A Training Grid Environment in VOCE

J. Chudoba, L. Fiala, J. Kmunicek, J. Kosina, T. Kouba, D. Kouril, M. Lokajicek, L. Matyska, M. Ruda, J. Svec
CESNET z.s.p.o., Praha, Czech Republic

Grids introduced a new way of information processing that utilizes many resources distributed among various organizations, in space and time. Mature middleware tools as well as complex applications have been developed recently, to efficiently utilize the power offered by the Grid systems. These achievements demonstrated that the potential of Grids is immense and a lot of various activities can benefit from their use. However, current Grids are very complex systems with a long and slow learning curve causing they are currently only used by large user communities that have enough resources for training their researchers. Therefore, the potential of current Grids is not fully utilized yet and especially small research groups that do not have resource to train their users are often discouraged from utilizing the Grids.

In this paper we describe an environment provided by the VO for the Central Europe (VOCE), which offers a generic Grid infrastructure open for all users from the Central Europe region. We will focus on training activities performed in VOCE and present a design and pilot implementation of a new infrastructure to support Grid users. This training infrastructure (t-infrastructure) is aimed to support effective training of Grid users in an environment which is as close to a standard production system as possible. We will describe all aspects of the t-infrastructure setup. Special care will be paid to identity management in such an infrastructure, in particular we will describe our solution for a certification authority that provides a formalized and traceable yet easy to use means for generating X.509 certificates.

A Universal API for Grids

Björn Hagemeier, Roger Menday, Bernd Schuller, Achim Streit

Highly dynamic features and security concerns often make it difficult for programmers to easily develop clients for Grid applications. There's a lot of overhead involved for each call to an underlying Grid resource, although many of these tasks are structured similarly. A universal, high-level API can alleviate the need to perform the tedious and repetitive tasks of accessing Grid resources and let the developer focus on the purely functional aspects of programming. The extra amount of work necessary to access a site is hidden behind a concise interface.

Another advantage of a cleanly defined API is the opportunity to implement for multiple backend Grid environments. The Roctopus API currently supports two implementations, one for Unicore 5 and more recently another one for the Web services based Unicore 6. Other backend implementations are conceivable enabling supporting other Grid infrastructures in the same uniform way.

Rather than designing a Task focused API, Roctopus takes a resource oriented approach to the definition of the Java interfaces. The linkages between the resources on the Grid are approached in a consistent manner. A resource is Locatable and possesses a Location. As such there is a generic mechanism in place for navigating the linkages. For example, a Site has a number Storages, a Storage contains a number of Files and a Site references its running Tasks. We see parallels to other systems with constrained interfaces, such as REST-based architectures, and UNIX which follows a philosophy of everything being a file. The proven elegance of these approaches is the inspiration for the design of the Roctopus API.

The API uses a Factory mechanism to get particular implementations. Multiple implementations of the API can co-exist within the lifetime of a program execution, and it encourages late binding to a specific implementation, selected at runtime. The process of implementing the Roctopus API for UNICORE 6 is examined, and furthermore the happy co-existence of implementations for version 5 and 6 of UNICORE is described.

Roctopus is a convenient toolkit for building web interfaces or command line environments for Grids. Here different underlying Grid infrastructures become browsable in a uniform way. Analysing and using Grid resources can then be as easy as browsing a UNIX file system. Furthermore, in the A-WARE project we are currently at the initial stages of using Roctopus as a basis for a Grid agent component to be plugged into a service bus, which is then used as for building higher-level orchestration services for Grids.

Adaptive, Component Based System Architecture for Monitoring Data Storing

Dominik Radziszowski, Krzysztof Zielinski
Department of Computer Science, AGH UST, Krakow, Poland

Contemporary computer systems, such as Grid, generates massive amount of information concerning their state, controlled processes and events occurring in reality surrounding them. Grid computing, emergency and telecommunications systems, sensor networks; they all, each second, are generating lots of monitoring information that has to be collected and stored. This situation is a real challenge for nowadays monitoring data storing systems, mainly in terms of universality and scalability. In this context, universality of systems is defined as an ability to collect and store any kind of monitoring data, adaptability to their changes and support for different monitoring modes. Scalability guarantee adequate efficiency while growing of incoming data stream and volume of the data already stored. Existing monitoring systems for Grid technologies like MDS (Monitoring and Discovery System), EDG NMA (Network Monitor Architecture), Ganglia, Nagios, MapCenter use traditional procedural or object oriented system model, others, like JIMS, do not support data storing. In authors' opinion such systems can be successfully build in a component architecture, using J2EE (Java 2 Enterprise Edition) technology. This approach should guarantee important, non functional system features: scalability, high availability, failover, load balancing, easy component replacement etc.

Authors made wide analysis of current monitoring systems architectures and their monitoring data models. On this basis functional and nonfunctional requirements for system USHER (Uniform Storage for Heterogeneous Environment monitoRing) have been specified in details. Two aspects are important in context of this system construction: (i) monitoring data collection and transport mechanisms, and (ii) monitoring data representation. This corresponds to two layers of the proposed system architecture. Integration on the 'communication' level can be achieved using e.g. JMX (Java Management Extension), specialized instrumentation module or protocol adapters. On the 'data' level, it requires introduction of a common data model. This model has been specified as object data model enhanced with metadata. The constructed system has two universal interfaces, identified in the requirement analysis phase: upload and query, used for storing and getting data and meta data respectively. The operations they provide relay on the proposed common data model and support interoperability with different monitoring systems.

The system offers a few types adaptability to: different monitoring environments, various resources types, possibility of dynamic resource attachment, dynamic attributes changes and different monitoring modes.

To prove realizability of assumed requirements and correctness of proposed solution, basic system model has been build using J2EE technology. The functionality achieved by the system has been positively validated. Results prove usefulness of chosen technology to creation of efficient system for massive storing of monitoring data from a wide range of monitoring systems. To make USHER available for Grid monitoring data storing, integration with JIMS monitoring system is planned in a near future.

The paper is structured as follows. After introduction, requirement analysis for adaptive, component based system for monitoring data storing is presented. Next, general system architecture is considered. In the next chapter, general data model and universal system access interfaces are described in details. As a case study system implementation in J2EE technology is presented. The paper is ended with conclusions.

An Approach to Restricted Delegation of User Rights based on the gLite Middleware

Stefan Piger, Christian Grimm, Jan Wiebelitz and Ralf Groeper
Regional Computing Center for Lower Saxony, University of Hanover, Germany

The delegation of user rights is an essential feature of Grid environments. By delegating their rights users authorize Grid services to act in their name. What hinders security sensitive communities like the medical community from using today's Grids is amongst others the unsolved security weaknesses of the implemented delegation mechanisms.
Delegation of user rights to Grid services is typically implemented by addition of authorization assertions to proxy certificates. These assertions are issued by an authorization authority like the Virtual Organisation Membership Service (VOMS) or the Community Authorization Service (CAS). VOMS and CAS are used to define VOs, roles and additional attributes and to assign them to Grid users. This information is used to define which services a user - or to be more precise her specific role in a VO - is allowed to use on the Grid. By transferring the proxy certificate and its associated - unencrypted - private key to the Grid service, the user enables the service to act in her name and with her rights. The service can now access every resource the user is entitled to use for the validity time of the proxy certificate.
While delegation is an essential feature in Grids, the use of proxy certificates poses certain risks. The risk we target in our work results from the unrestricted scope of user rights that are delegated to the Grid

services. Services normally do not require the complete scope of the user's rights for a compute job or an operation on a storage resource.
Restricting delegated rights helps minimizing potential damage caused by misuse of hijacked proxy certificates. Currently, no feasible solution to this problem is provided in Grid middleware.
In this paper we present an approach to limited delegation of user rights by use of restricted proxy certificates. We describe the policy language and the language elements our approach is based on, as well as two services that we have enhanced to police service access. Our implementation is based on the gLite Grid middleware.

The policy users define to restrict their delegated rights is included as a non-critical extension in the user's proxy certificate in addition to those created by VOMS. For this, we created a policy extension for non RFC 3820 proxy certificates as are used in gLite. The policy information itself is encoded in a XACML 2.0 data structure. The definition of the policy and its inclusion in the proxy certificate is done entirely in user scope without relying on centralized services. To encourage the acceptance of our approach we aimed to design this workflow in an effective and user-friendly way.

gLite based Grid environments currently rely on the IO-server and the Fireman Data Catalog service for global file access authorization. Our approach extends the IO-server by specific file access authorization based on user defined policies in the proxy certificate as described above. These policies carry information about the files and directories the user authorizes services to access in her name. For this, we modified the IO-server by adding a PDP that evaluates these policies and a corresponding PEP that enforces the decision of the PDP.
To enhance the security of the computing services in gLite we implemented a PDP for the Computing Element that evaluates these policies and decides about granting access to computing resources. The PDP is implemented as a plug-in to the authorization system LCAS and makes authorization decisions based on the unique job identification string (jobID) carried by each job. This jobID is created at submission time of each compute job by the UI and is registered at the Logging and Bookkeeping (LB) service. It is based on a URL like structure that contains the DNS entry of the LB service and a hash string. The authorization decision is made by comparing the jobID provided by the Grid middleware within the job and the policy extension inside the proxy certificate. The policy holds the ID of the job the user authorized to be executed. Thus the proxy certificate is bound to this specific job and cannot be misused to authorize the submission of additional jobs.

An Extension of Globus Toolkit for Grid Computing Optimization

Anatoliy Doroshenko and Konstantin Rukhlis
Institute of Software Systems of the National Academy of Sciences of Ukraine, Ukraine

The use of the Grid computing platforms is of growing spread now because this kind of systems allows using many types of computing systems to solve the most complicated problems in science and engineering. Furthermore Grid platforms can be used in heterogeneous and unstable networks where classical MPI-like platforms is ineffective because of the lack of monitoring tools, transactions support, tasks management and low security level. However most of the Grid platforms suffer from the lack of the service mobility, service directory subsystems, qualitative task scheduling and resource management, etc. For example, the absence of monitoring facilities leads to distribution of subtasks that is unaware of node performance and current throughput of communication channels, and service immobility results in the need of prior copying them and installing manually.

In this paper an approach is proposed to extend Globus Toolkit platform in OGSA-compatible manner with the aim to improve performance of parallel grid applications. More precisely, there is developed an extension with automated distributed grid service deployer, resident node performance measurement agent, directory service and application bridge for interoperability with non-grid applications. The approach is implemented as a framework on the base of Globus Toolkit, version 3.2.

The paper consists of three parts. In the first one there is an introduction where Grid platforms are analyzed, Globus Toolkit facilities are introduced and its main drawbacks are described.

In the second part there main mechanisms for the extension are presented including changes in the Grid nodes interaction process and new job execution workflow.

In the third part, the methodology for application of the developed framework to real computational tasks is illustrated with a sample task application. Comparative results of computational experiments with standard Globus Toolkit and the extended platform are provided.

Application of Pricing Engine for Electrical Energy Implemented for a Grid Environment

Gregor Pipan(1), Bostjan Slivnik(2), Jaka Mocnik(1), Uros Cibej(1), Marko Novak(1), Borut Robic(2)
(1) XLAB Research, Slovenia
(2) University of Ljubljana - Faculty of Computer and Information Science, Slovenia

The Europe is coming closer together, and therefore the need for collaboration arises in many different fields. One of them is Market of electrical energy. Even though the electricity itself is only a simple sine of voltage and current, there is no simple way to establish a marketing environment, which would allow a fair trade. In order to achieve this goal, each user (buyer or seller) has to be presented with the best possible price for a product. This price consists of energy costs, transfer costs and taxes. The problem presented is computationally demanding due to the fact, that each deal changes the whole environment (loads on the links and available products), and therefore has to be constantly re-computed. Because of this reason we have developed this solution for a Grid environment, which is able to re-compute the prices for all users in acceptable time. The paper consists of three parts; theoretical, implementation and results.

Theoretical part will describe the mathematical model used. The base of the model is planar graph, where the nodes present regions, and edges power-links between them. In each node we have offers and requests, which have to be interpreted in all the regions with the appropriate price, which depends on the price of the energy (originating offer), distribution matrix, which define the additional load for each link in the system, by given origin and destination node. After defining the mathematical model, we will determine the complexity of the computation, and will show, that the problem defined by this model is NP-hard, and in addition no approximation algorithm exist for it.

Then we will proceed with describing the implementation of the algorithm, which must comply with two requirements. First requirement is given by the nature of the application, and requires that algorithm is determinate, and second, that the computation cycle of calculating the prices does not require more than few seconds (2-4 seconds) of the wall time. The algorithm consists of several heuristics, which are used to reduce the solution space, and therefore provide a determinate algorithm, which is still fast enough to provide a solution in a requested time.

For Grid deployment a trivial parallelisation is used, which divide the given problem based on the natural cutting points, which are regions and products. We will also propose a possible parallelisation, which could be used, if required and enable division of the problem beyond the natural division points. The paper finishes with the results of execution times of different parallelization approaches, which have been described.

The paper concludes with a proposal for a generalisation of the algorithm, in order to be applied to other areas of interest.

