1. Regional Customization of EGEE Infrastructure in Central Europe

Marcin Radecki, Tomasz Szepieniec, Aleksander Kusznir, Partyk Lason, and Andrzej Ozieblo
Academic Computer Center CYFRONET AGH, Krakow, Poland

EGEE project [1] aims at building the global grid infrastructure and provides reliable grid software layer to make it possible. While main stream of development focuses on services that constitute 'a backbone' of the infrastructure there is still a need for extending them to enable new class of applications as well as facilitate usability and maintainability of such an environment.

The main role of ACK CYFRONET AGH is to coordinate operation of the infrastructure in the Central European (CE) Region, this includes scheduling for introducing new software releases to a production environment. We take this opportunity and together with our partners in the region (especially GUP and CESNET) build the procedure for introducing the new services to the grid middleware.

The new services contributes to grid middleware in twofold way: (1) they extend the capabilities by integration of software developed in other grid projects, for example OCM-G, GPM, glogin from CrossGrid project; (2) they tune the infrastructure to support new classes of applications like MPI support.

Integration of new software into production environment should be done in the way which not destabilize the whole environment. For that reason we apply software certification procedure. The software is first evaluated and prepared to integration on so called certification testbed [2] which is composed of grid resources dedicated mainly for testing, building and validation of the software. There are three clusters involved in software certification located in Cracow, Linz and Prague that provide resources for simulating production grid environment to test the software. Automatic tools determine infrastructure stability while the developer can run the software and test it.

Prepared software enrich the LCG software and during process of upgrading resource centers in the CE region is deployed onto all the sites. Customized infrastructure is available for the members of VOCE [3] - a Virtual Organization accepting persons from students to scientists willing to take their first steps in grid technologies of EGEE project. The whole regional infrastructure includes almost 1000 CPUs and 30 TB of storage in 14 resource centers. The numbers takes into account about 120 CPUs maintained by CYFRONET in two sites: Zeus and Ares one based on IA32 and IA64 respectively.

[1] EGEE project homepage, www.eu-egee.org
[2] Certification Testbed organization webpage:
http://grid.cyfronet.pl/egee/tiki-index.php?page=Certification+Testbed+Main+Page
[3] Daniel Kouril, Ludek Matyska: The VOCE Environment.
http://www.egee.hu/grid05/download/day_4/egee05-voce.pdf

2. VIROLAB - A Virtual Laboratory for Decision Support in Viral Diseases Treatment

Peter Sloot (1), Charles Boucher (2), Marian Bubak (3,4), Alfons Hoekstra (1), Pawel Plaszczak (5), Alice Posthumus (6), David van de Vijver (2), Stefan Wesner (7), Alfredo Tirado-Ramos (1)
(1) Universiteit van Amsterdam
(2) Universitair Medisch Centrum Utrecht
(3) Institute of Computer Science AGH, Krakow,
(4) ACC CYFRONET, AGH, Krakow
(5) Gridwise Technologies, Krakow
(6) Virology Education B.V.
(7) Universitaet Stuttgart

The main objective of the ViroLab project is developing a Virtual Laboratory for Infectious Diseases that facilitates medical knowledge discovery and decision support for e.g. HIV drug resistance.

Large, high quality, clinical and patient databases have become available which can be used to relate genotype to drug-susceptibility phenotype. The relevant data has two main characteristics: it spans all temporal and spatial scales from the genome up to the clinical data; it is inherently distributed over various sources (virological-, clinical- and drugs-databases) that change dynamically over time. Using a Grid-based service oriented architecture, we vertically integrate the biomedical information from viruses (proteins and mutations), patients (e.g. viral load) and literature (drug resistance experiments), resulting in a rule-based decision support system for drug ranking. The Virtual Laboratory supports tools for statistical analysis, visualization, modelling and simulation, to predict the temporal virological and immunological response of viruses with complex mutation patterns to drug therapy. The Virtual Laboratory provides the medical doctors with a decision support system to rank drugs targeted at patients. It provides the virologists with an advanced environment to study trends on an individual, population and epidemiological level. By virtualizing the hardware, compute infrastructure and databases, the virtual laboratory is a user friendly environment, with tailored workflow templates to harness and automate such diverse tasks as data archiving, data integration, data mining and analysis, and modelling and simulation. HIV drug resistance is one of the few areas in medicine where genetic information is widely used for a considerable number of years. Large numbers of complex genetic sequences are available, in addition to clinical data. ViroLab offers a unique opportunity as a blueprint for the potentially many diseases where genetic information becomes important in future years.

References
[1] VIROLAB - EU IST STREP Project 027446; www.virolab.eu

3. CHARON System - Framework for Applications and Jobs Management in Grid Environment

J. Kmunicek [1], P. Kulhanek [2] and M. Petrek [2]
[1] Institute of Computer Science, Masaryk University, Brno, Czech Republic
[2] National Centre for Biomolecular Research, Brno, Czech Republic

Here we present a generic system for utilization of application programs in the EGEE Grid environment - the CHARON system. Charon was developed by computational chemistry community in the Czech Republic to provide easy manageable, comfortable and modular environment to fulfill specific requirements of computational chemistry application users. It currently offers an alternative to standard LCG/EGEE environment in application-generic Virtual Organization for Central Europe (VOCE).

Present-day implementation of Charon system is completely compatible with scripting roots of the EGEE Grid environment, provides comfort computational jobs management by encapsulation of available LCG/EGEE middleware enviroment, support for smooth administration of large amount of computational jobs and enables easy retracing of already finished calculations. Compared to widely-spread graphical user interfaces (i.e. portals) Charon is oriented towards users requiring simple but feature-rich and powerful command line and scripting interface that offers support for tens and hundreds of jobs withing a single research project.

Taking into account the cornerstones on which the system is built up - modularity and generality - it is not targeted only for molecular modelling purposes but represents a generic application framework easily adoptable for broad set of generic application areas and their specific programs. Moreover Charon also permits to utilize resources from non-EGEE Grids as well. Therefore Charon is expected to be one of useful tools available for promoting utilization of Grid environment for general public purposes.

4. Early Experiences on Application Checkpointing for Realizing QoS Provision

Felix Heine, Matthias Hovestadt, Odej Kao, Oliver Marquardt and Kerstin Voss
Paderborn Center of Parallel Computing (PC2), Universitaet Paderborn, Germany

The provision of dependability and predictability will be vital for realizing next generation Grid systems. If the commercial customer should be attracted to use existing Grid systems for computing his business critical jobs, he must be able to rely on getting the results in time. In this context, a Service Level Agreement (SLA) is a powerful instrument for defining a comprehensive requirement profile. This way, the customer can not only specify his particular demands concerning quality and quantity of resources, but also demand for service quality characteristics, e.g. the adherence to a fixed deadline for job completion.

Services at Grid middleware layer (e.g. broker services) are responsible for negotiation of such an SLA and runtime handling. However, solely focusing on Grid middleware services is not sufficient. Since local resource management systems (RMS) are providing their resources to the Grid, Grid middleware has to realize jobs by means of these local RMS. If these RMS provide best effort service only, the capabilities of Grid middleware in providing dependability and predictability are quite limited.

The EU-funded project HPC4U (IST-511531) targets on realizing a highly predictable Grid fabric for future Grid systems. At this, a system will be developed which is able to negotiate on SLAs for new job requests, provide the negotiated level of service quality, and operates fault-tolerant in respect to resource outages within the local administrative domain. This fault-tolerance will be realized by means of application transparent checkpointing mechanisms, such that fault-tolerance can be provided for arbitrary (even commercial) applications.

