Papers accepted for CGW'04 /in alphabetical order by titles/

1. A novel approach to protect Grids with firewalls

D. Göhr, C. Grimm, N. Jensen, S. Piger, J. Wiebelitz*
Regional Computing Center for Lower Saxony, University of Hanover, Germany
**L3S Research Center, University of Hanover, Germany*

Motivation
The communication requirements of common Grid middlewares, with their extensive demand for unhindered communication, run contrary to the concept of legacy firewalls. The devices are normally statically configured to accept or deny certain packets or communication streams. Advanced firewalls include application-level gateways that forward only packets that fit in the course of events of the related protocol. But in the near future, an implementation of Grid protocols (for example Grid FTP) in application-aware firewalls is unlikely, as these protocols are not commonly used in the Internet and are momentarily of limited commercial interest. To leverage the use of firewalls in the Grid communities, one must take novel approaches. Today, dynamic configuration in terms of a controlled opening of firewall resources is one of the major issues in Grid security.

Previous work
One way is to configure firewalls dynamically, without the necessity to alter the operating system, by means of using proxy system. Grid applications that must open a communication channel to a system which is guarded by one or more firewalls could request the opening of certain ports for the duration of the session from the proxy system. The proxies would configure the firewalls and restore the previous configuration after the end of the session. The advantage is the total transparency to the involved firewalls. The disadvantage is the complexity of the proxy system because of the required extensive knowledge of different configuration protocols of the firewalls.

Another way is the use of an in-path signalisation, which uses a common, authenticated protocol. Applications or communication sources could use the protocol to signal the need for certain communication paths to middleboxes, for example firewalls. The IETF MIDCOM working group has specified mechanisms in RFC 3303 and 3304.

A novel approach
A drawback of the approaches is that one must modify existing components. Better, software would establish an authenticated, temper-proof communication channel between two Grid nodes, and not affect "Grid-unaware" firewalls. The complete Grid communication would take place over this tunnel. Our proposal is to use IPSec AH in conjunction with existing LDAP directories for the Grid X.509 certificates. IPSec can use the certificates to set up an authenticated tunnel to the destination Grid node. Firewalls can accept traffic to the nodes because IPSec implementations are sufficiently secure. Only minor modifications of the rules of the firewall are necessary. Another benefit is that "application-unawareness" reduces load on the firewall.

The paper summarizes the advantages of the approach, which is a feasible way to enhance Grid security.

2. A System for Distributed Computing Based on H2O and JXTA

Pawel Jurczyk(1), Maciej Golenia(1), Maciej Malawski(1,2), Dawid Kurzyniec(2), Marian Bubak(1,3), Vaidy S. Sunderam(2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) Emory University, Atlanta, USA
(3) ACC CYFRONET AGH, Krakow, Poland

H2O is Java-based, component-oriented, lightweight resource sharing platform for metacomputing [1]. It allows deployment of services into container not only by container owners, but by any authorized clients. As a communication mechanism, H2O uses RMIX.

JXTA technology is a set of open protocols that allows any connected device on network to communicate and collaborate in P2P manner[2].

The main goal of this work is to build a uniform global computational network using H2O distributed computing framework and JXTA P2P technology. This computational network will give users new possibilities in building and utilizing of distributed computing systems, namely H2O kernels behind firewalls will be accessible and group management in JXTA will bring us possibility of creating virtual groups of kernels, which enables dynamic ad-hoc created collaborations.

Our current implementation of H2O over JXTA allows user to export kernels having JXTA endpoints. This allows H2O metacomputing applications to seamlessly run across private networks and NATs, by using JXTA as an underlying connection technology. Communication between H2O kernels within JXTA network was made possible by adding a JXTA socket provider to RMIX. JXTA socket factories are used by RMIX to enable remote method invocations in the P2P environment.

At present, we focus our work on discovery of H2O kernels. We plan to create a service that holds information about currently registered kernels from local and JXTA network. Next, we will elaborate a mechanism of measurement of delay times in network from Name Service to H2O kernels.

References
[1] D. Kurzyniec, T. Wrzosek, D. Drzewiecki, and V. S. Sunderam, "Towards Self-Organizing Distributed Computing Frameworks: The H2O Approach", Parallel Processing Letters, vol.13, no.2, pp. 273--290, 2003.
[2] The JXTA Project, http://www.jxta.org/

3. Abstract Workflow Composition in K-WfGrid Project Environment

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

This paper presents a new tool supporting workflow composition for Grid applications [1]. The Workflow Composition Tool (WCT) is developed to tackle with dynamic workflow problem. In order to provide a possibility of runtime workflow rescheduling or optimization, the tool is designed to compose abstract workflows of applications. Therefore we define a new element building abstract workflows: a service class which contains (logically) all the published services implementing a certain interface.

As a service class is just a description of an interface - a piece of functionality provided by any service within the class - the WCT tool is concerned only on functional composition of abstract workflows. Each element of workflow is added after successful matching of the requirements with the (functional) capabilities of a particular service class. The multiplication of the resultant abstract workflows may be caused by many possible service classes conforming to a certain set of requirements.

The main input to the WCT is a description of data (results) which should be produced by the future application. It is also possible to upload an incomplete workflow as an input to complete it. The main output of the composition process is a description of several abstract workflows, each in a distinct document. During its operation the tool extensively uses external service registry which provides descriptions of available service classes (implemented service interfaces). The WCT contacts the registry and queries it in order to obtain interesting descriptions.

This paper is a discussion on the most important issues and problems related to a process of abstract workflow composition in the Grid. It also presents how such a tool may cooperate in a wider environment of workflow construction and execution on an example of European K-WfGrid project [2].

References
[1] M. Bubak, T. Gubała, M. Kapałka, M. Malawski, K. Rycerz, Workflow Composer and Service Registry for Grid Applications. FGCS Journal of Grid Computing, accepted
[2] K-WfGrid: www.kwfgrid.net

4. Access to Grid Services for Mobile Users

Piotr Grabowski, Bartosz Lewandowski, Jarek Nabrzyski
Poznan Supercomputing and Networking Center, Poland

The article examines the problem of giving the Grid users possibility to access their applications and resources, from any place, using mobile devices. According to our approach the devices (mobile phones, PDAs) are incorporated as clients of Grid services. Moreover, because of well-known limitations of mobile devices and "heavy weight" of protocols, standards and technologies used in Grids, our approach assumes adopting a gateway between the client and the Grid. This central point in our architecture is called Mobile Command Center (MCC). The MCC is written as a portal portlet with separate presentation layers for mobiles and standard web browsers - this allows us to reuse portal services. Within the aforementioned model, communication between clients and the gateway is performed using the HTTP protocol (native for Java 2 Micro Edition (J2ME) enabled mobile devices) in the client/server architecture. The communication between the Gateway and Grid Services is also performed in the client/server architecture where the MCC acts as a client, whose requests are served by GSI secured Web Services on the Grid side.
The gateway (which is exactly aware of mobile client limitations that are presented during initial handshake) gathers the requested information from different Grid Services, adopts it to client needs and abilities and sends back to the mobile device. The Grid Services that are accessible from the gateway can be divided into two groups. The first group consists of standard GridLab grid services, which responses have to be translated inside the MCC. This group can be represented by the GridLab Resource Management System (GRMS) which is used for application/simulation steering. The second group consists of services dedicated to use the mobile devices. An example of these is the Visualization Service for Mobiles (VSfM) which can produce visualizations

(scaled down in resolution and color depth) that can be displayed on mobile devices screens. Another service that is accessible from the mobile device via the MCC gateway is Message Box Service(MBS). It is used for storing, sending and managing different kind of messages/notifications for users. It can be also used for sending notifications from different grid applications and registering new visualizations by the GridLab Visualization Service. The newly registered visualization can be than accessed from the mobile device with the use of the MCC and the VSfM.

5. Adapting insecure applications to Grid environment based on the experience of enhancing DataTurbine with the GSI security layer

Pawel Plaszczak (1), Chris Wilk (1) and Matt Miller (2)
(1) Gridwise Technologies
(2) Creare

RBNB DataTurbine from Creare is a dynamic data server, providing high performance data streaming and data stream dispatching. So far, DataTurbine has mostly been used inside firewalls where performance had priority over security.
To cope with the open grid requirements of the NEESgrid project, Gridwise Technologies helped to integrate the GSI security functionality with the DataTurbine. Using the X.509 certificates and the GSI scheme, users can now securely stream data between the servers belonging to their virtual organization (VO). What's more, the new Grid security layer is transparent and the users with properly set certificates can operate their old streaming applications exactly as they did before.
In this paper we will analyze our development and integration work trying to answer the question: how much effort is needed to transition a typical insecure application into a grid-aware one, introducing as little changes as possible to the installations already in operation.

6. Adaptive and Integrated Monitoring System for Knowledge Grid Workflows

Hong-Linh Truong(1), Bartosz Balis(2), Thomas Fahringer(3), Marian Bubak(2,4)
(1) Institute for Software Science, University of Vienna
(2) Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland
(3) Institute for Computer Science, University of Innsbruck, Austria
(4) Academic Computer Centre - CYFRONET, Krakow, Poland

With the dynamics and the diversity of the Grid, Grid monitoring systems have to collect and handle diverse types of data. Moreover, they must be capable of self-management in order to cope with the structure of the Grid which is changed frequently. Despite tremendous effort has spent on developing monitoring systems for the Grid, still existing monitoring systems support limited types of sensors and do not focus on self-management aspects.

We present an adaptive and integrated monitoring framework which is intended to collect and deliver arbitrary monitoring information of a variety of resources and workflow applications. The monitoring system follows the architecture of sensor networks and peer-to-peer systems. Sensors are adaptive and controllable; they will use rules to control the collection and the measurement of monitoring data, and to react appropriate with the change of underlying systems. It will also be possible to enable, disable or change behavior of sensors via an external request. Both event-driven and demand-driven sensors are supported. Sensor managers are organized into super-peer model and monitoring data is stored in distributed locations. Thus, the monitoring system can cope with the dynamics of the Grid, and monitoring data is widely disseminated and highly available.

The monitoring framework unifies diverse types of monitoring data, such as performance measurements of applications, status of hardware infrastructure components, workflow execution statue, etc., in a single system. A generic sensor library supporting adaptive and controllable, event-driven and demand-driven sensors, and an event infrastructure will be developed so that any clients can publish and subscribe for certain events which could occur in monitored entities. Clients will discover the monitoring service which provides the data of interest, and will be able to query and subscribe a variety of types of data by using a uniform interface.

7. Adaptive Matchmaking in Distributed Computing

K. Bertels, K. Sigdel, B. Pour Ebrahimi, S. Vassiliadis
Computer Engineering Laboratory, ITS, TU Delft, The Netherlands

The sharing of resources is a main motivation for constructing distributed systems in which multiple computers are connected by a communication network. The problem in such systems is how resource allocation should be done in the case where some resources are lying idle and could be linked with other overloaded nodes in the network. This assumes some kind of matchmaking that is the process of finding an appropriate provider of

resources for a requester. Given the variety of approaches for matchmaking, it is important to be able to determine the conditions under which particular mechanisms are more performing than others. A framework that defines a set of criteria can be used to assess the usefulness of a particular mechanism. Examples of such criteria are scalability, flexibility, robustness and throughput. Previous research of matchmaking mechanisms shows that systems having completely centralized or completely localized mechanisms each have their deficiencies. On the continuum from centralized to p2p mechanisms we are interested in designing a mechanism that enables the network to change its internal matchmaking mechanism from p2p to a more centralized form or vice versa whenever that is required. This should allow the distributed system to adapt itself dynamically to changing conditions and always have the most appropriate infrastructure available. Evidently, the framework mentioned before will provide the necessary check points to induce such modifications to the infrastructure. This approach boils down to, for instance the idea of multiple matchmakers where the entire system can be partitioned into segments such that every segment has a matchmaker. Agents can then interact with each other in a local neighborhood, and matchmaking in different segments can take place in parallel. Toward this aim, we need to find the conditions under which either approach is best suited. Previous studies on centralized matchmaking show that there exists some kind of population size beyond or below which either no improvement can be generated or the matching efficiency goes down
respectively. So the influence of the population size of each segment should be studied in view of matching rate, matching time and matching quality. Different matching functions that vary in complexity and information used for matching, should be evaluated based on the above mentioned criteria. The current research also takes into account varying circumstances under which the network operates. Events such as highly unequal task and resource distribution, communication failures, etc. are introduced and studied. The current research will do large scale experiments using environments such as PlanetLab and Globus. These platforms build infrastructures to enable extensible, co-operated, and secure resource sharing across multiple domains through the concept of virtual machines.

8. Adaptive Services Grid Towards PlanetLab Research in Poland

Marcin Pilarski
Research and Development Centre of Telekomunikacja Polska, Warsaw, Poland

Goal of Adaptive Services Grid is to develop a prototype of an open development platform for adaptive services registration, discovery, composition, and enactment. ASG aims at creating an open development platform for services composition and enactment rather than another computational grid infrastructure. This paper describes the forthcoming use case scenarios and the motivation for the usage of PlanetLab as a testbed. PlanetLab is a geographically distributed platform for deploying planetary-scale network services. PlanetLab gets access to one or more isolated "slices" of PlanetLab's global resources via a concept called distributed virtualization. The aspects as historical timeline, current status and the background activity of the PlanetLab research will be also presented.

9. Advanced authentication service for complex grid environments

Bartlomiej Balcerek, Tomasz Kowal
Wroclaw Centre for Networking and Supercomputing, Wroclaw University of Technology, Poland

We present a centralized authentication service for Globus based grids, initially developed for SGIGrid project. The approach is to deliver an advanced authentication-related remote methods for grid users and services.

Throught an integrated authentication server, we propose a centralized credential repository and service for Globus' GSI supplied grid environments, as replacement of authentication data dispersion introduced by GSI standard. This approach enables easy management and maintenance of user credentials and gives a possibility to deliver advanced authentication methods for standard and non-standard grid entities, like RAD, VLAB or VUS, which are developed within SGIGrid project. The server is implemented in the Web Service technology and allows to invoke methods such as generation of proxy certificates or verification of user credentials.

References:
1. SGIgrid: Large-scale computing and visualization for virtual laboratory using SGI cluster (in Polish), KBN Project, http://www.wcss.wroc.pl/pb/sgigrid/
2. Tomasz Kowal, Bartlomiej Balcerek, SGIGrid project report: "Authorization and authentication server Authproxyd" (in Polish)

10. An approach to the performance visualization of distributed applications

Wlodzimierz Funika (1), Marcin Koch (1), Allen D. Malony (2), Sameer Shende (2), Marcin Smetek (1), and Roland Wismueller (3)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Department of Computer and Information Science, University of Oregon, Eugene, USA
(3) Fachgruppe BVS - Universitaet Siegen, Siegen, Germany

A key issue in designing and deploying effective distributed applications is the ability to monitor the execution and measure its performance. A standard approach to this problem, based on pre-execution instrumentation and post execution measurement, data analysis and visualization, features huge volumes of monitoring data whose processing is very time-consuming. As a result, there is a distinct need for performance analysis systems that can perform on-the-fly monitoring, and also allow application control and interaction at run-time.

J-OCM is one of the few monitoring systems that provide these features. It can supply performance data from a distributed application to any tool which requests the services provided by J-OCM via J-OMIS, a Java-related extension to the OMIS interface specification. Such an approach allows J-OCM to co-operate with many tools and evolve separately, without loss of compatibility. Up to now there were no complete performance monitoring tools which used J-OCM for performance analysis. As a result, it has been impossible to make a full use of on-line application monitoring for performance measurement and analysis. On the other hand, there exist very powerful tools for computational steering, such as SCIRun, that apply advanced visualization of complex data in real-time.

The paper presents an approach to visualize performance data provided by J-OCM. We use the TAU performance system with the Paravis performance visualization package built with the SCIRun environment. TAU has mainly been used on single parallel systems for both post-execution and online analysis. Here we aim at making TAU/Paravis collaborate with J-OCM. TAU/Paravis package consists of several modules which are used to build a complete visualizing tool. Such a component approach makes the whole system very flexible and easy to extend. We use this ability to extend the package.

We implemented additional modules placed between J-OCM and TAU/Paravis. These modules form a separate SCIRun package. Because TAU and J-OCM are data-incompatible, modules should provide necessary data conversion. Additionally, they allow the developer to choose which elements of application are to be monitored. The scope of performance visualization may comprise selected nodes of the distributed system, JVMs, threads, classes and methods. There are modules to gather different kinds of performance data, like threads synchronization, methods execution time, dependencies in method calls, and communication (Remote Method Invocation - RMI). Data can be summarized over a time interval or can be momentary.

At the end, we present an expansion into the monitoring of distributed Java applications based on web services. Recently, web services are becoming a more and more popular paradigm of distributed programming and this is why the ability to measure its performance is so important. We intend to check the adaptability of J-OCM based monitoring tools to the environments in which web services are provided.

References:

1. W. Funika, M. Bubak, M.Smętek, and R. Wismüller. 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. A. D. Malony, S. Shende, and R. Bell, "Online Performance Observation of Large-Scale Parallel Applications", Proc. Parco 2003 Symposium, Elsevier B.V., Sept. 2003.
3. C. Johnson, S. Parker, "The SCIRun Parallel Scientific Computing Problem Solving Environment" Proc. Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999.

11. Analysis of Implementation of Next Generation Grids Visions in FP6 Grid Technology Projects

Marian Bubak (1,2), Piotr Nowakowski (2)
(1) Institute of Computer Science AGH, Krakow, Poland
(2) Academic Computer Center CYFRONET AGH, Krakow, Poland

A group of experts in the area of Grid technologies and their applications has presented its visions of directions of research in this field in two previous reports [1,2]. These reports, starting with an analysis of requirements and usage scenarios, present a kind of research agenda and the list of the most important topics from three points of view: end-user, architectural, and software.

Following the slew of Grid projects implemented as part of the IST priority within the 5th Framework Program,

the EC has decided to carry on with Grid-related research work as part of the 6th FP, starting in September 2004 [3]. These Grid research projects, 12 in all, can broadly be divided into the following groups:
- underpinning research and Grid architectures,
- applications driving business growth and scientific innovations,
- new technologies for problem solving,
- building the European research area.

The paper aims at presenting the emerging trends in European Grid research. An analysis of the adherence of the objectives declared by these new projects to the Grid Expert Group recommendations and proposed research agenda is presented.

