monitoring information and terms of past SLAs.

28. Integration of OCM-G into the JIMS Infrastructure for the Monitoring of HEP Application

Wlodzimierz Funika (1), Krzysztof Guzy (2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland

One of the tasks in the EU IST int.eu.grid project [1] is to create a pilot application [2] which exploits Grid computational resources, based on the HEP application coming form the ATLAS project. So far, in a local processing model, data about events, describing collisions collected form a particle accelerator, were sent for preprocessing to local computational farms. The amount of these data, however, is so large that there is a strong demand for extra computational power and to meet this demand, Grid resources can be used.

Bringing the HEP application to the Grid requires providing some support from middleware services. The Real-Time Dispatcher (RTD) is a software solution dedicated to facilitate the delegation of computations to the Grid infrastructure. It coordinates distribution of data obtained from the ATLAS system to remote processing tasks(PT). The remote processing task is a grid job which is submitted in advance and is waiting for data to work on. RTD must decide which one from PTs gets the data. Therefore RTD needs to get information on the network load and resource usage of worker nodes where PTs are running to make scheduling decisions.

As a main monitoring data provider for RTD, the JIMS system [3] is chosen. JIMS – JMX-based Infrastructure Monitoring System – will be deployed on each site of the Virtual Organization intended to support the HEP application. JIMS will measure the available bandwidth to sites where events' data are processed. In addition to the network load, RTD needs to get some data about worker nodes status: CPU load and memory usage, very frequently and rapidly. The OCM-G [4] is a grid-enabled application monitoring system which is able to provide information on nodes where grid application's processes are running. For RTD purposes, OCM-G can be integrated with JIMS using JIMS' extension mechanism to its monitoring capabilities with MBeans (Managed Beans) components. MBean runs in a JIMS monitoring agent (MBean server), where it can be dynamically loaded at runtime, to implement some monitoring functionality. Therefore, in order to provide the monitoring data for RTD, there was created an MBean which JIMS is capable to use for this purpose. The MBean hides the OCM-G system and provides the data about nodes' CPU load and memory usage through its exposed interface. OCM-G system collects monitoring data using its components – local monitors. A local monitor is started, whenever a PT is launched on a worker node. Thus, the MBean, being an OCM-G client, can obtain monitoring data, via querying the high-level OCM-G component, i.e. data on host's computational resources utilization coming from the worker nodes where PTs are running. With help of the JIMS infrastructure the data from all sites are supplied to RPT, which efficiently fulfills its function of optimization of event distribution.

Acknowledgements. This research is funded by the EU IST Project int.eu.grid IST-031857.

References

1. EU IST int.eu.grid project's web page: http://www.interactive-grid.eu/
2. L. Dutka, K. Korcyl, K. Zielinski, J. Kitowski, R. Slota, W. Funika, K. Balos, L. Skital and Bartosz Kryza, Interactive European Grid Environment for HEP Application with Real Time Requirements, in: M. Bubak, M. Turala, and K. Wiatr (Eds.). Proceedings of Cracow'06 Grid Workshop, Oct 15-18, 2006, Cracow, Poland, ACC Cyfronet AGH, 2007, pp. 11-20.
3. Krzysztof Zielinski, Marcin Jarzab, Damian Wieczorek and Kazimierz Balos, JIMS Extensions for Resource Monitoring and Management of Solaris 10, in Vassil N. Alexandrov and G. Dick van Albada and Peter M. A. Sloot and Jack Dongarra, editors, Computational Science – ICCS 2006, 6th International Conference, Reading, UK, May 28-31, 2006, Proceedings, Part IV, volume 3994 of Lecture Notes in Computer Science, pages 1039-1046, Springer, 2006.
4. OCM-G project home page: http://grid.cyfronet.pl/ocmg/

29. A Toolkit for off-line Performance Visualization of Grid Applications Based on OCM-G and Paraver

Maciej Zientarski, Lukasz Gruszka, Wlodzimierz Funika
Institute of Computer Science AGH, Krakow, Poland

Visualization as a graphic representation of abstract data relayed in text and numbers makes it easy to see certain patterns. In grid computing where data processing takes place in a highly distributed manner it is essential to have possibility to see the whole execution workflow. It enables the user to visually relate computing time to communication time, frequency of executed probes or library functions to their duration, communication delay to volume, or simply to observe how and where all processes are currently running.

OCM-G [1] is an on-line monitoring system that provides application-related execution and infrastructure data on-the-fly. It has been successfully developed for message-passing oriented applications and adapted for other programming paradigms, e.g. for Grid superscalar [3]. To visualize monitoring data, an external GUI tool is required. Within the EU IST CrossGrid, a performance measurement tool, G-PM was developed which complies to the OMIS interface underlying the design of OCM-G. G-PM allows for a number of advanced measurements and visualization modes, including support for user-defined metrics, oriented towards on-line mode of analysis. It does not provide off-line analysis.