HPC4U is a 36 month project, which has started in mid 2004. At the current point of time, the first technical workpackage has been successfully concluded. The HPC4U system is able to checkpoint non-parallel applications and migrate the checkpointed application to a spare resource within the same cluster system. Within the second workpackage, the checkpointing capabilities will be enhanced to also cover MPI-parallel applications and the migration to remote cluster systems within the same administrative domain.

Process checkpointing is crucial for the overall success of the project, since it represents the limiting factor for providing fault-tolerance to the running application. In case of a failing resource the resource management system can use previously generated checkpoints to resume the computation, thus limiting the impact on the application completion. Beside a resource management system and a subsystem for process checkpointing, the HPC4U system will also comprise subsystems for network and storage. The storage subsystem will snapshot the data partition of an application at process checkpointing time. The network subsystem complements the process subsystem for handling parallel applications, since it allows to checkpoint the entire network state of the application (e.g. protocol queues or in-transit packets). This way, HPC4U guarantees consistency between the original and the restarted process, because the checkpoint dataset comprises the process checkpoint, as well as network and storage state.

This paper will first describe the project's demands on checkpointing technology and point out how HPC4U technology will advance the current state of art. In the main part, the paper will present the project's experiences on using checkpointing solutions for parallel and non-parallel applications, covering aspects of compatibility, impact on performance of application, and required time for restart in case of resource outages.

5. Integrating Dynamic Clusters in CLUSTERIX Environment

Roman Wyrzykowski (1), Konrad Karczewski (1), Gerard Frankowski (2), Radoslaw Krzywania (2), Kazimierz Balos (3), Jacek Kosinski (3)
(1) Institute of Computer and Information Sciences, Czestochowa University of Technology, Poland
(2) Poznan Supercomputing and Networking Center, Poland
(3) Academy of Mining and Metallurgy AGH, Cracow, Poland

The main objective of the CLUSTERIX national grid project is to develop mechanisms and tools that allow for deployment of a production grid with the core infrastructure consisting of local PC-clusters based on 64-bit Linux machines. Local clusters are placed accross Poland in independent centers connected by the Polish Optical Network PIONIER, providing dedicated 1 Gb/s channels. Currently the core infrastructure of the CLUSTERIX comprises 250+ Itanium2 CPUs located in 12 sites. At the same time, the computing power of the CLUSTERIX environment can be increased dramatically by connecting additional clusters. These clusters are called dynamic ones because it is assumed that they will be connected to the core infrastructure in a dynamic manner, using an automated procedure. For example, machines located in university laboratories and used for teaching purposes can be connected to the CLUSTERIX core at night time or during weekends.

This paper presents the concept and details of implementation of the method for the integration of dynamic PC-clusters in the CLUSTERIX grid environment.

First of all, the architecture of the CLUSTERIX infrastructure has been tailored to implement this functionality in an efficient and secure manner. In particular, each local cluster in the core is provided with a dedicated firewall/router. Such a solution allows for a balanced implementation of the integration procedure giving the possibility to choose the most appropriate core cluster to establish connection with the corrresponding firewall of a dynamic cluster.

Before starting the integration procedure, the necessary components of the CLUSTERIX middleware have to be installed on the dynamic cluster. Besides the Globus Toolkit and local batch system, these middleware include at least the Virtual Users' Accounts System (VUS) and security components. Also, the firewalls in both the core clusters and dynamic ones should be properly configured. Finally, in the CLUSTERIX core it is necessary to have the monitoring system properly installed and configured. The important requiremnt to this system is ability to cooperate with monitoring agent installed on a dynamic cluster, and the CLUSTERIX resource broker.

During the dynamic cluster integration, it registers in the grid monitoring system. It initializes a script running on a chosen firewall in the core. This script is responsible for the creation of a dedicated ssh tunnel between the firewall and the access node of the dynamic cluster. The necessary components of the CLUSTERIX middleware are then started, including the resource broker. As a result, the dynamic cluster becomes a part of the CLUSTERIX environment. When the cluster is taken down, a special procedure is applied to inform the monitoring system that it should initiate performing appropriate actions, e.g. migration of unfinished jobs.

The connection of dynamic local clusters to the CLUSTERIX backbone opens possibilities to access a shared environment with the extraordinary computational power, and dedicated applications designed to take advantage of the power delivered by the computational grid. For example, an experimental installation with 802 Itanium2 CPUs offering a peak performance of about 4,5 TFLOPs has been created.

6. CLUSTERIX Data Management System: Methods of Improving Its Scalability, Security and Fault Tolerance

Lukasz Kuczynski, Konrad Karczewski, Roman Wyrzykowski
Institute for Computational and Information Sciences, Czestochowa University of Technology, Poland

Grid applications deal with large volumes of data, which should be accessed in a fast and secure manner. Consequently, effective data management solutions are vital for the success of grid technology. The CLUSTERIX Data Management System (CDMS) has been developed in the CLUSTERIX Polish grid project, based on the analysis of both the end-user's requirements and existing implementations. During its development, a special attention has been paid to making the system reliable, secure, and scalable, aiming at the same time at creation a user-friendly grid data storage system.

CDMS has been provided with a variety of solutions aiming at provision of a fast and scalable access to data, as well as a required level of security, for both the transferred and stored data. In particular, the high level of security for stored data has been provided through the development of an advanced mechanism for the efficient control of access to data, based on system privileges and access lists. The transport subsystem of CDMS includes a variety of transport agents responsible for the implementation of effective data transfer mechanisms, like data stripping and their ciphering on the fly. Such an approach allows also for the implementation of so called chain-transfer, in order to provide a balanced network loading.

A highly modular design of CDMS makes it possible to replace standard decision modules, based on heuristic procedures, by alternative modules possessing elements of artificial intelligence, e.g., neuron networks, evolutionary algoritms. Moreover, owing to the application of mechanisms of dynamic loading of modules, which are available in majority of modern operating systems, the functionality of CDMS can be dynamically changed without the necessity to stop the data broker activity.

The architecture of CDMS assumes the distributed operation model. It is planned that in forthcoming versions of CDMS this feature together with the modular design of the system will assure its higher fault tolerance through elimination of a single point of failure. In particular, multiple instances of the data broker will be running concurrently, and their coherence will be provided by a synchronization subsystem. This paper presents the concept and details of implementation of this mechanism, which are planned to be adopted in CDMS.

Finally, the important advanatage of CDMS for end-users is that all low-level details related to operation of CDMS and data transfers are concealed behind the abstract layer of a virtual file system. The implementation of this layer has required the development of a dedicated set of tools. As a result, while integrating the CLUSTERIX Data Management System with end-user's applications, it looks like a standard file system.
For example, taking into account Grid specific networking parameters, CDMS tries to optimize data throughput via replication and replica selection techniques.