Attaching Dynamic Clusters to CLUSTERIX Grid

J. Kwiatkowski, M. Pawlik, G. Frankowski, R. Wyrzykowski, K. Karczewski
Institute of Applied Informatics, Wroclaw University of Technology, Poland
Poznan Supercomputing and Networking Center, Poland
Institute of Computer and Information Science, Czestochowa University of Technology, Poland

The increase of computer networks speed paired with the ubiquity of inexpensive, yet fast and generously equipped hardware offers many organizations an affordable way to increase the available processing power. Clusters, hyperclusters and even Grids, not so long ago seen only in huge datacenters, can now be found helping many small organizations in solving their computational needs. CLUSTERIX is a truly distributed national computing infrastructure with 12 sites located across Poland. The computing power of the CLUSTERIX can be increased dramatically by connecting additional clusters. These clusters are called dynamic because it is assumed that they will be connected to the core infrastructure in a dynamic manner, using an automated procedure.

Several conditions should be satisfied to provide the attractiveness of the dynamic cluster concept for its potential users. First of all, the attachment and detachment procedures should be simple and automated. After the initial fulfillment of conditions necessary to provide integration of a dynamic cluster (installed software, initial verification of the cluster performed only once), its operator should be able to attach and detach the cluster automatically by calling a single command. It requires to develop the attachment and detachment procedures invoked as a reaction to this command. The unified architecture of local clusters in the core has been tailored to implement this functionality in an efficient and secure manner. In particular, each local cluster is provided with a dedicated firewall/router whose public network interface is the only access point to the external network. This solution allows for a balanced implementation of the attachment procedure giving the possibility to choose the most appropriate local cluster to establish connection.

Every dynamic cluster can be connected to one of 12 ports corresponding to local clusters in the CLUSTERIX core. The monitoring system, installed on a dedicated node, is responsible for deciding which port the dynamic cluster will be connected to. So at the beginning, the access node of the dynamic cluster should contacts this node. The monitoring system notifies the chosen local cluster about the emerging dynamic cluster, and transfers to the dynamic cluster the public IP address of the access node in the core. According to principles of IP addressing adopted in the CLUSTERIX project, each local cluster possesses a certain subclass of private addresses from 10.x.x.x pool. Inside this subclass, we distinguish one subclass for the computational network of the local cluster, and 16 separate subclasses for dynamic clusters. This means that maximum 16 dynamic clusters may be integrated with a given local cluster at the same time. The procedure of detaching the dynamic cluster is much simpler. The dynamic cluster must inform the monitoring system, which in turn invokes a set of configuration steps in the firewall assigned to the dynamic cluster.

BackupGRID: Using Desktop Nodes to Provide a Grid Storage Service

Filipe Araujo(1), Patricio Domingues(2), Radoslaw Januszewski(3), Gracjan Jankowski(3), Rafal Mikolajczak(3), Luis Moura Silva(1)
(1) Dep. Engenharia Inform´atica, Univ. Coimbra, Portugal
(2) School of Technology and Management - Polytechnic Institute of Leiria, Portugal
(3) Poznan Supercomputing and Networking Center, Poland

Although usually regarded as cumbersome and timestealing tasks, the importance of backups is only fully appreciated when they are needed, that is, after an hardware mishap (e.g. disk crash), a software failure (e.g. file system corruption) or a user mistake (e.g. wrongly overwriting. In recent years, the advent of the Internet has permitted many online commercial backup solutions. Examples include Connected , XDrive and LogMeIn, just to name a few. All of these solutions are centralized, with the backup providers managing large arrays of dedicated storage devices. In the first part of this paper we present BackupGRID, a middleware system aimed to provide a wide-scale grid service for data backup, harvesting the terabytes of unused disk storage that can be found in desktop computers connected to the Internet. The proposed system aggregates nodes into storage pools, organized within institutions, companies, or by informal groups of users. We assume that the storage pools offer their backup services in exchange of some goods such as money, services or access to multimedia content . Our system relies on a highly decentralized architecture, where clients send requests directly to storage pools, either to backup or to restore files. The only centralized component of our architecture is a trust manager server, which has the following purposes: authenticate users, manage a web service that lists the available storage pools and manage the public evaluations of the storage pools. The amount of information stored in the trust manager server and the traffic that goes through it is quite limited. In this way, our system can scale to an almost arbitrary number of storage pools and clients and yet uses a very simple trust management scheme. One interesting aspect of BackupGRID is that clients need to incur expences to use the services. In this way, we eliminate one of the greatest problems in peer-to-peer systems, which are the free-riders and, at the same time, we give a strong motivation to users having some free space in their computers to make those space available. In the second part of the paper the proposed system will act as a model for further deliberations on specific aspects of storing data. From our point of view the most interesting aspect is to study the feasibility of using the peer-to-peer systems such as BackupGrid for storing the images of checkpointed applications in distributed computation environment. As the storing and accessing images of applications implies some requirements and differs significantly from other patterns of using the backup infrastructure this is a good opportunity to study the feasibility of using peer-to-peer solutions for checkpoint purposes.

Combining a Virtual Grid Testbed and Grid eLearning Courseware

Kathryn Cassidy, Jason McCandless, Stephen Childs, John Walsh, Brian Coghlan, Declan Dagger
Department of Computer Science, Trinity College Dublin, Ireland

Training activities in realistic settings are important contributors to successful eLearning. For Grid courseware, however, training users on a live production infrastructure is not ideal. The bursty nature of group training activities can affect throughput of real production jobs. Similarly, a heavily-loaded live infrastructure with lengthy response times can cause delays in processing even simple example training jobs; the delay is disruptive for students and can give the impression that the system is unreliable.

A dedicated teaching environment which closely replicates the live infrastructure would be a more desirable solution, but, physically replicating a self-contained subset of a grid infrastructure is not feasible. The advent of high-performance, user-friendly, non-commercial virtualisation technologies for commodity off-the-shelf computers, however, has enabled researchers to build non-trivial virtual testing and testbed environments. In addition, advanced networking techniques enable the networking component to be extended to present a

realistic replica of the network of an Grid infrastructure, whilst providing some limited connectivity to the real production network.

We show how the priciples of virtualisation have been applied to develop a teaching environment that realistically simulates the workings of a production EGEE Grid. The simulation gives the student the look and feel of using the real environment, without impinging on production resources. We outline how our tool, GridBuilder, aids site administrators; enabling them to quickly and easily create, manage and destroy whole grid sites or just single nodes that are preconfigured with the Grid middleware and a user-selected site profile. As well as simplifying the process of creating and managing sites, this can enhance the administrator's understanding of how the various Grid services fit together.

We also show how the Adaptive Personalised eLearning Service (APeLS), suitably extended with Grid security, can deliver adaptive courses to users with live exercises which they can perform in the virtual teaching environment. We give examples of how the adaptivity to the user's preferences and abilities can be used to enhance both a basic (introductory) and an advanced (R-GMA) Grid course.

Keywords: Grid, eLearning, Education, Training, Virtualisation, Testbed

Comparison of Grid Accounting Concepts for D-Grid

Gabriele von Voigt, Wolfgang Müller, Claus-Peter Rückemann
RRZN University of Hannover, Germany

This paper gives an overview of current grid accounting approaches and evaluates them in the context of the German grid initiative (D-Grid). In order to do so, the advantages and disadvantages of the most used accounting approaches, namely the Accounting Processor for Event Logs (APEL), Distributed Grid Accounting System (DGAS), Grid Accounting Services Architecture (GASA/GridBank), Grid Based Application Service Provision (GRASP), Grid Service Accounting Extensions (GSAX), Nimrod/G and SweGrid Accounting System (SGAS), are determined.

The development of the definition of the D-Grid accounting architecture has to take into consideration conceptual, technical and legal questions and to consider special requirements, resulting from the virtualisation of organisations and resources.

Technical criterions are:
-- determination and/or definition of generally obligatory calculation units,
-- delivery of calculation units,
-- allocation of calculation units to users,
-- customized presentation of resulting fees,
-- supply a multiplicity of users, both from the scientific range and from the economy,
-- services will be offered by a multiplicity of providers.

The evaluation of the accounting approaches was based on one hand on a comprehensive survey within the German Grid infrastructure, including grid communities and thus potential users as well as computing centres and thus resource providers. On the other hand technical requirements were taken into account, which were given by the D-Grid infrastructure.

The four approaches SGAS, GASA, DGAS, APEL inherit the most promising concepts, but also within these four, there is a slight advantage for SGAS. SGAS has its special strength in interoperability, ability for integration, portability, accounting beyond one community, supporting standards, security, fault tolerance, precision, administration, maintainability and verification, where as APEL is able to handle the customer specific accounting data as well as supports various metrics, and especially GASA and DGAS have an advantage in the possibility of the accounting and pricing of heterogeneous resources. The exact comparison has been described in detail [1]. This has been the basis for the development of a D-Grid accounting architecture [2] and the concept of the D-Grid accounting system, which is currently under way.

[1] Rückemann, C.-P., M. Göhner, 2006: Bewertung bestehender Accounting-Ansätze. D-Grid, Fachgebiete Monitoring, Accounting und Billing im D-Grid-Integrationsprojekt, 2006. 16 Seiten. http://dgi.d-grid.de/index.php?id=118&filename=mab_accounting_bewertung.pdf&dir=FG2/koordination_mab&task=download&mountpoint=2
[2] Rückemann, C.-P. und M. Göhner: Konzeption einer Grid-Accounting-Architektur. Fachgebiet Accounting (FG 2-7) im DGI, August 2006, D-Grid, Fachgebiete Monitoring, Accounting und Billing im D-Grid-Integrationsprojekt, 2006. 28 Seiten.
http://dgi.d-grid.de/index.php?id=118&filename=mab_accounting_konzeption.pdf&dir=FG2/koordination_mab&task=download&mountpoint=2

Data Grids: A Collaborative Semantic Model with Hybrid Namespace

Dalia El-Mansy (1), Ahmed Sameh (2)
(1) Department of Computer Science, The Southern Methodist University, Texas, USA
(2) Department of Computer Science, The American University in Cairo, Egypt

The Data Grid, like all other collaboration models, has strict rules for contributors to follow and many criteria to abide with. Namespace is one of the rules that govern the contributor. Usually, the namespace, or any other globally enforced rule, is stored and maintained centrally to be available to all contributors. Simple solutions like mirroring are adopted to avoid potential bottlenecks.

Some fields are not ready for abiding with such kinds of global rules. For instance, scientific research taxonomy (down to topics and areas of interest) is highly dynamic. Topics are confusingly interdisciplinary and sometimes have institutional influence. Still, researchers need to collaborate with others in other institutional environments.

The intention is to design a hybrid namespace collaboration model for existing organizations to pay lower cost (initial and running) in order to join it. Data resources will be allowed to have colliding names. The contributing organization will be free to set any rules for its internal users to follow (such as taxonomy). However, it will have to take care of a very simple interface to the intended model in order to introduce new contributed resources or to query external ones. Traditional data grids will join this model (represented by proxy agents) to provide their content while their owners can join as users of relevant servers to explore relevant resources to keep their content up to date.

A hierarchical namespace will be maintained in order to uniquely identify data resources. (Full names are the local names prefixed with the ID of owner data grid for uniqueness.) If the full name is queried, the corresponding resource is sought in the wide group of collaborating data grids. If the local name is queried, it is fetched in the local data grid. Similar and related topics in different places in the hierarchy will be linked for robustness. Otherwise, a comprehensive search should be performed all over the whole hierarchy.

Defining and Running Parametric Study Workflow Applications by the P-GRADE Portal

P. Kacsuk, G. Sipos, A. Toth, Z. Farkas, G. Kecskemeti and G. Hermann
MTA SZTAKI, Budapest, Hungary

The P-GRADE portal plays more and more important role in the life of various Grid user communities. After several successful demos at the biggest conferences and Grid user forums in Europe, Asia and the US, the representatives of several Grids and Grid based Virtual Organizations have approached and requested to support their communities by the Portal. As a result, the P-GRADE portal is already the official portal of the VOCE (Virtual Organization Central Europe), HunGrid (Hungarian VO of EGEE) and the eGrid (Economic Grid) VOs of the EGEE Grid. It also provides service for the users of GILDA (the Grid training infrastructure of EGEE), Croatian Grid and Turkish Grid infrastructures. Moreover, the P-GRADE portal is the official portal of SEE-GRID which operates a Grid infrastructure in the South-East Europe region. Besides LCG-2 and gLite based production Grids the portal is successfully used as service for the GT2 based UK National Grid Service (NGS) and it was also successfully connected to the GT4 based Westfocus Grid (UK). Our latest achievement is that the P-GRADE Portal has been connected to the ARC middleware, thus now it is able to execute Grid applications in the Nordugrid too. After a successful demonstration at the Supercomputing'05 exhibition the representatives of the US Open Science Grid and Teragrid also expressed their interest to connect the portal to their Grid. Consequently, the P-GRADE Portal is now connected to both OSG and Teragrid, reaching the users of many large production Grid infrastructures of the World. Moreover, recently the GIN (Grid Interoperation/Interoperability Now) Grid of GGF is supported by the portal enabling the simultaneous access to all of its resources coming from different Grids.