On the other hand, sometimes it is beneficial to analyze the execution of a grid application off-line via replaying its trace collected when running. OCM-G provides a service for trace collection which allows to gather the monitoring data which corresponds to the analysis needs. After a thorough analysis of existing performance tools, we have chosen the Paraver tool [3] as one which provides a large number of advanced performance visualization modes based on a well-known space time diagram. Paraver is an off-line visualization tool i.e. it requires data for a whole computation process provided at once so it can visualize the already completed execution. To make OCM-G and Paraver work together, there was a demand to develop a tool adapting the data collection operation and monitoring data coming from OCM-G to the requirements of Paraver operation and input format. OCMG2PRV provides a set of tools that steer and automatically convert data gathered by OCM-G to Paraver input files.

OCMG2PRV provides a easy to use mechanism that enables the user to visualize all MPI functions calls. One of its extra features is possibility to visualize communication times, communication participants, and user-defined problems on a time line. In a most simple scenario it does not require any additional code injection into a probed application. MPI functions that should be visualized can be specified in a configuration file so that OCMG2PRVfilters only selected executions.

Acknowledgments. This work is supported by EU IST CoreGRID project IST-2002-004265.

References

1. OCM-G monitoring system http://grid.cyfronet.pl/ocmg/
2. R.M. Badia, R. Sirvent, M. Bubak, W. Funika, C. Klus, P. Machner, M. Smetek, Perfomance Monitoring of Gris Superscalar with OCM-G/G-PM: Integration Issues, in S. Gorlatch, M. Bubak, T. Priol (Eds.), proc. of the CoreGRID Integration Workshop, Oct 19-20, 2005, Krakow, Poland, ACC Cyfronet AGH, 2006, pp.321-331
3. Paraver visualization tool http://www.cepba.upc.es/paraver/index.html

30. Recent Extensions in Application Monitoring System OCM-G

Tomasz Duszka (1), Jakub Janczak (1), Tomasz Szepieniec (1,2)
(1) Institute of Nuclear Physics PAN, Krakow, Poland
(2) ACC Cyfronet AGH, Krakow, Poland

The main motivation of the user adapting their applications to grid environment is possibility to compute more efficient. Thus, the performance of the application adapted to the new environment is crucial for user's satisfaction. Detailed performance study on the application results on finding and fixing performance bottle-necks. This is especially important for parallel jobs. To meet these requirements OCM-G [1] monitoring system was developed. In this paper we described the extensions to OCM-G that was introduced recently.

A pool of services was extended by a new ones including monitoring load on nodes, free disk space, memory and processor usage, list of opened files by a job. Additionally there is a way to look on output files of jobs before these are done (a new tool called ocmg-tail is provided). For more control on running jobs user is able to send some input files to job after it has been submitted. For the constant frequency based task (which include especially synchronous requesting the system for specified value), there was provided timer object.

Another new feature of OCM-G which inevitably one should have in mind is a new command token, which noticeably increases the system powerfulness. This new feature makes OCM-G component a parent process for original application process – thanks to this any other process can be spawned, controlled by reading and writing to its standard descriptors. Having this possibility it is now possible to rerun the submitted application without the need of waiting for the new instance being submitted.

Through many new features introduced in OCM-G there could not be omitted new compatibility code added in latest version. GSI 2, supported before, has been supplemented with full version 3 and 4 support. Additionally, it is possible to run new versions of OCM-G on bare cluster environment again.

To increase the popularity and usability for the developers a new way of connecting to the OCM-G has been provided. OCMGJAPI, as it was called, is a JAVA API based on three layers. The first one allows the developer connecting to the OCM-G using either GSI or MCI connection type and sending bare messages. The second provides a set of monitoring system objects (such as nodes or applications), which are responsible for performing corresponding requests. As the system objects are stateful (can be attached or not attached), the third layer provides the API side statefulness, what noticeably decreases the requests number.

New features of OCM-G introduced in latest versions highly increase its usefulness in everyday use. These has proved their value in OCM-G based frameworks developed for application within BalticGrid Project [2]. The OCMGJAPI introduces a new abilities for developer to create a monitoring frontends, extending its spectrum to J2EE applications, portals and more.

References

1. OCM-G Home Page: http://grid.cyfronet.pl/ocmg/
2. IST BalticGrid Project, web page: http://www.balticgrid.org/

31. JFlooder – Application Performance Testing with QoS Assurance

Tomasz Duszka, Andrzej Górecki , Jakub Janczak, Adam Nowaczyk, Dominik Radziszowski,
Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland

Contemporary distributed systems, such as Grids, must guarantee high efficiency, scalability and failover. Attainment of these features requires complex tuning of numerous highly coupled configuration parameters at system, middleware and application level. Because even little, outwardly not significant, change of one parameter may affect system efficiency and even stability, tuning of distributed system must be validated by series of load tests. Planning, implementation, execution, presentation and comparison of results of these tests are not easy tasks; it is a main reason why in many installations these tests are skipped or reduced to simple stress testing. Well performed load tests are build upon QoS (Quality of Service) requirements and can help in detecting system behavior under different load – only results of such tests can be compared and may answer the question whether system tuning was correct and, what is more important, how the system will perform in production.

Popular tools available for generation of system load and results acquisition like Grinder or JMeter help a little but suffer from lack of automation in results comparison, dynamic load characteristic changes and support for QoS requirements. Imperfections of available application performance testing induce authors to create a new tool – JFlooder. JFlooder is a program that performs load tests of SOA conform application and middleware running on distributed infrastructure (cluster and grid).