7. Inter-application Control Through Global States Monitoring on a Grid

J. Borkowski, D. Kopanski, M. Tudruj
Polish-Japanese Institute of Information Technology, Warsaw, Poland

This paper presents methods for organizing execution control of Grid applications by means of predicates computed on globally monitored application states. This new Grid control paradigm [1] has been embedded in a graphical Grid-application design environment (Grid PS-GRADE) [2], which is currently under implementation. It enables designing graphically inter applications control on a Grid in a much more flexible way. Grid-level application state monitoring is done by a special kind of structural application elements called synchronizers. Grid-level synchronizers collect state reports from applications, construct consistent Grid-level global states, evaluate predicates on them and send control signals to applications. Control signals can modify applications behaviour in asynchronous manner. A similar control method based on monitoring of global process states by application-level synchronizers has been also embedded internally inside Grid applications [3]. In this way a flexible user programmable control infrastructure has been created, which enables asynchronous control at any desired granularity level. In this infrastructure application-level synchronizers co-operate with Grid-level synchronizers thus providing new structurized methods for conveying intra and inter application control decisions.
In this paper the following co-operation schemes included into the proposed Grid environment will be discussed:
1. Simple workflow. A selected application starts executing after a set of selected applications is completed. Example: complicated scientific computations performed stage by stage in different computer networks.
2. Alternative workflow. One of several applications is selected for execution depending on the results (state) of former applications. Example: one of two available program packages is run depending on computation results performed so far.
3. Supporting workflow: A set of curently executed applications require activation of auxiliary applications, which will provide useful results. Example: In a coarse grain simulation of a system of moving objects a collision that appears, stimulates a change in the Grid application configuration. An application which models the collision in a detailed way (with a fine granularity of events) is activated. After the results of the detailed simulation of the collision are known, the coarse grain simulation continues.
4. Partial canceling of workflow: Applications that become superfluous from the global grid-level point of view are stopped. Example: The exhaustive parallel search on a Grid for the optimal solution in a solution space

is stopped or restricted when the search provides a satisfactory solution.
5. Workflow coupling: A global status of many applications is monitored and control directives are distributed to particular applications as needed. Applications compute parameters that are subject to mutual exchange. Some parameters in meta-level applications are updated with the use of results of some auxiliary computations. Example: An exhaustive search run on a Grid using Branch & Bound method is supported by heuristic search algorithms, results good for search bounding that come from the heuristic algorithms are introduced to the B&B application.
The paper shows how the described application co-operation schemes can be implemented in the proposed Grid system, which supports the control based on global computation state monitoring using an infrastructure provided by application and Grid level synchronizers. Resulting efficiency improvement is evaluated.

Bibliography
1. M. Tudruj, J. Borkowski, D. Kopañski, Parallel program graphical design with program execution control based on global application states, 3rd International Symposium on Parallel and Distributed Computing, ISPDC 2004, Cork, Ireland, July 2004, IEEE Computer Society Press, pp. 240-247.
2. D. Kopanski, J. Borkowski, M. Tudruj, Grid applications control based on synchronizers, Cracow Grid Workshop '05, Cracow

8. Discovery Service for JMX-enabled Monitoring System - JIMS Case Study

Krzysztof Wojtas, Leszek Wasilewski, Kazimierz Balos, Krzysztof Zielinski
Institute of Computer Science AGH, Krakow, Poland

One of the significant problems in the domain of Grid monitoring is keeping possible accurate image of the set of available clusters. Although there are many monitoring services with cluster discovery facility, there is a lack of such solutions with ready-to-use implementation in the form of Java/JMX (Java Management Extensions) component communicating through Web Services. Such component could be used in other monitoring frameworks for the purpose of dynamic cluster discovery, as a discovery service for network of mobile devices, and for many other dynamically changing environments. The paper shows the solution of the problem in the scope of resource monitoring in the Grid, consisting of many clusters. Such environment is supposed to allow the Grid configuration to be variable and dynamically change over time. New clusters can be added, switched off or some kind of malfunction can make them unavailable. Such events should be handled by the system without any administrator activities. This is the genesis and one of the most important requirements for Global Discovery Service (GDS) module, which is described in this article. Implementation of the GDS module is based on Global Discovery Service Protocol specification, which is the underlying theory and has been created to fulfill system prerequisites. The protocol requires, that every cluster has its own operating instance of GDS. It is also assumed that one of the instances acts as a Global Registry (GR) and keeps information about the configuration of clusters in the Grid. The election of GR is performed dynamically and in case of its unavailability or network communication failure, new GR is reelected. Protocol supports situations when all clusters are raised in the same moment and no GR is selected yet. In such case, which is similar to the case when GR fails to respond on heartbeats, election algorithm is used and new GR is chosen. The protocol provides solution in most common critical situations that can be met in communication in the Internet, what is described further in the paper. Global Discovery Service is only one of the services, which can be deployed in JIMS monitoring system. It allows Grid Monitoring Client application to read the Grid topology configuration and makes the Grid topology information accessible for every system module, which is configured for using the GDS. Successful implementation and deployment of the GDS modules for use with JIMS monitoring system in Grid networks (CrossGrid, Clusterix) proves that the presented solution is reliable and can be used in other projects. The paper is organized as follows: Section 2 contains requirements for the Global Discovery Service. Section 3 describes detailed concept of the presented solution and specifies the GDS Protocol. Section 4 covers the issue of protocol implementation using Java language, JMX and Web Services. JIMS case study: the implementation and issues of GDS module deployment in JIMS monitoring system, is described in Section 5. Finally, related work is presented. Paper is ended with conclusions.

9. WSRF2OWL-S: A Framework for Generating Semantic Descriptions of Web and Grid Services

Marian Babik (1), Ladislav Hluchy (1), Jacek Kitowski (2), Bartosz Kryza (3)
(1) Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
(2) AGH University of Science and Technology, Poland
(3) ACK Cyfronet AGH, Poland

Web Service Resource Framework (WSRF) is a recent effort of the grid community to facilitate modeling of the stateful services. Design and development of the WSRF service based systems is quite common and there are several emerging WS initiatives, which tries to automate the process of discovery, composition and invocation of such services. The semantic web services are a typical example, showing the potential of how ontological modeling can improve the shortcomings of the service oriented computing. One of the major obstacles in the process is the development of the ontologies, which describe web and grid services. Although, there are numerous standards for modeling semantic services, there are very few frameworks and tools, which can help automate the process of generating the semantic descriptions of services.

The article presents a framework, which can semi-automatically generate the OWL-S descriptions for both stateful and stateless services based on the Web Service Description Language (WSDL) and corresponding annotations. Such functionality is inevitable in the grid environment hosting a vast number of services, which have to be semantically described in order to enable automated discovery, composition and invocation. Design and implementation of the framework will be described and relevant use cases will be shown to demonstrate the functionality. The framework is based on the well known OWL-S API and extends it to cover the WSRF services, OWL-S extensions and OWL-S annotations.

We will highlight the number of issues, that we have faced during the design and development of the framework. Further, we'll provide an application scenario based on the flood-forecasting simulations showing how this framework can be used to create semantic descriptions of the WSRF services.

10. Multiple HLA Federate Processes in Grid Environment

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

The subject of this talk is a Grid management service called HLA--Speaking Service that interfaces actual High Level Architecture (HLA)[1] application with the Grid HLA Management System (G-HLAM) [2]. The design of the architecture of G-HLAM is based on the OGSA concept that allows for modularity and compatibility with Grid Services already being developed. The group of main G-HLAM services consists of a Broker Service which coordinates management of the simulation, a Performance Decision Service which decides when the performance of any of the federates is not satisfactory and therefore migration is required, and a Registry Service which stores information about the location of local services.

On each Grid site supporting HLA there are local services for performing migration commands on behalf of the Broker Service as well as for monitoring of federates and benchmarking. The HLA-Speaking Service is one of the local services interfacing federates running on its site to the G-HLAM system. HLA Speaking Service is responsible for execution of an application code on the site it resides and manages multiple federate processes. The version for single federate process was described in [3].