References
[1] Next Generation Grid(s). European Grid Research 2005-2010. Expert Group Report, June 2003,
[2] Next Generation Grids 2. Reqirements and Options for European Grids Research 2005-2010. Expert Group Report, July 2004, http://www.cordis.lu/ist/grids/
[3] European Grid Technology Days 2004, http://www.nextgrid.org/events/
http://www.cordis.lu/ist/grids/

12. Building grid services using UNICORE middleware

J. Jurkiewicz (1), P. Bała (1,2)
(1) ICM, Warsaw University, Warsaw, Poland
(2) Faculty of Mathematics and Computer Science, N. Copernicus University, Torun, Poland

Grid middleware is designed to provide access to remote high performance resources. The most important paradigm is seamless access to the resources. The user obtains number of tools ranging from the set of scripts to the graphical application launched on the client workstation. The second approach motivated development of the UNICORE grid middleware. In result user obtains advanced UNICORE Client which allows user to access distributed resources.

The significant advantage of the UNICORE client is flexible graphical user interface which, through plugin technology, allows for easy development of the application specific interfaces. The CPMD, Gaussian, Amber, Gamess or DataBase Access plugins are good examples.

Unfortunately, this technology cannot be used in the Web Services environment and adaptation of the application specific interfaces, if not done properly, will be a time consuming task. In order to overcome these disadvantages, we have developed web services tools to access UNICORE grid. The main idea of the presented approach is to move application specific interface (plugin) from the UNICORE Client to web services environment. The solution is based on the portal hosting environment which allows to access resources, similarly as it has been performed by the UNICORE Client.
The user obtains similar functionality, however access to the distributed resources is easier and allows to hide unnecessary details. Presented approach brings grid services functionality to the UNICORE middleware.

At the moment, the main communication protocol is based on the existing components, in particular Unicore Protocol Layer (UPL) adopting Abstract Job Object paradigm.
In the future, the standard web services protocols (SOAP based) will be used, which will provide high interoperability with other grid services based solutions such as Globus.

This work is supported by European Commission under IST UniGrids project (No. 004279). The financial support of State Committee for Scientific Research is also acknowledged.

13. Capability and Attribute Based GRID Monitoring Architecture

Jiri Sitera, Ludek Matyska, Ales Krenek, Miroslav Ruda, Michal Vocu, Zdenek Salvet, Milos Mulac
CESNET z.s.p.o., Praha 6, Czech Republic

The Grid Monitoring Architecture (GMA), as proposed by GGF, is an architecture specification for general Grid monitoring infrastructure. While designed as a basis for interoperable systems, most actual implementations do not allow inclusion of third party infrastructure elements. We propose a GMA extension, adding support for (virtual) overlay infrastructures, through explicit meta-description of data requirements and infrastructure element capabilities.

The extension consist of three main parts: data attributes, infrastructure components capabilities and match-making process. The key part of our proposal is the idea that all data (events) should have associated attributes, i.e., metadata description of required behavior of the monitoring infrastructure when processing this data. The metadata description is thus extended to describe not only data types and structures but also

constraints for data handling and usage. The data attributes are complemented by component capabilities. They describe infrastructure element features and specific behavior for data processing.
The extended GMA also defines match-making process which virtually connects data with components while taking into account both the expressed data requirements and declared component capabilities. This match-making functionality is a part of generalized directory service. The data are thus guaranteed to be seen and manipulated only by such parts of the infrastructure, that are capable of appropriate processing or that can guarantee trust, persistence or other requirements.

Major advantage of the proposed extension, called capability based GMA(CGMA), is that components with different capabilities and features can coexist and provide their services relatively independently, but under one unifying framework (which can be seen as a meta-infrastructure). The goal is to provide a way how to incorporate and use specialized and optimized components rather than look for ways how to create general enough components fitting all, often contradictory, needs.

In CGMA, data sources are not only describing types of data they are dealing with but also conditions under which the data could be passed to the infrastructure. The producers adopt data attributes and register them together with their own capabilities while consumers register their capabilities as part of their search for appropriate data sources. The directory (its match-making part) is using attributes and capabilities to find appropriate components for proper consumer/producer interaction. As more complex infrastructure elements, with both consumer and producer capabilities, are added, the same process is able to create a path for ``event flow'' through the infrastructure. This way, specific overlays on top of the monitoring infrastructure are created and maintained.

Proposal is based on experience gained during the development of Logging and Bookkeeping (LB) service for the EU DataGrid and recently EGEE projects. We will show motivation and key advantages of the extended monitoring architecture on real LB use cases, explaining how it can solve the deficiencies of current GMA implementations like R-GMA.

The paper will be concluded with a proposal for capability based GMA prototype. The proposal builds on the R-GMA relational model and extends its mediator with the match-making capabilities. This implementation will be shown to cover all the LB use cases, while leaving space for co-existence with other GMA implementations, including the R-GMA itself.

14. Checkpoint/Restart mechanism for multiprocess applications implemented under SGI Grid Project

Radosław Januszewski, Gracjan Jankowski, Rafał Mikolajczak
Poznan Supercomputing and Networking Center, Poland

One of the most required tools allowing to obtain the high availability and fault tolerance level for Grid computing in the HPC and HTC areas is the checkpoint - restart and migration mechanism. This mechanism can be used when there is a need to turn off the production system for maintenance purposes or when the system fails. The next very important issue is the ability for dynamic load balancing between computational nodes. To achieve process migration in distributed systems you need to put the migration aware checkpointing mechanism on the nodes. There are a few system level checkpoint/ restart tools which are available for commercial operating systems (eg. for IRIX and UNICOS), moreover there are some packages for 32-bit Linux system as well.
The paper describes the checkpoint restart package which was developed as a part of the SGIgrid project ("High Performance Computing and Visualization with the SGI Grid for Virtual Laboratory Applications" project nr 6 T11 0052 2002 C/05836). The project was co-funded by the State Committee for Scientific Research and SGI.
We present the architecture of the tools which were developed on the SGI Altix 3300 server system with four Intel Itanium2 processors working under Linux OS based on kernel version 2.4. Our package provides kernel level checkpointing. It consists of a set of tools and kernel modules that allow saving the state of the job and restarting it later.
Additionally we describe in detail solutions for some interesting issues that we encountered during this project e.g. the virtualization mechanism which ensures coherency of the recovered application, surrounding environment and resources. The solutions will be considered in two aspects: as a general checkpoint/restart problem and implementation problems pertaining to Itanium limitations and features. The document will cover some portability subject matter in context of virtualization and checkpointing mechanisms.
There is available to download the checkpointing package designed for kernel version 2.4.20-sgi220r3. The package has implemented the following functionality:
* provides support for single and multi processes applications
* provides support for 32 and 64 bit applications
* provides support for interprocess communication
* virtualization mechanisms are solving migration issues

Key Words: Grids, checkpointing, system architecture, resource virtualization.

15. Choosing and integrating various Grid resource managers with GT4 - the early experience

Pawel Plaszczak and Dominik Lupinski
Gridwise Technologies

This presentation will serve as an introduction to one of our research projects related to grid technologies available on todays market and also those that are about to be ready. The paper documents the early experiences in integrating Globus Toolkit 4 and state of the art industry schedulers.

In the long term, our research will give us an insight in the heterogeneous Grid solution for computationally intensive industrial applications. As far as the date of the conference is concerned we are going to present our results as a work in progress, though.

First, we are going to portray our environment used in conducting this research. The set of virtualized computers have been achieved with the "User Mode Linux" software.

Next, we will depict the operation of compute clusters functioning under various resource management schemes. Our research covers scheduling systems such as PBS/Torque and SUN N1 Grid Engine.

Next, we put emphasis on the effort necessary for integration of those schedulers. The beta version of the Globus Toolkit 4 will be evaluated as the solution, with the stress of the latest GRAM version, also compared to the alternative solution from Globus 2.4.

As the last integration step, a central installation of the Community Scheduler Framework (CSF) will be evaluated as a final layer interacting with end-users of this set of VO's resources.

16. CLUSTERIX Data Management System

Lukasz Kuczynski, Konrad Karczewski, Roman Wyrzykowski
Czestochowa University of Technology, Poland

Nowadays grid applications deal with large volumes of data. This creates the need for effective data-management solutions. For the CLUSTERIX project CDMS (Clusterix Data Management System) is being developed. User requirements and analysis of existing implementations have been the foundation for development of CDMS. A special attention has been paid to making the system user-friendly and efficient, allowing for creation of a robust Data Storage System.

Taking into account grid specific networking conditions - different bandwidth, current load and network technologies between geographically distant sites, CDMS tries to optimize data throughput via replication and replica selection techniques. Another key feature to be considered during grid service implementation is fault-tolerance. In CDMS, modular design and distributed operation model assure elimination of a single point of failure. In particular, multiple instances of Data Broker are running simultaneously and their coherence is assured by a synchronization subsystem.

17. CRO-GRID Infrastructure: Project Overview and Perspectives

E. Imamagic, B. Radic, D. Dobrenic
University Computing Centre (Srce), University of Zagreb, Croatia

We have witnessed tremendous success of production grids and various grid applications in last decade. Motivated by previous and current grid initiatives and raising science community demands for computing resources, project CRO-GRID was initiated. CRO-GRID is a national multi-project whose goals are introduction of grid and cluster technologies to science community and industry and development of distributed computing environment and grid applications.

CRO-GRID multi-project consists of three projects: CRO-GRID Infrastructure, CRO-GRID Mediator and CRO-GRID Applications. CRO-GRID Mediator develops distributed middleware system based on latest Web Service Resource specifications.
CRO-GRID Applications is responsible for design and implementation of applications by using CRO-GRID Mediator system and existing grid middleware, programming environments and tools.

CRO-GRID Infrastructure is responsible for building underlying cluster and grid infrastructure for two other projects to use. Furthermore, CRO-GRID Infrastructure will assure infrastructure maintenance and optimization and provide support to grid and cluster related projects. Status of project CRO-GRID Infrastructure is described below.

In first six months thorough investigation of existing cluster technologies has been completed. Investigation was focused on particular cluster subsystems (such as job management systems, monitoring tools, etc.) and followed by selection of the most appropriate cluster technologies. Selected cluster technologies were implemented on five clusters placed on scientific centers and universities in four cities. In coordination with project Giga CARNet, clusters were connected with gigabit links.

Currently we are analyzing contemporary grid technologies and projects. We are planning to finish grid technologies evaluation and decide which will be used by the end of this year. In the same time we will implement basic grid services on existing clusters.
Afterwards we will start upgrading grid functionalities based on specific needs of project CRO-GRID applications. In the following two years, we will focus our work on grid and cluster maintenance and optimization, community outreach and linking with international initiatives.

18. CrossGrid and EGEE Installations at Cyfronet: Present and Future

Patryk Lason, Andrzej Oziebo, Marcin Radecki, Tomasz Szepieniec
Academic Computer Centre - CYFRONET AGH Kraków, Poland

In keeping with the trend toward the grid technology, Cyfronet is actively involved in recent EU Grid projects. In these projects the centre undertakes a variety of activities which include providing testbed for software developers, maintenance of computing resources and support for partners across Europe. Moreover, many significant responsibilities lie on the Cyfronet's teams e.g. to guarantee appropriate level of services, ensure proper operation of computing infrastructure in the region and certify new resources as they are deployed.
To fulfil the mandates a considerable hardware infrastructure have been established, including 40 dual processor Intel Xeon based nodes (IA32), and also 20 dual Itanium-2 based nodes. Currently, the IA32 processors are used by the CrossGrid and LCG projects while the IA64 part is being utilized for initial installation for the EGEE project. In the future we plan to merge all the computing resources into one resource pool to be used by a range of applications.
During last years we have gained much experience which allows us to play a key role in propagation of the Grid technology in our region. The plans are to improve comprehension among scientist of the advantages and possible profits that can be gained from the grid technology.

19. Data archivization aspects with SGI Grid

Renata Słota(1), Łukasz Skitał(2), Darin Nikolow(1), Jacek Kitowski(1,2)
(1) Institute of Computer Science, AGH-UST
(2) Academic Computer Center CYFRONET AGH

The SGI Grid project[1] aims to design and implements broadband services for remote access to expensive laboratory equipment, backup computational center and remote data-visualization service. Virtual Laboratory (VLab) experiments and visualization tasks very often produce huge amount of data, which have to be stored or archived. Data archivization aspects are addressed by Virtual Storage System (VSS). Its main goal is to integrate storage resources distributed among computational centers into common system and to provide storage service for all SGI Grid applications. VSS can use variety of storage resources, like databases, file systems, tertiary storage (tape libraries and optical jukeboxes managed by Hierarchical Storage Management - HSM systems).

In the paper we present data archivization functionalities of the VSS, its architecture and user interfaces.

VSS offers standard operations, which let user to store and retrieve datafile organized in meta-directories. Besides this standard operations, some archivization specific functionalities are implemented: access time estimation, file fragment access, file ordering and fast access to large files residing on tapes. Access time depends on storage resource type and state. It can vary from milliseconds to minutes or even ten of minutes for data archived on tape libraries. VSS offers Access Time Estimation for HSM systems, which provide approximate access time to given file. User can order file or file fragment, which means to inform VSS, when he will need to access the file. If file resides on tapes or other slow media, the system will copy it to cache, what will minimize access time to the file. Access to large files residing on tapes is accelerated by incorporating file fragmentation. Automatic replication has been implemented as a method of data access optimization. Implementation details of VSS have been presented in [2].

VSS is equipped with following user interfaces: Web portal, Text console, Java API, VSSCommander (java api

application). Web portal is VSS interface, which hides all system details and provide easy way of using VSS. Text console is an interfaces for VSS developers, it provides more details then others interfaces. Text console is not intended to by used by end users. Java API is programmers interface, which allows easy VSS client applications development. VSSCommander is a java application written using VSS Java API. VSSCommannder offers access to VSS in manner similar to WinSCP or Windows Commander.

The VSS as a part of SGI Gird is already in deployment phase and is used as a production system, which functional and performance tests are also presented.

[1] - SGI Grid - http://www.wcss.wroc.pl/pb/sgigrid/en/index.php
[2] - D. Nikolow, R. Słota, J. Kitowski, Ł. Skitał, "Virtual Storage System for the Grid Environment", 4th Int. Conf. on Computational Science, Kraków, June 6-9, 2004, LNCS vol.3036, pp. 458-461. http://www.cyfronet.krakow.pl/iccs2004/

20. Data Management in the FloodGrid Application

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

We would like to present the data management tasks and tools used in a flood prediction application of the CROSSGRID project. The project is aimed towards improving support for interactive applications in the Grid. Apart from providing several tools for development and support of Grid applications, it also contains a testbed with several testing applications. One of these is the FloodGrid application.
The grid prediction application of CROSSGRID, called FloodGrid aims to connect together several potential actors interested in flood prediction - data providers, infrastructure providers and users. The application consists of a computational core, a workflow manager, two user interfaces and a data management suite. The project is based on the Grid technology, especially the Globus toolkit 2.4 and 3.2 and the EU DataGrid project. architecture and used technology.
The application's core is a cascade of several simulation models. At the beginning of the cascade is a meteorological prediction model, which receives its boundary conditions from outside of the cascade and computes temperature and precipitation predictions. These are then used in the second stage of the cascade - a hydrological model, which computes flow volume in selected points of the target river basin. This is then processed in the last stage of the cascade, in a hydraulic model. This model uses actual terrain model to compute water flow in the area. Where the water hits area outside of the river basin, a flood is expected.
The computational core is interfaced to the environment and to FloodGrid users by several other components - the FloodGrid portal, which provides user interface to the whole application, workflow service for the control of the cascade and a data management suite.
The data management software of FloodGrid is equipped with facilities that support transport and storage of input data for the simulation cascade, cataloguing of all available files in the environment and easy access to them. It also provides radar imagery and data from measurement stations directly to the FloodGrid portal. It builds on the software available in the CROSSGRID testbed, which is interconnected with the portal and augmented with a metadata catalog for easy lookup of needed files. The whole suite consists of this metadata catalog (in the form of an OGSI grid service), data delivery software and EDG replica management (part of the testbed). All these parts are connected with the user interface of FloodGrid (the portal) and may be used from there, as well as automatically by the workflow service and simulation jobs. The metadata catalog is divided into two parts - a reusable grid service interface and an application-specific relational database. The interface may be used with any application, provided that the underlying database follows provides several tables with information about its structure in a form expected by the interface. The interface itself is accessible via several service calls for metadata management. Each metadata item describes a file (identified by its GUID from the EDG replica manager). New metadata items may be added, existing items may be removed and guid lookup is performed by specifying a set of constraints.

21. DIGITAL SCIENCE LIBRARY FOR NUCLEAR MAGNETIC RESONANCE SPECTROSCOPY

Przemysław Rek(1), Mirosław Kopec(1), Zofia Gdaniec(2), Łukasz Popenda(2), Ryszard W. Adamiak(2), Marcin Wolski(3), Marcin Lawenda(3), Norbert Meyer(3), Maciej Stroiński(3)
(1) Computer Centre of Technical University of Łódź, Łódź
(2) Institute of Bioorganic Chemistry PAS, Poznań
(3) Poznań Supercomputing and Networking Center, Poznań

The need (and possibilities) for electronic publishing and presenting of the digital content grew along with the development of the Internet network. Nowadays users have the possibility of using more and more sophisticated tools designed to create, browse and search for electronic documents. These tools perform an important role in the global information infrastructure, and can benefit to the education and scientific environments.

The Digital Science Library (DSL) is created on the base of the Data Management System (DMS), which was developed for the PROGRESS project. Its main functionality, which is storing and presenting data in grid environments, was extended with the functions specific to the require-ments of the Virtual Laboratory. In its concept the Digital Science Library allow to store and pre-sent data used by the virtual laboratories. This data can be input information used for scientific experiments as well as the results of performed experiments. Another important aspect is the capa-bility of storing various types of publications and documents, which are often created in the scien-tific process. This type of functionality, which is well known to all digital library users, is also pro-vided by the DSL. One of the main examples of using DSL in practice is the Virtual Laboratory of NMR.
This work presents the general assumptions, project design and implementation of the Digital Science Library for the purpose of Nuclear Magnetic Resonance Spectroscopy (NMR) data. NMR spectroscopy is a unique experimental technique that is widely used in physics, organic and inor-ganic chemistry, biochemistry as well as in medicine.
The analysis of one- or multidimensional homo- and heteronuclear spectra obtained in the course of NMR experiment can provide information about the chemical shifts of the nuclei, scalar coupling constants, residual dipolar coupling constants and the relaxation times T1, T2. All of these data can be stored in the presented database, which also offers tools for performing quick and op-timal search through the repository.
Compared to the other NMR databases available through the Internet, like BioMagResBank, NMR data-sets bank, NMRShiftDB, SDBS and Spectra Online, DSL is more suitable for teaching, since it contains an entire range of the information about the performance and analysis of the NMR experiment.