JFlooder consists of agents running on workers nodes, responsible for generation of desired load and central graphical management console. The console is responsible for agent coordination during test execution in order to obtain required system load characteristic (plain, with load picks etc) and on-line or off-line results gathering and presentation. Plugin architecture of the application allows for very elastic test definition. Build-in plugins allows for easy definition of tests performing HTTP/RMI requests for accessing Web Services and execution of JRuby scripts. Build-in metrics covers system response time, system load and operation correctness. Custom plugins may perform any activity and measure any user-defined metrics. Results are presented in table and charts both as raw data and as aggregates (deviation, average, min, max etc.) in specific time periods.

The unique ability of the program is a possibility of detection of 'optimal operation point' for a system under test. Optimal operation point is a highest system load under which it can still guarantee given QoS parameters. JFlooder starts with a low system load and perform tests collecting metrics (with special interest in QoS parameters), next it automatically increases the load by starting additional agents and does the testing again – this process is repeated up to the moment in which QoS parameters are violated – optimal operation point is detected. This mechanism is very helpful for detection of system characteristic in a new or tuned environment. It without much effort answer the question what is the maximum load for which QoS parameters are kept.

JFlooder has also been successfully used for testing adaptation mechanisms of a component system. It is actually improved and will be available as open source project in the near future.

32. A Performance Visualization Tool – Candle

Tomasz Duszka (1), Jakub Janczak (1), Tomasz Szepieniec (1,2)
(1) Institute of Nuclear Physics PAN, Krakow, Poland
(2) ACK Cyfronet AGH, Krakow, Poland

Understanding, the performance behavior of parallel application running in grid environment require specialized tools. Few years of experience with building the application monitoring system OCM-G and the performance visualization tool G-PM [1], link with a modern programming tools resulted in a successor of G-PM called Candle [2] developed in the frame of BalticGrid Project [3]. In this paper we described a idea of Candle and indicate most important changes respect to G-PM.

First important improvement is that, the results of monitoring are kept in a storage – a new layer of performance data management. Each storage is an abstract repository of performance data, that can be saved for future analysis. Measurement uses predefined metric (IE. probe result or node load) to gather performance data and inject them to the specified storage. From now we can visualize these data using built in tools, but additionally can do more, by defining a "transformation". This transformation is defined using mathematical formulas (with many useful functions like integral, derivative) so it is capable of doing virtually anything in a simple and clearway. Also one can define an aggregation between storages which can i.e. calculate average node load for each site. Visualization subsystem provides a user with a large set of highly configurable charts. User is able to set between others: the way how data is being collected, the contexts which should be visualized, the way the data is being treated (some storage data combinations are possible).

Being a system build on top of OCM-G, Candle uses innovative way of connecting to MPI monitoring subsystem – its foundations lays in use of just created OCMGJAPI project. OCMGJAPI brings a new quality in this area introducing new levels of abstraction for underlying system and as a result significantly increases development comfort. The GUI strongly utilizes the Swing framework but powered up with flexdock library – thanks to it Candle is able to give the user a chance to configure his/her workspace anyway he/she wants – for example it's easy to look at load of some nodes while defining new transformation of incoming data and adding a new Database storage at the same time.

Candle combined with OCM-G makes an outstanding and unique solution for MPI applications monitoring being computed in grid. Being a quite easy to use tool provides a wide range of powerful features, such as flexible visualizations and transformations engines. As Candle is based on Java language and widely used technologies such as Swing or JFreechart it is very easy to extend its functionality and to use the tool on all platforms.

References

1. Balis, B., Bubak, M., Funika, W., Wismueller, R., Radecki, M., Szepieniec, T., Arodz, T., Kurdziel, M., Grid Environment for On-line Application Monitoring and Performance Analysis, Scientific Programming, vol. 12, no. 4,2004, pp. 239-251.
2. Candle – Performance Visualization Tool, web page: http://grid.cyfronet.pl/candle
3. IST BalticGrid Project, web page: http://www.balticgrid.org/

33. A View on Site Efficiency with Batch System Analysis Tool

Marcin Radecki, Michal Zajac
ACC CYFRONET AGH

In recent years, an emergence of world-wide grid infrastructure composed of thousands of processors located in a number of grid sites can be observed [1], [2]. One of the main goals of a grid site is to maximize utilization of resources by developing and implementing suitable resource allocation policy for supported Virtual Organizations. Although there are many tools giving possibility to observe load at a site like Ganglia [3], Lemon [4], however they seem to lack robust features focused on a site policy adherence.

A utility called Batch system Analysis Tool (BAT) is being created in scope of BalticGrid project [5]. BAT consists of a web interface [6] where the user can ask queries to the system which then performs specific operations on the data received from the database and present it in graphical form. The database itself contains information about jobs i.e. identifier, time queued and started, used resources etc. The administrator of a site can check if the policy regarding jobs is implemented correctly e.g. whether jobs get the assigned priority or if a VO is really using dedicated resources. Using this information the site administrator can identify opportunities to maximize resources utilization. On the other hand, the VO managers can inspect if their VO(s) are supported at sites at the agreed level.

BAT supports multiple grid sites. The data about jobs is taken by agents distributed over grid sites, sent to the server by encrypted connections and finally is stored in the database. To avoid infinite growth of the database some data retention mechanisms are in place. SSL-secured connections ensure data integrity, similarly a web interface is accessible via https for users with defined roles, which allows to present only the relevant data to authorized users.