We present the functionality of the HLA-Seaking Service with an example of N-body simulation of dense stellar system. Such simulations remain a great challenge in astrophysics and there is a need for a computer infrastructure that will significantly improve their performance. We believe that the Grid is a promising environment for such requirements, since it offers the possibility of accessing computational resources that have heretofore been inaccessible.

Acknowledgements:
This research is partly funded by the EU IST Project CoreGRID, the Polish State Committee for Scientific Research SPUB-M grant, and by the Dutch Virtual Laboratory for e-Science project (www.vl-e.nl).

References:
[1] HLA specification, http://www.sisostds.org/stdsdev/hla/
[2] K.Rycerz, M.Bubak, M.Malawski, and P.Sloot. A Framework for HLA-Based Interactive Simulations on the Grid SIMULATION, 81(1):67--76, 2005.
[3] K.Zajac, M.Bubak, M.Malawski, and P.M.A Sloot. Towards a Grid Management System for HLA-Based Interactive Simulations. In: S.J. Turner and S.J.E. Taylor, editor, Proceedings Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2003), pages 4-11, Delft, The Netherlands, October 2003. IEEE Computer Society.

11. Fine-grained Instrumentation and Monitoring of Legacy Applications in a Service-Oriented Environment

Bartosz Balis(1), Marian Bubak(1,2), Krzysztof Guzy(1)
(1) Institute of Computer Science, AGH University of Science andTechnology, Krakow, Poland
(2) Academic Computer Centre CYFRONET AGH, Krakow, Poland

Grid systems based on modern service-oriented architecture (SOA) still heavily use legacy code. For needs of debugging and performance measurement there is necessity to monitor such legacy applications. The basis of monitoring of applications is instrumentation which in a service-oriented environment can not be static -- we usually do not have opportunity to instrument application source code and deploy it at runtime. We propose a solution for instrumentation which is: (1) dynamically enabled and disabled, (2) fine-grained to enable monitoring at the level of code regions, (3) accessible through a standardized instrumentation service to expose instrumentation functionality to arbitrary tools and services. Current efforts do not yet address those problems.

We present a framework to instrument and monitor legacy applications available via a service interface. In this effort, we employ several existing systems and specifications. The OCM-G system [1] is useful for monitoring of MPI applications. We extend the OCM-G to support the concept of Standard Intermediate Representation (SIR) [2] to have an abstract view of application as a convenient way for the user to pick individual code regions to be instrumented. We have designed an instrumentation service compliant with a standardized language WIRL (Workflow Intrumentation Request Language) for specifying instrumentation requests. The mentioned functionality is integrated with the GEMINI monitoring infrastructure [3], which provides us with opportunity to build custom sensors in order to collect information about any entity we want to monitor (legacy application in our case).

Our approach to instrumentation is to combine source code instrumentation and binary wrapping with the dynamic control of the measurement process at runtime. The instrumentation is inserted statically via patching of source code or binary libraries, while activation and deactivation of the instrumentation is done at runtime. We provide a tool to automatically insert probe functions at defined places into source files and to generate SIR descriptions of the code. The OCM-G system can dynamically enable/disable instrumentation probes. We include SIR into the application executable from where it can be retreived by the OCM-G. The OCM-G itself is hidden behind a custom sensor compliant with the GEMINI infrastructure. Since the OCM-G uses the OMIS interface [5] for requests for monitoring information, we translate requests expressed in WIRL into OMIS. The custom GEMINI sensor delivers information to Monitoring Service of GEMINI from where it can be obtained by interested clients.

Acknowledgements. This research was partially funded by the KWfGrid Project and K-WfGrid SPUB-M.

References
[1] OCM-G - Grid-enabled OMIS-Compliant Monitoring system, http://www.icsr.agh.edu.pl/ocmg
[2] Clovis Seragiotto, Jr., Hong-Linh Truong, Bernd Mohr, Thomas Fahringer, Michael Gerndt, Tianchao Li, 'Standardized Intermediate Representation for Fortran, Java, C and C++ Programs', APART Technical Report Workpackage 1, http://www.fz-juelich.de/apart
[3] B. Balis, M. Bubak, J. Dziwisz, H.-L. Truong, and T. Fahringer. Integrated Monitoring Framework for Grid Infrastructure and Applications. In P. Cunningham and M. Cunningham, editors, Innovation and the Knowledge Economy. Issues, Applications, Case Studies, pages 269-276, Ljubljana, Slovenia, October 2005. IOS Press.
[4] OMIS - On-line Monitoring Interface Specification. Version 2.0. Lehrstuhl fur Rechnertechnik und Rechnerorganisation Institut fur Informatik (LRR-TUM), Technische Universitat Munchen. http://wwwbode.informatik.tu-muenchen.de/~omis

12. Real Time Monitoring of Web Service Applications

Marian Bubak [1,2], Wlodzimierz Funika [1], Dominik Dziok [1], Marcin Koch [1], Allen D. Malony [3], Marcin Smetek [1] and Roland Wismueller [4]
[1] Institute of Computer Science, AGH, Krakow, Poland
[2] ACC CYFRONET AGH, Krakow, Poland
[3] Department of Computer and Information Science, Univ. Oregon, Eugene, USA
[4] Fachgruppe BVS, Universitaet Siegen, Siegen, Germany

The paper presents results of monitoring a real web service application. It introduces the issue of performance monitoring and visualization of distributed applications using the J-OCM monitoring system [1] and the TAU performance visualization environment [2]. Since J-OCM provides a raw OMIS-based [3] interface it has been integrated with TAU within the SCIRun environment [4]. SCIRun is a kind of problem solving environment which provides the framework for implementing modular systems.

The tool we present in this paper is addressed to developers of the distributed systems interested in upgrading the performance of the system they develop. It allows to observe the execution of every part of the distributed application: hosts, application servers, web services, and the operations implemented.

The primary goal of the paper is to present the result of real time monitoring of the example distributed application. The application we monitor executes on multiple homogenous physical machines. The design of the application assumes both domain and functional decomposition of the problem. The application itself implements the common operations on 2D matrices, such as multiplication, addition, transposition, and inversion.

We perform operations on random data and observe the behavior of the application at each stage of processing the request by the web server. We monitor and visualize the time consumed and also the size of the messages sent.

Finally, we perform some general tests on the monitoring system. They are intended to measure the impact of the tool on the performance of the monitored application.

Acknoledgements. This research was partially funded by the KBN grant 4T11C 032 23.

References
[1] W. Funika, M. Bubak, M.Smetek, and R. Wismueller. An OMIS-based Approach to Monitoring Distributed Java Applications. In: Yuen Chung Kwong, editor, Annual Review of Scalable Computing, volume 6, chapter 1. pp. 1-29, World Scientific Publishing Co. and Singapore University Press, 2004.
[2] TAU's 3-D Profile Visualizer - ParaVis, University of Oregon, Computer and Information Science http://www.cs.uoregon.edu/research/paracomp/tau/tauprofile/dist/paravis/
[3] Ludwig, T., Wismueller, R., Sunderam, V., and Bode, A.: OMIS -- On-line Monitoring Interface Specification (Version 2.0). Shaker Verlag, Aachen, vol. 9, LRR-TUM Research Report Series, (1997) http://wwwbode.in.tum.de/~omis/OMIS/Version-2.0/version-2.0.ps.gz
[4] C. Johnson, S. Parker, "The SCIRun Parallel Scientific Computing Problem Solving Environment". In: Proc. Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999.
[5] W. Funika, M. Koch, D. Dziok, M.Smetek, and R. Wismueller. Performance visualization of Web Services using J-OCM and SCIRun/TAU. The 2005 International Conference on High Performance Computing and Communications, 2005.