As P-GRADE portal becomes more and more popular among users we have received important feedbacks asking for new features of the portal. One of those requests was the support of parametric studies at the workflow level. The idea of parametric study applications is that the same workflow should be executed with a large number of different files. Moreover, different input ports must be fed by different number of files and

the portal should be able to automatically generate the cross-product of these input files and run the workflow for each element of this cross-product. Obviously handling large number of workflows and files raises many new problems to be solved. Here we mention only some of them just to illustrate the problems:

1. Where to place the necessary input files?
2. Where and how to store the output of each run of the workflow?
3. How to prevent flooding the Grid and the portal by a single user?
4. How to specify the parametric study workflows in a way that simply extends the specification of simple workflows?
5. How to manage the large number of workflows by the portal?

The main principles of supporting parametric study application by P-GRADE portal are as follows:

1. Any port of a PS-WF (parametric study workflow) can be used to feed many files to the WF. Such a port is called as PS-port and distinguished from the ordinary input ports both in the UI and in the inner representation of the WF. For each PS-port in a PS-WF there is a unique integer identifier (an ordering number starting from 0) generated by the portal.
2. A PS-port represents a set of input files that are stored in the same directory of a SE (storage element). It is the responsibility of the user to place these input files into the SE before submitting the PS-WF. Such a directory must not store any other files, only the input files belonging to the associated PS-port.
3. If there are several PS-ports in a PS-WF, then the portal RS (run-time system) takes care of producing the necessary cross-product of the input files of these PS-ports.
4. For each element of the cross-product the RS generates an executable WF (e-WF). The internal representation of an e-WF is the same as the normal WF.
5. Once the RS generated an e-WF it submits this e-WF in the same way as normal e-WFs are submitted (since they are the same).
6. The number (N) of e-WFs that are generated for parallel execution is the decision of the portal. When a PS-WF is submitted, the portal RS generates N e-WFs of the cross product and submit them simultaneously to the Grid. Once an e-WF is completed the portal RS generates the next element of the cross product and the related e-WF and submits it into the Grid.
7. An extra global parameter of a PS-WF is the target output directory of the workflow results. The target output directory must be in a SE.
8. Once an e-WF is completed the portal moves the zipped result into the target output directory. As a result not more than N partial WF results should be stored on the portal simultaneously for one PS-WF. Any post-processing of the results is the task of the user and not of the portal.
9. To avoid the flooding of the Grid and portal by a single user, one user can submit only one PS-WF at a time. The next one can be submitted if the previously submitted PS-WF is completed. Moreover, the portal administrator can set the maximum number of e-WFs that can be simultaneously generated from a single PS-WF as well as the maximum number of jobs that can be submitted by the portal to the Grid.

The paper will describe in detail how the new PS support feature of P-GRADE portal has been designed and implemented. Both the user interface and the run-time support system will be explained in the paper.

Description of a Lightweight Bartering Grid Architecture

Cyril Briquet, Pierre-Arnoul de Marneffe
Department of Electrical Engineering and Computer Science, University of Liege, Belgium

Automated resource exchange and negotiation between participants of a Virtual Organization, or Peers of a Peer-to-Peer Grid, is an important feature of Grid computing because it enables scalable cooperation between entities under separate administrative control. Automated negotiation and accounting of resource consumption have been studied, and market-based resource exchange methods have been proposed. However, there currently exist few simulators of resource exchange accounting or actual Grid middlewares supporting automated priced negotiation between separate entities.

To this end, we propose a Lightweight Bartering Grid (LBG) architecture suitable to the development of Peer-to-Peer Grids with an automated priced resource exchange capability. Bartering is a decentralized, non-monetary, market-based resource exchange method that has recently been shown to be appropriate for automated priced resource exchange between Grid Peers. Only Bag-of-Tasks Jobs are currently considered, and resources are exchanged between Peers to execute work units at the Task level.

We present the LBG architecture as well as a functional simulator and a middleware under development that both instantiate it. The main components of a Peer are a set of loosely coupled managers that take care of the different aspects of operating a Peer: Request Manager (that enqueues requests from Peer users as well as from other Peers), Resource Manager (with an accurate representation of the resources owned by the Peer and a modeled representation of the resource capacities that can be supplied by other Peers), Task Manager (that ensures Task completion), Negotiator and Scheduler.

The actual implementation is realized in the Java environment, and care was taken to let it architecturally open to different accounting, negotiation, scheduling and queuing algorithms. Once an algorithm is coded in Java and implements the expected interfaces, it can be simulated with accuracy. An interesting feature is that the code of the Peer managers is used by both the simulator and the middleware. Therefore, any version of a simulated algorithm is also immediately available for use in the middleware without any modification.

Development of a Billing Framework for D-Grid

Wolfgang Müller, Claus-Peter Rückemann, Gabriele von Voigt
RRZN University of Hannover, Hannover, Germany

This paper contains the description of the development of a transparent and comprehensible billing framework for grid-computing within the German grid Initiative D-Grid. The vision is to establish an appropriate billing solution for the mutual provision and sustainable usage of grid resources. Therefore, a billing framework for D-Grid has been specified with special interest to the synergy for science and economics.

Within the setup and administration of a grid infrastructure, participants want to share in a fair way various resources (such as hardware, software) as well as the required resource management (such as maintenance and administration). To realise this basic idea of grid-computing in a commercial context, it is necessary to establish a comprehensive billing solution. For the owners of grid resources it is essential to:
- generate incentives for the constant provision of grid services to meet the needs of grid resource consumers,
- enabling them to quest offers of resources of other providers when their own capacities are insufficient,
- allowing them to maximize their resource utilization by offering a competitive service access price in order to attract consumers, and as a consequence,
- reducing their total costs of ownership (TCO).

On the other hand, in a competitive and transparent market the users (resource consumers) have the option of choosing the providers that best meet their requirements or have the option to choose between rental or self-procurement of computing resources.

Accounting data are the basis for the realization of a billing-framework. This includes accounting and prognosticating of resource usage data about hardware-, software-, distribution- and transmission costs for grid resource providers (GRP) and grid resource consumers (GRC).

Currently there is no standardized procedure for accounting and especially for billing in existing implementations. There exist only a few prototype implementations of a Grid billing solution, for example the "GridBank/GASA" framework in the context of the Australian GRIDBUS project, the "SweGrid Accounting System" (SGAS), the "Data Grid Accounting System" (DGAS) or the "Metadata Catalog Service" from the UK e-Science program [1]. None of the investigated implementations fits to all communities or could be easily ported into an open D-Grid environment. Therefore a billing framework for D-Grid has been specified [2]. The major features are the ability to handle various currencies, the existence of clearing institutions, the necessary security, the mechanisms of price building and the interaction between user, resource broker, resource provider and clearing institution, the payment after job submission as well as the procedure of settlement.

References:
[1] Müller, W., Falkner J., Mucha M. 2006: Billing-Anwendungsfälle und Anforderungen, FG2-8 Billing im D-Grid. In: Rückemann, C.-P., (Herausgeber): Ergebnisse der Studie und Anforderungsanalyse in den Fachgebieten Monitoring, Accounting, Billing bei den Communities und Ressourcenanbietern im D-Grid. Fachgebiete Monitoring, Accounting, Billing im D-Grid-Integrationsprojekt, 1. Juni 2006, D-Grid, Deutschland, 2006. 141 Seiten. Seiten 99-112. http://dgi.d-grid.de/index.php?id=118&filename=mab_studie_ergebnisse.pdf &dir=FG2/koordination_mab&task=download&mountpoint=2
[2] FALKNER, J., W. MÜLLER und M. MUCHA: Erste Konzeption des D-Grid-Billing-Frameworks. Fachgebiet Billing (FG 2-8), DGrid, Fachgebiete Monitoring, Accounting und Billing im D-Grid-Integrationsprojekt, 2006. 19 Seiten. http://dgi.d-grid.de/index.php?id=118&filename=mab_billing_framework_vorlaeufig.pdf &dir=FG2/koordination_mab&task=download&mountpoint=2

Distributing a n-body Problem Algorithm at Large-Scale over a Multi-Sites Grid using JavaSpace

Virginie Galtier
Supélec, Metz, France

Sun introduced JavaSpace, a grid middleware inspired by the Linda system tuple-based model. It is proposed as a Jini service, enabling Java programs to exchange objects through a virtual shared memory. Object retrieval is done by associative lookup. The API is both simple, yet rich enough to enable fast and easy development of a large range of distributed applications. Communications between client and services instances use Java remote method invocation (RMI) and Java serialization. Previous evaluation studies have concluded that because of those relatively inefficient underlying communication mechanisms, JavaSpace was best suited to distribute applications with a high computation-to-communication ratio. In contrast with those works which were mostly conducted on a reduce number of geographically-close nodes or micro-bench oriented, our objective is to evaluate the performance gains obtained when distributing a real and complete long-range data interaction application, on a large number of processors both on clusters and multi-sites grids.

As a test application, we choose to distribute the n-body algorithm. This classical problem, which finds applications in areas such as astrophysics, molecular dynamics or plasma physics, consists in finding the future positions of N bodies exercising mutual attraction. When all pairwise forces are computed directly, each time step requires N(N-1) operations and each body computation unit needs to communicate with the N-1 others, which contributes to make the n-body algorithm an interesting candidate for our study. To reduce the computation time demanded by each step, the algorithm was distributed over P machines according to the following strategy: the master places P groups of N/P bodies with their initial position and speed in the JavaSpace and puts each of the P workers in charge of a group. At each step, a worker reads from the JavaSpace the P-1 other groups for the previous step, it computes the new positions of its own group and writes the updated group in the JavaSpace. At each step, a garbage process gets rid of intermediate results as they become useless. This implementation, which arranges the bodies into P groups rather than N individual bodies, performs an optimization of the number of communications between a worker and the JavaSpace and eases the matching process when retrieving an object from the JavaSpace. This optimization supposes an overall homogeneity of the involved nodes performance, hypothesis that reasonably stands in the case of a production grid (as opposed to a desktop peer-to-peer grid).

Our testbed consists in nodes from Grid'5000, a French research grid platform gathering 1665 nodes from 14 clusters distributed over 9 sites across France. We've tested our implementation with up to 391 workers, located within the cluster hosting the JavaSpace service or distributed on 1 to 5 remote sites (preliminary tests showed that for this application on Grid'5000, the network was not a bottleneck: 4 workers per site on 5 remote sites lead to similar performances as 10 workers per site on only 2 sites for instance). We've first found out that distributing the algorithm using JavaSpace was easy. It was as well efficient: good speed-ups (close to ideal) were measured. Yet, the optimal number of processors to use appears to be quite sensitive and the speed-up quickly degrades over that limit. Our experiments revealed a simple empirical law to adequately choose this number depending on the problem size: knowing the number of workers that achieves a 90% efficiency for a problem of given size N, to maintain that level of efficiency when N doubles, the number of processors must double as well. This law is useful for an optimal usage of a shared or leased grid. Lastly, our scalability measurements show as well that the theoretical execution time (O(N^2)) can linearized (O(N)) provided enough resources are available.

Dynamic Firewalls and Service Deployment Models for Grid Environments

Gian Luca Volpato, Christian Grimm
RRZN - Regional Computing Centre for Lower Saxony, Leibniz Universitaet Hannover, Germany

Firewalls with solid implementation and correct configuration block malicious and unwanted traffic while allowing safe and desired communication. Accurate security policy definitions and consistent firewall configurations allow benign applications to seamlessly traverse firewalls. However there may be situations where such applications are nevertheless blocked by inbound/outbound access control policies. This may happen when firewalls are overzealous in blocking malicious code (filtering rules are too restrictive) and when firewalls are based only on static configuration rules that do not accommodate the specific requirements of novel applications.

In contrast to the classical client-server paradigm that is mainly supported by today's firewalls, the future Grid environments will demand more and more often the ability to:
- start connections from hosts located outside of the protected network;
- establish connections from/to ports not known in advance to the firewall.

The first requirement calls for firewalls to allow incoming traffic from possibly unknown sources. The second

requirement calls for firewalls to dynamically open ports (for incoming or outgoing traffic) and close them as soon as they are no longer necessary.

To overcome, at least partially, these limitations we propose and analyze two completely different, yet complementary approaches.

The first method is based on the extensions and modifications of firewall implementations, as proposed by different Grid research communities, with the purpose of enabling and simplifying traversal by authorized applications. From a logical viewpoint a so called "dynamic firewall" should implement an access control mechanism such that the protected network appears to be completely closed to external systems, but internal Grid services still responds to trusted clients. In this way port probing and other network mapping attacks would be unsuccessful.
We compare in this paper three proposals for the implementation of a dynamic firewall. Dyna-Fire and CODO (Cooperative On-Demand Opening) have a similar architectural model, based on the introduction of a signaling protocol between applications and firewalls. GCB (Generic Connection Brokering), as the third proposal, introduces instead a new component in the network, the connection broker. In our preliminary evaluation CODO appears to be the most promising solution. However, it was not designed with the purpose to fit all the particular requirements of a Grid environment. The most important drawbacks of CODO are here described, together with suggestions for further improvements.

As a consequence of the partially dissatisfying implementations of dynamic firewalls, we introduce a second approach toward a firewall-friendly deployment of a Grid infrastructure. Our idea proposes a homogenous and consistent deployment of Grid services across all sites participating to a Grid project. Given the large amount of middleware installation options, each one demanding its own specific firewall configuration, an easy and secure deployment is practicable only when all entities involved in the project are coordinated by a central board with which they agree on a single deployment model.
In our discussion we choose Globus Toolkit 4 as the middleware for the implementation of a Grid infrastructure. We intend to focus our attention on the impact of the different deployment options on the configuration of the firewalls located at the Resource Centers. The outcome of this analysis is a set of reference models for firewall-friendly Globus deployment at Resource Centers and Virtual Organizations.