Batch Analysis Tool allows to draw on-fly, highly configurable graphs where the user can select custom time period, specify VO(s) and other advanced parameters. The graphs for recent periods contain details at a maximum level of precision registered at the database.

Aside of the data about running and waiting jobs, BAT shows tasks which were not started within specific (user-defined) time period since their submission time. A few examples of other possible analysis are:

- VO efficiency – measure a ratio of CPU time to wall time for jobs within a VO,
- Job efficiency – same ratio but jobs are divided to four groups depending on execution time, and displayed on 20bars – each describing range of 5% efficiency,
- Sites summary – pie graph with statistics of summary wall time for each VO (or all of them) as a function of site, or vice versa.

This paper presents how the site administrator can use BAT tool for their day-to-day work. Paper begins with a section on architecture and functionality of system, next a site overview graph is presented with details and finally a use case of finding a performance bottleneck at a site is presented.

References:
1. EGEE project web site, http://www.eu-egee.org
2. DEISA project web site, http://www.deisa.org
3. Ganglia website, http://ganglia.sourceforge.net/
4. Lemon website, http://lemon.web.cern.ch/lemon/index.shtml
5. BalticGrid project web site, http://www.balticgrid.org [6] BAT web interface http://grid.cyfronet.pl/bat/

34. Application Management in Earth Science

Viet Tran, Ladislav Hluchy
Institute of Informatics, Slovak Academy of Sciences, Slovakia

Earth Science (ES) is an all-embracing term for sciences related to the planet Earth, covering a large and diverse user community. The major disciplines use mathematics, physics, chemistry, and biology to study behavior of the Earth and their spheres. Examples of typical ES applications are meteorology, hydrology, geology, and Earth observations.

Although ES communities actively participate on Grid projects including DataGrid, EGEE I and EGEE II, the number of applications using Grid in operation mode is still very small. There are still big gaps between ES applications and Grid technologies. The DEGREE project is aiming as increasing the collaborations and closing the gaps between ES communities and Grid developers.

In the DEGREE project, the requirements of ES applications have been collected and analyzed from samples of 21 typical ES applications. Important missing technologies that are required for full operation of ES applications have been identified. Some of the most critical ones are near-real time job executions, reliability and quality of service, intelligent workflow management, license management.

The ES communities does not stop only on giving requirements, they also actively cooperate with middleware developers and influence the development process. The specification for improvements for middleware developers is being proposed for middleware developers. Test suites have been done also for realistic testing and validation of the middleware for ES communities.

Two test suites have been proposed for the application management: flood application with focus on workflow management and earthquake application with focus on distributed job management. The test suites aim at providing well-documented test specifications and real applications plus data to the Grid developers for testin g and validation of middleware.

35. 3D Geovisualization Service for Grid-oriented applications of Natural Disasters

Eva Pajorová, Ladislav Hluchý, Jan Astalos
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

The state-of-the-art visualization libraries and tools (both 2D and 3D) offer many features that can satisfy virtually all needs of a typical scientific application. However, the approaches of visualization toolkits are differ. Our aim is to design a framework that will define a unified way of cooperation between Grid visualization applications and visualization clients. Working with the Grid technology implies increased complexity on one hand, and user demands for high interactivity on the other. The duration of the computations in the Grid environment being exceedingly long, the user naturally wants to see the intermediate results and requires means to modify the running computations if the intermediate results are not satisfactory. Our current purpose is to create unified framework that provides means for run-time monitoring of the running Grid applications and displaying their intermediate results in the client application.

This paper describes creation the basic grid tool for mentioned framework, concrete 3D Geovisualization service for Grid_based assessment applications of natural disasters. A lot of international projects oriented on natural disasters utilise grid computing and within grid solution raises requirement of visualization service for displaying their intermediate results and for presentation the final results. Such service requires unified standards like integration of input data formats and especially creation of unified visualization tool. It should integrate visualization requests of any kind of application oriented on computing of natural disasters. In case it is grid computing it has to be established also submit workflow, which controls execution of this visualization service. Development of all executable modules and solution of all above mentioned grid computing specific problems was subject of our scientific work presented in this article. The Geovisualization servis as well as submit workflow were tested on the applications solved in project MEDIGRID and on natural disasters applications solved in our institute The natural disasters like fires and floods become subject of science in research institutions more and more frequently. Topic of a lot of projects is how to prevent such disasters. Many applications from this area are using different kinds of simulation tools, which are producing output data for displaying the results of the computation. The purpose of 3D Geovisual service is to model and display results of various simulations of natural disasters like fire spread in time, its intensity and erosion or floods in time or landslides as well. The output of the service is terrain covered by ortophotomap or it is only shaded according to the altitude. On such prepared terrain is displayed the simulation of the fire spread in time, its intensity and erosion, flood water spread in time and soon. They are visualized statically or dynamically by adding red faces for the fire, blue faces for the water and brown faces for the landslides. Output of the service can also be the files representing the virtual reality of natural disaster, which can be presented by CAVE system.