22. Discrete Particle Simulations using shared memory clusters

W.Dzwinel and K. Boryczko
Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland

We discuss the use of current shared-memory systems for discrete-particle modeling of heterogeneous mesoscopic complex fluids in irregular geometries. This has been demonstrated by way of mesoscopic blood flow in various capillary vessels.
The plasma is represented by fluid particles. The fluid particle model (FPM) is a discrete-particle method, which is a developed, mesoscopic version of molecular dynamics (MD) technique. Unlike in MD, where the particles represent atoms and molecules, in FPM fluid particles are employed. The fluid particles mimic the "lumps of fluid", which interact with each other, not only via central conservative forces as it is in MD, but with non-central, dissipative and stochastic forces as well. Because of three times greater communication load per FPM particle than per MD atom, the reconfiguration of the system becomes very time consuming. The other blood constituents are made of "solid" particles interacting with harmonic forces. The tests were performed for the same system employing two million fluid and "solid" particles. We show that irregular boundary conditions and heterogeneity of the particle fluid inhibit efficient implementation of the model on superscalar processors. We improve the efficiency almost threefold by reducing the effect of computational imbalance using simple load-balancing scheme.
Additionally, in employing MPI on shared memory machines, we have constructed a simple middleware library to simplify parallelization. The efficiency of the particle code depends critically on the memory latency. As an example of application of small, shared-memory clusters and GRID environment in solving very complex problems we demonstrate the results of modeling red blood cells clotting in blood flow in capillary vessels due to fibrin aggregation.

23. Distributed evolutionary algorithm plugin for UNICORE system

Waclaw Kus, Tadeusz Burczynski
Department for Strength of Materials and Computational Mechanics, Silesian University of Technology, Poland

The shape optimization problem of structures can be solved using methods based on sensitivity analysis information or non-gradient methods based on genetic algorithms[5]. Applications of evolutionary algorithms in optimization need only information about values of an objective (fitness) function. The fitness function is calculated for each chromosome in each generation by solving the boundary - value problem by means of the finite element method (FEM)[3,8] or the boundary element method (BEM)[3]. This approach does not need information about the gradient of the fitness function and gives the great probability of finding the global optimum. The main drawback of this approach is the long time of calculations. The applications of the distributed evolutionary algorithms can shorten the time of calculations[1,6].

The use of grid techniques in optimizations can lead to improvements in hardware and software utilization. The other advantages of grids are simple and uniform end user communication portals/programs.
The first evolutionary optimization tests [4] were performed using Condor package[2]. The idea presented in the paper is to prepare group of plugins and programs for evolutionary optimization of structures using UNICORE environment[7]. The distributed evolutionary algorithm plugin for UNICORE environment is

presented. The plugin works with evolutionary software prepared in our department for linux platform. Some numerical test of optimization of structures are presented in the paper.

[1] T. Burczynski, W. Kus, Optimization of structures using distributed and parallel evolutionary algorithms Parallel Processing and Applied Mathematics, PPAM2003, Revised papers, Lecture Notes on Computational Sciences 3019, Springer, pp. 572-579, 2004.
[2] Condor, High Throughput Computing, http://www.cs.wisc.edu/condor/
[3] M. Kleiber (red.), Handbook of Computational Solid Mechanics, Springer- Verlag, 1998.
[4] W. Kus, T. Burczynski Computer implementation of the coevolutionary algorithm with Condor scheduler, KAEIOG 2004, Kazimierz, 2004.
[5] Z. Michalewicz, Genetic algorithms + data structures =evolutionary algorithms. Springer-Verlag, Berlin, 1996.
[6] R. Tanese, Distributed Genetic Algorithms. Proc. 3rd ICGA, pp.434-439, Ed. J.D. Schaffer. San Mateo, USA, 1989.
[7] UNICORE Plus Final Report - Uniform Interface to Computing Resources, Joint Project Report for the BMBF Project UNICORE Plus, Grant Number: 01 IR 001 A-D, 2003.
[8] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method. The Basis, Vol. 1-2, Butterworth, Oxford, 2000.

24. Distributed visualization system based on web-services

Witold Alda(1), Bartłomiej Balcerek(2), Maciej Dyczkowski(2), Grzegorz Kowaliczek(1), Stanisław Polak(1), Rafał Wcisło(1), Adrian Zachara(1)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Wroclaw Centre for Networking and Supercomputing, Wroclaw, Poland

In the paper we present the architecture and implementation of the general-purpose distributed visualization system based on web-services tools. The data files, which contain information for 3D visualization can reach enormous size in current computer simulations and experiments. Practically they can only be gathered and stored on the remote servers with large storage devices.
System presented in the paper can interactively convert files stored on remote servers into common X3D format, possibly extracting only parts which are temporarily needed, thus reducing the number of bytes transferred to client's workstation. In the implementation layer we use contemporary Java-based tools, such as J2EE, EJB (Enterprise Java Beans) and Jboss application server on the remote side, as well as Java3D graphic library for local (client) rendering. The architecture of the system utilizes design patterns, component technology and XML notation in order to achieve clarity and flexibility of the project. The current version the system is supplied with sample readers/converters of vastly used PDB files - for storing molecular information of proteins - and DEM files - suitable for keeping elevation data of digital 3D geographical maps. However, due to modular architecture of the system, it is fairly easy to add new readers.

25. Dynamic Grid Scenarios with Grid REsource Management System in action

Krzysztof Kurowski, Bogdan Ludwiczak, Jarek Nabrzyski, Ariel Oleksiak, Juliusz Pukacki
Poznan Supercomputing and Networking Center, Poland

GridLab Resource Management System (GRMS) is an open source meta-scheduling system, developed under the GridLab Project. It provides mechanisms for resource management for Grid environment: job submission, job migration, job control, application checkpointing, gathering information about submitted jobs, etc.
GRMS operates on behalf of a user using his or her credentials for authentication and authorization. In cooperation with other Grid middleware services and using low level core services GRMS is trying to deal with dynamic nature of Grid environment.
GRMS has some basic mechanisms which allow to implement more complex scenarios for job management. Basic pieces of job management are: job submission and job migration witch application level checkpointing. Based on that and with strong assistance of dynamic resource discovery mechanism implemented within GRMS it is possible to provide more advanced and dynamic scenarios based on job rescheduling methods. The reschedule policy checkpoints and migrates already running jobs in order to release the amount of resources required by a job pending in the GRMS queue. The rescheduling method that is used, consists of several steps. At first resources that meet static requirements have to be found. If there are such resources available, list of jobs that are running on that resources and were submitted by GRMS is created. Then the system tries to determine the migration of which jobs can bring the required result. This step consists of two actions. First, GRMS searches for jobs after termination of which the pending job could be started immediately. Second, selected jobs are analyzed again to check which of them can be migrated to other available resources, taking into account the requirements of these jobs and the resources available at the moment.
In the next step the best job for migration is chosen by evaluation of available machines and jobs found for migration. Selected job is then checkpointed and migrated to the new location and original job is submitted to the machine on which migrated job was running.
Presented rescheduling technique is very useful and brigs much better results comparing to other possible

mechanisms - keeping job in a queue, or submitting job to overloaded machine. We will also compare this method to a popular backfilling mechanism.

26. Experiences with the Globus Toolkit on AIX and deploying the Large Scale Air Pollution Modelling application as a grid service

Ashish Thandavan, Cihan Sahin, Vassil N Alexandrov
ACET Centre, University of Reading, UK

The Globus Toolkit is the one of the most popular software toolkits available today for building grids that let people share their computing, storage and other resources securely online. It has evolved significantly from its earliest version in 1998 to the version based on the open-standard Grid services to the version currently under development based on WSRF.

In this paper, we outline our experiences with using the Globus Toolkit on our IBM SP2 and IBM pSeries cluster, both running AIX and LoadLeveler. As a member of the UK's EPSRC-sponsored OGSA Testbed project, we worked on installing and testing OGSA based middleware, Globus Toolkit v3 on our resources and investigated the deployment of the Large Scale Air Pollution Modelling application as a grid service.

27. GRID applications control based on synchronizers

D. Kopański, J. Borkowski, M. Tudruj
Polish-Japanese Institute of Information Technology (PJWSTK), Warsaw, Poland

The paper presents how program execution control based on global application states can be extended to coordinate parallel applications executed on a Grid. Special control processes called synchronizers are introduced into applications. Processes report their execution states to synchronizers, which construct consistent application states based on timestamps accompanying state messages. Synchronizers compute control predicates on consistent states, and upon them, take control decisions concerning application behavior. The decisions are transformed into control signals. Synchronizers dispatch control signals to processes causing there the desired reactions in program execution control. The signals arrived can activate an associated code in the form of "distributed interrupts", or they can cease execution of the current computational task to make room for a new one. These ideas have been included into the parallel program graphical design system PS-GRADE (a modified version of the P-GRADE system [KDL01]) for programs executed in clusters [TKB04].
This paper extends the described ideas towards the Grid level. There are two types of synchronizers in the Grid level system: application-level synchronizers and Grid-level synchronizers. Application level synchronizers know local application states and they can report them to the Grid-level synchronizers by sending state reports over Grid communication network. The programmer decides which application states should be passed to the Grid-level synchronizers. Application states are assembled at the Grid level to form Grid level consistent states, using similar rules as at the application level. Control signals generated by the Grid-level synchronizers are passed down to application level synchronizers, which send proper control signals to its application processes. The programmer must specify reaction details. Program execution control based on Grid-level synchronizers extends the workflow concept [LO04] by providing additional program execution control. In the workflow paradigm, application execution control is based on standard macro data flow strategy. Our aim is to be able to specify conditions, which could globally control the behavior of applications distributed on a Grid and launch adequate control actions. We use Grid services for sending state reports (receiving signals) by application level synchronizers to (from) Grid-level synchronizers located outside local clusters. Synchronizers are integrated with dedicated Grid services based on Open Grid Services Infrastructure. The Globus Toolkit v3 (GT3) is used to implement the web services infrastructure. The SOAP protocol is used for signal and state message delivery between GT3 sites. GridFTP is used for transferring program code and data files to requested sites.
Application program management is provided by the Globus GRAM services. Grid services and synchronizers communicate based on TCP sockets. Implementation of the described concepts based on the PS-GRADE system is under way.

References
[KDL01] P. Kacsuk, G. Dózsa, R. Lovas, The GRADE Graphical Parallel Programming Environment, in Parallel Program Development for Cluster Computing: Methodology, Tools and Integrated Environments, Nova Science Publishers, 2001, pp. 231-247.
[LO04] R. Lovas et al. Workflow Support for Complex Grid Applications: Integrated and Portal Solutions, 2nd European Across Grids Conference, Nicosia, Cyprus, 2004.
[TBK04] M. Tudruj, J. Borkowski, D. Kopanski, Graphical Design of Parallel Programs with Control Based on

Global Application States Using an Extended P-GRADE System, in Distributed and Parallel Systems: Cluster and GRID Computing, Kluver Int. Series in Eng. and Comp. Sc., Vol. 777, 2004, pp. 113-120.

28. Grid approach to heat transfer simulation in atomistic-continuum model

Witold Alda(1), Michał Białoskórski(2), Remigiusz Górecki(2) and Jarosław Rybicki(2)
(1)Institute of Computer Science, AGH, Cracow, Poland
(2)Technical University of Gdansk, Gdansk, Poland

In the paper we present simulation results of heat transfer in 3D systems plates modeled by coupled atomistic and continuum systems obtained on cluster of computers. Distribution of the calculations has been performed using both message passing and shared memory techniques. Bridging the scales and combining molecular and continuum approach has been intensively investigated for many years now. Attention has been mostly paid to mechanical deformations and crystalline defects and their propagation through the atomistic-continuum boundary. A number of methods has been developed, partly presented in the overview in [1].

In the presented work we compare and analyse two models for the less developed heat coupling. The first on is based on Xiao and Belytschko approach [2] in which we introduce the interface region with overlapping molecular and continuum representations. The continnum part is described by finite elements, while atomistic one is modeled by classic Molecular Dynamics. We couple atomistic and continuum parts through a bridging domain, assuming roughly 100 atoms per element. The heat transfer in the bridging domain is represented by the minimization of linear combinations of Hamiltonians for discrete and continuum parts. The temperature there is defined through linear combination of kinetic energies of particles and nodes respectively.

In the second model we simply add temperature to nodes in the continuum model as a hidden degree of freedom. Again we use a bridging domain and the iterative procedure for heat transfer from the particles to the nodes.

In the simulations we use a single 3D sample modeled in half using particles and in half using continuum representation. In the particle model we use non-equilibrium molecular dynamics with two heat baths placed on the ends of the system. Special attention has been paid to the analysis of the influence of the grid scale and the integration time-step in the continuum system, in order to enable smooth energy transfer and avoid rapid growth of energy in the atomistic part near the particle-continuum boundary.

Bibliography
[1] Curtin W.A. and Miller R.E., 2003, Modelling Simul. Mater. Sci Eng. 11, R33
[2} Xiao S.P. and Belytschko T. 2004, Computer Methods in Applied Mechanics and Engineering, in press.

29. Grid Inventory and Roadmap

Ali Anjomshoaa (1), Ariel Oleksiak (2), Mark Parsons (1), Paul Graham (1), Matti Heikkurinen (3), Heinz Stockinger (3), Kurt Stockinger (3), Jarek Nabrzyski (2), Maciej Stroinski (2), Jan Weglarz (2)
(1) EPCC, Edinburgh Parallel Computing Centre, Scotland
(2) PSNC, Poznan Supercomputing and Networking Center, Poland
(3) CERN, Switzerland

Nowadays Grid computing is becoming a very popular research area and attracts attention of many scientists and industrial players all over the world. In Europe, there are plenty of both national and EU-funded projects working on Grids. Many of these Grid projects have been funded by the European Commission through the Framework 5 IST programme. Most of these projects have already finished or is going to finish soon.
Hence, outcome of their work, called often a "first generation Grid", contains many solutions and tools that are mature enough to create a solid background for further development. However, results of these projects need to be consolidated and promoted.
In order to categorise and summarize existing projects in Europe, we have developed a taxonomy and applied it to 20 FP5 Grid projects. We briefly describe the projects and provide an overview of current Grid activities in Europe. Furthermore, we describe noticed trends based on both the European Grid activities as well as progress of the world-wide Grid community. The work we present here is a unique source of information that aims to help to promote European Grid development.

30. Grid Organization Memory for Knowledge Management for Grid Environment

K. Krawczyk (1), R. Slota (1,2), M. Majewska (2), B. Kryza (1), J. Kitowski(1,2)
(1) Academic Computer Centre Cyfronet AGH, Cracow, Poland
(2) Institute of Computer Science AGH-UST, Cracow, Poland

The EU funded project, called Knowledge-based Workflow System for Grid Applications (K-WfGrid), focuses on the knowledge based approach to developing scientific and commercial applications for the grid. The grid organizational memory, as planned, is responsible for knowledge modeling and implementing it in the form of ontologies, knowledge storage in the distributed knowledge base and finally knowledge searching and accessing. The special attention is given to the grid services management. The semantically rich metadata in the form of ontology will capture the services characteristics and put them into the knowledge base. These metadata will describe possible features of grid services and also other elements to make the seamless integration of grid information possible. The information and data maintained by the grid middleware and grid applications will not be copied to the knowledge base. Their meaning and also the way to access this information will be described.

The paper will present results of initial work done in the field of knowledge modeling. We will show the way of OWL-S usage in our system in order to describe and search for required grid services. The resources and application ontology will also be presented. Furthermore, the initial design of the knowledge base engine framework will be presented. The way of accessing the knowledge base will be discussed as well as different possibilities of the storage. The reliability, failover and the security issues will be presented.

31. Grid Software Installation Tools

Marcus Hardt, Harald Kornmayer, Ariel Garcia
Forschungszentrum Karlsruhe GmbH, Germany

Software installation in a Grid Environment is still an open problem. The basic grid software - called middleware - usually requires administrative privileges for installation. Application software can be installed by unprivileged users. Consequently it requires different treatment than middleware.

The deployment of middleware was addressed in all big grid middleware projects like CrossGrid or EGEE. Sophisticated roll out procedures for homogenous deployment of middleware to all participating sites have been developed. It is centrally specified what software component will be installed at which grid element. To make sure that the deployed software does not compromise the system security or stability issues, intensive acceptance testing measures are required.

This central approach is also being used for application software. But it introduces a big amount of involved overhead. Thus the responsibility for application installation was delegated towards communities or virtual organisations.

Different procedures for software deployment evolved in Alice and in LCG. In both projects the focus is on enabling virtual organisations to deploying registered software to the grid. Still, the decision of which software is going to be installed is centrally made. ne software manager is in charge of doing acceptance tests per VO. A lightweight way for application installation by the individual grid-user is not yet addressed.

This paper describes the different ways for software installations on the grid outlined above in detail. Furthermore, this contribution will point out the limitations of the software installation approaches that were presented. A new approach for installing software will be presented and discussed. This approach is characterized by being minimally dependent on other software and on the runtime environment. It shall be deemed to be the logical extension from middleware installation towards ab end user oriented installation.

32. Grid testing using virtual machines

Stephen Childs, Brian Coghlan, David O'Callaghan, Geoff Quigley, John Walsh
Trinity College Dublin, Ireland

Realistic testing of Grid software requires a testbed configuration that closely resembles the deployment environment. A full testbed representing a single site must therefore include enough machines to run gateway servers, compute clusters and services such as replica location, MDS, etc; a testbed representing a multi-site grid would require proportionately more machines. This would be beyond the reach of smaller sites wishing to develop new Grid software; even large institutions may be slow to commit significant numbers of machines exclusively to testing.

We propose the use of Virtual Machines (VMs) to enable realistic Grid testing with relatively modest resources. Each physical machine can host multiple servers running in VMs, thus reducing the hardware commitment.