Economic Virtualization of ICT Infrastructures

Jochen Stoesser, Arun Anandasivam, Nikolay Borissov, Dirk Neumann
Institute of Information Systems and Management, Universität Karlsruhe, Germany

The management of ICT infrastructures is facing tremendous challenges. In the past, organizations built up powerful infrastructures driven by "one-application-one-platform" style of development. However, these powerful computing resources are confronted with low system utilization, high costs of capital, and an increasingly dynamic environment. Grid Computing provides a promising approach to solving this dilemma. By "virtualizing" the underlying ICT infrastructure, the Grid aims at offering organizations high-quality ICT services at low costs.

The Grid creates a paradigm shift in ICT resource management by transforming computing resources from being a costly and inflexible asset to a dynamic service which is used and paid for if required. This leads to the new business model of "utility computing" with specialized so called "utility providers", built on economies of scale and scope, selling these ICT services "on demand". The currently most prominent approach to pricing these "utilities" is the subscription model where the requesting organization subscribes to a service as specified in so called service level agreements (SLAs) and is charged periodically for its usage. However, the subscription model is not fully able to achieve efficient resource utilization since the actual demand for Grid resources will generally not coincide with the service levels as specified in the SLAs and thus charging is somewhat independent from the actual usage. Furthermore, due to the long timeframe of this contracting scheme, both organizations with temporarily idle resources as well as organizations with fluctuating demand will be unwilling to participate in the Grid. A further possible source of inefficiency are the central scheduling algorithms which are currently used to solve the logistic problem of deciding what ICT resource should be assigned to what task or request at what time in order to maximize resource utilization. These central scheduling algorithms have problems when organizational boundaries are crossed and information about demand and supply are manipulated.

In this paper we propose market mechanisms that will enable the establishment of an economically sound and technically feasible Open Grid Market. In this Open Grid Market, Grid resources can be traded spontaneously to achieve an efficient allocation of given resources. In contrast to the subscription model with temporarily inflexible SLAs, the Open Grid Market will apply the so called metering pricing model where organizations only pay for their actual usage. Consequently, the Open Grid Market avoids the potential drawbacks of the subscription model and SLAs and encourages organizations to provide their idle resources to other organizations in the Grid in return for the payment of the market price. Market forces take over the

task of establishing an efficient allocation of user demands to resources and vice versa, provide incentives to contribute to the Grid, and hence increase overall utility.

The design of the market mechanisms is a crucial task in moving Open Grid Markets to their full potential. Current market approaches for Grids often rely on single-sided auctions (e.g. English auction) where one resource owner offers Grid resources to multiple requesters. This mechanism will generally not lead to an efficient allocation of resources between owner and requester. On the supply side of the Grid market, demand and thus liquidity is split up between the several auctions and resource owners respectively, because requester will only concentrate on certain auctions and will not consider all possibilities. On the demand side, resource requesters face the so called "exposure problem". They do not have the chance to set up their offer for resources. Many tasks and applications require a combination of resources (bundles) such as processing power and storage for their execution. Traders might not be able to obtain all of the necessary resources if these are traded in disjoined auctions. To overcome these economic drawbacks, the market mechanisms proposed in our paper rely on a centralized double auction which supports combinatorial bids for bundles from both resource owners and requesters. Therefore the mechanism concentrates demand and supply and mitigates the exposure problem.

In addition to the economic design of the market mechanisms, in this paper we will also propose a general technical design which describes the location of the Open Grid Market in the Grid architecture, its components, and the interaction and interdependencies with other Grid components.

Enhancing the G-PM Performance Measurement Tool by Space-time Diagram

Wlodzimierz Funika, Jacek Duell, Pawel Jojczyk, Robert Strack
Institute of Computer Science AGH, Krakow, Poland

The application programmers need tools which help them to design algorithms for mentioned above distributed systems and verify correctness of its results. There is a lot of programs which helps in finding bottlenecks in execution of the tasks running on one machine. But what with tasks which are distributed across many processors? It does not seem so easy to find out if resources are used efficiently or if network is overloaded. Modern tools should have possibility of monitoring program execution and displaying gained information in easy readable and flexible way also in distributed environment. The example of such a tool is G-PM. It enables grid programmer to monitor the execution of his/her program. It allows to carry out precise measurements using built-in metrics and to visualize results with minimum impact on performance.

This paper makes an overview of the idea of G-PM. It brings closer metrics, calculations automatically made on them and provides various visualizations of measurements. But we focus on a new feature that was planned to be added to G-PM. Until now metrics in G-PM were capable of merging their results (i.e. summing or averaging). It was very useful for most of users needs, but it was impossible to figure out with big precision the exact time of events. By default all measurements was averaged on a given interval of time. It was necessary to add a new solution which in a simple way shows more precisely interactions between running tasks - passing messages, delays spent on sending and receiving data or synchronizing the execution of an algorithm. This motivates to use a space-time diagram to visualize occurrences of those events.

Space-time diagram is a graph which shows interactions between tasks or threads. Originally was used in particle physics and has been adapted for computer science needs. Now it is a common way of visualization of interactions in parallel programs (therein grid applications). One can imagine each task as a straight line (timeline) with intervals marked on it. These intervals are linking events in which the task participates. When such events require more than one task (i.e. sending data requires a sender and a receiver) proper intervals are connected with each other with slanting line.

One can think that the only thing we had to do was to translate monitoring information provided by OCM-G into a space-time diagram. But there are some traps which wait for unaware programmer. What to do with data which consists of large amount of information incoming from OCM-G or how smartly to join the mentioned intervals (OCM-G only stores "really necessary" information about called MPI procedures). Furthermore, the diagram should be interactive and let the user to get more detailed information about the captured events like an amount of data received during the transmission or state of a particular process (waiting on barrier, blocking on transmission or system call etc.) and time scope of each process state. It also should be really useful so requires easy navigation.

The paper also presents the advantages and disadvantages of space-time-based monitoring functionality. It is important to notice that analyzing the execution of a distributed task results in a lost of performance. Often such deep measurements (which are necessary for creating space-time diagram) can overload processors and network and as result perturb those measurements. The idea which brings us closer to solve this

problem is using "cache". The incoming messages are doubly cached - for the first time by the grid infrastructure and for the second time in G-PM (before flushing it onto the screen). The paper also gives results of overhead measurements.

Acknowledgments. The research is partly supported by the EU IST CoreGRID and int.eu.grid projects.

References:
[1] R. Wismueller, M. Bubak, W. Funika, B. Balis. A Performance Analysis Tool for Interactive Applications on the Grid, Intl. Journal of High Performance Computing Applications, vol. 18, no. 3, pp. 305-316, 2004.
[2] M. Bubak, W. Funika, R. Wismueller, T. Arodz, M. Kurdziel. The G-PM Performance Measurement Tool for Interactive Grid Applications, in: Bubak, M., et al. (eds.), Proceedings of Cracow'03 Grid Workshop, October 27-29, 2003, Cracow, Poland, ACC Cyfronet UMM, 2004, Kraków, pp. 227-234.

Execution Management and SLA Enforcement in Akogrimo

Antonios Litke, Kleopatra Konstanteli, Vassiliki Andronikou, Sotirios Chatzis and Theodora Varvarigou
Dept .of Electrical and Computer Engineering, National Technical University of Athens, Greece

The Grid can be viewed as a distributed, high performance computing and data handling infrastructure, that incorporates geographically and organizationally dispersed, heterogeneous resources (computing systems, storage systems, instruments and other real-time data sources, human collaborators, communication systems) and provides common interfaces for all these resources, using standard, open, general-purpose protocols and interfaces. The coordinated execution of computational intensive tasks and some cases of real-time distributed applications are some of the paradigms that are of particular interest to be potentially deployed on Grid and mobile Grid infrastructures. Especially, if we consider that Grids can be used as dynamic information systems to enable business models for utility and pervasive computing, we can identify the significance for the design and creation of novel schemes and architectures for the efficient management of such systems. Such applications are candidates to migrate to Grid and mobile Grid solutions.
In order to do this, the Grids should be equipped with the relative mechanisms to achieve efficiency. The specific architectures that will be designed for this purpose should involve as basic building blocks those mechanisms that will guarantee Quality of Service (QoS) attributes which are mandatory for the commercial exploitation of these applications through the establishment of Service Level Agreement (SLA) contracts.
In this paper, we present a scheme for advancing and managing QoS attributes through execution management in the Akogrimo project. In order to achieve this, the execution environment of the Grid infrastructure establishes and exploits the synergies between the various modules of the architecture that participate in the management of the execution and the enforcement of the SLA contractual terms. The approach that is followed is based on a layered structure of a Grid infrastructure which positions the grid services middleware in collaboration with both the application services and the network services in order to provide an adjustable QoS to the requests of the clients. The components that manage and control the execution in the grid environment interact with the suit of the SLA related services exchanging information that is used by the services to provide the quality framework of the execution with respect to the agreed contractual terms.
The Grid Infrastructure Layer is implemented using the Globus Toolkit version 4 (GT4). This open source grid middleware provides the necessary functionality required to deploy a fully operational grid system. The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability.
The grid services related with the SLA contracts management are developed using the WSRF.NET platform. Implementing a sub-module of the grid infrastructure layer, not depending on the GT4 tools, on a different platform than the GT4 was a compelling implementation and technological challenge, since it checked whether these two popular grid services development tools, implement the WS-related specifications in a transparent and interoperable way. During the implementation procedure several minor inconsistencies (especially in terms of the WS-Notification specification) have been handled by explicitly editing the SOAP messages within the service's source code.
The full paper gives an architectural overview of the layered approach that is followed in Akogrimo and presents the detailed interaction between the core services that are involved in the management and enforcement of the SLA contractual terms, as they have been designed and implemented. Moreover, it discusses implementation issues that have been faced during the development phase. Finally, it concludes with a discussion of the current practices in the field of SLAs in Grids and of potential future research.

Keywords: Open Grid Services Architecture, Service Level Agreement, Execution Management Services

Extending Grid Components with Monitoring Support

Ondrej Krajicek, Ludek Matyska
Masaryk University, Brno, Czech Republic

Management of jobs, data, users and access rights through virtual organizations are well established parts of currently available grid middleware toolkits. To make the necessary management decisions, all these components rely on information provided by some kind of monitoring of the infrastructure (the grid fabrics, its services and components) and the data available to them. Most available grid monitoring systems like R-GMA [2] or Mercury [3] are based on the generic Grid Monitoring Architecture (GMA) [1], defined by GGF. This architecture defines the components of Grid monitoring systems and their interactions, but the actual data collection and manipulation is not specified. The gathering of monitoring data is usually based on well known-metrics provided by lower level middleware layers or on agent-based monitoring of grid resources, but there is no established way how the monitoring data are generated in the first step.

The primary data are usually collected either through explicit instrumentation (extension of logging facilities of individual monitored components), through external agents analyzing local logs provided by the components, or through external agents gathering the information in some other way (mostly polling the local system status). The full instrumentation is the most extensive source of monitoring information, as the instrumentation can be put directly into the code to its most appropriate sections. At the service level, this kind of instrumentation could provide internal status data and specific events not available otherwise. The instrumentation can be done directly by the programmer of a specific service of middleware component (or even an application), but the is cumbersome, time consuming, error prone and usually tightly coupled with just one specific monitoring system.

Therefore we propose a novel approach, based on the implementation extensions. The goal is to provide uniform programming model and tools for authors of all the grid components (including applications). This programming model and tools will provide means to incorporate monitoring into the components without increasing implementation complexity and with added flexibility, without introducing dependence on and particular monitoring toolkit. We propose a Monitoring by Contract approach, based on the ideas of Design by Contract [5] and Axiomatic Semantic [6]. Monitoring by Contract involves extending the programming model with constructs, which allow simple encapsulation of monitoring. These extensions can be either declarative or imperative. The declarative approach, which also offers more potential and is the desired state of the art, defined extensions in terms of contracts, like preconditions and postconditions on certain programming language elements. On the other hand, the imperative approach allows to define extensions in the form of modules or class libraries, which encapsulate the individual aspects of monitoring.

Both approaches will be described, demonstrated and compared.

References:
[1] GFD.7 -- A Grid Monitoring Architecture, http://www.gridforum.org/documents/GFD.7.pdf
[2] R-GMA -- Relational Grid Monitoring Architecture, http://www.r-gma.org/
[3] Mercury Grid Monitoring System, http://www.lpds.sztaki.hu/mercury/
[4] Ganglia Monitoring System, http://ganglia.sourceforge.net/
[5] Building bug-free O-O software: Introduction to Design-by-Contract(TM), http://archive.eiffel.com/doc/manuals/technology/contract/
[6] Winskel, G. The Formal Semantics of Programming Languages, The MIT Press, 1993. ISBN: 0-262-23169-7.

Finding Relations in Web Pages

Dariusz Zbik and Cezary Biernacki
SoftwareMind.pl

Our goal is to present our experience with finding relations between entities (persons, companies) within web pages. Input data of our system is a large volume of web pages. The determining of relations could be divided into two main stages. The first one is the natural language processing (NLP) of the web pages. During this part the system, in a parallel manner, finds entities (persons, companies) within pages. In the second stage the system tries to find relations between entities based on their coexistence. The results of computing are stored in the JXT semantic web database.