36. Interactive Air Pollution Simulation in int.eu.grid

Branislav Simo (1), Martin Gazak (2), Ladislav Hluchy (1)
(1) Institute of Informatics, Slovak Academy of Sciences, Slovakia
(2) MicroStep-MIS

The paper presents the implementation of an interactive, grid, radioactive air pollution application. The application enables the user to simulate the spread of a pollutant particles released to the atmosphere using the Lagrangian trajectory model. The model is computationally intensive and based on the number of particles simulated requires a computation time from several minutes up to two hours to simulate a given scenario.

At the beginning the model was a sequential application running on Windows OS. In order to run in the grid infrastructure provided by the int.eu.grid project, the model has been ported to the Linux OS what required a new implementation of certain library functionalities provided by the Borland C++ Builder – the IDE used for application development on Windows. The model was further modified to use a parallel computation using the MPI standard to achieve performance increase. Parallelism has been implemented by dividing the particles into equally sized subsets and simulating them independently on individual computational nodes of the job. Such division was quite straightforward as the particles are simulated independently by the model.

One of the features being developed in the int.eu.grid project is the support for grid application interactivity. Interactive applications require a way to enable the user to send commands to the application and the application to send the responses to the user. What the commands are and how is the application state presented to the user is up to the application developer.

The user-application connection is realized by setting up a data channel between the application running in the grid and a special application specific plug-in running inside of the Migrating Desktop (MD) – a Java based graphical user interface (GUI) running as a client on the user's machine, whose basic functionality comprises accessing grid resources and managing user's jobs and data. The channel passes all the data from the standard output of the application to the plug-in and data sent to the channel are passed to the standard input of the application. In case of MPI application the channel takes the output from all MPI processes comprising the job, but the input is connected just to the master process.

Starting jobs from the MD allows the user to request the setup of the interactive channel, which is then connected to the application plug-in. Application plug-in has to explicitly support interactivity and support it in application specific way, so it usually has to be implemented for each application from scratch. For more advanced application that create complex interactive visualization using the Visualization ToolKit (VTK) there is a higher-level feature enabling to capture the visualization as a MPEG4 video stream and send it to the client plug-in, which in turn captures the keyboard and mouse commands and sends it to the application.

Our application needed just the raw interactive channel. When considering how to express the state of the simulation to the user we decided to send pictures created by the application running on the grid showing the map of the simulated area with particles plotted on it. Such image is sent for each simulation step in the JPEG format. It is decoded by the plug-in and shown to the user. Each step of the simulation the user can send two commands – to split the particles or to terminate the application. Particle split results in doubling the count of the particles, thus increasing the precision of the simulation from that point on, but also raising the computational requirements.

The interactive channel allows the users to view the development of the simulation scenario in real-time and allows them to influence the simulation interactively.

37. New System of Parallel and Biologically Realistic Neural Simulation

G.M. Wojcik, W.A. Kaminski, J. Ruthe, D. Stanislawek, M. Zukowski and M. Falski
Institute of Computer Science, Maria Curie-Sklodowska University, Lublin, Poland

Mammalian brains are one of the most complicated complex systems in the known Universe. Human brain consisting of 10^11 neural cells (neurons) has 10^14 degree of complexity and it is responsible for the origin of consciousness, mind, intelligence and other higher psychical functions. Even partial understanding of the brain functionality would allow for treatment of many diseases, creation of new generation of neural prothesis and explaining a wide range of cognitive phenomena lasting for centuries in mankind interest.

From the physics point of view each neuron is a complicated computational unit and can be simulated in computers. However, in the biologically realistic Hodgkin-Huxley (HH) model each neural cell is represented by set of several (sometimes several thousands) non-linear differential equations describing the behaviour of so-called neuron-equivalent electrical circuit. These equations can be solved only in numerical way. Simulations of neural networks consisting of thousands of biologically realistic HH neurons is always power consuming and good parallelisation of algorithms is required. Nevertheless we are still unable to simulate the whole brain in satisfactory detail.

There are several neural simulators (i.e., GENESIS, NEURON) using parallelisation techniques for modelling, however, their effectiveness is relatively low. They use script language that creates the model during the simulation and often the process of model creation takes longer than the simulation itself. Models implemented in this way cannot be used for solving other problems that they were originally dedicated to, their modification is relatively difficult. Simulators have been developed by many people for several years, mainly in C or C++, which also must have had some influence on their stability and efficiency.

In this paper we present the idea of new system for neural simulation. The main task of the project was to design the software making possible elastic development of large models by different groups of researchers. As natural consequence of high computational requirements, the system is optimised for parallelisation with use of cluster and heterogeneous grids. Using Java, software engineering techniques and parallelisation optimisation we created the system consisting of two main and separate modules: the model constructor and the simulator. We discuss the system architecture and implementation. The comparison of the GENESIS and our system's effectiveness is presented in some detail as well.

38. HLA Component Model on the Example of Multiscale Simulation

Katarzyna Rycerz (1,2), Marian Bubak (1,3), Peter M.A. Sloot (3)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland
(3) Faculty of Sciences, Section Computational Science, University of Amsterdam, The Netherlands

Building simulation system from modules of different scale is interesting, but non-trivial issue. One of the main challanges are the complex time interactions that can appear between elements of such applications [1]. High Level Architecture (HLA) [2], especially its time management service can be efficiently used to distribute and communicate multiscale components. Another important aspect worth to be addressed is the ability of flexible and transparent creation of distributed multiscale simulation system according to user needs. The issue of interoperability, software reuse and flexibility is directly addressed by Grid and Component technologies. Component technologies are specifically oriented on creating mechanisms for easily plugging and unplugging pieces of functionality to/from complex application. This mechanism can be also useful for distributed simulation system consisting of elements from different scale that can be join together according to the actual application(user) needs.