Disk images of each node type can be stored to allow new nodes to be quickly instantiated. Automated configuration management tools can be used to make changes specific to a particular test environment.

Not only can the entire functionality of a grid access gateway be hosted on one physical test machine, but a small number of physical hosts with VMs can be used to create and evaluate virtual clusters of various scales. Multiple sites can be assembled, and these may even be (as in the Grid-Ireland TestGrid) faithful replicas of the real sites. All the national servers for a country may be replicated in VMs on another physical host.

We describe feasibility studies conducted at the Grid-Ireland Operations Centre. Preliminary results show that using the Xen VM, the performance of virtual sites is close to that of real sites: the small CPU overhead due to virtualisation is compensated for by increases in network performance. New nodes can be initialised in minutes, and automatic configuration means that they can quickly be brought into an operational state.

The use of VMs brings the creation of realistic testbeds within reach of smaller institutions and can act as a catalyst for the development of new Grid tools and services that can be quickly deployed in the real world.

33. GRIDdy - Globus toolkit framework

Boris Turk(1), Marjan Sterk(2), Gregor Pipan(1), Roman Trobec(2), Ales Stimec(1), Igor Rozman(2)
(1) XLAB - Research, Ljubljana, Slovenia
(2) Jozef Stefan Institute, Ljubljana, Slovenia

This paper presents framework named "GRIDdy" and its to use for developing and running grid applications. The framework is intended for parallelization of computationally intensive applications whose input data can be split and the results merged, i.e. applications that do not require a large amount of communication. A sample application based on combinatorial search is also presented. GRIDdy is based on Globus toolkit 2, Java CoG (Commodity Grid), gridFTP and Apache Ant. Java was used to enable portability and interoperability on heterogeneous grids.

Globus Toolkit and existing tools like Java CoG are relatively complex. GRIDdy was designed to simplify and accelerate the development process and enable the programmer with no technical knowledge on Globus to focus on the application itself. Application intended for running on GRIDdy framework consists of the following parts. First, the data is divided into several pieces. In what fashion this is done depends on the nature of the problem and thus the programmer has to write the corresponding code by extending the existing GRIDdy classes. In the next step the framework transfers each portion of data and all the java classes needed for data processing to specific nodes in grid using GridFTP. Allocating parts of data to grid nodes depends on the structure of grid and thus is not part of the framework but left to the programmer. In the third step the framework executes the uploaded classes that were designed to process different parts of data. If a specific computer is not accessible or the processing does not finish in the specified timeout, another node is chosen. In the last step all the partial results are first collected by the framework and then merged together into a final solution. The latter operation is in tight connection with splitting data and is therefore left to the programmer.

The electrocardiogram (ECG) application was adapted for use in GRIDdy framework. During an ECG measurement, multiple electrodes are placed on patient's chest to measure the electric potentials. The start activity of the heart can be localized using these potentials. A combination of electrode positions provides good localization accuracy if a small change of location of the start activity causes a significant change of the measured electrical potentials. The nature of the model used to simulate the measurements makes exhaustive search a reasonable choice. The GRIDdy-based ECG application splits the set of possible combinations to subsets, which are then distributed to nodes in the grid. The result of each task is the best combination from the corresponding subset. Merging the results consist of simply choosing the best of these results on subsets.

Results of test runs of the GRIDdy framework and ECG application will be presented in the paper. Preliminary results on two nodes of a grid show that using the grid speeds up the computation by a factor from 1.5 to 1.9 on mid-sized problems that take a few minutes to execute.

34. GridMiner: A Framework for Knowledge Discovery on the Grid - from a Vision to Design and Implementation

Peter Brezany (1), Ivan Janciak (1), Alexander Wöhrer (1) and A Min Tjoa (2)
(1) Institute for Software Science, University of Vienna, Austria
(2) Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria

Knowledge discovery in data sources available on Computational Grids is a challenging research and development issue. The GridMiner project (www.gridminer.org) aims, as the first Grid research effort, to cover all aspects of this process and integrate them as an advanced service-oriented Grid application.
Currently, the suite of Grid data analysis services includes sequential (clustering, sequence discovering),

parallel (OLAP) and distributed (mining decision trees and rules) implementations which are able to deal with standard data sources like, for example, the XML WebRowset format. Additionally the service for discovery of association rules can operate directly on OLAP data cubes.
A special research effort of our project deals with the integration of all needed services into a workflow, which is executed by an appropriate engine. Due to its data exploration tasks, GridMiner needs a highly dynamic workflow concept, where a client can compose the workflow according to its individual needs. In our approach, we designed a new specification for dynamic service composition called the Dynamic Service Composition Language (DSCL), which is based on XML. Further, we developed a workflow engine called the Dynamic Service Composition Engine (DSCE). DSCL allows the description of a workflow consisting of various Grid services and the specification of their parameter values. DSCE can be controlled interactively by a client, which has the possibilities to execute, stop, resume or even to change the workflow and its parameters. Moreover, DSCE can be used for processing in batch mode.
Knowledge discovery is a highly interactive process and so it is essential that GridMiner provides a powerful, flexible and simple to use graphical user interface (GUI). The GUI allows interactively constructing workflow descriptions and visualising the results of knowledge discovery tasks. The results of data mining tasks are represented by the Predictive Model Markup Language and the results of OLAP queries in our own XML schemas. The GUI is supported by the knowledge base, which consists of the following building blocks: ontologies, metadata, rules and facts. We proposed our own special ontology for data mining domain, for data sources and activities used in the process of knowledge discovery. The knowledge base allows to store and share newly discovered knowledge for future knowledge reuse.
The knowledge discovery process is supported by a novel Grid Data Mediation Service, which allows integrating heterogeneous relational databases, XML databases and comma separated value files into one logically single homogeneous virtual data source. The newly developed concepts for the mediation service have been implemented by reusing the standard reference implementation of Grid Data Services, namely OGSA-DAI, proposed by the DAIS Working Group.
The GridMiner system has been implemented as a research prototype completely built on top of the Globus Toolkit Version 3 and standard web technologies.

35. Grid-wide Intrusion Detection

Stuart Kenny and Brian Coghlan
Computer Science Department, Trinity College Dublin, Ireland

We describe SANTA-G (Grid-enabled System Area Networks Trace Analysis), an instrument monitoring framework that uses the R-GMA (Relational Grid Monitoring Architecture). We describe the CanonicalProducer, the R-GMA component that allows for instrument monitoring, and how it would be used to construct the basis of a Grid-wide intrusion detection system.

The Grid Monitoring Architecture (GMA) of the Global Grid Forum (GGF), consists of three components: Consumers, Producers and a directory service, which in the R-GMA is referred to as a Registry. R-GMA is a relational implementation of the GMA developed within the European DataGrid (EDG), which harnesses the power and flexibility of the relational model. R-GMA creates the impression that you have one RDBMS per Virtual Organisation (VO). However it is important to appreciate that the system is a way of using the relational model in a Grid environment and not a general distributed RDBMS with guaranteed ACID properties. All the producers of information are quite independent. It is relational in the sense that Producers announce what they have to publish via an SQL CREATE TABLE statement and publish with an SQL INSERT and that Consumers use an SQL SELECT to collect the information they need.

SANTA-G is a generic template for ad-hoc, non-invasive monitoring with external instruments. It is intended to use the SNORT functionality of the SANTA-G network monitoring tool to construct a system that will allow for Grid-wide intrusion detection. The enabling technology for the template is the CanonicalProducer. A demonstrator of this concept, developed within the CrossGrid project, is a network monitor that allows a user to access logs stored in libpcap (a packet capture library) format through the R-GMA. Examples of tools that generate logs in this format are Tcpdump, an open-source packet capture application, and SNORT, a network intrusion detection system.

SNORT logs alerts in response to suspect packets. These alerts will then be published to the R-GMA via the SANTA-G network monitor. By using other R-GMA components, such as an Archiver, alerts published from multiple sites can be aggregated to form a Grid-wide intrusion log. Custom coded Consumers can then query this log for specific alert patterns, triggering their own alerts if a pattern is detected, and thereby generating Grid-wide intrusion alerts. It is also intended to complement the SNORT data with information from other security components. Work is currently underway extending the SANTA-G network monitor for use with such tools as Tripwire and AIDE (Advanced Intrusion Detection Environment).

36. G-Vid - A Dynamic Grid Videoservice for Advanced Visualization

Martin Polak, Dieter Kranzlmueller, Jens Volkert
GUP, Joh. Kepler University Linz, Austria

This paper describes our recent efforts in using state of the art video coding techniques for providing interactive visualization, streamed content creation and transformation on the grid. Our grid-based video service G-Vid offers the ability to create high quality (and thus) computationally expensive images of the user's visualization request. For this, G-Vid utilizes a grid-empowered centralized or decentralized streamed content creation service by providing simple but powerful interface to the user.
Based on its modular nature, G-Vid can accept arbitrary types of input data to create multilevel streamed video output data streams, using the latest MPEG4 coding/decoding technology.
In contrast to other existing streaming video solutions, such as Darwin StreamingServer or WindowsStreamingServer, G-Vid is very flexible regarding its inputs and the ability to capture and provide feedback channels to the underlying content creating applications. This allows to seamlessly integrate interactivity through the grid for users acting on the streamed video output.
An optimal look and feel is provided by dynamically adapting the output characteristics of the presented material depending on the users client abilities and the actually available transport bandwidth on the grid.

37. HPC4U: Providing Highly Predictable and SLA-aware Clusters for the Next Generation Grid

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

Research on Grid Computing lead to numerous Grid middleware systems, e.g. UNICORE and the Globus Toolkit. However, current Grid architecture and implementations lack many essential capabilities which would be necessary for a future large scale Grid system. For that reason, a group of experts has been convened by the European Commission, asked to identify research priorities for realizing the Next Generation Grid (NGG).

Applications in these NGGs will demand the Grid middleware for mechanisms to enable a flexible negotiation of specific levels of Quality of Service(QoS). In this context, a QoS guarantee may range from the reservation of resources for a given time span, which is required for the orchestrated usage of distributed resources, up to guarantees for an advanced level of Fault Tolerance.

A Service Level Agreement (SLA) is a powerful instrument for describing a job's requirement profile. It is the exact statement of all obligations and expectations within the business partnership between the resource provider and the Grid user as its customer: it describes which resources should be provided in what amount for how long and in which quality. It also encompasses the price for resource consumption, resp. the penalty fee for violating the agreement.

Many research projects already focus on SLA functionality within the Grid middleware. However, it is not sufficient to add SLA mechanisms like negotiation or monitoring to Grid middleware systems only. As Grid middleware systems base on local Resource Management Systems (RMSs) for execution of Grid jobs, also these RMSs have to be able to guarantee the contents of a negotiated SLA. Comparing the capabilities of current Resource Management Systems on the one side, which are at best able to reserve resources for a fixed time span, and the requirements of future Grid systems on the other, a gap between both sides becomes apparant.

The goal of the HPC4U project (Highly Predictable Cluster for Internet Grids) is to provide an application-transparent and software-only solution of a reliable Resource Management System. It will allow the Grid to negotiate on Service Level Agreements, and it will also feature mechanisms like process and storage checkpointing to realize Fault Tolerance and to assure the adherence with given SLAs. The HPC4U solution will act as an active Grid component, using available Grid resources for further improving its level of Fault Tolerance.

In this paper, we describe the architecture of the HPC4U solution. It consists of an SLA-aware resource management system, which negotiates the SLAs and monitors their execution. It interfaces with lower level building blocks for job-transparent checkpointing and restart, storage virtualization and checkpointing, and network failover. The RMS will be delivered as open source, while the lower level blocks will be available as commercial products. However, comparable products may be integrated as plug-in modules.

HPC4U (IST-511531) is a EU-funded project within the sixth framework program (FP6). It started on 01.06.2004 as a 36 months research project with 8 partners from 6 countries.

38. Improved communication for the parallel applications on the grid

R. Metkowski, P. Bała
Faculty of Mathematics and Computer Science, N. Copernicus University, Torun, Poland

Grid computing become a new approach to the use of computer resources. At the same time, Beowulf type clusters started to play important role as computing platform for the academic and scientific users. Unfortunately there are not many tools available that support distributed computing in grid environment, especially for the parallel jobs run across available computational resources.

In contradiction, parallel computing, limited to the massive parallel systems, or recently to the single, tightly connected clusters is well established.
The number of standard tools exists, for example widely used communication libraries such as MPI or PVM. These tools provide currently good performance while used on the dedicated parallel systems as well as on the tightly coupled PC clusters. The communication efficiency significantly decreases at the clusters build with standard Ethernet interconnect, or while distributed resources are used. It is known that the main reason which prohibits efficient parallel simulations comes from the high communication latency, rather than from limited communication bandwidth. The another difficulty comes from the fact that most of large clusters is located behind firewall and is using private address space.

In order to allow for parallel simulations on the grid we have extended existing MPI tools to support local IP addressing and firewalls. In the distributed environment consisting of number of clusters The MPI communication is handled by the dedicated daemons running on the cluster front-ends. The role of the daemon is to establish communication between clusters and redirect communication to the intracluster channel. The communication pattern is redesigned and all messages are passed through the daemons rather that in the form of the point - point messages.

Presented design allows for handle of local addressing scheme and allows for optimization of the communication which reduces latency. Developed tool allows to achieve resonable performance of the parallel jobs on the grid.

This work is supported by the State Committee for Scientific Research.

39. Intel Itanium 2 Processor - New Standard for Computing Servers

Pawel Pisarczyk, ATM, Poland

40. InteliGrid project: A vision of engineering on the grid

Matevz Dolenc, Vlado Stankovski, Ziga Turk
University of Ljubljana, Faculty of Civil and Geodetic Engineering, Ljubljana, Slovenia

The original idea behind grid computing is to support collaborative problem solving in virtual organizations (VO). A challenge for collaboration infrastructures is to support dynamic VOs that collaborate on the design, production and maintenance of products that are described in complex structured product model databases.

Such VOs are typical for industries with long supply chains like automotive, shipbuilding and aerospace. Perhaps the most complex dynamically changing VOs and are in architecture, engineering and construction (AEC). Semantic interoperability of software and information systems belonging to members of the VO is essential for efficient collaboration within the VO.

We believe that the current state of the art - the Web Services paradigm, is too fragile and tangled for efficient collaboration in AEC. Grids provide the robustness but need to be made aware of the business concepts that the VO is addressing. The grid itself needs to commit to the product's and process's ontology thereby evolving into an ontology committed semantic grid. To do so there is a need for the generic business-object-aware extensions to grid middleware, implemented in a way that would allow grids to commit to an arbitrary ontology. These extensions are propagated to toolkits that allow hardware and software to be integrated into the grid. This is expected to be done in the InteliGrid project. This paper presents projects baseline, hypothesis and expected results as well as early developments towards the above described goal.

41. Interaction with Grid-Enabled Applications using SSH Session Server Framework

Tomasz Kuczynski (1,2), Piotr Kopta (1), Roman Wyrzykowski (1)
(1) Czestochowa University of Technology, Poland
(2) Poznan Supercomputing and Networking Center, Poland

This paper presents the SSH Session Server Framework. The original goal of this software is to create a flexible portal interface that can be easily extended and adopted for utilization with different applications. The main pressure is put on separation of the visualization part from the application's logic, as well as giving possibility of framework extension at run-time. Supporting VRML, X3D and SVG output presentations is also one of the most important features.

It is necessary to provide adaptation of ready end-user applications, and easy utilization of them seamlessly on multiple hosts. These requirements constrain us to use SSH as a communication channel.

All of the above leads us to extend concepts which have been validated during the Web Condor Interface implementation and utilization. New feature is using XML application extensions, called by us "parsers", which allow us to describe rules of user-application interaction, including input data format, output data parsing and visualization. In combination with Web Condor Interface concepts: SSH sessions and pseudo terminals, parsers and application managers allows us to gain pseudointeraction in Grids.

The SSH Session Server Framework pulls together with the GridSphere Portlet framework - the outcome of the GridLab project. This in turn allows to use a variety of build-in features such as user's secure space, chart generation, etc.

The SSH Session Server Framework consists of two main parts: (i) SSH Session Server Portlets and Services, (ii) SSH Session Server. Utilization of the SSH Session Server allows us to gain persistence and interaction possibilities.

42. Interactive Access in the UNICORE environment

Csaba Anderlik, Jan-Frode Myklebust
Parallab, University of Bergen

High Performance Computing (HPC) resources are usually batch oriented, i.e., there is a local resource management system which takes care of the execution of incoming tasks based on the available resources and the resources requested by a given task. Therefore, Grid middleware technologies are also often batch system oriented. Typically the user prepares a job (a generalized description of the tasks to be performed and the requested resources), submits it to the Grid, and then retrieves the outcome after the job has finished. There is no interaction with the job while the job is running.
In this paper, we present a mechanism that provides interactive access functionalities for UNICORE software. Although some of the implementing components are dependent on functionalities of UNICORE, the structural ideas can be generalized applied to other middlewares as well. The results presented in this paper are the outcome of the participation of Parallab in the EU-funded project EUROGRID (01-11-2000 to 31-12-2003).

Naturally, many users of HPC systems, users of simulation applications or users who need debugging capabilities, expect some kind of interactive access built in the Grid that enables the user to interact directly with a running job. Interactivity has several levels:
1) interactive steering and control of jobs. By introducing explicit job control tasks, the user can define branches and conditions in a task graph, that are being evaluated during job execution;
2) interactive steering and control of application. By providing input and reading output of an application while it is executing in the batch system, the user can direct the further progress of an application;
3) terminal-based shell access. This provides editing capabilities and full shell access to the target computing resource.

The first level of interactivity is implemented in the standard UNICORE client, while we implemented the other two levels as extension plugins for the client.
The second level of interactivity has been implemented using the special file called FIFO. The application running in the batch system reads from a FIFO, and writes its output to a file. The user sends input to the FIFO by submitting simple ECHO ExecuteTasks to the computing resource on which the application is executing.

The implementation of the terminal based shell access (or IA SSH feature) is based on the interfaces in the Gateway and NJS used by the Alternative File Transfer mechanism. These interfaces allow services to be added to a Gateway in a manner that is consistent with the UNICORE setup. These services provide some implementation-dependent functionality whose initiation will be controlled by the same authentication setup as for UNICORE Protocol Layer (UPL) requests.