13. Monitoring Web Services with the J-OCM Monitoring System

Lechos³aw Trebacz, Wlodzimierz Funika, Piotr Handzlik, Marcin Smetek

The monitoring of an application is aimed at observing, analyzing and manipulating the execution of an application, which gives information about different aspects of its life.

The paper presents a Java-related monitoring system for distributed applications which are oriented towards the use of web services. In this paper, we focus on the issue of building a monitoring platform based on a well-defined interface between a monitoring system organized as middleware and tools that use the means provided by the monitoring middleware. Web Services as paradigm of distributed programming represent a new platform on which developers can build distributed applications featuring interoperability, platform-independence and language-neutral Web protocols. The three main technologies used in Web Services are: SOAP protocol, WSDL language and UDDI.

The one of many platforms which supports deploying web services is Axis with Tomcat. Apache Axis is an implementation of the SOAP protocol and enables to create web services on Tomcat. Our system extends the J-OCM monitoring system, and calling the functionality of this extension will comply to the same syntax as that of J-OCM. J-OCM constitutes a low-level layer of the architecture enlarged by a special extension for web services. On one hand we aim at the observation and analysis of SOAP messages exchanged with a Web Service (by dynamic instrumentation of Axis code), on the other hand our system is intended to monitor the characteristics of the remote call, for example: request time, execution time, response time.

The extension is created with use of JDK 1.5.0 which provides a better instrumentation as well as a comprehensive Java Virtual Machine Tool Interface (JVMTI), to collect data from JVM. The main goal of our paper is to describe an extension to JOCM system, which enables the monitoring of Web Services-based applications. This extension should: provide information about running Web Services and their number, provide information about main stages during a web service life cycle, for example: the event of receiving a request from the client, the events of starting particular services, the duration of SOAP message parsing etc., makes it possible to manipulate particular services, for example: enables stopping any operation at any moment, enables calling a method of a web service from within the monitoring system etc., makes accessible all information about a web service, makes it possible to check, which stages of a web service life cycle take most of CPU time, to analyze the performance of this application, provide information about an errors during running Web Service.

This extension is intended to work on-line, because it is very important especially in the monitoring of Web Services, to provide prompt information for the tools operating on the application.

This research is partially supported by the KBN grant 4T11C 032 23.

Bibliography
1. Ludwig T., Wismueller R., Sunderam V., Bode A.: OMIS - Online Monitoring Interface Specification (Version 2.0). Shaker Verlag, Aachen, vol. 9, LRRTUM Research Report Series, 1997
2. Marcin Smêtek: OMIS-based Monitoring System for Distributed Java Applications, M.Sc. thesis, AGH, Krakow, 2003
3. http://ws.apache.org/axis/java/architecture-guide.html

14. A Case Study of Interoperability in a Distributed Tools Environment

Marcin Bialek, Wlodzimierz Funika, Piotr Pegiel, Marcin Smetek
AGH, Krakow, Poland

Nowadays, more and more complex programs and computer systems are being created. Often these programs are specialized - programmers concentrate on very narrow areas of knowledge, what restrains their products from reaching broader audience. In many cases a solution can be to combine several good programs into a large system. Very often, however, simple information exchange between programs is not sufficient and a higher level of cooperation of individual programs (which is called "interoperability") is required. A classic example could be the cooperation of the debugger and the code editor.

Applications' cooperation requires a very well designed system of information exchange. In our solution, we use JINEXT, an interoperability extension to the OCM monitoring system, underlied by the OMIS interface that enables communication between many programs in a distributed environment. Additionally, JINEXT allows each cooperating application to retain transparency of its implementation. As a result the applications do not need to know anything about each other and they are still able to co-operate.

Our case study focuses on a debugger and a source code editor. To meet the conditions that may occur in real life, we decided to combine the application we wrote with the program that was developed completely independently. To do this we used the well known "gdb" as the debugger and created the source code editor.

The idea of the tools cooperation using JINEXT is fairly simple - it uses a mediator's template that allows the communication of individual tools to be controlled in an easy and flexible way. In our case we register different kinds of tools (the editor and the debugger) in the JINEXT system. After registration, every program is assigned "groups of interests" , e.g. the editor is listening for the event of entering a new line generated by the debugger. Each tool can also listen for the events from the tools of his own type - in our example the editor listens for the events of starting the process of debugging triggered by other editors.

Since the "gdb" program doesn't have any JINEXT cooperation functions implemented (undoubtedly, most programs don't), we used a concept of proxy to translate the editor's events and to raise events depending on the "gdb" state. In this way a quite universal scheme was designed that can be a basis of running other programs that are not adjusted to work with OMIS/JINEXT interface.

The benefits of using a simple but powerful, uniform cooperation between many tools are evident when it comes to using the programs which are being created independently, especially in large scale distributed environments in a way that enables secure, flexible and scalable communication between them.

This research is partially supported by the KBN grant 4T11C 032 23.

Bibliography:

1. A. Janik: Providing interoperability for distributed Java applications. M.Sc. thesis. AGH, Krakow, (2004)
2. T. Ludwig, R. Wismueller, V. Sunderam, and A. Bode: OMIS -- On-line Monitoring Interface Specification (Version 2.0). Shaker Verlag, Aachen, vol. 9, LRR-TUM Research Report Series,(1997)
3. R. Wismueller, J. Trinitis and T. Ludwig: A Universal Infrastructure for the Run-time Monitoring of Parallel and Distributed Applications. In: Proc. Euro-Par'98, Southampton, UK, September 1998, LNCS 1470 , pp. 173-180, Springer-Verlag, (1998)

15. Dynamic Instrumentation of Java Distributed Applications within the J-OCM Monitoring System

Pawel Swierszcz, Wlodzimierz Funika, Marcin Smetek

Java is a very efficient and easy-to-use platform for distributed applications development. Its features such as system platform independence, dynamic and network oriented architecture, robustness as well as the growing number of common standards make it a language of choice for many projects. However, an increasing complexity of created software and requirements for high stability and high quality of applications make it desirable for a developer to inspect, monitor, debug or in any way alter Java programs' behavior on-the-fly.

The main goal of this paper is to present a design of system for instrumenting Java classes at runtime.

Instrumentation is a process of modifying program's functionality by adding fragments of code at specific locations that implement some new functionality. This allows programmer to enhance classes with logging, monitoring, caching or any other capabilities that are required at the time. Altering an application's behavior happens while it's up and running and is completely transparent – in other words does not modify original features, but just adds some new ones.

The research under discussion extends a distributed Java-oriented monitoring system, J-OCM, which derives from the On-line Monitoring Interface Specification (OMIS) compliant OCM system. J-OCM adds Java awareness and allows for creating a wide range of monitoring tools and interactive access to running Java applications. It is used as a communication and transport framework, provides information about hosts (nodes) and JVMs running the application as well as implements low-level class hot-swapping service using Java's new JVMTI (Java Virtual Machine Tool Interface).

The process of instrumentation performed by the system being described consists of several stages. The first one is choosing which application classes are to be modified and analyzing their contents. A class model is created in form of a SIR tree which is a format for representing applications' code structure. Then comes a design of units of additional functionality called instruments. Each instrument implements some feature (e.g. writes out a current class field value) and is connected to program hooks which are locations in the code where instrument code can be injected. Having prepared a design of instrumentation the developer can move to the third stage and perform actual modifications to class files. New fields are added to the classes, their internal structures are modified and new bytecode instruction sections are introduced to classes' methods – accordingly to previously crafted instruments and their binding to SIR elements. Then original program classes are replaced in JVMs with their instrumented versions. Every object of the instrumented class instantiated from that point on as well as static calls to class methods will have and perform the extended functionality.