The component approach [3,4] to distributed simulations is becoming more and more popular. An interesting proposition of integration Gris Service and Component approach to distributed simulations is presented in [5].HLA [2] is a standard for large scale distributed simulations. Apart from many advanced features specyfic for such applications (like time, data and ownership management), HLA offers the ability of plugging and unplugging varius simulation models (also with different internal types of time management) to/from complex simulation system. This feature can be used to create HLA-based component model, where components are independent simulation models that can be dynamically joined into a coherent whole. In HLA terminology each component of a distributed simulation is called a federate and can form federation with other federates. Each federate is able dynamically join to a federation and resign from it. This approach differs from other popular component models (e.g. CCA [3], MOCCA [4]), as the federates are not using direct connections. Instead, all federates within federation are connected together using tuple space, which they can use for subscribing and publishing events and data objects.

Two main goals of our reseach was to (1) show how HLA standard, especially time management mechanism can be useful for multiscale simulations and (2) how such application can benefit from Grid and Component technologies. For the first purpose we analyzed the typical time interactions between multiscale components basing on modules taken from from Multiscale Multiphysics Scientific Environment (MUSE) [6], which is sequential simulation system designed for calculating behaviour of dense stellar systems like globular clusters and galactic nuclei. For the second purpose we are investigating an Grid environment that supports HLA component model. As a Grid platform hosting HLA components we have choosen H2O environment. The designed system enables the user to dynamically set up multiscale simulation comprised of chosen HLA components residing on the Grid. A user is able to decide which components can participate in federation and dynamically plug and unplug them to/from running simulation system. The example application is build from simulation modules taken from MUSE.

The experiments conducted have shown that distributing multiscale simulation using HLA time management features can be beneficial for such applications. Additionally, we have presented the idea of HLA component model, which enables the user to dynamically compose/decompose distributed simulations from multiscale elements residing on the Grid. We have also build preliminary prototype of the system supporting such component model.

In the future we plan to extend the designed environment, so a user will also be able to decide how components will interact with each other (e.g. by setting up appropriate HLA subscription/publication and time management mechanism) and to change nature of their interactions during simulation runtime.

Acknowledgements. This work was supported by EU IST-2004-004265 project CoreGRID, the Polish State Committee for Scientific Research SPUB-M and the Foundation for Polish Science.

References

1. K. Rycerz, M. Bubak, P.M.A. Sloot Using HLA and Grid for Distributed Multiscale Simulations Proceedings of International Conference of Parallel Processing and Applied Mathematics (PPAM'07), Gdansk, September 2007, LNCS (to appear)
2. High Level Architecture specification – IEEE 1516
3. R. Armstrong et al. The CCA Component Model for High-Performance Scientific Computing. Concurr. Comput.: Pract. Exper., 18(2):215--229, 2006
4. Malawski, M., Bubak, M., Placek, M., Kurzyniec, D., Sunderam, V.: Experiments with Distributed Component Computing across Grid Boundaries. In Proceedings of the HPC-GECO/CompFrame workshop in conjunction with HPDC 2006, Paris, France, 2006
5. X. Chen, W. Cai, S.J. Turner, Y. Wang: SOAr-DSGrid: Service-Oriented Architecture for Distributed Simulation on the Grid. Principles of Advanced and Distributed Simulation (PADS) 2006: 65-73
6. MUSE Web page http://muse.li/

39. Saleve: Supporting the Deployment of Parameter Study Tasks in the Grid

Peter Dobe, Richard Kapolnai, Imre Szeberenyi
BME IK, Budapest, Hungary

We present Saleve, a developers' tool to aid the creation of parallel running parameter study (PS) applications. It is prepared to cooperate with several distinct parallel computing systems, including different types of batch schedulers or Grid middleware systems. Without the need of knowing technical details of any specific middleware, Saleve enables for researchers to develop new parallel programs easily, but it also makes it possible to modify existing sequential programs in order to exploit the parallel infrastructure, for example a continent-wide computing Grid.

With its web service based client-server architecture, Saleve is an easy-to-use, extendible system conforming to the most up-to-date standards of Grid technology. In the Saleve configuration, the original parameter study application written in C is merged with the Saleve client library. Thus the Client is the program splitting the input parameter domain, performing the resource-demanding calculations, and finally aggregating the subresults. In the simplest case, the Client can be run locally as a stand-alone application. In the other mode of operation the Client uploads itself to a Saleve Server, witch can submit the task to a local scheduler or to a Grid. In this manner Saleve hides the details of the Grid or cluster infrastructure from non-IT expert users.

Each of the incompatible parallel computing technologies is handled by a plugin of the Server. To adapt Saleve to a new infrastructure, one has to develop a new plugin which takes care of the submission and the management of a task. Splitting up the input, creating jobs for subdomains and summarizing the subresult remains the duty of the Client. Recently we have developed a plugin for gLite, the middleware of the EGEE project.