The chosen solution has the following advantages:
- It is a fast solution that enables real-time code development and debugging.
- It is very well integrated in (and fully compliant with) the UNICORE infrastructure.
- No modifications are required to the TSI script.
- The users can be authorized based on the information stored in the UNICORE User DataBase.
- The IA Gateway and NJS modules are loosely coupled with the third-party software (J2SSH) used to set up the connection between the Gateway and target machine. Any Java-based SSH implementation can easily be incorporated in these modules.

The final paper will describe the above in more detail.

43. INTERACTIVE TASK INVOCATION IN THE VIRTUAL LABORATORY

Marcin Okoń, Marcin Lawenda, Norbert Meyer, Dominik Stokłosa, Tomasz Rajtar, Damian Kaliszan, Maciej Stroiński
Poznan Supercomputing and Networking Center, Poland

The Virtual Laboratory (VLab) is a distributed environment, providing a remote access to the various kind of rare and expensive scientific laboratory equipment, and other computational resources [1]. Usually, this is achieved by presenting a GUI of a remote application, responsible for controlling a given scientific device. And the majority of data processing applications also require the interactive user communication. That is why the VLab must handle the scheduling and executing of the interactive applications accordingly.

In general, there are two main kinds of tasks in the Virtual Laboratory: experiments and computational tasks. The latter can be divided into regular (batch) jobs and interactive/visualization tasks (the ones performed in the real time, directly by the users - via the GUI). The biggest difference (and difficulty) between those types is that - in the interactive and experimental tasks - the time slot reserved for running the task on a computational machine or scientific device must be synchronized with user preferences, considering specific work hours, daily schedule etc. In case of experiments the deciding factor is the current device availability, workload, maintenance periods etc. This kind of data can often become very non-deterministic.

Every of those aspects had to be considered during the design phase of the mechanism presenting the users with the graphical interface of the actual computational (or visualization) application, allowing them to perform their interactive task. Because in the Virtual Laboratory all computational tasks submitted in the VLab system are transferred to the Globus via the GRMS [2] module - an important part of the GridLab project [3] - this proposed solution had to consider the dynamic nature of Grid systems, different possible program execution location, resources availability and - last but not least - the security issues.

The solution, implemented in the Virtual Laboratory system is as follows:
The interactive task is scheduled to the Grid via the Grid Gateway Module, which contacts the GRMS and passes the task and the references to the input data, all described in the xml format. GRMS decides where the interactive task should be executed (via the Monitoring and Discovery System - MDS) and reserves a slot for a VNC [4] session. All this information is returned back to the VLab system (via the Monitoring module).
The establishing of the connection is initiated by the GRMS, which starts the appropriate VNC Manager module on a computational machine. The VNC Manager uses the Virtual User System (VUS) [5] to dynamically create temporary user account and gather accounting information for the VLab user for whom the application will be run.
In the next step, the VNC Manager allocates an empty slot (VNC session port and user password) and returns this information to the Monitoring. Using this data, user connects to the remote application via Secure VNC (SVNC) viewer by connecting to the waiting VNC Server - managed by the VNC Manager. The session can be terminated by the user request or by the GRMS, in case the reservation period expires. Other option provides the mechanism to prolong user session on request (only if it is feasible).

The security aspects focused mainly on the securing the VNC client connection to the VNC server. This is achieved by dynamic control of the server (i.e. starting it only when the scheduled task is about to start, dynamic password generation etc.) and by the secure tunnelling the VNC connection, using the Zebedee program [6]. Furthermore, the connection between all modules in the system is also encrypted using SSL certificates.

Although this solution was developed for the Virtual Laboratory System, it was designed in a way that allows it to be reused in any other project dealing with the same problem. It has a modular, easy to install architecture, all the necessary functionality and works closely with other well known Grid solutions.

44. Interactive visualization using the UNICORE grid middleware

K. Benedyczak (1,2), A. Nowiński (1), K. S. Nowiński (1), P. Bała (1,2)
(1) ICM, Warsaw University, Warsaw, Poland
(2) Faculty of Mathematics and Computer Science, N. Copernicus University, Torun, Poland

Recent grid activity has been significantly oriented towards utilization of the computational resources for the data processing and number crunching rather than visualization. This approach was also influenced by the fact that computational intensive applications in general have no dedicated user interfaces. Some expensive commercial graphical tools are available but they are treated as separate applications run locally on the users workstation. Using these applications the user can prepare input or perform advanced postprocessing. The results are stored in the files used as input for another job run in the grid environment which is used to transfer input and output files and to submit and control job.

Progress in the networking allows now for efficient transfer of large data files with the acceptable performance suitable for the interactive visualization which goes behind traditional file transfer schema. However, it is commonly believed that latency and communication overhead caused by the grid middleware prohibits on-line visualization in the grid environment.

In this paper we have extend existing capabilities of the UNICORE grid middleware in order to provide mechanism for interactive visualization. The mechanism of the UNICORE gateway plugins has been used to stream visualization data from the computational server to the visualization workstation. The authorization and authentication as well as job control are performed using UNICORE tools providing standard grid capabilities. The important advantage of the presented solution is good security model and ability to work with the firewalls and local networks. No additional ports, except one used by the UNICORE gateway must be opened.

The visualization itself, is provided by a visualization plugin, allowing user to choose among several generic visualization techniques. Any data stream in the recognized format may be downloaded from execution space while job is running, and then presented in custom way in client's window. The result are fully featured 2D or 3D data viewer, allowing visualize and analyze data quickly in a single user environment.

The interactive visualization tools were tested using high performance applications such as molecular dynamics, proving ability to perform interactive visualization in the grid environment.

This work is supported by European Commission under IST UniGrids project (No. 004279). The financial support of State Committee for Scientific Research is also acknowledged.

45. Interoperability support for Java-oriented on-line tools

Wlodzimierz Funika, Arkadiusz Janik, Marcin Smetek
Institute of Computer Science, AGH, Krakow, Poland

Nowadays, creating new applications has become a very difficult process. In order to accelerate and to minimize the number of bugs done during developing process, it is necessary to use monitoring tools supporting this process. However, using different tools may even make situation worse if tools are not able to cooperate with each other. This feature is called interoperability. In this paper we will present an extension to On-line Monitoring Interface Specification (OMIS) that provides a platform for the interoperability of monitoring tools. Since Java is getting a more and more important platform used in distributed computing, we focused on the tools-monitoring system interface based on Java-oriented On-line Monitoring Interface Specification (J-OMIS) and called the extension JINEXT (Java INteroperability EXTension).

A number of different interoperability models can be found such as ToolTalk, Eureka Software Factory, Purtillo, and LARKS. OMIS also provides a kind of interoperability support for monitoring tools. However, not all of problems have been solved in it. Common problems can be divided into two main categories: structural conflicts (e.g. running the tools concurrently, allowing them to work at the same time sharing the same resources) and logical conflicts (e.g. maintaining the consistency of the system or making changes made by some tool invisible for other tools).

When designing JINEXT we used the mediator model solution, which supports software modularization and makes a behavior evolution consistent. JINEXT acts as an additional layer between a tool and a monitoring system, processing tool's requests and maintaining synchronization between tools. Our solution introduces a new token type, called jtool_ which inherits from the p_ process token. It means that each tool which uses OMIS to monitor the state of a target application is treated as an ordinary process, what allows manipulating it's operation. Each tool must implement a single interface defined by events and actions. An interface clearly defines which services can be executed by a tool and which events this tool can raise. A mediator is a kind of connection between the event generated by a tool and the action which should be executed by another tool whenever the event occurs.

A prototype of JINEXT compliant system is called JINTOP (Java INTerOPerability) and implemented as a dynamic library loaded by the monitoring system whenever JINEXT's services are required. The results of tests carried out on example tools (a profiler, debugger, editor, and compiler) are interesting. In our scenario the compiler tool is notified whenever the editor tool changes the source code of an application. A recompilation of the code makes the debugger restart the application, while the profiler tool continues measurements if the application is paused/resumed. The overhead value due to using JINTOP instead of direct OCM calls is acceptable (ca. 3-5\% for several interoperating tools). The next step in this research is to extend it to be grid compatible.

1. Roland Wism"uller: Interoperability Support in the Distributed Monitoring System OCM, In: R. Wyrzykowski et al., (eds.), Proc. 3rd International Conference on Parallel Processing and Applied Mathematics - PPAM'99, pages 77-91, Kazimierz Dolny, Poland, September 1999, Technical University of Czestochowa, Poland. Invited Paper.
2. M. Bubak, W. Funika, M.Smetek, Z. Kilianski, and R. Wismueller: Architecture of Monitoring System for Distributed Java Applications. In: Dongarra, J., Laforenza, D., Orlando, S. (Eds.), Proceedings of 10th European PVM/MPI Users' Group Meeting, Venice, Italy, September 29 - October 2, 2003, LNCS 2840, Springer, 2003

46. Java Based Component Expert Framework for Automatic Application Builder

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

Automatic Application Builder (AAB) for automatic service selection for scientific and business grid enabled workflows proposed in [1], requires Component Expert Architecture as a framework for service selection in a knowledge based environment. However, most of the contemporary state-of-the-art knowledge driven solutions are written in Java programming language, and thus it implies that AAB framework should be written in that language.

In this paper a comparison of Java implementation of CEA elements with C++ one, used previously in the CrossGrid project [2] is discussed.

Acknowledgement
The work described in this paper was supported in part by the European Union funded projects CrossGrid IST-2001-32243 and K-WfGrid IST-2004-511385. AGH Grant is also acknowledged.

References
[1] L.Dutka and J.Kitowski. Automatic Application Builder for Grid Workflow Orchestration. In M.Turala M.Bubak, M.Noga, (eds.), Proc. Cracow Grid Workshop'03, October 27-29, 2003, Cracow, pp.131-135, ACK Cyfronet AGH, Cracow 2004.
[2] Crossgrid - development of grid environment for interactive applications, 2001. EU Project no.: IST-2001-32243, http://www.crossgrid.org.

47. JIMS - the Uniform Approach to Grid Infrastructure and Application Monitoring

Kazimierz Balos, Krzysztof Zielinski
Deapartment of Computer Science, AGH University of Science & Technology, Poland

Object oriented resources representation makes new possibilities of infrastructure and application monitoring. Assuming that all resources and applications are represented in a unified manner, there can be created one system that provides common view for monitored infrastructure and applications. An example implementation of such a system is the JIMS monitoring system, which stands for the JMX-based Infrastructure Monitoring System. Though its name originates from its primary purpose of execution hosts monitoring, it can be easily adopted for further grid infrastructure and applications monitoring and management. Article presents uniform approach to monitoring grid infrastructure and managing jobs running in the queueing systems, including grid middleware and applications running within Java Virtual Machines. Due to the fact that JMX (Java Management Extensions) have been recently included in the standard release of JVM and since version 5.0 it's an integral part of Java language, all applications can be fully monitored including memory maps, heap sizes and other details concerning running processes and threads. For grid middleware monitoring, there is described special module installed in Computing Element for Sun Grid Engine, PBS or other grid engines monitoring. Using DRMAA standard for grid middlewares, there are performed actions for reading parameters, which cannot be simply accessed using C API or Java API bindings for grid engines. Concept of uniform system for infrastructure and applications monitoring is described in Section 2. Section 3 describes solutions for applications attachments to monitoring system. Possible applications of applications' monitoring, including regular Java applications as well as the applications running within the context of application servers, are

described in Section 4. System evaluation, future work and related projects are described in Section 5. Paper is ended with conclusions.
Keywords: infrastructure and application monitoring, JMX, Sun Grid Engine

48. K-WGrid - Knowledge based Workflow system for Grid Applications

Marian Bubak (1,2), Thomas Fahringer (3), Ladislav Hluchy (4), Andreas Hoheisel (5), Jacek Kitowski (1,2), Steffen Unger (5), Gianni Viano (6), Konstantinos Votis (7) and K-WfGrid Consortium
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Academic Computer Centre CYFRONET AGH, Krakow, Poland
(3) University of Innsbruck, Austria
(4) Institute of Informatics of the Slovak Academy of Sciences, Bratislava, Slovakia
(5) Fraunhofer Institute for Computer Architecture and Software Technology, Berlin, Germany
(6) Softeco Sismat SpA, Genova, Italy
(7) LogicDIS S.A., Athens, Greece

This project, started on 1 September 2004, during following 30 months will address development of a person-centric grid environment easy to program, to configure and to manage, based on standards for software and protocols.

We see the coming Grid as a large, distributed collection of Grid or Web services which may offer computing, data storage, networking, or specialized simulations. Due to dynamic nature of the Grid, workflow composition is a challenging task because the system has to deal with resource unreliability and unpredictability, what is closely related to fault tolerance and scheduling. In existing Grid environments little attention has been given to distributed Grid workflow applications and Grid systems do not use the accumulated monitoring information in order to assist the user in constructing efficient workflows.

The main objective of the Project is the knowledge-based support for workflow construction and execution so we will develop methodology and build a system that
- semi-automatically composes a workflow of Grid services,
- executes the composed workflow application,
- monitors the performance of the Grid infrastructure and applications,
- analyses the resulting monitoring information,
- captures the knowledge contained in the information,
- reuses the joined knowledge gathered from all participating users in a collaborative way in order to efficiently construct workflows for new Grid applications.

The Project will close the gap between monitoring and workflow construction - our grid system will be able to learn how to perform tasks and subsequently will employ this knowledge in workflow construction.

The K-WfGrid will combine together and extend current achievements in Web Services, Semantic Web, Grid services and agent technologies to enable easier composition, execution and monitoring of workflows for real-life applications. The K-WfGrid system will assist users in composing powerful Grid workflows by means of a rule-based expert system. The system will adopt ontological descriptions of the environment and applications, and it will develop a distributed knowledge base, a network of intelligent Grid services, as well as sophisticated performance monitoring and analysis, advanced user interfaces, and knowledge discovery. It will use a dynamic, just-in-time approach to workflow construction, where services will be selected at the time when they are needed, based on up-to-date runtime information about the status of Grid resources.

Besides the flood application, the project will implement an enterprise resource planning application for the financial sector. This application will include enterprise financial and accounting systems, manufacturing systems, supply chain management and electronic commerce, all integrated into a single enterprise-wide distribution. The third application of the K-WfGrid falls into the domain of coordinated traffic and transport networks management.
We expect that many environmental, industrial, state and municipal authorities will be the potential users of technologies developed within the Project.

K-WfGrid Consortium is coordinated by Fraunhofer Institute for Computer Architecture and Software Technology; Berlin, Germany, and, besides of this institution, there are 3 academic partners: Institute for Computer Science, University of Innsbruck, Austria, Institute of Informatics of the Slovak Academy of Sciences, Academic Computer Centre CYFRONET of the AGH University of Science and Technology, Cracow, Poland, as well as two industrial partners: LogicDIS S.A. from Greece and Softeco Sismat SpA from Italy. There is already a group of associated partners going to work together with us.

References

[1] www.kwfgrid.net

49. LGF: A Framework for Semi-Automated Adaptation of Legacy Code to Web Services

Bartosz Balis(1), Marian Bubak(1,2) Michal Wegiel(1)
(1) Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland
(2) Academic Computer Centre CYFRONET AGH, Krakow, Poland

The focal point of the research conducted within the scope of the Legacy to Grid Framework (LGF) project [1] is a semi-automated adaptation of legacy software to grid services environment. Our work concentrates on the development of tools intended to significantly reduce the programmatic effort involved in transforming traditional libraries and applications to their service-based equivalents.

There are a few attempts at tackling the problem of smooth migration to grid services technology [2] [3]. Nonetheless, all of the solutions that we are aware of possess considerable shortcomings. Their deficiencies pertain to, among others, poor scalability, insufficient level of security, and lack of versatility.

We propose an architecture designed from ground up in conformance with fundamental grid requirements. Its most distinguishing features are:
- universality (it can accommodate a wide variety of legacy software),
- scalability (it guarantees sound response to temporary peaks in workload),
- security (authentication and authorization are employed and vulnerabilities connected with open incoming ports are eliminated).

Our solution supports most frequently used design patterns, including asynchronous and concurrent method invocations as well as transactional processing. In addition, it offers mechanisms of load-balancing and transparent fail-over realized with the help of high-level process migration.

We implemented a prototype version of a framework that allows for code-generator-driven porting of C/C++ software to OGSI-compliant grid services. It provides a set of utilities that aid with the development of the necessary interposition layer that mediates between backend and client tiers. Performance measurements that we carried out clearly demonstrate that the communication overhead introduced in our architecture is perfectly tolerable in case of computationally-intensive applications.

Framework usability was verified on the basis of two real-world test cases:
- OCM-G (distributed system for on-line monitoring of grid applications) [4],
- C-BLAS (numerical library)

which served as a proof-of-concept of the overall solution.

References

[1] B. Balis, M. Bubak, and M. Wegiel. Adaptation of Legacy Software to Grid Services. In: Bubak, M., G.D. van Albada, P.M.A. Sloot, J.J. Dongarra (eds): Proc. International Conference on Computational Science, ICCS 2004, Krakow, Poland, June 2004, LNCS 3038, pp. 26-33, Springer, 2004.
[2] Huang, Y., Taylor, I., Walker, D., Davies, R.: Wrapping Legacy Codes for Grid-Based Applications. In Proc. International Workshop on Java for Parallel and Distributed Computing, Nice, France, 2003.
[3] Kuebler, D., Eibach, W.: Adapting Legacy Applications as Web Services. IBM Technical Report. http://www-106.ibm.com/developerworks/webservices/library/ws-legacy/
[4] B. Balis, M. Bubak, M. Radecki, T. Szepieniec, and R. Wismüller. Application Monitoring in CrossGrid and Other Grid Projects. In: M. D. Dikaiaokos (ed): Grid Computing. Proc. Second European Across Grids Conference 2004, Nicosia, Cyprus, January 2004, LNCS 3165, pp.212-219, Springer, 2004.

50. Liquid State Machines and Large Simulations of Mammalian Visual System

Grzegorz M. Wojcik, Wieslaw A. Kaminski
Maria Curie-Skłodowska University, Lublin, Poland

Mammalian brains are built of about one hundred billion neural cells (neurons). We can simulate the processes occurring in brain with wide variety of available neural models (i.e. Hodgkin-Huxley model). Nevertheless, it is still not possible to simulate the whole brain (or even any larger part of it) by modeling and considering the work of each neuron separately. That is why we are in strong demand of any model treating the brain as a one complex system. The new idea of modeling the brain as a whole was given by Wolfgang Maass (2002) and since then it has been called Liquid State Machine (LSM). In that model we can treat large groups of neurons

(called microcircuits) as units with some computational power. The work of each unit resembles dynamics of the liquid. However, even after such simplification we are in need of parallel simulations and cluster applications.