In view of the lack of connections between web pages the first stage of computing is parallel in a natural way. Each page could be processed separately. This kind of computing could be parallelized without any advanced tools. Our system uses many instances of web-applications running on independent machines. There is one additional web-application which manages the computing. It takes care of computing application configuration and delegates tasks to compute. In this case the task comprises a set of pages to process. From the manager to the computing unit only URLs are passed. For each page the computing

application gets page content from the local storage and starts processing. Internally we use ANNIE/GATE to find entities within the page, this processing is rather slow. Entities discovered by NLP are stored in a file in the distributed file system (DFS). A status of the page processing is sent back to the application which manages computing.

The second stage of the system processing is more difficult and hence more interesting case of parallel computing. To discover relations we decided to use Hadoop which is an open source version of the Google's MapReduce concept. The input data for this stage are the pages processed during the previous stage. The input data is stored on DFS which is strongly connected with MapReduce.

The relation determining is divided into steps. The first step assigns unique id number for each entity. The assigned ids connect all occurrences of the same entity within all processed pages. To achieve it using MapReduce concept, each entity within the page is treated separately. Entity string is used as the output key of the map process. The reduce process of the first step collects all occurrences of the same entity and assigns one id for them. Each reduce process has a separate scope of id numbers. Additionally, the reduce process computes statistics of entity occurrences. The output of this step is composed of pairs: entity with id number as a key and page identification as a value part of the pair. During the second step all entities from the same page are grouped in the same reduce process. In this way we could recreate the input pages being equivalent and, additionally, enriched by the global id for each entity. In the next step of processing the map process of our system tries to guess which entities are close enough to from a relation. If the entities seem to be related, the appropriate relation is formed as the output of the map process. The key part of the map output is a pair of the related entities ids. The value part of the output is the localization of the relation evidence. The reduce process groups all occurrences of the relation evidences from a pair of the entities in a one place. It allows to decide, using statistics, if the relation should be formed. In the next steps the system prepares the output files in the RDF format.

Fine-grained Security Management in a Service-oriented Grid Architecture

S. Mueller (1,2), A. Hoheisel (1), B. Schnor (2)
(1) Fraunhofer Institute for Computer Architecture and Software Technology, Berlin, Germany
(2) Institute for Computer Science, University of Potsdam, Germany

One major reason for the lack of acceptance of Computational Grids in the industry are security concerns. Conventional Grid security architectures, such as GSI (Globus Toolkit), focus on the service provider's perspective and do not cover all concerns of the service user, for instance the management of user credentials is delegated to services which are not under full control of the user (e.g. myProxy). Another drawback is that common Grid security systems do not cover the fine-grained authorization of services, taking into account the methods, their parameters, and the message flow in deciding whether a user or another service is authorized to access the service. This is particularly important for Grid workflow systems.

This paper presents a security approach, which uses fine-grained and role-based security mechanisms in combination with restricted delegation of privileges, in order to overcome the drawbacks of current implementations. The design of the new security architecture provides a simple and convenient integration of legacy web services without the need of modification.

Our basic approach is to secure standard web services. In the first place this is done by ensuring the authentication and optional the confidentiality of the message using the WS-Security standard. The second step is to perform a two-sided authorisation. Given the perspective of a service provider the following criteria may incorporate into a authorisation decision: The subject on behalf the request is made, the group membership of the subject, the requested service, the action to perform and the intermediate services the request has already passed through. On the other hand, everytime an intermediate service (e.g. a workflow service) is delegating a request, it is the interest of the user to stay in control of what is done in his name. That is the reason why the approach enables the user to specify the same set of authorization criteria a service is provided with. In this case the intermediate service represents the role of the requesting subject. All authorisation policies are expressed and enforced using the OASIS standard XACML (eXtensible Access Control Markup Language).

The paper describes the theoretical concept and a prototype implementation and demonstrates the applicability with a case study based on a real-world postprocessing workflow from the media industry. We plan to apply this security infrastructure in several international and national projects such as CoreGRID, K-Wf-Grid, MediGRID, and Instant Grid.

First Steps of a Monitoring Framework to Empower Risk Assessment on Grids

Nicolas Lerch, Holger Nitsche
Paderborn Center for Parallel Computing (PC2), Germany

Grid technologies have made a big step forward to commercial environments by using Service Level Agreements (SLA). At the moment SLAs are accepted or rejected without or only minimal risk assessment. One reason for a missing comprehensive risk assessment are inadequate data sources. Our monitoring framework is intended to fill this gap.

Cluster management software or grid middleware often install their own monitoring subsystem (MSS). All these systems differ in the number of probes, the way data is collected and processing of this data. However they face a common problem; they are closed systems and were never designed to share data with other MSS. It is a common case that different MSS acquire the exact same data on one node. E.g. a cluster front-end node running grid, cluster and hardware MSS all collecting CPU, Network and disk information. Beside the node and network overhead this information is presented as different data-sources to a risk assessment system. This is an example for the general problem: whenever a new view on a grid is needed we have redundant information and incompatible data from our data-sources.

Our approach is to unify the data-collection on the node level, to transmit collected data from the node over only one channel and to distribute collected data from a unified monitoring server to other (monitoring) applications. We do not intend to develop a new monitoring front-end, but to feed other applications (e.g. MDS, ganglia, risk assessment) with collected data. This leads to coherent data through all applications and the ability to combine different data for new views, e.g. node fan speed with job submission view.

We examined some popular cluster (Ganglia, PCP, Nagios) and grid monitoring (Globus MDS, R-GMA, Sun Grid Engine Monitoring) systems for available probes, communication and data managment. We decided to implement a Unified Monitoring Framework (UMF) as a Client-Server-System. By implementing a simple Plugin-API for probes, the Unified Monitoring Client (UMC) can easily make use of existing probes. Nagios is used as an example front-end for the Unified Monitoring Server (UMS). The UMC is the intelligent part of the framework with its own scheduler and priority queues for collecting and transmitting data to the UMS. In case of communication abortion between UMC and UMS the collected data is stored locally on the node and transferred to the UMS after reconnect. The received data is stored in a Database by the UMS. The front-end fetches relevant data from the RDBMS by use of a Translator-process. Configuration data for each UMC is stored and changed on the UMS. Changes can be applied to the UMC during runtime. Communication is secured by Secure Socket Layer and can be configured to be Socket (LAN) or HTTP (WAN).

The monitoring framework is still in development and tests are performed on a 200 node cluster at PC2. In a small experiment in the D-GRID (German Grid) DGI-project the functionality of this approach for grid monitoring was tested. The results where very promising. The next step is the extension of the framework and it's integration in the AssessGrid Project (www.assessgrid.org).

Formalizing Security Requirements for Grids

Syed Naqvi (1), Philippe Massonet (1), Alvaro Arenas (2)
(1) Centre d'Excellence en Technologies de l'Information et de la Communication, Belgium
(2) CCLRC Rutherford Appleton Laboratory, UK

The Grid promises to make distributed computing more effective and more reliable for businesses as they require continuous innovations not only to survive in this competitive world but also to meet their customers' demands of low cost yet reliable products, services and solutions. Security is an important precondition for the acceptance of the Grid in business applications. Due to the specific nature of the Grid - loose coupling, dynamicity, resource sharing, reconfigurability in real-time, etc. - its security requirements are somewhat different from those of traditional distributed systems. Grid applications need to be flexible in order to be adaptable to the changing needs. Policies can be used to reconfigure them to adapt to the required changes in the goals or in the environment.

The grid security requirements are driven by the roadmap of Open Grid Services Architecture (OGSA). It provides a high level abstract strategy that allows well-defined protocols and interfaces to be defined for security services and permits an application to outsource security functionality by using a security service with a particular implementation to fit its current need. However, OGSA roadmap falls short of providing any formal means of expressing security requirements of the Grids in general and of its specific applications in particular. In this paper we have carried out a formal analysis of security requirements for semantic grid services to explore how these requirements can be expressed as metadata associated to these services.

We describe a method to specify security requirements and derive security policies. Our approach uses the KAOS requirements engineering method to formalise the security requirements for Grids. The derivation of adaptable security policies from the security requirements is explained with the help of examples.

In this paper, we elaborate our approach by presenting a case study of Grid Data Management System (GDMS). The problem addressed is to assure fault tolerant and secure management of distributed file systems. Fault tolerance is attained by keeping sufficient number of replicas at different grid nodes; whereas the secure management of these files requires adequate encryption strength during the inter-node files transfer. The various parameters involved in attaining these requirements are formally presented in this paper. The implementation of this approach in the real systems requires formal derivation of security policies from the requirements model. The derived policies are quite abstract and need to undergo refinement process so that operational policies can be obtained. These operational policies can be directly implemented on a system - grid data management system in the context of this work. Security policies define the types of security measures that are used and what scope those measures have but not how those measures are designed or implemented. System security policies are derived from security requirements that specify the risks and threats that must be countered. These policies are system-specific and reflect the threat environment and the security problems assumed by system designers. For the refinement of higher level policies into operational policies, we need a formal representation for objects, their behaviour and organisation; a technique for refining high-level goals into more concrete ones; and finally a means of inferring the combination of operations that will achieve the concrete goals. However, the refined goals cannot be directly used in policies without first identifying the operations that will achieve them.

* This research is supported by the European Network of Excellence CoreGRID (project reference number 004265). The network aims at strengthening and advancing scientific and technological excellence in the area of Grid and Peer-to-Peer technologies. The CoreGRID webpage is located at www.coregrid.net

g-Eclipse - an Integrated Workbench Tool for Grid Application Users, Grid Operators and Grid Application Developers

Harald Kornmayer
Forschungszentrum Karlsruhe GmbH, Germany

The g-Eclipse project, which started on 1 July 2006, aims to build a general, integrated workbench toolset for Grid users, operators and developers. The approach of the g-Eclipse project is different from most web-portal based projects by developing a fat client framework. The g-Eclipse framework will be based on the Open Source Eclipse framework. The objectives of the project are the delivery of an extensible framework for different Grid actors: Grid application users will be able to access the Grid in a desktop-like manner and their common use cases will be supported, Grid resource providers and operators will be supported by configuration tools based on visualisations and Grid applications developers will be supported with development, deployment and debugging tools as well as with a workflow editor. The g-Eclipse project aims to provide a middleware independent framework. In the first year, the exemplary support for the gLite middleware from the EGEE project will be delivered. The g-Eclipse framework will integrate existing tools from the CrossGrid Project, such as components of the Migrating Desktop (MD), the GridBench suite and the Grid Visualization Kernel (GVK). g-Eclipse aims for an extensible framework to be able to offer integration support for third-party Grid tools as well.

The paper will discuss the g-Eclipse approach in comparision to the standard web-server based portal approach. The requirements from different Grid actors will be discussed as well as their implications for the development process. A first overview of the architecture will be given.

Grid-Based Business Partnerships using Service Level Agreements

Mike Boniface, Stephen Phillips, Mike Surridge
IT Innovation Centre, University of Southampton, UK

Traditional academic Grids are based on collaborative resource sharing usually organised by service providers, who agree to share their resources to create a single 'resource pool' that can meet the needs of a common user community. However, this involves service providers and users signing up to a 'virtual organisation' with a common objective (that of the relevant user community), and agreeing unified policies for managing and granting user access to the shared resources. This arrangement is very expensive to establish and operate, does not meet the business needs of commercial service providers or consumers, and leaves no room for participants to compete openly on price or quality of service.
The SIMDAT project [1] is developing generic Grid technology targeted at business users from several representative industry sectors. As part of SIMDAT, we have developed an SLA (Service Level Agreement)

Management service for GRIA [2] that allows service providers and customers to trade resources (applications, data, processing, storage) under the terms of bilateral SLAs. Service providers are able to operate independently of each other, and compete as necessary to provide services to paying customers. Customers are able to control which services they consume, how much they are used, and by whom.
The SLA Management service allows service providers to advertise SLA templates that are proposed by customers during SLA negotiation. The SLA describes quality of service (QoS) and other commitments by a service provider in exchange for financial commitments by a customer against an agreed schedule of prices and payments.
Service providers deploy application services appropriate to their business operation. These services generate usage reports using their own QoS criteria which may be qualitative (e.g. error conditions) or quantitative (e.g. processing time, data transferred). The SLA Management service uses these reports to monitor customer usage and the level of commitments from existing agreements compared with available capacity. It can then automatically decide whether to enter into a new SLA (with a given QoS) when requested by a customer; whether a requested service is covered by an existing SLA, and whether the service can be provided within the capacity of the provider; which SLA(s) should be breached when capacity is about to be exceeded, and how to reduce loads in the corresponding application service(s); when a consumer is exceeding the limits of an SLA, and how to prevent this by reducing service consumption in the corresponding application service(s); and what level and quality of service is actually delivered, and how much to bill for this.
To enter into a new SLA, a customer must be authorised to bill services provided under the new SLA to a GRIA charging account, and the account must have sufficient credit left to cover the scale of services that would be supplied under the SLA. This means the human manager of GRIA services can control how much service will be provided to which customers, but he only has to make one business decision: what credit limit to allow on the customer's trade account, if any. After that, everything is automated, which makes GRIA services very responsive to new user needs, yet inexpensive to operate for providers. This is critical in a business Grid where human resources can easily become the largest cost when running services - in GRIA these costs are minimised, so services can be profitable for providers while still being affordable for customers.
Within SIMDAT, GRIA has been successfully deployed to support business partnerships in the aerospace, automotive, and pharmaceutical sectors. The SLA Management service is part of GRIA's Service Provider Management package, which is available for download, free and open source, from www.gria.org.