The range of use for the described system is not limited to only monitoring or profiling. Instruments can implement almost any kind of service including object state persistence (for example to relational database), caching with external caching system.

This research is partially supported by the KBN grant 4T11C 032 23.

Bibliography:
[OMIS] W. Funika, M. Bubak, M.Smetek, and R. Wismüller. An OMIS-based Approach to Monitoring Distributed Java Applications.
[JOCM] M.Smetek. OMIS-based Monitoring System for Distributed Java Applications. M.Sc. thesis. AGH, Krakow, 2003
[JVMTI] Java Virtual Machine Tool Interface, Sun Microsystems, http://java.sun.com/j2se/1.5.0/docs/guide/jvmti/
[JVMSPEC] The Java Virtual Machine Specification, Sun Microsystems.
[SIRSPEC] C. Seragiotto, Jr., Hong-Linh Truong, B. Mohr, T. Fahringer, M. Gerndt, Standarized Intermediate Representation for Fortran, Java, C and C++

16. Monitoring the Distributedness of Java Applications

Grzegorz Wianecki, Mateusz Wojcik, Wlodzimierz Funika, Marcin Smetek
AGH, Krakow, Poland

The increasing complexity of the distributed systems is causing an increasing demand for tools allowing to analyze the system's behavior and performance. Some aspects of the system cannot be analyzed statically and need to be observed during the system execution by monitoring.

The monitoring of distributed systems needs to take into consideration such issues as identifying and correlating the events and resources from different system's nodes. This brings an additional level of abstraction to the system representation and the monitoring system needs to be specialized for a particular system.

An important requirement for distributed systems development is the portability of the application, not only on the level of source code but also on the binary and platform level. Due to these characteristics Java is becoming popular language used to develop the distributed systems.

J-OCM is a monitoring system, an implementation of J-OMIS specification, allowing to monitor Java applications. J-OCM implements a set of events and services allowing to analyze the application state and execution flow. Tools using J-OMIS specification can be connected to the J-OCM and retrieve the required data.

Up to now the support for distributed applications in J-OCM was limited. The system can monitor several virtual machines and hosts but the data retrieved was treated independently and did not allow to be correlated to form a complete picture of a distributed application.

This paper presents an extension to J-OCM system, designed to monitor the RMI based applications and RMI internal mechanisms with emphasis on Activation. Activation is a mechanism allowing to delay resource allocation for remote objects until they are used for the first time. It also provides a mechanism to free the resources and store the object state until it is used next time.

We have designed additional events and services and implemented them to support the RMI mechanisms. The basic monitoring events are raised from within remote invocations. Other RMI elements monitored are RMI registry, Activation and its subsystems. A special attention was payed to Distributed Garbage Collector. Activation events allow to monitor the registration of objects and groups, activation flow, the objects going inactive and dynamic class loading.

This research is partially supported by the KBN grant 4T11C 032 23.

Bibliography

[1] Marcin Smetek, "OMIS-based Monitoring System for Distributed Java Applications", M.Sc. thesis, AGH, Krakow, 2003
[2] Grzegorz Duda, "Monitoring RMI calls in Java applications", M. Sc. thesis, AGH, Krakow, 2003
[3] The Java Virtual Machine Specification, Sun Microsystems.
http://java.sun.com/docs/books/vmspec/

17. Fault Tolerance and Data Synchronization in Grid Registry for Workflow Applications

Marian Bubak (1, 2), Marek Kasztelnik (1), Cezary Gorka (1), Maciej Malawski (1, 2), Tomasz Gubala (2)
(1) Institute of Computer Science, AGH, Kraków, Poland
(2) ACC CYFRONET AGH, Kraków, Poland

Semantics-based service discovery and application composition is a challenging problem in today and future Grid systems. Grid Registry is Java-based, distributed, scalable, semantics-based and Grid-enabled registry storing information about Grid or Web services. It allows to build tools that automatically find needed services and connects them to the workflow. Consequently, user defines only what he or she can give to the input and what is needed for the output. Afterwards, needed services are found and linked to create a workflow that does the computation. The first version of the Grid Registry was described in [1].

The Registry is a system that does not have centralized point of administration, it uses standard, open, general-purpose protocols and interfaces. This software allows user to add, remove or search information about services. The most basic element of the registry is single node. This term is used to express a computer system that acts like a provider and makes all functionality of the registry available for the user. This solution cannot have any duplicated information stored in registry. That is why when something wrong happens with one or more nodes data stored there is unreachable. Situation like described above was the reason to develop a fault-tolerance mechanism for the registry.

To solve this problem, we designed and developed new version of the Grid Registry [2]. The new version adds fault-tolerance and data synchronization mechanisms, which allow user to use registry even when something wrong has happened with some nodes. This paper presents the concepts of backing up data, query redirection, load balancing and, in case of error, data synchronization. What is more, the performance tests that present comparison between two versions of the registry and show behavior of the new Grid Registry are included.

Acknowledgments.
This research was partially funded by the CoreGRID EU IST Project and by the CoreGRID SPUB-M.

References:

[1] Marian Bubak, Tomasz Gubala, Micha³ Kapalka, Maciej Malawski, Katarzyna Rycerz, Workflow composer and service registry for grid applications, Future Generation Computer Systems, vol. 21, no. 1, 2005, pp. 79-86.
[2] Marek Kasztelnik, Cezary Górka, Marian Bubak (Supervisor). Tools for development of workflow-based grid applications. Master of Science Thesis, AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Institute of Computer Science, Kraków, Poland, August 2005.

18. Using MOCCA Component Environment for Modeling of Gold Clusters

Marian Bubak (1,2), Maciej Malawski (1), Michal Placek (3)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Krakow, Poland
(3) Faculty of Physics and Computer Science AGH, Krakow, Poland

Component-orinted programming is becoming popular paradigm for building distributed environments such as Grid or Peer-to-Peer systems [1]. Distributed component approach is an interesting alternative to service oriented architectures and there is a need to apply it to solve real problems from science, engineering and industry.

In this paper, we show how MOCCA component framework [2] is used for modeling of gold clusters. Clusters of atoms are very interesting forms between isolated atoms or molecules and solid state, and therefore research in this field may be very important for the technology of constructing nanoscale devices. Modeling of clusters invloves several energy minimization methods such as Molecular Dynamics Simulated Annealing (MDSA) or L-BFGS, as well as choosing an empirical potential [3]. These methods are highly compute-intensive, and the optimal result depends on the number of possible iterations and initial configurations for each simulation run.

As a solution to this compute intensive problem, we decided to use the MOCCA distributed component framework [2]. MOCCA implements the Common Component Architecture (CCA) standard [4], which is designed for scientific applications. As a middleware, MOCCA uses the H2O resource sharing platform, which offers mechanisms for running and communicationg components in distributed environment. Following the component-based approach, we decompose our problem into separate components responsible for various simulation methods and additional ones for preparing initial configurations. The components expose interfaces, called ports, which are used for exchanging data (atom coordinations in this case) and also for application control. Various methods can be combined by connecting appropriate components into an application while distribution of components on many machines allows for faster obtaining of results.

This application shows that the distributed component-based approach can lead not only to efficient utilization of distributed resources but that it also facilitates the application programming by offering the possibility of composition from well defined building blocks.