In Maass? model there are simple integrate-and-fire neurons. In the referred application we will use several LSMs build of biologically more realistic Hodgkin_Huxley neurons. We will construct the simulation in the GENESIS package which is also available for the PVM and MPI environments. For the needs of our project we will make the model of mammalian visual system consisting large number (reaching even half million) of cells. Results, time and efficiency of simulations conducted on the local cluster will then be compared with the research done on six_processor machine (Sun Enterprise 3000 Ultra).

51. Modeling of Data Replication in Data Grids

Uros Cibej, Bostjan Slivnik and Borut Robic
University of Ljubljana, Faculty of Computer and Information science, Ljubljana, Slovenia

Data replication is a well known technique used in distributed computing. Recently data replication has also been used in grid computing, but there is a lack of theoretical background in the algorithms used. We propose a gradual advance towards better algorithms by formally analysing simplified models of grid systems. In this article we present a static model of the data replication problem on data grids, which is the first step towards quality algorithms in real data grids. We show that this problem is NP-hard and non-approximable.

More preciselly, a data grid is a grid which is used to manage huge amounts of data and data intensive tasks. In such systems, data replication is a technique of placing multiple copies of data on different servers within a data grid. Data replication improves data access time and provides fault tolerance by maintaining copies of data objects (e.g. files, databases or parts of databases) at different locations. If the replicas of objects are well placed, each computational task can access the nearest replica of the needed object, thus reducing its execution time. As a result, the throughput of the whole system can be increased.

Current approaches to data replication are mainly ad hoc strategies, i.e. the replication strategies have very little or no theoretical background.

To address this problem from a theoretical point of view, a formal theoretical model of a data replication problem within a data grid will be presented in the paper. The model includes a graph representing a data grid, a set of data objects to be replicated, and a goal function for evaluating feasible solutions. The model incorporates all relevant elements and parameters: a grid network topology, communication capabilities of network links, storage capacities of grid sites, and sizes of objects. Finally, a goal function, when minimized, favours those replications where every site in a grid has, in average, a fast access to each data object.

Finally, the formulated optimization problem which models data replication is proved to be NP-hard and non-approximable. Nevertheless, the presented model of data replication is the starting point for constructing and evaluating heuristics for solving the problem of data replication.

52. Multicriteria Driven Resource Management Strategies in GRMS

Krzysztof Kurowski, Jarek Nabrzyski, Ariel Oleksiak, Juliusz Pukacki
Poznan Supercomputing and Networking Center, Poland

In order to take into account various local policies and to satisfy different preferences of users (stakeholders) involved in job management processes in large grid environments we have proposed a set of new multicriteria driven resource management strategies.
All these strategies have been implemented as a multicriteria framework, called the MCEvaluator, within the main broker service in the GridLab Resource Manageent System (GRMS).
Furthermore, all these strategies have been successfully integrated with other grid middleware services and tested during real distributed simulations performed on the heterogeneous resources in the GridLab testbed.

The aim of this paper is to show various possibilities of using multicriteria approaches to create more general strategies and models for solving optimization processes in grid environments. Moreover, some preliminary results will be presented to proof the concept of using a multicriteria decision support to automate resource management processes and minimize stakeholders participation in the entire process.

53. OCM-G - Advanced, Grid-Enabled System for On-line Application Monitoring

Bartosz Balis (1), Marian Bubak (1,2), Marcin Radecki (2), Tomasz Szepieniec (2), Roland Wismueller (3)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Academic Computer Centre - CYFRONET, AGH, Krakow, Poland
(3) Electrical Engineering and Computer Science, University of Siegen, Germany

We present the OCM-G (OMIS-Compliant Monitor for the Grid), a monitoring system for Grid applications. The current version of the OCM-G is the result of a three-year development [1] in the framework of the EU-IST Project CrossGrid [2]. The OCM-G supports on-line monitoring of grid applications [3]. The unique capabilities of the OCM-G, among others, include:
(1) support for multi-site applications;
(2) techniques for data rate reduction such as selective run-time instrumentation, local preprocessing, buffering, and profiling, which ensure low overhead and high responsiveness, enough even for monitoring of interactive applications;
(3) services-driven rather than fixed-metrics-driven design: the OCM-G provides a large number of monitoring services which allow for flexible constructing a variety of metrics, rather than offering a limited set of metrics with fixed semantics;
(4) extendibility: the OCM-G can be extended with additional services, loaded dynamically at run-time;
(5) compact and secure design: the OCM-G runs as a set of processes, which while using a lightweight and fast socket-based mechanism for communication, does not impose any extra requirements with respect to potential security holes, such as additional open ports on firewalls;
(6) design as autonomous infrastructure exposing a standard interface OMIS: the services of the OCM-G are available via a standard interface, which minimizes the effort of porting OMIS-compliant tools across platforms (basically only the OCM-G needs to be ported), and since the OCM-G is an autonomous "service" (not just a library linked to tools), interoperability of multiple tools monitoring a single application via the OCM-G is feasible.

All the above-mentioned features were tested and proven in practice in monitoring of real applications [4].

References
[1] Bubak, M., Funika, W., Balis, B., and Wismueller, R.: Concept for Grid Application Monitoring. In: Proceedings of the PPAM 2001 Conference, Vol. 2328 of Lecture Notes in Computer Science, pp 307-314, Naleczow, Poland, September 2001, Springer.
[2] The CrossGrid Project (IST-2001-32243): http://www.eu-crossgrid.org
[3] Gerndt, M. et al.: Performance Tools for the Grid: State of the Art and Future. APART White Paper. Research Report Series
[4] 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, accepted.

54. OpenMolGRID: QSAR/QSPR appications in Grid environment

Sulev Sild, Andre Lomaka, Uko Maran
University of Tartu, Estonia

The quantitative structure activity/property relationship(QSAR/QSPR) methods are commonly applied to develop models for the prediction of various chemical properties and biological activities. In practice, the predictive QSAR/QSPR models are often used for assessing the safety of chemicals, the design of novel materials and discovery of new drugs. These methods require a significant amount of computational resources when applied to large and complex datasets. The Grid computing approach is well suited for speeding up the QSAR/QSPR applications.

We present the Open Computing Grid for Molecular Science and Engineering (OpenMolGRID) system that is based on the UNICORE Grid middleware and integrates various applications for molecular design and engineering tasks. For example, the applications are available for accessing chemical data, semi-empirical quantum chemical calculations, molecular descriptor calculation, QSAR/QSPR model development, chemical structure generation, etc. The system provides a solution for constructing automated workflows based these distributed application resources. In particular, the QSAR/QSPR applications in the OpenMolGRID system are described.

55. Optimized Java computing as an application for Desktop Grid

Richard Olejnik(1), Bernard Toursel(1), Marek Tudruj(2), Eryk Laskowski(2)
(1) Université des Sciences et Technologies de Lille, LIFL UMR CNRS 8022, France
(2) Institute of Computer Science Polish Academy of Sciences, Warsaw, Poland

During two last decades, the inter-connected computing resources show an exponential development. This evolution is accompanied upstream by continuous improvement of the quality of computing resources and downstream by an unprecedented increase in the complexity of computation. The evolution of fast networks of workstations gives rise to a new architectural alternative for the parallel processing and makes it possible to pace this challenge on a medium scale computer networks with Desktop GRID computing [1,2]. It is a new paradigm for programming and software systems, which exploits unused computing resources in the Intranet environments and can deliver tremendous computing power to solve many complex problems.

In this context, we propose a completely Java compliant system built around the object oriented programming environment. To enable efficient execution of heterogeneous applications with irregular and unpredictable execution control we propose DG-ADAJ (Desktop Grid Adaptive Distribution of Applications in Java), a middleware platform, which has mechanisms ensuring automatic adaptation of the elements of the application, in response to computation evolution and to the variations of the resource availability. It will control granularity of computations and distribution of the applications on the Desktop Grid platform of execution, thus, unburdening programmers with these problems. DG-ADAJ, being a programming environment for Java applications, provides a run-time environment, which optimizes dynamic placement of the application objects on Java Virtual Machines implemented on the DGrid. This placement is based on new mechanisms of observation of the activity of objects and relations between them. Initially, DG-ADAJ is an extension of the ADAJ system [3], built for cluster computing. Our current work is to re-engineer ADAJ mechanisms to extend them for larger scale distributed computing. Currently we work on the following problems:

- extension of ADADJ mechanisms so that it can simultaneously manage several applications (inter-applications) in wide and heterogeneous environment (currently ADAJ can optimize only one intra-application),
- extension of initial distribution optimization of application objects on computers (JVMs) [4] towards the DGrid and mechanisms for dynamic run-time migration,
- evaluation of distributed computations built using software common components (CCADAJ project), undertaken on the DGrid platform.

For evaluation of the proposed system concepts, the French GRILLE 5000 platform will be used as soon as it is installed. We hope to observe in the proposed Desktop Grid computing environment the similar efficiency of the continuous CPU resource monitoring and object distribution mechanisms, which have been observed in the case of cluster computing.

References

[1] F. Cappello, Ontology of Desktop Grids: a Pragmatic View from the XtremWeb Experience, GRID 2003 - Desktop Grids: Critical Systems and Applications Research, Phoenix, 17 November 2003.
[2] D. Kondo and all, Characterizing and Evaluating Desktop Grids: An Empirical Study, 18-th International Parallel & Distributed Processing Symposium - IPDPS, Santa Fe, April 2004.
[3] V. Felea, B. Toursel, Middleware-based Load Balancing for Communicating Java Objects, NATO Advanced Workshop on Concurrent Information Processing and Computing, Sinaia, Romania, July, 2003.
[4] E. Laskowski, R. Olejnik, B. Toursel, M. Tudruj, Scheduling Byte Code-Defined Dta Dependence Graphs of Object Oriented Programs, Int. Conference on Parallel Computing in Electrical Engineering, Dresden, September, 2004.

56. Optimizing of data access using replication technique

Renata Słota(1), Łukasz Skitał(2), Darin Nikolow(1), Jacek Kitowski(1,2)
(1) Institute of Computer Science, AGH-UST, Poland
(2) Academic Computer Center CYFRONET AGH, Poland

With the emerging of the grid technologies large scale computational problems become able to be solved with a speed not possible before. Next, the data grids have appeared to address the storage oriented problems associated with these large scale computations. In data grids the data is geographically distributed but accessed through a common interface providing a logical entity to the user. With the highly heterogeneous data grid environment in terms of connection bandwidths and storage elements, the problem of data access optimization appears. One method to cope with this problem is data replication. Two replication method categories can be distinguished: static replication and dynamic replication. In the case of static replication the decision about replica creation or removal is done manually by the system administrator or the client accessing the data. The dynamic replication is done automatically by appropriate grid component in order to follow the changing data access pattern of users and thus reduce their access time.

The polish grid project Clusterix [1] - National Cluster of Linux Systems is designed for use with computational and data intensive applications. The purpose of the project is to develop set of tools and procedures allowing to built productive Grid environment based on local PC clusters (64bit or 32bit) spread in independent supercomputing centers. Project use Pionier (Polish Optical Internet) as a network layer.

We present our research concerning optimizing of data access using replication technique done within the Clusterix project. This research continue our previous study on data access optimization [2] carried out within the CrossGrid project. In the paper we show part of the Clusterix system architecture focusing on the subsystems which are important in point of view of data access optimization and replication. These subsystems are: Clusterix Data Management System (CDMS), Network Resource Management System (NRMS) and JMX-based Infrastructure Monitoring System (JIMS). CDMS implements dynamic replication methods. There are three replication methods used: initial replication, replication on demand and replication based on statistic analysis. Implementation of replication and optimization methods is divided into two

subsystems: optimization subsystem and replication subsystem. We show example case studies of optimization and replication, with an emphasis on project components cooperation (CDMS,NRMS,JIMS) and detailed benefits and costs analysis. Preliminary test results are presented.

In the conclusion we assess the implemented replication as a data access optimization method.

[1] - Clusterix - http://www.clusterix.pcz.pl
[2] - K. Stockinger, H. Stockinger, Ł. Dutka, R. Słota, D. Nikolow, J.Kitowski, Access Cost Estimation for Unified Grid Storage Systems, 4th International Workshop on Grid Computing (Grid2003), PhoenixArizona, November 17, 2003, IEEE Computer Society Press, 2003.

57. Parallel Clustering of Large-Scale Noisy Multidimensional Datasets

M. Kurdziel and K. Boryczko
Institute of Computer Science, AGH University of Science and Technology, Krakow, Poland

Cluster analysis aims at dividing data into groups (clusters) of related points. It has application in different fields e.g. pattern recognition, statistics, analysis of biological data, economy etc. Modern clustering algorithms face a number of issues. In particular the number of clusters may be not known apriori and the clusters can differ in sizes, shapes and densities. Moreover, additional difficulty may be introduced by existence of noise or outliers within the clustered data.

Traditional clustering algorithms assume that each of the data points belongs to one of the clusters. Consequently they are not suited for clustering of noisy data. Moreover, the algorithm often requires the number of clusters to be provided as a parameter, or assume particular clusters' shape (e.g. spherical).

In this article we present a new clustering algorithm that aims at discovering clusters in noisy, multidimensional data. We employ the Shared Nearest Neighbour graphs to find so called core points which span the clusters. The we use the Mutual Nearest Neighbour distance measure to partition the core points into clusters.

We present parallel implementation of our algorithm that is suited for shared memory machines. The experimental results validate the effectiveness of our approach, and demonstrate that it scales almost linearly with the number of processors.

58. Parallelisation in Computational Fluid Dynamics - Performance Issues

Jacek Rokicki, Michał Wichulski, Robert Wieteska
Warsaw University of Technology, Institute of Aeroanutics and Applied Mechanics, Warsaw, Poland

Practical problems in Computational Fluid Dynamics require enormous computer resources if accurate and reliable solutions are needed. This is because the partial differential equations are highly nonlinear, the physical phenomena are multiscaled and at the same type the geometries are very complex. Typical computational meshes contain millions of cells, yet even such detailed spatial resolution is still not sufficient to effectively predict certain phenomena.

Reduction of computational time is possible(at the present level of algorithm development), only through application of parallel processing. This however is not straightforward, and the parallel efficiency depends on few important factors:
- the choice of parallelisation method
- proper load balancing between processors
- limiting the volume of communication
- the choice of proper programming tools

In the present paper this issues are discussed, for the case when parallelisation is based on the domain decomposition. The theoretical model of such parallelisation will be presented. The model allows for quantitative predictions of acceleration and efficiency (as a function both of the problem size and number of processors used). Load balancing issues will be discussed for both structured and unstructured meshes. The performance predictions will be verified experimentally on various parallel platforms. New possibilities and problems arising when using grid architectures will be additionally discussed.

59. Performance Measurements for Interactive Applications in CrossGrid

Marian Bubak(1,2), Wlodzimierz Funika(1), Roland Wismüller(3), Tomasz Arodz(2), Marcin Kurdziel(1)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Academic Computer Centre - CYFRONET, Krakow, Poland
(3) Fachgruppe BVS - Universitaet Siegen, Germany

With Grid computing, a new level of complexity has been introduced to the development process of distributed applications. Grid enabled applications are supposed on a diverse set of hardware and software platforms. At the same time, computing should be as cost effective as possible. Usually, performance drops as a result of an imbalance in computation distribution, flaws in the schema of application synchronisation operations These situations can be fully observed only during an application run in an environment, thus grid-oriented on-line performance monitoring systems become indispensable tools for development of high performance Grid applications. Moreover, the monitoring should not introduce a noticeable overhead as this might disturb the application's execution, leading to perturbed results.

A performance monitoring tool for Grid applications, G-PM, designed to fulfil the above-stated demands is being developed in the CrossGrid project. For gathering of the raw monitoring data the G-PM tool uses the OCM-G monitoring service. The performance data are requested from the OCM-G and then analysed by G-PM during the application's execution. On top of the data returned by OCM-G, a set of standard, built-in metrics is defined, like communication volume and delay, or overhead of file I/O.

The tool differs from related approaches in the respect that it combines the support for on-line standard performance measurements with the support for user-defined metrics and application-specific instrumentation. This enables to adjust the tool's operation to the specific analysis needs meaningful in the context of the application. Once such a metrics, specified with a simple definition language, has been entered in the tool's user interface, it can be used in the same way as the built-in ones.
The paper is intended to present the final status of G-PM.

References
1. Wismüller, R., Bubak, M., Funika, W., Arodź, T., Kurdziel, M., Performance Measurement Model in the G-PM Tool, in: Bubak, M., van Albada, G. D., Sloot, P. M. A., Dongarra, J. J. (Eds.), Computational Science - ICCS 2004. 4th International Conference, Kraków, Poland, June 2004, LNCS, no. 3036, vol. I, Springer, 2004, pp. 462-465
2. R. Wismüller , M. Bubak, W. Funika, T. Arodź, M. Kurdziel , Support for User-Defined Metrics in the Online Performance Analysis Tool G-PM, in: M. D. Dikaiakos (Eds.), Grid Computing. Second European AcrossGrid Conference, AcGrids 2004, Nicosia, Cypus, January 2004, Revisited Papers, LNCS, no. 3166, Springer, 2004, pp. 159-168.

60. Performance monitoring of Java applications using Web Services

Wlodzimierz Funika, Piotr Handzlik, Lechoslaw Trebacz
Institute of Computer Science, AGH, Krakow, Poland

Monitoring of application is observing, analyzing and manipulating the execution of the application, which gives information about threads, CPU use, memory use, about methods and classes. A particular case is monitoring of distributed applications where additional issue is the performance analysis of nodes.

The paper presents a review of Java-oriented monitoring platforms for distributed applications which are oriented towards the use of web services. Java, as a relatively, object-oriented, secure and portable language, is also a flexible and powerful programming system for distributing computing. Distributed programming in Java is supported by remote method invocation (RMI), object serialization, reflection, a Java security manager and distributed garbage collection. The Java oriented monitoring platforms to be analyzed are: Jiro Based Grid Infrastructure Monitoring System, Legacy Software - Grid services, Java Management Extensions (JMX), derivatives of OMIS - compliant monitoring systems (J-OCM, OCM-G). 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 monitoring facilities provided by the monitoring middleware.