SIMDAT has received research funding from the European Commission under the Information Society Technologies Programme(IST), contract number IST-2004-511438.

[1] www.simdat.org
[2] www.gria.org

GVOSF: Grid Virtual Organization Semantic Framework for Knowledge Support

Bartosz Kryza(1), Lukasz Dutka(1), Renata Slota(2), Jan Pieczykolan(1) and Jacek Kitowski(1,2)
(1) Academic Computer Center CYFRONET AGH, Cracow, Poland
(2) Institute of Computer Science, AGH University of Science and Technology, Cracow, Poland

One of the most important questions in today's business world is cooperation between business parterns, understood as collaborative work or research within project teams, etc. Contemporary business scenarios are very challenging, in terms of complex projects which often involve more than one company or supplier [1]. This requires mechanisms which can empower collaboration between partners, who are often distributed geographically across continents, speak different languages, use different technologies or have different cultures.

One of recently proposed concepts is to create a collaboration network for partners. Such a network is based on the Internet and connects together shared resources, allowing them to dispose more computational power to solve other parties problems. However, this approach, which is limited only to computational calculations is not enough for modern business settings. Contemporary business organizations or crisis management teams like police, fire-brigades need a simple and very fast way of creating collaboration network in a time shorter than an hour. The Grid and Virtual Organizations [2] are a realization of this requirement - an infrastructure which spins-up together IT resources, giving scientists an illusion of single, very efficient metacomputer. They also need a way to collaborate by sharing not only their computational power but also more abstract resources like information or even knowledge or experience.

Many problems related to ad-hoc creation of a VO are mostly related to heterogeneity of resources shared by VO members. Not only computer equipment is different, but also data formats, service descriptions, knowledge repositories - this requires a method of mediation between IT infrastructures of VO members. Such a method is necessary to provide connectivity and make collaboration possible and efficient.

The authors propose a semantic enabled framework which can be set up on the top of existing Grid infrastructures and VO management tools. The idea behind this solution is to use ontological representation of metadata to annotate resources shared by partners, declare requirements of VOs, support business processes modeling and support for domain specific knowledge management. With such annotations it is possible to perform semi-automatic creation and maintenance of Virtual Organizations in the Grid. In order to provide mediation between resources ontology similarity algorithms will be used, which facilitate higher level of interoperability. The proposed framework will be based on the Grid Organizational Memory knowledge base [3], developed within the K-Wf Grid project. GOM generic interfaces for knowledge management will be extended with specific ones, which will make VO management easier and more robust.

The article will describe high level architecture of mentioned semantic framework. It will identify the basic VO management activities like creation, evolution and termination. Knowledge management mechanisms will be described in the context of VO management, which will allow for storing and publishing of gathered knowledge and experience. Security issues, especially trust management will be investigated. In the summary of the article prototype implementation evaluation conclusions will be presented.

References

[1] Burn, J., Marshall, P. and Wild, M.: Managing Knowledge for Strategic Advantage in the Virtual Organisation, in proc. of ACM SIGCPR, 1999.
[2] Foster, I., Kesselman, C., and Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. International Journal of Supercomputer Applications, 2001
[3] Kryza, B., Slota, R., Majewska, M., Pieczykolan, J. and Kitowski, J.: Grid Organizational Memory - Provision of a High-Level Grid Abstraction Layer Supported by Ontology Alignment, Future Generation Computer Systems, In print, 2006

IEBS - Intelligent ExaByte Storage based on Grid Approach

Lukasz Dutka(1), Barbara Palacz(2) and Jacek Kitowski(1,2)
(1) Academic Computer Centre CYFRONET AGH, Cracow, Poland
(2) Institute of Computer Science, AGH University of Science and Technology, Poland

The need for disk drive capacity has grown exponentially over time. Today, it is becoming essential not only to provide even greater data storage, but also to ensure the data's easy access from multiple machines simultaneously. This paper discusses a novel approach to this problem, by designing a system supplying usable storage of ExaByte order of capacity.

The system is based on a data grid approach, which deals with controlled sharing and management of large amounts of distributed data. Grid computing is designed to solve problems too big for a single computer. At the same time, IEBS solves the issue of storage requisition, by braking it down onto thousands of separate hard disk drives ~1TB capacity each, which form together a virtually single computer storage. All disk drives are equipped with simple processing powers and Ethernet interfaces. They have their own IP addresses and work together over UDP/IP network, with the help of several additional devices. Therefore, the physical layer of our grid environment is made up of multiple resources connected by a network. Above this layer, there is a layer implementing the system's functionality, which includes data sharing, access authorization, easy monitoring of the system's reliability and performance, as well as great system scalability.

The general idea of data management in IEBS is that the entire ExaByte storage is divided into logical units, called blocks. The system organizes the blocks and stores information about where each block is located and what data segment is written into it. When a specified data segment is needed, the system simply indicates at which disk and at which address the segment is located. Such approach allows realization of all basic operations on data, which are reading, writing, updating, deleting and replicating data. Some of these operations are also available with several supplementary features that result from the intelligence of the storage. For instance, the system keeps track of all the physical locations of its hardware, thus it can satisfy a user's request for a precise location of a specified file. Moreover, thanks to the built-in processing power in disk drives, IEBS is capable of performing many advanced data processing scenarios, including redundancy support operations on demand, such as an on-line or off-line file replication and so on.

IEBS is a storage system model, which can resolve the problem of a great storage requisition with relatively low cost of construction. Some already-existing solutions, related to this issue, are NAS and SAN. The first one mentioned is a storage-centric design, where a main server handles all processing of data. Such approach does not allow physical data distribution. SAN, on the other hand, runs over proprietary protocols and has an expensive, complicated infrastructure that requires well-trained personnel. Neither of the solutions incorporates intelligent storage, supporting distributed replication of files, nor they take into account as much initial storage as an ExaByte.

The IEBS solution is dedicated for institutions needing to store and process large amounts of data, and

requiring a storage system which can be easily extended, geographically distributed and maintained by many independent authorities, similar to the grid approach.

Influence of Virtualization on Grid Application Deployment Process - CCM case study

J. Cala, K. Zielinski
Department of Computer Science, AGH UST, Krakow, Poland

Recent progress in technologies of computer resources virtualization influences grid computing in many ways. This seemingly obvious conclusion conveys, however, two possible meanings and goals of the virtualization. From the one point of view, grid computing very much concerns building of a virtual organization (VO) over computational, communication and storage resources, which supports their efficient and transparent sharing between groups of users. From the other, computer resources virtualization might be perceived as a process of exposition of physical resources in the most convenient form, supporting their utilization and control by applications or end-users.
Grid as a VO is a large-scale structure of computer resources which complexity and implementation techniques strongly depend on the level of the resource virtualization. Grid considered as a VO specifies resources, access rights, and sharing policies on different abstraction levels, from global to local ones. That is why the process of deployment of an application in a grid is a sophisticated multi-step procedure.
A promising way to deal with this issues is application of SOA approach and, in particular, component based software engineering. The goal of this paper is to analyze how the choice of a level of computer resources virtualization might influence the deployment process of a component based application. The level of virtualization changes the information model that has to be taken by the deployment procedure. The technical details of this procedure are platform specific but we refer to platform independent model defined in Deployment and Configuration of Component-based Distributed Application Specification (D&C) by OMG, which seems to be a good starting point for general considerations.
Deployment procedure, as defined in the specification, consists of five steps. It begins with installation of a software package and ends with launching it over selected resources in the target domain. This paper turns the most attention to one of these steps - the planning phase, which is a process of matching the requirements of a software package to be deployed with the resources of the target execution environment, and deciding how components of the application are going to be arranged in the environment.
An important issue, closely related to planning, is possible representation of resources and requirements in context of different virtualization levels. For each property which is a part of a component's requirement there has to exist a corresponding satisfier property among the attributes of the matching resource. Therefore, it is evident that properties of both resources and requirements have to be expressed at the same abstraction level corresponding to the selected virtualization technique. Each virtualization technique introduces, however, specific information model, different methods of management and monitoring, hence might have direct influence on the deployment process.
In the end of the article there is shown a sample software package with descriptions provided for the chosen virtualization levels. The general, D&C description model is mapped to CORBA Component Model which is, currently, the only technology adopting D&C specification. The provided examples are compared and a short discussion is presented concluding the advantages and disadvantages of the proposed descriptions as well as assessing their complexity and its impact on development and planning phases. The paper is ended with conclusions.

Infogrid: Grid Middleware Using an Information System

Oliver Lyttleton, Brian Coghlan, Eamon Kenny and Geoff Quigley
Trinity College Dublin, Ireland

One of the least desirable aspects of grid computing has been protocol and API inflation, particularly the growth of proprietary protocols. We propose reduction to a single protocol, one that accesses information over a grid, and is implemented with a simple API. All the lower level detail of any protocols will be subsumed into the protocols used to exchange the information, so the only protocol in use will be the information systems native information access protocol, and the only API will be the native access API. In this hypothesis, higher-level protocols are composed from schemas and information exchanges. The schemas define the structure of the information, but not its content.

R-GMA is a monitoring and Information system that provides a global view of data distributed across a grid system. It models the information's structure using the relational data model (SQL-92). We use R-GMA as

the information system that will be used to exchange information, and the R-GMA API will be the InfoGrid API which subsumes any other APIs. In some cases R-GMA code will replace grid middleware code, and in other cases R-GMA code will wrap grid middleware components which need to send/receive information via the R-GMA information system. By way of example we propose to simplify TCDs Metagrid middleware by replacing transport layer APIs with the Infogrid API, which will reduce the "API inflation" which bedevils grid computing at present.

We have also modified the LCG WMS to create a "Infogrid CE" that uses the Infogrid protocol. The LRMS used by the CE is Libra. We wish to take advantage of the proportional share scheduling algorithm used by Libra, which is superior to PBS in some circumstances. Libra is an economy-based job scheduling system for clusters, developed by Rajkumar Buyya. We have modified the LCG jobmanager so that extra attributes can be defined in the JDL for the Estimate, Deadline, and Budget attributes used by the Libra batch system, and these values will be used as arguments by the jobmanager when the CE submits a job to the LRMS. We have also modified the LCG WMS to replace its Job Controller with R-GMA code, so that communication between the RB and the CE is achieved via the R-GMA information system using the InfoGrid protocol.

Keywords: MetaGrid, LCG WMS, R-GMA, Libra, Information System

Integration of Grid Applications in MediGRID

Jürgen Falkner, Anette Weisbecker
Fraunhofer IAO, Stuttgart, Germany

The key to successfull use of Grid infrastructures is the integration of applications and their adaption to the requirements of Grid environments ("gridification"). Usually Grid applications are not developed from scratch but are advancements of already available applications. The sense of Grid technologies here is to boost the capacities and capabilities of such applications as well as to simplify the access to such applications and to enlarge the number of users of such applications.
In order to adapt applications to Grid environments a lot of factors have to be taken into consideration. These range from hard factors like available hardware and middleware to soft factors such as user skills, the predicted number of users, the purpose of use and security considerations.
Therefore methodically sound procedures to analyse the needs and wishes of future users within a certain Grid community and the technical circumstances are needed as well as reproducible methods to analyse existing applications with regard to their gridification and future use in combination with Grid workflow management, job management, scheduling and many other services.
MediGRID is one of the Grid communities within the German e-Science program "D-Grid" and involves users and applications from the areas of biomedical informatics, medical image processing and clinical studies. Affiliated MediGRID resource providers operate an own Grid infrastructure for this community on which the communities' applications are executed.
During the initial phase of the project a fact finding and requirements analysis was started with a couple of surveys, which was then extended through a number of workshops with users and application providers as it showed that certain information could in practice not be gained by the means of surveys.
These were conducted in order to investigate necessary information like the structure of applications and application workflows, the prospected number of users, the advantages that were to be reached via Grid technologies compared to the capabilities of the applications as they had been and many more.
This paper gives an overview over the strategies and procedures to conduct requirements analyses for the establishment of Grid environments and the gridification and deployment of Grid applications. It discusses not only the methods of analysis but also the factors that need to be taken into consideration and gives practical examples from the experience of the MediGRID project.

InteliGrid Document Management System

Matevz Dolenc (1), Krzysztof Kurowski (2), Michal Kulczewski (2) and Alexander Gehre (3)
(1) University of Ljubljana, FGG, Ljubljana, Slovenia
(2) Poznan Supercomputing and Networking Center, Poznan, Poland
(3) Technical University of Dresden, Dresden, Germany

A challenge for grid collaboration infrastructures is to support dynamic virtual organisations (VOs) that collaborate on the design, production and maintenance of products. InteliGrid project is addressing the engineering end-user requirements of the architecture, engineering and construction sector as well as from industries with long supply chains like automotive, shipbuilding and aerospace.