Acknowledgements. This research was partially funded by the CoreGRID EU IST Project and by the CoreGRID SPUB-M.

References
[1] European Research Network on Foundations, Software Infrastructures and Applications for Large Scale Distributed, GRID and Peer-to-Peer Technologies. http://www.coregrid.net/
[2] M. Malawski, D. Kurzyniec, V. Sunderam, MOCCA - Towards a Distributed CCA Framework for Metacomputing, Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Joint Workshop on High-Performance Grid Computing & High-Level Parallel Programming Models - HIPS-HPGC, April 4-8, 2005, Denver, Colorado, USA, IEEE Computer Society Press, 2005,pp. 174a.
[3] N.T. Wilson and R.L. Johnston: Modeling Gold lusters with an Empirical Many-body Potential, Eur. Phys. J. D 12, 161-169 (2000)
[4] CCA forum. The Common Component Architecture (CCA) Forum home page, 2005, http://www.cca-forum.org/.

19. Experiments with Distributed Training of Neural Networks on the Grid

Marian Bubak (1,2), Maciej Malawski (1), El¿bieta Richter-Was (3,4), Grzegorz Sala (3,5),Tadeusz Szymocha (3)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) ACC CYFRONET AGH, Kraków, Poland
(3) Institute of Nuclear Physics, Polish Academy of Sciences, Krakow, Poland
(4) Institute of Physics, Jagiellonian University, Kraków, Poland
(5) Faculty of Physics and Applied Computer Science AGH, Kraków, Poland

Neural networks proved to be efficient and accurate means for data classification and prediction in wide area of applications. Once the suitable network topology is selected, and the network is trained on a representative data sample, then the usage of such network is fast and produces good results. However, the process of neural network training is a highly compute-intensive and time consuming task.

As a solution to this problem, the distribution of the computation on a cluster of machines can lead to significant improvement in decreasing computation time. Other experiences with neural networks show [1], that also Grid environment can be very well suitable for distributed training. What is more, the process of selecting the best structure of the neural network requires many experiments with various parameters, therefore utilizing the resources available on the Grid can make this task less time consuming and more convenient for researcher.

As our target application are will include High Energy Physics, we based our work on such tools as ROOT and Athena and tested them in the Grid environment. We also observed, that training of neural networks on the Grid requires many repeated tasks such as job preparation, submission, monitoring of status and gathering results. Performing them manually is time consuming for the researcher, therefore preparation of tools for automatizing such tasks can facilitate the whole process considerably.

In this paper we describe our experiences with neural network training on the Grid and propose how such process can be facilitated. As a testbed for our experiments we are using infrastructures of LCG and EGEE projects.

[1] O. Ponce, J. Cuevas, A. Fuentes, J. Marco, R. Marco, C. Martínez-Rivero, R. Menéndez, D. Rodríguez, Training of Neural Networks: Interactive Possibilities in a Distributed Framework, Lecture Notes in Computer Science, Volume 2474, Jan 2002, Pages 33 - 40

20. Performance Aspects of Running Thermomechanical Applications in CLUSTERIX Environment

Tomasz Olas, Roman Wyrzykowski
Institute of Computer and Information Sciences, Czestochowa University of Technology, Poland

In this paper we present performance results for the NuscaS package running in the CLUSTERIX grid environment. This environment is a production cluster grid comprising local PC-clusters based on 64-bit Linux machines. Local clusters are placed accross Poland in independent centers connected by the Polish Optical Network PIONIER. NuscaS is an object-oriented package for FEM analysis, designed at the Czestochowa University of Technology to investigate thermomechanical phenomena. In previous works, the performance of NuscaS on clusters has been extensively studied, for various cluster architectures and different programming models.

Because of the hierarchical architecture of the CLUSTERIX infrastructure, it is not a trivial issue to adopt an application for its efficient execution in the CLUSTERIX environment. This requires parallelization on several levels corresponding to the grid architecture, and taking into account heterogeneity in both the computing power of different nodes, and network performance between various subsystems. Another problem is a variable availability of grid components. In the CLUSTERIX project, to run NuscaS jobs across several local clusters, the MPICH-G2 tool based on the Globus Toolkit is used as a grid-enabled implementation of the MPI standard.

In this paper, we describe details of algorithmic and software developments necessary to run NuscaS jobs as distributed meta-applications, using resources of several local clusters managed by the MPICH-G2 environment and CLUSTERIX resource management system. The subsequent part of the paper is devoted to the experimental study of performance of the NuscaS package in the CLUSTERIX environment. The first performance results are rather promising; they confirm the possibility of using such cluster grids for running computation-intensive applications. Based on these results, we have initiated works on modeling the performance of the NuscaS package in grid environments. Such modeling allows for analyzing and predicting the performance of running this complex scientific application on various grid computing platforms.

21. Modern Numerical Codes for Modelling Astrophysical Plasma on Multi-processor Computers

Malgorzata Selwa, Krzysztof Murawski
Institute of Physics, UMCS, Lublin, Poland

In the age when exploration of the Universe results in a number of peculiar physical phenomena their theoretical understanding is vitaly required.
A simple theoretical way to describe these phenomena is with a use of magnetohydrodynamic equations. As a consequence of their intrinsic complexity, most of theoretical models cannot be solved analytically and they require numerical treatment. There are a number of codes which solve these equations numerically. However, only few of these codes are adopted to parallel platforms. We apply the codes FLASH, VAC and WARP3 to simulate oscillations of magnetic structures in the outer layer of the solar atmosphere.
We compare the performance of these codes on the local cluster of the grid CLUSTERIX and test these codes with respect to their speed, simplicity of implementation on the cluster and limitations in the considered models.
Finally, we successfully verify results of the numerical simulations with the observational findings.

22. Grid-based Simulations of MAMMALIAN Visual System

Grzegorz M. Wojcik and Wieslaw A. Kaminski
Division of Complex Systems and Neurodynamics, Maria Curie-Sklodowska University, Lublin, Poland

Large biological neural networks are examined. Ensembles of simulated microcircuits model behaviour of mammalian visual system in some detail. All neural cells are simulated according to Hodgkin-Huxley theory. In that model each neuron is treated as a set of several non-linear differential equations. Good simulation of large groups of Hodgkin-Huxley neurons usually requires high computational powers. The modular structure of visual system seems to be appropriate task for grid computations.

In this paper we report first results of CLUSTERIX grid application to modelling of vision processes. MPGENESIS simulator is used for all simulations. We investigate networks consisting 256 thousands of Hodgkin-Huxley neurons. First simulations were run on the local cluster with 24 nodes. Consequently, in the next stage of experiments, we checked the time of simulation for larger number of processors, using CLUTERIX grid resources.

Such number of simulated neurons allowed us to observe liquid computing phenomena. In theory cortical microcircuits are treated as Liquid State Machines (LSM). The work of each machine resembles behaviour of particles in a liquid. Though, we also present some results confirming the thesis that neural liquids tend to be in different states for different, changing in time stimulations and that such biological structures can have computational power. Separation abilities of the investigated model will be discussed in some details.

23. AAB - Automatic Service Selection Empowered by Knowledge

Lukasz Dutka (2) and Jacek Kitowski (1,2)
(1) Institute of Computer Science,AGH-UST, Cracow, Poland
(2) Academic Computer Centre CYFRONET AGH, Cracow, Poland

Nowadays grid systems become more important for scientific as well as for business world. Unfortunately the complexity of grid systems makes them very difficult in exploitation by non technical staff. In K-WF Grid we try to significantly simplify access to the grid systems making the access way more user friendly. In this project, users simply expresses what they need, what is the problem they need to solve, and system constructs workflow based of available in the grid system services, providing the expected results.