The gLite plugin has risen pressing questions which led to the re-design of some aspects of the Saleve Server internals. First, the Server has to authenticate itself towards the Grid, which can be arranged by the plugin or by the user. Both approach has advantages and drawbacks. Second, an EGEE job has to be monitored to be sure it successfully runs and to re-submit it on failure. To achieve this, the Server needs to take care of its jobs proactively, and check them regularly.

In this paper we introduce the usage of Saleve. We give a quick overview of its plugin mechanism and the aspects of designing a new plugin. Then we plan to discuss the proposed solutions to the issues risen by the gLite plugin.

40. Towards a System for Interactive Parameter Sweep Applications on the Grid

Adianto Wibisono, Frank Terpstra, Victor Guevara, Zhiming Zhao, Adam Belloum and Marian Bubak
Informatics Institute, University of Amsterdam, the Netherlands

One type of applications that may benefit from what is offered by the Grid are parameter sweep applications. They involve processing of independent jobs, performing similar tasks on different data sets or exploring ranges of scenarios and parameter spaces. Parameter sweep experiment is an important task in the domains of system modeling and optimization. A straightforward approach is to execute the system with collection of pre-selected parameters and then choose the most interesting parameter settings based on some metrics on the observed simulation result. Most of the time, the interesting range of the parameters to be studied are unknowns, and though there are numerous existing method to automatically guide the parameter search, in actual experiment, trial and error is unavoidable. Performing parameter sweep experiments is exploratory by nature, and when the parameter space becomes large, such approach requires intensive computing and storage power; employing heuristic mechanisms in locating most interesting parameter region is crucial.

A number of parameter sweep framework are already developed, e.g., [2,3,5,7]. A typical parameter sweeping toolkit has two parts: a task planner and a parameter searching component. The task planner schedules computing tasks based on different combinations of parameter values, and the parameter searching tool optimizes the regions of parameter space and provides feedback to the task planner. So far, most of the systems focus on the task planner, e.g. in Nimrod/G [2], Pegasus [4], and little attention has been paid to the searching component. In Nimrod/O, several optimization algorithms have been included, e.g. Simplex, P-BFGS, Simulated Annealing etc. However, the selection of these algorithms is static in the experiment, which means the parameter searching component can not adapt strategies in one experiment. Since the searching component has little knowledge on the specific problem domain, static chosen optimization algorithm will not be sufficient for a large problem. Including human expertise in the runtime loop of parameters weeping will be crucial.

We conducted study of common practices in performing parameter sweep experiments based on literatures [1, 6,9-11] and experiences from our project (VL-e) use cases [8]. Based on this study we identify the need for interactivity during execution of parameter sweep experiments, types of interactive actions needed to steer parameter sweep execution and what is expected from each of this actions.

We propose a solution which enables interactive execution of parameter sweep experiments with a human in the loop. We describe a framework where scientist performing parameter sweep study could monitor and steer the direction of the parameter sweep executions. This new feature to monitor and to intervene with running parameter sweep simulation is something what is missing in existing frameworks for performing parameter sweep applications such as Nimrod/G [2], Condor-G [5], AppLes Parameter Sweep Template (APST) [3] or P-Grade [7].

Preliminary design of framework that would support interactivity and a prototype that provides a initial version of this interactivity will be presented. The framework consist of a front end component, where user can submit description of parameter sweep experiments (parameter space, observation space and sampling policy), parameter search coordinator component, which will manage the parameter space exploration, and a visualization and interaction component to allow user perform steering of parameter sweep execution. For resource management and submission to the grid resource we will use components from WS-VLAM [12], grid enabled workflow system from VL-e project.

This work was carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl). Part of this project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ). The authors of this paper would like to thank all the members in the VL-e SP 2.5 for discussions.

References

1. S. Ayyub, D. Abramson, C. Enticott, S. Garic, and J. Tan. Executing large parameter sweep applications on a multi-VO testbed. In Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on, pages 73-82, May 2007.
2. R. Buyya, D. Abramson, and J. Giddy. Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid. In High Performance Computing in the Asia-Pacific Region, 2000.Proceedings. The Fourth International Conference/Exhibition on, volume 1, pages 283-289, vol.1, 14-17 May 2000
3. H. Casanova, F. Berman, G. Obertelli, and R. Wolski. The apples parameter sweep template: User-level middleware for the grid. In Supercomputing, ACM/IEEE 2000 Conference, pages 60-60, 04-10 Nov. 2000.
4. E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, and M. Livny. Pegasus: Mapping scientific workflows onto the grid. In European Across Grids Conference, pages 11-20, 2004
5. J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke. Condor-g: a computation management agent for multiinstitutional grids. In High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on, pages 55-63, 7-9 Aug. 2001.
6. E. Huedo, R. Montero, and I. Llorente. Experiences on adaptive grid scheduling of parameter sweep applications. In Parallel, Distributed and Network-Based Processing, 2004. Proceedings. 12th Euromicro Conference on, pages 28-33, 11-13 Feb. 2004.
7. M. Kacsuk. Parallel program development and execution in the grid. In Parallel Computing in Electrical Engineering, 2002. PARELEC '02. Proceedings. International Conference on, pages 131-138, 22-25 Sept. 2002
8. S.D. Olabarriaga, A.J. Nederveen, and B. O'Nuallain. Parameter sweeps for functional mri research in the "virtual laboratory for e-science" project. In Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on, pages 685-690, May 2007.
9. A. Salman, S. Turovets, A. Malony, J. Eriksen, and D. Tucker. Computational modeling of human head conductivity, In International Conference on Computational Science (1), 2005. LNCS vol 3514, pages 631-638, May 2005.
10. V. Stankovski and W. Dubitzky. Special section: Data mining in grid computing environments. Future Generation Computer Systems, 23(1):31-33, Jan. 2007.
11. W. Sudholt, K.K. Baldridge, D. Abramson, C. Enticott, S. Garic, C. Kondric, and D. Nguyen. Application of grid computing to parameter sweeps and optimizations in molecular modeling. Future Generation Computer Systems, 21(1):27-35, Jan. 2005.
12. A. Wibisono, D. Vasunin, V. Korkhov, Z. Zhao, A. Belloum, C. de Laat, P.W. Adriaans, Bob Hertzberger: WS-VLAM: A GT4 Based Workflow Management System. In International Conference on Computational Science (3), LNCS vol. 4489, pages 191-198, 2007