In engineering in general, the ability to securely access diverse data sources in a collaborative environment

is becoming an essential requirement for optimising the design, development and maintenance phase of the engineering product life-cycle. While a product model based exchange of information is slowly replacing the traditional RDMS and file based storage solutions it is true that a majority of the communication in a typical engineering project is still document based. It is therefore essential to provide end-users with a powerful but easy-to-use document management system that would enable easy, on-demand, personalised, and secure access to document based information in a collaboration environment with a goal of supporting general engineering end-user scenario.

The InteliGrid document management system (IDMS) provides a generic, grid based, ontology enabled document management solution that provides client as well as server side components with well-defined web services interface that enables remote access the the underlying document management services. The ontology based implementation of the system enables the realisation of various new end-user scenarios, from advanced fine-grained access rights management to enabling semantic based document retrieval. The system currently supports two different back-end document storage systems - RDMS and WebDAV based servers. By providing WebDAV based back-end storage the end-users can also use any third-part WebDAV compliant application to directly access stored documents.

In this paper the relevant sub-systems of the InteliGrid semantic grid architecture that enable the ontology based document management system are presented. The underlying end-user scenarios that led to the development of the system as well as the implementation details of the developed system are also highlighted.

Interactive European Grid Environment for HEP Application with Real Time Requirements

Lukasz Dutka(1), Krzysztof Korcyl(1, 3), Krzysztof Zielinski(1, 2), Jacek Kitowski(1, 2), Renata Slota(2), Wlodzimierz Funika(2), Kazimierz Balos(1), Lukasz Skital(1), Bartosz Kryza(1)
(1) Academic Computer Centre CYFRONET-AGH, Cracow, Poland
(2) Institute of Computer Science, AGH University of Science and Technology, Cracow, Poland
(3) Institute of Nuclear Physics PAN, Kraków, Poland

The objective of the Interactive European Grid (int.eu.grid) project is the deployment of an advanced Grid empowered infrastructure in the European Research Area. The infrastructure is specifically oriented to support the execution of interactive demanding applications, guaranteeing interoperability with existing large e-Infrastructures like EGEE by providing basic common middleware services. The initiative exploits expertise generated by the EU CrossGrid project to provide researchers an interactive and simultaneous access to large distributed facilities through a friendly interface with powerful visualization.

We plan to use the interactive grid environment to support processing of data coming from the largest particle accelerator - the Large Hadron Collider (LHC). The amount of data generated by the detector requires an implementation of a multi-stage filtering system to select only those interactions which may be interesting for physics phenomena studies. The current architecture of the High-Level Trigger (HLT) part of this filtering system requires building a massive computer farm of order of 1000s processors employed just for the selection of the most interesting collisions, therefore it heavily involves network infrastructure.

Our research, formerly focused on the feasibility studies on using networking resources (GEANT and NRENs) for high volume, real time data transfers, required for the HLT application, proved that the concept of using remote processing for real-time detector monitoring and data filtering is feasible from the networking perspective. The further research have shown that exploitation features of int.eu.grid enable to employ grid infrastructure to realize HLT filtering tasks when it is necessary. The features of interactive grid might be helpful in exploitation of remote resources when the local processing resources become insufficient. In such a case, the ATLAS operator will initiate the process of searching the GRID for available resources. This includes processing power and network resources which are sufficiently effective accessible.

In order to solve HLT filtering we plan to develop tools for monitoring grid infrastructure helping in smart distribution of tasks to remote site taking into account current status of both the network and the computation nodes. At the same time application monitoring is aimed to provide data on the current status of tasks and their execution progress. This data is intended to underlay making decisions within tasks' lifetime.

It is important to remark that the software design and development effort required to adopt HLT application approach to grid processing of data has to take into account Quality of Service data and real time demands. Such an approach can be used for many other purposes in other fields including astronomical or medical applications which very often require computational power in a very similar way and the power is available in the grid but lack of support for interactivity and real time grid respose deferred them to request for the grid power.

In this paper we discuses architecture of the grid solution for the HEP application taking into account challenges of QoS and real time requirements in execution. The developed solution is able to adapt dynamically and transparently to the computational power needs driven by real-time events as well as to continuously changing infrastructure features (like network throughput or current site load) on the fly.

The work described in this paper was supported in part by the European Union through the IST-2006-031857 project int.eu.grid funded by the European Commission program.

Kerberos Authentication in Grid Environment

Stanislaw Kulczycki, Krzysztof Wilk
GridwiseTech

In this paper we evaluate deployment of user authentication means to Grid environment. The authentication means we use are provided by Kerberos version 5. Kerberos v5 is an industry standard of network authentication protocol. It is defined by RFC 4120 document and implemented by multiple vendors. In our case we rely on Kerberos tickets for authenticating network users who are allowed to access Grid Stack. Prior to accessing Grid Stack each user must obtain a Kerberos ticket from previously defined Kerberos Ticket-Granting Server. The ticket is akin to enterprise ID and every user is expected to show it before using Grid Stack services. Following enterprise ID analogy, every ticket is valid only for a precisely specified amount of time but can be renewed after its expiry date.

Grid Stack is a term coined for the purpose of this paper. Grid Stack is a loosely coupled set of applications we use together for various purposes. It consists of four software layers as following (in order of decreasing user proximity): Grid Portal, Distributed Resource Manager, Distributed Filesystem and IP level encryption. Grid Portal is the uppermost layer of Grid Stack. It provides single point of entry to lower stack layers. It is the only stack layer that is visible to regular users of Grid Stack. Distributed Resource Manager (DRM) is a piece of software that facilitates execution of user applications on a pool of worker nodes, also known as execution nodes. All the nodes are managed by DRM software. DRM software makes possible running user applications, including parallel ones, on multiple nodes at the same time. IP level encryption increases security of entire network communication flow. It is implemented by IPsec framework that is an optional part of IPv4 and an obligatory part of IPv6.

In our research we examine the necessity of each of the aforementioned Grid Stack layers. Then we ponder the possibility of adding more layers. We also evaluate advantages and disadvantages of the layers, and provide appropriate rationale for our conclusions. We also share results of tests we conduct in our testbed. Our main testing purpose is to verify if Kerberos can be used effectively in Grid environment. From software testing point, we conduct software tests that belong to several test categories: stress tests, performance tests, grey box tests and integration tests. We think that if the tests prove that such a generic solution as Kerberos performs well, it can be used as an alternative to very specific, Grid authentication solutions e.g. MyProxy. Therefore we put emphasis on testing of Kerberos-driven authentication in network circumstances common to Grid use e.g. frequent network packet delays, significant packet losses, network traversal overhead imposed by firewalls. We also evaluate Kerberos server replication feature. Last but not least, we measure time overhead imposed by various encryption algorithms used by Kerberos authentication and IPsec encryption of network communication flows.

Knowledge of the Grid: a Grid Resource Ontology

Michael Parkin(1), Sven van der Berghe(2), Oscar Corcho(1), Dave Snelling(2), John Brooke(1)
(1) eScience North West Centre, School of Computer Science and Manchester Computing, The University of Manchester, UK
(2) Distributed Services Research Group, Fujitsu Laboratories of Europe Ltd. (FLE), Hayes, Middlesex, UK

The Grid computing vision is to enable "coordinated resource sharing" [ANATOMY] through the creation of application-independent middleware and protocols. To this end a number of middleware systems have been developed, e.g. Globus toolkit [GLOBUS], Unicore [UNICORE], gLite [GLITE]. These systems are all mainly trying to implement the functionality required for a computational Grid but use different syntax and components to express such functionality. In the EU FP6 Grid programme two projects [GRIP, UNIGRIDS] have concerned themselves with interoperability between Globus and Unicore. Among the methods for achieving interoperability they developed an ontology for describing Grid Resources and the methods of accessing these Grid resources, this has been termed a Grid Resource Ontology (GRO) (1). The original GRO developed for GRIP used an ontological mapping approach, terms used in Unicore and the GLUE schema [GLUE] (used in many Globus projects) were mapped to each other, the mapping indicating that

they had the same meaning, this is essentially a dictionary approach. It was found that this led to the loss of a considerable amount of the functionality of both systems, since they had implicit architectural assumptions, which lay behind the use of the naming of resources. This architecture expressed how the resources were accessed and used. Consequently the most recent version of the GRO contains terms and relations that are able to express this architecture in an abstract way. Since most current Grids provide similar functionality despite what individual Grid middleware is used, the abstraction is possible in theory and is being realised in the GRO. For example the GRO has the abstract concept of an AccessPoint where access to resources is obtained. In Unicore this is called a Gateway, in Globus it is called a Gatekeeper and the detailed mechanisms of gaining access are different in the two systems but in the abstract sense they perform the same function.
The GRO is grounded on the client/resource provider view characteristic of Service Oriented Architectures [OGSA]. This gives rise to two fundamental design issues: one, it must allow us to express resource requirements in an abstract, resource (and middleware) independent form in order that clients can form descriptions of abstract actions they wish to be carried out on the Grid. Secondly, the ontology must express, again in an abstract, resource and middleware independent manner, the resource provider's capabilities. We refer to the process of translating between these two forms as incarnation, a term that was originally used by the Unicore Grid middleware, literally meaning "the act of causing to exist" (2). In an SOA a service grounds a request enabling it to be physically enacted e.g. as the creation of a file, a database search, the creation of a process or a batch job.

We also describe how our work relates to work aiming at the creation of a common, structured set of terms (a vocabulary) to describe Grid applications and middleware. The GLUE schema, Unicore Abstract Job Objects, the DMTF Common Information Model and the OGSA Glossary [GLOSSARY] are good examples of existing vocabularies. However, the examples listed lack some of the characteristics of ontologies (defined as "formal, explicit specifications of a shared conceptualisation"): they either lack formality in their definitions, which makes their usability and reusability more difficult, do not express a shared point of view of Grid applications and middleware, are not explicit, or they do not cover all the concepts and of components that a Grid application may have. What is new in our work is the ability to use a formal ontological language standardised by W3C (OWL or the Web Ontology Language) to express the abstract relationships and objects used in developing a Grid.
In this paper we present the GRO and describe how it links to work on a Semantic Grid architecture [sOGSA] developed in the FP6 OntoGrid project. This allows richer interoperability between different Grids and provides a basis for more flexible and more dynamic creation of infrastructures to enable Virtual Organisations.

Acknowledgements
This work is supported by the EU FP6 OntoGrid project (STREP 511513) funded under the Grid-based Systems for solving complex problems, and by the Marie Curie fellowship RSSGRID (FP6-2002-Mobility-5-006668).

References
[ANATOMY] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid. International J. Supercomputing Applications, 15(3), 2001.
[GLOBUS] I. Foster, C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. International J. Supercomputer Applications, 11(2):115-128, 1997.
[UNICORE] D. Erwin, D. Snelling. UNICORE: A Grid Computing Environment. In Proceedings of Euro-Par 2001: Parallel Processing, 7th International Euro-Par Conference, Sakellariou R, Keane J, Gurd J, Freeman L. (eds.). Springer Lecture Notes in Computer Science: 2001. 825-834.
[GLITE] http://glite.web.cern.ch
[GRIP] R. Menday, P. Wieder. GRIP: The Evolution of UNICORE Towards a Service Oriented Grid. In Proceedings of the Cracow Grid Workshop 2003.
[UNIGRIDS] http://www.unigrids.org
[OGSA] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich. The Open Grid Services Architecture, Version 1.0. Global Grid Forum (GGF), Informational Document GFD-I.030. 29 January 2005.
[GLUE] The GLUE Schema. http://glueschema.forge.cnaf.infn.it/
[GLOSSARY] J. Treadwell (Ed.). Open Grid Services Architecture Glossary of Terms. Global Grid Forum (GGF) Document GFD-I.044. 25 January 2005.
[SOGSA] O. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, C. Goble. An Overview of S-OGSA: A Reference Semantic Grid Architecture. Journal of Web Semantics 4(2):102-115. June 2006

(1) Available at http://www.unigrids.org/ontology.html
(2) Oxford English Dictionary. July, 2006.

Large-Scale Evolutionary Optimization on the Grid: Multiple-Deme Genetic Algorithm in the Globus-Based Environment

Adam Padee, Wojciech Padee, Krzysztof Zaremba
Institute of Radioelectronics, Warsaw University of Technology, Warsaw, Poland

Numerical optimization performed with evolutionary algorithms is becoming increasingly popular for solving complex problems. This includes areas like: parameter optimization in High Energy Physics, advanced scheduling problems, electronic circuits design, or many others. Although evolutionary algorithms demand significantly higher CPU power than the classical, gradient-based methods, they perform much better on objective functions with multiple local optima. Another advantage of evolutionary algorithms over classical optimization techniques is easier parallelization.

There are several approaches to the construction of parallel evolutionary algorithms, but most of them are suitable for homogenous environments like parallel supercomputers or single clusters. This paper proposes a robust and flexible architecture of the genetic algorithm with distributed population, that can be easily adopted to the heterogeneous environment of the computational Grid. It is designed as a hierarchical hybrid of the two different algorithms: parallel SSGA (Steady State Genetic Algorithm) on the level of clusters, and a multiple-deme evolutionary algorithm (also called the Island Model) on the level of whole Grid.