The process of workflow construction and execution is divided in several phases including constructing abstract workflows [1] and converting them into a real ones. The abstract workflow describe only what has to be done in order to lead to a specific solution, but the real workflow refers which exact instance of services have to be called in order to produced some results.

The paper discusses technical issues of automatic conversion an abstract workflow into a real one supported by knowledge. Since, the tool converting the abstract workflows into the real ones (AAB - Automatic Application Builder) is based on Component Expert Architecture (CEA for more see [2]) the process
of service matching is made by rules run by an expert system. In order to make the process more accurate we decided to exploit semantic techniques for description of the current system context - called in CEA as a call environment, as well as we want to derive benefit from existent in K-WF Grid semantic description of the current status of the grid world kept as a knowledge and served by Gird Organization Memory [3]. Moreover, the paper presents some first results of integration AAB with GOM, and processing current context in order to make the automatic services matching required in the phase of conversion abstract workflows into the real ones.

REFERENCES

[1] T.Gubala, M.Bubak, M.Malawski, K.Rycerz: Semantic-based Grid Workflow Composition, 6th Int. Conf. on Parallel Processing and Applied Mathematics PPAM05, Poland
[2] L. Dutka, R. Slota, D. Nikolow, and J. Kitowski. Optimization of Data Access for Grid Environment. In 1st European Across Grids Conference, Lecture Notes in Computer Science, Universidad de Santiago de Compostela, Spain, February, 13-14 2003. Springer
[3] Bartosz Kryza, Marta Majewska, Renata Slota and Jacek Kitowski. Unifying Grid Metadata Representations Through Ontologies, In: Proc. of 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005, R.Wyrzykowski et.al. eds., 2005, To be published. Poznan, Poland

24. Refinement of Case Retrieval for Grid Service Performance Prediction Through Semantic Description of Input Data

Zoltán Balogh, Emil Gatial, Michal Laclavík, Ladislav Hluchý
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

The article deals with performance prediction of Grid services. The performance of Grid services is estimated based on the past cases. Authors have identified that in many applications the description of Grid service input data should be considered as important part of the feature vector. Authors propose the refinement of case retrieval process through semantic description of discrete features and service input data. Instance based learning is used to
estimate the performance of Grid services.

25. Grid Virtual File System – User Centric Approach to Data Management in the Medigrid Project

Marek Ciglan, Branislav Simo, Martin Maliska, Ondrej Habala, Ladislav Hluchy
Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

Data management plays an important role in the distributed systems, such as scientific Grids are. In this paper we describe grid data management system developed for the Medigrid project. The Medigrid project aims to create a distributed framework for multi-risk assessment of natural disasters. The project will integrate in the grid framework models of forest fire behavior and effects, flash floods, erosion and landslides.
In this paper, we focus on two aspects: the design of the Medigrid data layer and the presentation layer for data management services.

First, we present the architecture of the Medigrid data layer and its components. We also describe how we have solved several challenging problems that arise during the project development. One of the essential requirements of the project was the need to support the heterogeneity of computational resources in the Medigrid testbed. Some applications that must be operational in the Medigrid system are bound to the Windows operating systems, others to the Linux platform. This fact has proven to be a significant obstacle, as the standard grid data transfer tool, GridFTP is operational only on Linux based systems. Moreover, we could not use other non-grid transfer tools, as the transfer mechanism must be integrated with grid security infrastructure, to ensure the safety of the data sets.
Another system requirement was to provide mechanism for fine grained security policies definition to prevent unauthorized access to proprietary data.

In the second part of the paper, we describe the presentation layer of the Medigrid data services, which serves as the interface for users' interaction with the system. As the majority of future users of the Medigrid system are not IT experts, the complexness of the distributed grid data management could be an important impediment in the adoption of the system. We have proposed and implemented Grid Virtual File System (VFS) - an extension of Replica Location Service and Metadata Catalog Service. VFS allows creating structures of virtual directories for simplification of the logical organization of the data files distributed across the grid. VFS hides from user most of the data management related operations that must be performed in grid environment (e.g. data replication, manipulations with the catalog services – RLS, metadata catalogs). This concept permits the user to view and manipulate the files in the grid in much the same way he works with the local file system on his workstation. We believe that presenting new and complex system to the users in a way similar to the system they are already familiar with, brings the necessary simplification of the software usage. VFS will be also integrated with the job submission user interfaces, allowing the simplification of the data sets definition for the grid jobs. Finally, we discuss potential impacts and long-term benefits of VFS exploitation in the grid systems.

26. Managing Semantic Metadata in K-Wf Grid with Grid Organizational Memory

Bartosz Kryza (1), Jan Pieczykolan (1), Marian Babik (2), Marta Majewska (3), Renata Slota (3), Ladislav Hluchy (2) and Jacek Kitowski (3)
(1) ACC Cyfronet AGH, Cracow, Poland
(2) Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
(3) Institute of Computer Science, AGH University of Science and Technology, Cracow, Poland

Recent developments in Grid computing and Semantic Web technologies lead to attempts of using these two technologies together. However in order to find a clear use case of how these technologies can actually benefit from each other, more effort is needed in bringing the two communities together. The K-Wf Grid project [1] is one of such efforts, performing research in the area of knowledge management and workflow based Grid applications.

In order to support automatic composition of workflows from services registered in the Grid, one needs powerful metadata formalism in to express both Grid and Application related concepts and a metadata service that can effectively store and publish this information.

In K-Wf Grid project, the metadata are expressed using ontologies defined using Web Ontology Language (OWL). The ontologies are used to define the schema of metadata of every aspect of the Grid, such as application, resources, data or services. The individuals of schema concepts represent particular entities in the Grid environment like files or services.

The metadata service is being developed specifically for this project and is called Grid Organizational Memory (GOM) [2]. GOM is in fact an extensible framework for publishing and querying semantical metadata in the Grid. It provides interfaces for querying and modifying the underlying knowledge base. Its plug-in based architecture allows for easy extensibility of its functionality for instance with new query languages or custom configurations for storing OWL ontologies. The Grid Organizational Memory has distributed architecture [3], which allows to store and provide different kinds of metadata separately (e.g. data and services), using different reasoners and storage back-ends, optimized for a particular kind of metadata.

In this paper, we present an architecture and design of the K-Wf Grid knowledge base component called Grid Organizational Memory, tailored specifically for managing semantic metadata in the Grid environment. A general explanation of how the semantic description of various aspects of the Grid improves the functionality of basic use cases in the Grid is discussed. An evaluation of some use cases and initial performance results obtained with several storage and reasoning configurations are described and conclusions summarize the paper.

Bibliography

[1] http://www.kwfgrid.net/

[2] Krawczyk, K., Slota, R., Majewska, M., Kryza, B., Kitowski, J., Grid Organization Memory for Knowledge Management for Grid Environment, in: Bubak, M., Turala, M., Wiatr, K. (Eds.), Proceedings of Cracow Grid Workshop - CGW'04, December 13-15 2004, ACC-Cyfronet UST, 2005, Kraków, pp. 109-115.

[3] Kryza, B., Majewska, M., Krawczyk, K., Slota, R., Kitowski, J., Unifying Grid Metadata Representation through Ontologies, PPAM Conference, LNCS, Springer Accepted, 2005.