41. A Complex Automata Model of HIV-1 Co-receptor Tropism: Mutation Rate Prediction

G. Ertaylan and P.M.A. Sloot
Section Computational Science, Faculty of Science, University of Amsterdam,
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
(g.ertaylan@uva.nl, p.m.a.sloot@uva.nl

Early infection with human immunodeficiency virus (HIV) is characterized by the predominance of CCR5-tropic (R5) virus. However, over the course of infection the CXCR4-tropic (X4) virus appears in the later stage of the infection in approximately 50% of the infected individuals and usually precedes an accelerated CD4+ T cell depletion with rapid disease progression.

The reason for this phenotypic switch and effects on the disease progression is still not clear. The interaction/competition between virus populations, co-receptor designation of various target cells and its effect on cell entry, the localization of the infection in certain tissue types, different clearance rate of X4 and R5 variants are some factors known to complicate the picture.

To investigate the effect of 'the longitudinal niche change' on co-receptor tropism, we developed a Complex Automata Model which consists of identically configured T helper cells and a freely mutating virus population with various co-receptor usages for R5, R5X4 or X4, where R5X4 stands for the intermediate population.

The results of our experiments show that the error threshold for the HIV-1 mutation rate is about 30 times the actual mutation rate of HIV-1, which is remarkably close to recently reported in-vitro values.

Our results indicate that the diversity in the virus population, accumulated during the disease course was sufficient to overcome the challenges in co-receptor designation of target cells. Therefore the changes in the environment and target cell range/conformation seem to be the main candidates for driving factor of the co-receptor switch in HIV disease course.

This work was done within the Virolab project using the Virolab Virtual Laboratory experimental environment, for more details see: www.virolab.org and http://www.science.uva.nl/research/scs/.

42. VIROLAB: A Distributed Decision Support System for Viral Disease Treatment

P.M.A. Sloot, A. Tirado-Ramos, G. Ertaylan, Breanndan O Nuallain, D. van de Vijver, C. Boucher, M. Bubak

The HIV drug-resistance interpretation systems are used routinely throughout the world in a clinical setting. More knowledge is rapidly becoming available upon which clinical decisions could be made. This knowledge, information, data and evidence from many sources are combined within a Decision Support System (DSS) to provide coherent judgements on drug-susceptibility. At the core of the DSS is a HIV drug-resistance interpretation system incorporating knowledge from the principal systems (Stanford HIVdb, Rega, ANRS, Virolab) in use throughout the world.

We describe an improved rule-based language which has adequate expressiveness and enjoys a fully-specified, formal semantics, allowing for automated reasoning over rule sets. Among the questions which can be addressed are:

- Ambiguity: Is the rule set internally ambiguous? Does it allow more than one interpretation?
- Completeness: Does the rule set have complete coverage?
- Consistency: Are there rules in the set which make contradictory predictions?
- Redundancy: Do some rules of a rule set subsume others?
- Dissonance: How do rule sets differ in their predictions?
- Predictive power: Can one rule set make more specific predictions than another or can it make predictions in cases where the other is silent?

The formal language which we present has a well-defined semantics that will allow for making judgements of the above kinds using reasoning that is either completely automated or at least semi-automated.

Furthermore recent findings have revealed the need to express multiplicative effects of certain mutations on drugs. The state of the art language for specifying HIV drug interpretation rules, ASI, in its present form, is limited to linear combinations of effects.

In future work we will use Bayesian hierarchical modelling to make predictive distributions in the presence of uncertainty. The full chain of analysis will combine Bayesian hierarchical modelling with probabilistic decision analysis based on utility attribution and/or multi-objective optimisation of such quantities as cost, chance and duration of survival or quality-adjusted life years.

This work was done within the Virolab project using the Virolab Virtual Laboratory experimental environment (See www.virolab.org). For more details see: P.M.A. Sloot; A. Tirado-Ramos; I. Altintas; M.T. Bubak and C.A. Boucher: From Molecule to Man: Decision Support in Individualized E-Health, IEEE Computer, (Cover feature) vol. 39, nr 11 pp. 40-46. November 2006.