We also describe our proposition to monitor applications based on Web Services. We do this with Jakarta Tomcat and AXIS. Jakarta Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. Tomcat is developed in an open and participatory environment and released under the Apache Software License. Apache Axis is an implementation of the SOAP protocol. We aim at the observation and analysis of SOAP messages (data) sended to and from a Web Service (requests and responses). Our system is intended to monitor the characteristics of the remote call, for example: request time, transport time, cache access time, response time. It should show a good scalability, simplicity of use and be transparent for the user. We will build an extension to J-OCM to monitor Web Services which will comprise all above operations. J-OCM uses the J-OMIS interface, which specifies three types of services: information services (providing information about an object), manipulation services (allowing to change the state of an object) and event services (to trigger arbitrary actions whenever a matching event takes place).

Our monitoring system contains three kinds of components: application monitor (AM), node's local monitor (LM), service monitor (SM). Application monitor is embedded in the application (in our case Web Service). It is used to perform monitoring activities in the context of the application. Node's local monitor is created on each node, where there are Web Services to be monitored. LM executes OMIS requests accepted from SMs and sends the results back. Service Monitor resides permanently and exposes the monitoring services to tools.

References

1. B. Baliś , M. Bubak, M. Radecki, T. Szepieniec, R. Wismüller, Application Monitoring in CrossGrid and Other Grid Projects, in: M. D. Dikaiakos (Eds.), Grid Computing. Second European AcrossGrid Conference, AxGrids 2004, Nicosia, Cyprus, January 2004, Revisited Papers, LNCS 3166, pp. 212-219, Springer, 2004
2. Bubak, M., Funika, W., Smętek, M., Wismüller R., Monitoring System for Distributed Java Applications, in: Bubak, M., Noga, M., Turała, M. (Eds.), Proceedings of Cracow Grid Workshop 2003, October 27-29, 2003, Cracow, Poland, ACC Cyfronet UMM, 2004, Kraków, pp. 237-244

61. Perun -- Fault-Tolerant Management of Grid Resources

Ales Krenek and Zora Sebestianova
Institute of Computer Science, Masaryk University, Brno, Czech Republic

The number of computing resources involved in MetaCentre, the Czech national grid project, has been constantly growing over the past years. In this environment we face generic problems of managing those resources:
- they fall into distinct administrative domains, and are distributed geographically,
- the resources are heterogenous, from PC clusters to supercomputers,
- the configuration is constrained with complex integrity requirements (e.g. a batch system server has to be aware of existing user accounts on the managed machines),
- due to the number of resources, the probability of all of them being up at the same time is almost zero.

We address these issues with our resource-management system Perun. Its principal idea is keeping an ideal centralised consistent image of the configuration of the managed resoureces, and attempting perpetually to force the resources to follow this image as closely as possible.

Our implementation keeps the ideal image in a relational database (Oracle), currently about 50 tables, with many both declarative and procedural integrity constraints. All the modifications to the configuration are done in terms of changes of the database. There exist several views on the data respecting the different administrative domains of the managed resources. We provide the resource administrators with both command-line and www interfaces for specific frequent tasks, as well as raw SQL access as a "catch all" tool.

Changes of the database are gathered by database triggers, stored for future reference, and evaluated by plugin modules to plan updates of _services_ on affected resources. The service is an atomic unit of configuration, e.g. the /etc/passwd and /etc/shadow files on a UNIX machine. The service script selects appropriate data from the database, sends them over the network to the managed resource, and instructs a daemon on the resource to perform the actual service update. Alternatively, the service script may update certain resources (e.g. Kerberos) directly. The system is customisable and extensible -- the database schema is not fixed and both change evaluation and service update scripts can be added in order to deploy new services.

Processing of certain workflows may also involve human actions. For this purpose the system gathers _notifications_, attemting to aggregate and sort the required tasks to make these mostly routine actions as easy as possible, and delivers them to the responsible people (currently via email, we consider developing an interface to a request-tracking system).

Security of such system is critical. In general we attempt to grant as minimal priviledges to either user or service identities as possible. There is no need for a single grid-wide "root" credential. We base the implementation on Kerberos but the system can be adapted to use different security mechanisms.

The system has been deployed in the MetaCentre project for several years. Currently it manages an access matrix of about 200 active users to 250 computers (including PC cluster nodes). The configured services include management of user accounts on all the machines, user directories on filesystems (both local and network like AFS), automounter maps, Kerberos realms, configuration of batch system, and access to the machine via X509 certificates.

Complete source code and a detailed installation and administration guide are available from the authors.

62. Porting Thermomechanical Applications to CLUSTERIX Environment

Tomasz Olas, Roman Wyrzykowski
Czestochowa University of Technology, Poland

NuscaS is an object-oriented package for FEM analysis, designed at Czestochowa University of Technology to investigate thermomechanical phenomena. Its functionality includes also parallel implementation on clusters.

This work is devoted to adaptation of the NuscaS package to the CLUSTERIX Grid environment. This environment is a production Grid comprising local PC-clusters based on 64-bit Linux machines. Local clusters are located in geographically distant independent centers connected by the Polish Optical Network PIONIER.

The proposed solution allows us to run tasks not only on a local cluster, but also across several local clusters using the MPICH-G2 environment. To submit tasks, an Internet portal is developed. Its functionality includes also visualization of results while the application is running, as well as its suspending and resuming.

63. Problems in the Implementation of Grid Security Services

Syed Naqvi, Michel Riguidel
Computer Sciences and Networks Department, Graduate School of Telecommunications, Paris, France

Security services are one of the most desirable characteristics of computational grids. Although these services were overlooked in the early stages of the grid evolution when the grid community was composed of dedicated computing researchers and the data was non-critical. Nowadays the swelling number of grid applications and consequent increase in the amount of critical data over grid has considerably raised the stakes for an efficient security architecture. Establishing in-depth security solutions for grid remains in its initial stages, as there are a number of impediments in the way of successful implementation of these security designs on a real grid. These problems have to be overcome in order to make the grid endeavor successful.

System developers periodically release patches to overcome the shortcomings of their previous release. These patches are generally released when some vulnerability present in their product is successfully exploited. The same practice is, however, not feasible for the grids due to the scope of the applications and the nature of the data stored over it. Certain grid applications like healthcare, where a patient's data has to be protected throughout its lifecycle, require a truly dependable security mechanism. In such applications the loss of information is irreversible and hence a well-designed security mechanism is required to persuade the already-skeptical potential users to participate in the global computing environment.

This article presents a thorough analysis of the various problems faced by the designers and developers of grid security solutions. These problems range from the non-availability of an adequate mechanism to simulate the grid security services to the grid specific constraints for the implementation of rigorous security solutions. The impact of these problems on the pace of the development of the grid security technologies is outlined and subsequently some remedial solutions are presented. Grid community's lack of experience in the exercise of the Common Criteria (CC), which was adopted in 1999 as an international standard for security product evaluation is also discussed, as the evaluation of grid security solutions requires excellent criteria to assure sufficient security to meet the needs of its users and resource providers.

NB: This research is supported by the European Commission funded project SEINIT (Security Expert Initiative) under reference number IST-2002-001929-SEINIT. The overall objective of the SEINIT project is to ensure a trusted security framework for ubiquitous environments, working across multiple devices and heterogeneous networks, in a way that is organization independent (inter-operable) and centered around an end-user. Project webpage is located at www.seinit.org

64. Processing RMI calls in the J-OCM monitoring system

Wlodzimierz Funika (1), Marian Bubak (1,2), Marcin Smetek(2), and Roland Wismueller(3)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Academic Computer Centre - CYFRONET AGH, Krakow, Poland
(3) Electrical Engineering and Computer Science, University of Siegen, Germany

Java technology has grown in popularity and usage. There are, however, performance problems which need to be identified with monitoring facilities. Remote Method Invocation (RMI) combines the problems of pure Java with those stemming from the Java distributed programming : downloading of stubs can generate a large network traffic; distributed garbage collection protocol and object serialization introduce a substantial overhead on the performance of RMI calls. Performance evaluation of RMI calls involves the analysis of the RMI structure and communication. RMI's key points from the viewpoint of performance analysis are: RMI call start/end on the client side and RMI call start/end on the server side.

From possible approaches: introducing an additional layer between the client application and the stub, using the events relating to the operation of JVM, extending Java classes, and modification of Java classes, we made use of the latter one. This method enables to monitor all key moments of an RMI call and provides information connected with RMI registry. For monitoring RMI calls we used the J-OCM which extends the OCM monitoring system that follows the idea of separation of a tool from a monitoring system through a standardized interface, OMIS. An RMI extension to J-OCM is intended to provide the notification of RMI-related

events to the J-OCM with instrumentation of the RMI key points. To realize the approach, two problems have to be solved: what functionality is to provided by the extension and how to relate the RMI Id on the client side to its Id on the server side. A set of additional services relating to RMI registry and RMI events has been introduced into J-OCM. To uniquely identify the client application with the implementation of a remote object when notifying the relevant events to the J-OCM, the client's and server's tokens assigned by J-OCM are used.

The tests of the extended J-OCM have shown an overhead for a 100 ms remote method implementation, which varies from 1 % to ca. 12 % depending on the number of calls. On a given hardware configuration, the timing of an 100-ms remote method comprises ca. 9.1 % of communication costs while the rest of time is spent in the proper remote method execution.

References

[1] 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, LNCS 1470, pp. 173-180, Springer, 1998

[2] M. Bubak, W. Funika, M.Smetek, Z. Kilianski, and R. Wismueller: Architecture of Monitoring System for Distributed Java Applications. In: Dongarra, J., Laforenza, D., Orlando, S. (Eds.), Proc. Euro PVM/MPI 2003, Venice, Italy, Sept. 29 - Oct. 2 2003, LNCS 2840, pp. 447-454, Springer, 2003

65. Quattor - Framework for Managing Grid-enabled Large Scale Computing Fabrics

Piotr Poznański (1), German Cancio (1), Rafael Garcia Leiva (2), Lionel Cons (1)
(1) CERN, Geneva
(2) Universidad Autonoma de Madrid (UAM), Spain

Quattor is a framework for managing large scale computing fabrics. It is a result of developments of the Fabric Management Work Package of the European DataGrid project.

Although fabric management components are not grid components themselves, they are essential to have a working grid. Our experience gathering requirements, developing and deploying the Quattor framework in the DataGrid testbed and the CERN Computing Centre shows that there is a real need to be able to install, configure and manage grid computing clusters correctly, automatically and supporting adaptability.

The framework provides automated and scalable configuration and installation of very large heterogenous computing fabrics. It has a modular architecture with a central configuration database and autonoumous agents running on fabric nodes. Configuration information is expressed in the high level description language called Pan. At present Unix derivatives such as Linux and Solaris are supported. Quattor has been shown to scale to thousands of computing nodes and offers a significant reduction in management costs for large computing fabrics. Quattor includes innovations compared to existing solutions which make it very useful for computing fabrics integrated into grid environments.

Since one year, the framework is used to manage more than 2500 nodes of production farms of the CERN Computer Center. It is adopted by other projects such LHC Computing Grid and institutes across Europe such as IN2P3/LAL, NIKHEF, RAL and UAM.

66. SECURE ACCESS MODEL TO GRID RESOURCES

Paweł Szychowski(1), Mirosław Kopeć(1), Marcin Lawenda(2), Marcin Wolski(2), Norbert Meyer(2), Cezary Mazurek(2), Maciej Stroiński(2)
(1) Computer Centre of Technical University of Łódź, Poland
(2) Institute of Bioorganic Chemistry PAS, Poznan, Poland

In the paper secure access model in the connection with Digital Science Library (DSL) services is presented. DLS is a part of the Virtual Laboratory (VLAB) system and base on the Data Management System (DMS) developed within the confines of Progress project. Authors focus their attention on ensuring secure connection and protected the data transfer to the local workstation. There was assumed that access interface was realized as a web portal.

Secure access to resources, data and processes is a major issue of grid systems, because such types of systems work on a public infrastructure supervised by different administrators. Scientists and the other users set meaningful store by them data what makes a challenge in an open and mobile environment. It is impossible to sure for all users' connection protection on a level of firewall rules [1].

To manage accessing to resources, proxy certificates are commonly used. The appropriate RFC standard has been issued lastly, in June 2004 [3]. It is due to, together with dynamically generated private key, replace, temporary, the original user certificate and key everywhere the authentication or authorization is needed. The packet of proxy certificate and key is called a credential.

Usually, a credential is made by a user himself. The user can use it in a next stage of grid computing and

shows the credential in every site. In this situation the user gives only one his password to his private key on the beginning. Further, using the credential doesn't require any password. So we have the SSO (Single Sign On) postulate fulfilled.

But in usual view, a user manages his credential himself. He makes it on the time period which he minds and stores it in a local repository. This approach isn't appropriate in most serious application providing costly resources. There, a right of setting access to resources should be given only granted managers. A user can only write to the manager to give him right accessing to a resource.

In the article we present such mechanism which is known as "ticket algorithm" or "token algorithm" [2]. There is a grant manager called "tool booth" which, for user request, issues credential giving an access to a resource. The credential is only generated when a designate policy lets do it. The certificate in the credential carries needed for authentication information, such as DN (Distinguish Name) of a user and optionally filled field PCI (Proxy Certificate Information) which allows relay authorization rules.

The "ticket algorithm" has been deployed inside the VLAB application particularly in the DLS subsystem. Using it VLAB users can effectively connect with data containers and download or upload data directly maximizing throughput. Thanks to security of "ticket algorithm" danger unauthorized transfer data is minor because each user has to present proxy certificate as his identity. Certainly, on the resources sites, servers, being able to interpreting proxy certificate, are required, e.g. GSIFTP or GASS servers from Globus Alliance Toolkit.

Contemporary, the algorithm is implemented in the DLS subsystem with servers in the Globus Toolkit 2.4. Presented model is compatible with the grid security concept i.e. basis on the strong cryptography and PKI infrastructure. Big advantages of this solution are authorization undeniably in data access and confidentially data transfer.

References

[1]. P. Szychowski, M. Lawenda, M. Wolski, N. Meyer, M. Kopec, C. Mazurek, M. Stroinski, "Bezpieczny dostep do uslug zarzadzania danymi w systemie laboratorium wirtualnego", Raport wewnetrzny PCSS nr RW-5/04, kwiecien 2004, Poznan

[2]. Manage credentials and access control in a grid application http://www-106.ibm.com/developerworks/library/gr-cred/?Open&ca=daw-gc-dr

[3]. RFC 3820 Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile (ftp://ftp.rfc-editor.org/in-notes/rfc3820.txt)

67. Security issues between Grid portals and Globus Toolkit 4

Pawel Plaszczak and Jakub Dziwisz
Gridwise Technologies

For non-experts, Grid portals are the only reasonable way to experience the Grid. They are great vehicles for hands-on experience for those decision makers who need understanding, but have no time for hacking the scripts and command line interfaces. However, the ease-of-use must not take priority over the security, safety and reliability of the system.

In this paper we want to share our early impressions on integrating GridSphere portal with the Globus Toolkit 4. Among security features provided by GridSphere portlet container are authentication and session management, role-based access control and user/group management. We will look at this integration project from the security standpoint, to see how these features are complementary with Globus GSI and its WS-security implementation. What else is needed to build truly secure Web-accessible Grid services?

68. Semantic Extraction and Transformation of Distributed Data in a Business Warehousing System

Maciej Zygmunt, Jakub Kozik, Krzysztof Sowa-Piekło, Jakub Sipowicz
ABB Corporate Research, Kraków, Poland

Data models are essential for managing data as a corporate asset. Without proper data models the business users and IT developers are not able to deliver comprehensive and precise representation of information structures, business rules and relationship among the data. Traditionally, in business warehousing (BW) there are several models required: starting with a subject area diagram, through a business data model, to the system and the technology models. Each of them defines different level of abstraction and transformation to provide a coordinated and fully integrated database schema. Both the data warehouses and grid systems aim at integrating data from multiple heterogeneous sources. This implies several issues related to building one consistent and reusable data model, which is fundamental for delivering information management systems. In the BW development the key step is called Extract, Transform, Load (ETL), which deals with data reformatting, reconciling and cleansing. The primary thing is the discovery of internal data relations in source systems and then building relations that are not explicitly expressed in the data model but are important to build a common data warehouse.

This paper presents an ontology-based approach to semantic data integration for building a business intelligence system supporting ABB manufacturing. It focuses on facilitation of ETL phase presenting the implementation and usage of semantic tasks in Microsoft Data Transformation Services (DTS). The Semantic Task was conceived to enable a semi-automatic extraction of a knowledge model used in a database on the basis of tables that describe "meta data" rather than any real entity (e.g. a table of types of resources). Such tables can be easily recognized, since they often have relatively few rows. OWL is used as a knowledge representation. It is expressive enough to describe object-oriented models. The intension is to integrate knowledge models of databases, rather than pure data.

The DTS semantic task presented in this paper uses our own ontology engine which is based on RDF and supports most of OWL constructs. In addition to automatic import of the source data model it allows marking of classes for importing to the model and marking classes for incremental data reading. The developer has also the ability to edit classes, their properties and instances as well as to add knowledge about the meaning of classes. A class in the model can be defined as a many-to-many relation, extension class, class table, properties table. The semantic task control (STC) can make use of any number of other models called base models that can be added, edited or deleted. Many predefined models can be used; examples can be EPRI CIM or SUMO. In the "Middle Model" the developer defines dependencies (i.e., subclass, superclass, disjoint class, equivalent class) among classes of the main model and base models. STC also supports browsing and previewing the content of source data as well as importing data into the warehouse. Additionally, having two models in OWL with specification of dependencies between classes and properties, the semantic task enables performing an automatic translation of data. As those models dependencies are described in OWL, they are browsable and reusable. What is more, due to the possibility of using subtyping it is possible to describe a partial correspondence. This approach does not eliminate a necessity of writing translation scripts, but reduces it significantly.

Experiences from the usage of the semantic task in developing a business intelligence system show that it shortens significantly the time required for building all data models required in data warehousing. It also provides better model management reducing maintenance costs by allowing discovery of source system data relations and definition of higher level abstractions with support for ETL automation.

69. Simulation of CA Models on Heterogeneous Parallel Computers

P. Jacewicz (1) and T. Zieba (2)
(1) Institute of Control & Computation Engineering, Zielona Gora, Poland
(2) Max Elektronik SA, Zielona Gora, Poland

Many of Cellular Automata (CA) modelling complex physical systems by use complex transition function, large-size lattice or both. These models have big requirements for calculation time and memory re-sources. Because states of CA cells are updated synchronously in each time step using a set of rules, is natural to use for simulation of these models, a parallel computers. Situation is easy when the computer consist of the processors with the same efficiency - lattice of the CA can be divided in proportion to number of the processors and each of them calculate new state for all cells in its part. Situation is more complicated in the case of heterogeneous parallel machine. For that case and proposed in advance methodology, the simulation slows down to the speed of the slowest processor.
In the paper we present the proposition for better use of resources in heterogeneous parallel machines. The solution consists of the analysis of calculation time measured for each of processors. Depending on time which was taken for the calculation and data transfer, the amount of cells is increased or decreased.
The proposition is illustrated by two example for some different cluster configurations and compared to the classical proposition described above.

70. Single Point of Access to Resources of HPC-Europa

Krzysztof Kurowski, Jarek Nabrzyski, Ariel Oleksiak, Dawid Szejnfeld
Poznan Supercomputing and Networking Center, Poland

The main goal of the research presented in the paper is to provide users with the Single Point of Access (SPA) to the resources of HPC Transnational Access centers in Europe. The centers include HLRS, EPCC, CINECA and CEPBA - some of the biggest HPC centers in Europe, and the research is done under the EU-funded HPC-Europa project. To this end the SPA project, led by PSNC, is building the HPC-Europa Grid Portal which will provide a transparent, uniform, flexible and intuitive user access to HPC-Europa resources hiding underlying complexity and heterogeneity.
One of the main requirements is to keep the autonomy of HPC centers allowing them to use their favorite middleware (Globus, Unicore, Web Services, etc.), local policies etc. In addition, detailed accounting information has to be stored and provided on demand for both limiting resource usage and charging purposes. We present here the main requirements and the proposed architecture of the SPA based on the GridSphere portal framework. We also discuss the interoperability issues and analyse to what extent a design of the uniform interface to diverse Grid middleware is possible

71. Stripped replication for the grid environment as a web service

Marek Ciglan, Ondrej Habala, Ladislav Hluchy
Institute of Informatics, Slovak Academy of Science, Bratislava, Slovakia

Grid technology, as an emerging paradigm for the distributed computing, requires a reliable and flexible support for data management tasks. The optimized access to the data resources, in such an environment, is needed to increase data availability and thus to increase overall performance of the whole grid system. A principle of the file replication, creation of copies of a single data source on multiple grid sites, was adopted in the grid technology as a key concept to extend the data accessibility.
When we need to transfer and process a file at a grid site, current state-of-the-art grid data management tools are able to access the cheapest replica of the required file, considering the position of requesting grid site, network load and replica site response delay. In this paper, we propose a technique for further acceleration of the replication process. We present replication method which doesn't transfer the data from only a single grid site, but is rather based on parallel transfers from multiple sites containing replicas of the desired file. From each site, in parallel, we transport only a portion of the given data source, obtaining the whole file at the end of the process.
We address the technique for file replication from multiple grid sites as a stripped replication. Using the stripped replication, we can expect time reduction of replica creation, compared to file transfer from a single grid site. However, transporting file stripes from multiple sites can eventually increase the transfer time, if the process isn't supervised and optimized. For example, if one of the file replica resides on the grid site with extremely low connectivity, attempt to transfer in parallel the identical potions of file from each site could results in decrease of overall replication speed. We propose an optimization algorithm for stripped replication management, which address the latter.
Current trend in the grid technology is to provide grid services in the form of the web services. The concept was introduced in Open Grid Services Architecture (OGSA) and was further extended in Web Services Resource Framework (WSRF). We present a prototype implementation of the stripped replication as a web service. Described replication optimization method is integrated, within the web service, with the Replica Location Service (RLS) a grid service which provides the information about data sources locations in the grid system. This allows a user to utilize stripped replication in an easy to use way, providing only the Logical File Name (LFN) of the data source which he wishes to replicate. Our implementation is made in Java programming language, it uses GridFTP as a file transportation mechanism, CoG 1.2 API libraries (provided by the Globus Alliance) as an interface to GridFTP service and RLS developed in EU Data Grid project. We discuss several interesting technical details of the stripped replication service implementation. Finally, we present results of stripped replication mechanism achieved by the prototype implementation of stripped replication algorithm. We compare them with the results of the standard replication tools and show interesting performance improvement.

72. Supporting HLA-based Distributed Iteractive Simulations on the Grid - Current Status

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

This talk presents the current status of the design of a system that supports execution of HLA distributed interactive simulations in the Grid environment. The most important components of the system are presented and actual performance issues are discussed. The archtecture is based on the Open Grid Services Architecture (OGSA) concept that allows for modularity and compatibility with Grid Services already developed.

As HLA [1] is explicitly designed as a support for interactive distributed simulations, it provides various services needed for that specific purpose, such as time management useful for time-driven ornevent-driven interactive simulations. It also takes care of data distribution management and allows all application components to see the entire application data space in an efficient way.

On the otherhand, the HLA standard does not provide automatic setup of HLA distributed applications and there is no mechanism for migrating federates according to the dynamic changes of host loads or failures, which is essential for Grid applications.

In our opinion, the Grid Services concept provides a good starting point for building and connecting independent blocks of different functionality of the HLA execution management system.

Our solution [2] introduces HLA functionality to the Grid Services framework extended by specialized high-level Grid Services. This allows for execution control through Grid Service interfaces; the internal control and data of

distributed interactive simulations flows through HLA. The design also supports migration of federates (components) of HLA applications according to environmental conditions.

References:
1.HLA specification, http://www.sisostds.org/stdsdev/hla/
2.Rycerz, K., Bubak, M., Malawski, M., and Sloot, P.: A Framework for HLA--based Interactive Simulations on the Grid in: Simulation:Transactions of the Society for Modeling and Simulation International. Special Issue: Applications of Parallel and Distributed Simulation in Industry.(accepted)

73. Testing Heterogeneous Computing Nodes for Grid Computing

Eamonn Kenny, Brian Coghlan, John Walsh, Stephen Childs, David O'Callaghan, Geoff Quigley
Department of Computer Science, Trinity College Dublin, Ireland

Grid-Ireland is working towards providing a fully remote managed infrastructure providing support to regional institutions. The national infrastructure is homogeneous and supports heterogeneous resources at each site. A variety of grid projects must be supported including CosmoGrid, CrossGrid, Webcom-G and Marine Grid. The computing resources at sites include an Origin 3840 SGI IRIX 6.5.17m and clusters running Red Hat 9, Red Hat 7.3, FreeBSD and Fedora Core 2.

EGEE have almost finished porting the current LCG2 grid implementation to Scientific Linux on 32-bit and 64-bit Intel architectures. In October 2003 Grid-Ireland began porting the LCG2/EGEE implementation to non-reference platforms: Red Hat 9, Fedora Core 2, Mac OS X 10.3, AIX 5.2L, IRIX 6.5.14 and 6.5.17m. Each module must be carefully ported preserving versioning identical to that of LCG2/EGEE. The easiest way to achieve this is to autobuild the software on a nightly basis checking out the code from the repository in CERN. Autobuilding minimizes the demand on human resources to maintain the software, and allows selective building of software components. The porting issues were found to be cumbersome and time consuming. Many issues could have been avoided if: (a) a dependency tree was provided for all packages, (b) code was POSIX compliant,(c) 32-bit and Red Hat 7.3 specific details were not hardcoded, (d) software was better compartmentalized.

The initial goal, with a suitable test-suite in mind, was to provide Globus and EDG job submission for simple single node executables on each ported worker node. This requires the porting of the VDT distribution of Globus plus a minimal set of LCG2 packages. Following this, MPI submissions and replica management must be provided to support both computationally and data intensive tasks. Since security and resource discovery by the resource broker will play a very important role in providing an efficient heterogeneous framework, VOMS and R-GMA support will be provided. Fast and clean deployment of the binaries to each worker node is desirable so a software repository plus installation scripts will be provided for each operating system. Initially the software will be installed on our testbed to ensure that deployment will be non-problematic. Once installed on the testbed the following will be tested: (a) basic Globus and EDG job submission, (b) EDG job submission using software compiled and run on the worker node (c) MPI submission, (d) replica management using the Java and later the new gfal based software, (e) basic R-GMA querying, (f) a complete set of benchmarks using scientific libraries.

EDG job submission has already been achieved under Fedora Core 2, Red Hat 9 and IRIX 6.5.14. The test-suite is under development and we present current progress.

74. The Grid Authorization Service - Internal Design, Interaction with External Services and Management

Marcin Adamski, Radosław Strugalski, Krzysztof Kurowski, Jarek Nabrzyski
Poznan Supercomputing and Networking Center, Poland

Security is one of most intensive and critical research issues in the field of Grid technologies. In order to use the Grid in practice, appropriate techniques to security problems have to be designed and efficient solutions developed. One of the most needed security tools is a consistent grid authorization service, capable of covering the complete Grid infrastructure. The development of a flexible and universal Grid Authorization Service was the primary goal of the security work in the GridLab Project.

Two mains features: the modular structure and the complicated internal data structure decide that the Grid Authorization Service (hereinafter referred to as the GAS) can be used in various grid solutions. GAS implements many different security scenarios (which use for example push or pull models). GAS is provided as a single logical point to define security policy for the whole Grid infrastructure.

One of the most important things which make the GAS a very good solution in many applications is the attitude to internal administration. This authorization service can be managed in numerous ways. A special portlet was

prepared to simplify the management of security policies stored in the GAS. There are also other administration tools available: the GUI client written using GTK libraries, and the command line client.

The core functionality of the GridLab GAS is currently ready and is utilized on the GridLab testbed. The GAS is applied to multiple scenarios with many GridLab services like: GRMS, mobile user services and the new Grid information service from GridLab - the iGrid. Based on the integration with these services we will demonstrate the way the other services or virtual organizations can benefit from the GAS.

75. The Workflow engine for the CrossGrid flood forecasting application

Martin Maliska, Branislav Simo, Ladislav Hluchy
Institute of Informatics, Slovak Academy of Sciences

During the development of the Flood forecasting application that is being developed within the CrossGrid project we have come across a requirement to execute several time consuming tasks in a single run. This requirement is caused by the fact that the Flood application consists of several simulation models - hydrological, meteorological a hydraulic. To create a workflow is good approach how to describe relationships among these models. Because the models are implemented as grid jobs, a decision to build workflow management system for grid workflows capable of running in the CrossGrid testbed has been made. First implementation of the workflow engine has been tied to a portal user interface, but now a separation has been made and the workflow engine has been implemented as a standalone grid service.
The workflow engine works with workflow templates and workflows. Each workflow consist of several jobs. Workflow templates define an order of execution of jobs and their default parameters. Workflows are concrete instances of workflow templates that may be scheduled for execution. The engine has four modules: grid service interface, workflow execution and monitoring layer, grid access layer and database access layer. Instead of using a workflow description language, we decided to simply store information about the workflow templates, their content and instances to a database.
Dependencies between the jobs are described by relationships between database tables. The jobs are configured by input and output parameters. The parameters are bounded to the resources (files, directories or variables). A sharing of resources between several parameters is restricted so only one of the parameters can be an output parameter.
Two different user interfaces have been implemented for the workflow engine in the context of the CrossGrid project. The first one is a portal interface based on portlets using the Velocity as a template engine. The second one is a plug-in for the Migrating Desktop - a Java client developed within the CrossGrid project that provides desktop environment for grid users. Because we wanted to create both these user interfaces user friendly, we decided to build them as similar as possible. Functionality of the user interfaces can be divided to operations on workflow templates and operations on workflows. Currently we support only selecting of existing workflow templates. Workflow instances can be created / destroyed and of course submitted for execution. During execution of a workflow, a state of each job of the workflow can be monitored and it is possible to stop the execution in every step.
We are working on integration of the application portal interface with the Open Grid Computing Environment (OGCE) in order to enable the cooperation of portal users by means provided by OGCE. We also need a tool that would enable a user to construct the workflows in easy way, so we are going to either integrate some of the existing tools or build our own one.

76. Towards a scalable grid ontology repository

Marian Babik, Ladislav Hluchy
Institute of Informatics, Dept.of parallel and distributed computing, Slovak Academy of Sciences, Bratislava

Using knowledge based systems in composition and execution of complex workflows is one of the challenging tasks in the recent grid projects.
Among the advantages of such approach are more flexibile collaborative environments, possibility to compose dynamic workflows, which are faster, reliable and can cover broader range of application domains as well as possibility to learn from the past experiences. Such system can enhance the possibilities of both application domain services and grid middleware services. However for such systems to work, it is necessary to share semantically rich data represented by a markup language. Such language has to describe an upper ontology for grid services and the corresponding groundings. Since the amount of the information, which has to be shared is enormous, finding distributed ways of sharing the data is indeed necessary.

In our work we present a scalable grid ontology repository system (OntStore), which unlike other ontology repositories is capable of storing and querying the ontologies and resource descriptions in distributed manner. We'll discuss the possible languages for the representation of the upper ontology and groudnings (i.e. DAML-S, OWL-S, WSMF, WSMO) and address the design issues for the proposed ontology repository. The repository is based on the distributed hash table (DHT) approach emerging in the P2P arena and well known

W3C standardized languages. The system decomposes the ontologies into the corresponding triples (subject, predicate, object) and uses the DHT system called Pastry to hash and store the elements. There are no super-peers in the system and both exact-matched and range queries can be routed. The system ensures, that queries do find the resources if the resources are available in the network. It also guarantees, that the number of routing hops for inserting a resource and resolving most queries are logarithmic to the number of nodes in the network.

Results from the experiments describing semantic data distribution and query routing performance will be shown. We'll also discuss the possible load-balancing techniques and address the decomposition of the ontologies.

The testing framework for the distributed ontology repository is based on the FreePastry and Jena. FreePastry is freely available implementation of the Pastry network. Jena is a well known java framework for building semantic web applications. We'll describe the simulation version of the testing framework, which is based on the Euclidean network topology and idealized node life. It emulates a network of nodes that are randomly placed in a plane, where proximity is based on euclidean distance in the plane. We'll also cover the distributed version, which can be implemented using either remote method invocation (RMI) or Wire. Wire is an implementation of the event-driven architecture based on sockets, and uses the non-blocking I/O support.

77. Transactional Grid Deployment

Brian Coghlan, John Walsh, Geoff Quigley, David O'Callaghan, Stephen Childs and Eamonn Kenny
Department of Computer Science, Trinity College, University of Dublin, Ireland

Grid-Ireland is based on a homogeneous infrastructure which manages heterogeneous resources. In order to guarantee quality of service to the grid sites within Ireland it is especially important that the infrastructure be consistent at all the sites. Here we present the globally applicable principles of transactional grid deployment and how we have applied them within Grid-Ireland.

We consider a case where a new release of some infrastructural grid software is incompatible with the previous release, as is often true. Once a certain proportion of the sites in a grid infrastructure are no longer consistent with the new release then the infrastructure as a whole can be considered inconsistent. Each grid infrastructure will have its own measures, but in all cases there is a threshold below which proper operation is no longer considered to exist. The infrastructure is no longer available. Thus availability is directly related to consistency. An inconsistent infrastructure is unavailable. Here we examine this subject, present some early estimates, propose a methodology, transactional deployment, that is intended to maximize the infrastructure availability, describe an implementation, and estimate the effect of the methodology on availability.

We present some technical details of our transactional deployment system. This system uses standard technologies and protocols such as Perl, PHP, XMLRPC and secure shell. Our implementation links in with a central CVS and RPM repository and provides a simple user interface to allow versions of the infrastructure to be selected and automatically deployed across the sites. Transactionality is applied to ensure that the infrastructure will make the transition between consistent states - failure to update at one site will result in all sites being returned to their previous state.

Another very important benefit of the transactional deployment system is the ease it brings to deployment. Transactional deployment allows for an automated idempotent push-button upgrade process, with no possibility of operator error. This is a major bonus when employing unspecialised staff to maintain the grid infrastructure.

Even from preliminary results, it is clear that the infrastructure has greatly enhanced availability with transactional deployment, improved from 87-93% to 99.8%, with the potential for 99.9% availability with a small amount of extra work.

78. Virtual User System for Globus based grids

Norbert Meyer, Paweł, Wolniewicz, Michał, Jankowski
Poznan Supercomputing and Networking Center, Poland

The management of user accounts in a Globus based grids, that includes many virtual organizations with hundreds or even thousands of user accounts is non-trivial. Adding user account and maintaining the grid-mapfile manually requires too much administration time, therefore some automatization is needed. From the grid economy point of view users and administrators should be able to track the resources used by user processes. The accounting is possible on a single computer system, but currently it is almost impossible to gather accounting for a whole grid environment. PSNC has introduced the Virtual User System (VUS) which meets the above requirements.

VUS is an extension of the system that runs users' jobs (e.g. queuing system, Globus Gatekeeper, etc.) and allows running jobs without having a user account on a node. This allows minimizing overhead related to creating and maintaining additional user accounts. On the contrary to other solutions, VUS assures an accurate security level achieved by user authorization and possibility of charging the user with costs of resource usage. Additionally, it respects local policy of sites and makes it possible for the local administrator to differentiate between local and remote users.

The first implementation of VUS was an extension to queuing systems (e.g. LSF) and it was successfully exploited 3 years ago in the Polish national cluster which connected several HPC centres in Poland. The current implementation is GRAM 'callout', a mechanism introduced in Globus Toolkit 3.2.

VUS extends the GT Gatekeeper and GridFTP services by the following features:
- flexible and configurable, fine-grained authorization with minimum administrative overhead. Authorization is done by querying plugins.
- virtual user accounts. There is a set on such accounts on each node, they are bound to users only for some time - there is no need to maintain an account on each node for each user!
- accounting. Once we know who and when used a virtual account, it is possible to charge him/her with costs of used resources.

VUS is used in various configurations in several national projects:
Clusterix (National Cluster of Linux Systems), SGIgrid (done in cooperation with Silicon Graphics) and international projects like GridLab(5 FP of EU) and CoreGrid (6 FP of EU).