First level, the parallel SSGA, operating on the level of clusters, is highly integrated with the local batch system (currently LSF and OpenPBS are supported). At the beginning, Master process is submitted to the selected cluster via global Resource Broker. Then the Master process sets the population size (depending on the number of CPUs available for the given VO) and takes care of submitting the proper number of Slave jobs directly to the batch system. Internal architecture of the Master contains a parent population (with fully developed individuals) and execution queue with individuals waiting for evaluation. As soon as one of the Slaves finishes it's job, the resulting individual is put to the parent population using a reverse tournament operator. Slave jobs are monitored by the Master and respawned if needed.

Second level, the multiple-deme genetic algorithm, is responsible for communication between clusters. It is fully distributed, so there is no central process - all the operations are performed by the Master processes at the clusters. Communication between demes can be done in couple of ways. The simplest model needs only a global filesystem (e.g. LFC in EGEE), which is then used by each of the Master processes to record migration candidates. Disadvantage of this approach is slow exchange of the information, which makes it suitable only for problems with complex objective function. Second model utilizes direct communication between Masters, either via MPICH-G2 or plain TCP/IP. It is much faster, but also more troublesome, as MPICH-G2 is not available in many Globus-based production Grids (e.g. EGEE). TCP/IP, on the other hand, requires a preliminary synchronization step (exchange of information about CE names and available ports, registered as description files in LFC).

Different communication topologies (ring, multi-dimensional mesh, hypercube) and migration rates are being tested. The problem appears to be quite interesting, because of different sizes of the demes and different characteristics of the communication channels. There is also a study on the differentiation of the genetic operators in the participating demes. Parameters describing operators' probabilities and ranges can be either set a priori in each deme, or independently adapted with the development of the population.

Finally, the test results are presented. The algorithm is tested on the well-known deceptive functions (Griewank and Rosenbrock) to test it's ability to avoid local optima and to traverse low-gradient areas. There is also an example production application optimizing a track reconstruction routine for High Energy Physics.

LDAP Gateway with Partial Caching (LDAP-PC)

Jiri Sitera
CESNET and University of West Bohemia, Czech Republic

Within a grid environment, a plethora of information sources offer access to information via specific APIs and protocols. An user is thus overloaded by the necessity to know many details and access protocols. Also, the data in some information sources are generated on demand, with a visible latency between the request and a reply. To remedy this problem we present the concept of the LDAP gateway with a partial caching. It is a tool intended to create a simple but powerful gateway between primary data sources and clients. Such gateway provides a unified view to data originally available by various APIs and protocols. The LDAP-PC extends the basic LDAP gateway with the ability to cache data in a hybrid way. Generally, two distinct models of the gateway behavior can be identified---forwarding model when all requests to data are just transformed to appropriate format and sent to the right data source, and a mirror model, when gateway stores all the data to the local LDAP database and requests are served from this local database. The LDAP-PC gateway supports both models and the actual behavior is derived from the scope of a query. If the query can be answered using just one LDAP database entry, it is transformed to a primary data source request. On the other hand, if more data sources (more LDAP entries) have to be accessed to answer the query, the reply is provided from the cache. The LDAP-PC cache semantics guarantees maximum cached entries age, as the whole cache is refreshed by regular updates. This also means that the combination of entries is meaningful, as all the data are from the same time slice.

The LDAP-PC is intended as an easy to deploy tool to be used everywhere the need for easy-to-use and unified access to data exists. While it is not complex and feature rich information infrastructure, it can act as a partial but well working solution addressing all key problems---single access protocol, data model, authentication options support, one authorization scheme, data syntactic transformations, etc. Thanks to its caching strategy, it can be also used to solve performance issues. Typical use of LDAP-PC can be demonstrated on a gateway to volume quota and usage records of an AFS filesystem. That data are easy to transform to LDAP model which is straightforward to use and completely hides peculiarities of the AFS protocols from users (developers of web based tools such as user support services, portal, etc.). LDAP-PC also provides appropriate authorization controls extending those provided by the AFS API. In this case the caching semantics takes place in a natural way: tools asking for actual disk usage of one user are handled by on-fly direct query to the AFS providing fresh value, whereas tools asking for disk usage of all volumes located on a particular server partition are processed from cache---much faster than real AFS query. The paper will be concluded by description of pilot deployment (both the usage of LDAP-PC and its pilot implementation details) and plans for future work (improved implementation, potential grid deployment areas etc).

Method for Mapping FEM Computation onto Cluster Grid Architectures

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

In the CLUSTERIX project, the MPICH-G2 middlware based on the Globus Toolkit is used as a grid-enabled implementation of the MPI standard. It allows for running multilevel parallel applications across local clusters. The CLUSTERIX infrastructure has a hierarchical architecture, with respect to both the memory access and communication. For example, communication between computers of the same site is much faster than comunication between nodes of different sites provided y a wide area network. It is not a trivial task to adopt the existing applications for effective use in meta-cluster.

In this paper, we present a method for mapping Finite Element Method (FEM) computations onto cluster grid architectures, taking into consideration their structure and characteristics. The method is based on using a two-level scheme of partitioning of FEM computational tasks, that allows for matching local clusters engaged in computations. This is achieved by a suitable decomposition of FEM meshes into submeshes assigned to separate clusters in the grid. These submeshes are then divided into smaller parts which are distributed among nodes in the same site. Such an approach allows for reduction of a numeber of messages transfererred between sites in the grid. This reduction is essential for improving the performance of a parallel application because of large dealys introduced by communication in widea area networks.

The proposed method for mapping FEM computations onto cluster grid architectures allows for the efficient execution of tasks on grids containing much more processors than in case of using the traditional approach. Moreover, using a fixed numeber of processors, is becomes possible to run efficiently even tasks with a smaller size. In the work, we also propose the performance model which makes possible to estimate benefits of using the proposed approach, in an analytic way, as well as preliminary performace results.

Monitoring of Jobs and their Execution for the LHC Computing Grid

Ralph Müller-Pfefferkorn (1), Reinhard Neumann (1), Torsten Harenberg (2), Matthias Hüsken (2), Peter Mättig (2), Markus Mechtel (2), David Meder-Marouelli (2)
(1) Technische Universität Dresden, Germany
(2) Bergische Universität Wuppertal, Germany

The new generation of high energy physics experiments at the large hadron collider at CERN will be launched in 2007. The urgent need of huge amounts of computing power to analyse its data can only be satisfied using Grid technologies, which are developed and deployed by the "LHC Computing Grid" collaboration. Although still under development, this Computing Grid consists already of over 28000 computers at 180 participating institutes. The experience has shown, that for such a heterogeneous grid of this size and the typical HEP scenario with hundreds of jobs per user a job based monitoring is an essential part both for daily usage and in the recovery of job failures.
LCG itself distinguishes only the following final job states for finished jobs: canceled, aborted, ok (done), ok (failed). Currently, the user can get only limited information about the progress of his job and on the command line only. Furthermore, under certain circumstances, he has even no access to the output of the job when it crashes. The latter makes error identification a difficult and annoying task.
Existing monitoring tools, e.g. Ganglia, keep track of system resources on the worker nodes and identify failures. But all these tools share the disadvantage that they are not job-based, i.e. they are not aware of the current status of the job execution. Additionally, the information produced by these tools is not associated

with a single job, so the end-user has to identify on which computer his job has run and look for problematic system conditions himself.

To overcome these limitations, we developed two new monitoring components.
1. A robust job execution monitor
The monitor runs parallel to a user job and collects information about the system environment, the status of the job and helps to identify possible failure conditions. The information is continuously provided to the user, keeping him informed about the ongoing process. The execution monitor itself consists of two main parts, which both are written in the Python programming language. On the one hand, this is the watchdog component. It supervises some important system resources, e.g. processor load, network traffic or free disk space. On the other hand, there is the script wrapper, which executes a given shell script stepwise and provides logging and analyzing data for every command. All these pieces of information are published via R-GMA in regular intervals and thus accessible to the user on the user interface. This approach provides the possibility to access all relevant information about the user job at any time, especially in case of an error condition. For the user interaction, a graphical tool has been developed.
2. A job and resource usage monitoring
The job and resource usage monitoring is aimed at users and resource providers. It consists of three components - the monitor for information collection and storage, the analyser and the visualiser. The current version is based on the existing LCG job monitoring components, which collect sampling information (e.g. CPU usage, memory usage, status) for single jobs and publish them in R-GMA. Future versions will be extended to provide additional information. In the Analyser the monitoring information is extracted from R-GMA, analysed and published for visualisation. Access to the information is offered by the visualiser - a Web based, graphical and interactive application. It provides the user with graphical representations of the information about his/her jobs in form of histograms, time lines, summaries etc. But these are not just static displays. By clicking into it the user can get detailed information about single jobs (e.g. the CPU usage as a function of time). The visualizer is integrated into a portal framework (GridSphere) to provide a user-friendly web access.

All this work is done in the HEP Community Grid project within the German D-Grid Initiative.

Multimodal Planet Visualization on the Metagrid

Ronan Watson, Soha Maad, Eamonn Kenny, Brian Coghlan
Department of Computer Science, Trinity College Dublin, Ireland

A critical success factor for the grid is its potential to move from a test bed to a production infrastructure. This can be achieved when the grid infrastructure is robust enough to support various compute and data intensive application domains. Moving towards a productivity phase will put the grid in a leading position with respect to its peer technologies and infrastructures. User interfaces, including visualization, will play a major role in this transition.

Here we describe a multimodal planet visualization application developed within a VR-based grid visualization framework. The latter was developed as part of a metagrid infrastructure designed to allow the grid user to transparently access multiple grids while supporting multimodal interactivity and advanced visualization. Whereas the term multimodal has been used in the Human Computer Interaction literature to describe various patterns of interaction with a content (using speech, video, virtual avatars, remote controls), in the context of this paper the term multimodal refers to the ability to accommodate various patterns of uses and objectives.
Our Multimodal Planet Visualization (MPV) grid application is integrated into existing tools and techniques for high end visualization and interactivity with a rendered geographical description of a selected part of the planet of particular interest to the user.
It currently supports: 1) 3D planet navigation; 2) Visualization of grid nodes over the globe - this visualization offers various level of detail starting from a site level down to a single physical machine level; Other possible interactions which our MPV might support would include 3) Earth Sciences 4) Risk assessment (e.g. for earthquake and other natural disasters); 5) travelers assistance in locating their geographical location (here accessibility via handheld device of the rendered scene is key); as well as many other uses that may arise when tackling a rich domain of application.

While the existing tools and techniques for Planet visualization are of a general purpose nature and offer limited compute and data intensive graphical visualization and interactivity, our MPV grid application leverages the power of the grid to offer: 1) more advanced visualization 2) interactivity with the rendered content; 3) and integration with key data intensive graphical applications that would benefit from the use of the grid .

The paper is divided into six sections. The first section overviews our grid visualization framework which is developed as part of a metagrid. The second section describes the background of our chosen application

including existing tools and utilities that our Multimodal Planet visualization depend on. Thirdly we describe the multimodal nature of our application and the various patterns of use that we intend to offer. In the fourth section we describe our experimental set up. The paper concludes with some preliminary results and future work.

Keywords: metagrid, multimodal, application, map, visualization

On Similarities of Grid Resources for Identifying Potential Migration Targets

Georg Birkenheuer, Sven Döhre, Matthias Hovestadt, Odej Kao, Kerstin Voss
Paderborn Center for Parallel Computing, University of Paderborn, Germany

SLAs are important instruments for defining all expectations and obligations within the business relationship between end-user and resource provider. As SLAs define beneath the provider's fee, a penalty for their violation, they pose a business risk for grid providers. For end-users, in case of important computations the economical damage through a failed SLA deadline might be far higher than the SLAs penalty. Therefore determining this risk for SLA failure is of great advantage supporting all stakeholders in their decision processes. The goal of the EC-funded project AssessGrid - Advanced Risk Assessment and Management for Trustable Grids - is to integrate these features into the Grid.

Resource monitoring and periodically updated risk assessments of running and scheduled jobs will at the Grid fabric layer pave the way for introducing risk management into the resource management system's scheduling process. This way, the system will be able to initiate precautionary fault-tolerance mechanisms like checkpointing and migration. If the scheduling system decides on migrating a job to a remote system, the target resource for this job has to be selected carefully. Alternative resources have to comply with the SLA requirements as well as additional ones due to the migration process. If a fitting resource has been identified, the resource provider may start an SLA negotiation on the continuation.

In this context, AssessGrid introduces an information retrieval component, which analyzes similarities and selects potential migration targets. This analysis is based on information provided by monitoring systems within the Grid infrastructure. The focus of this comparison is on static hardware and software characteristics due to their paramount impact on the success of the job migration. Dynamic aspects are also important for an SLA-compliant job migration, particularly regarding the quality of service aspects.

The information retrieval component is already available in a first version. It connects to the GridICE service in order to access published monitoring data within Grid systems. While analyzing the data published by these projects it became apparent that the total number of resources and their Grid composition varies significantly. In consequence, the analysis of similar resources for a given job often points out a strong interdependence between the size of a Grid and the number of potential migration targets. Our analyzing tests compare characteristics like CPU type, clock speed as well as the name or version of the operation system. Similar resources could have a CPU with higher clock speed, more RAM, but have to be of the same processor family. The results show that the number of equal nodes is significantly lower than the number of similar nodes. Furthermore it is shown that often not all compared monitoring data like CPU model or the version of the operating system is available. Accordingly, we started additional evaluations assuming particular non-provided data aspects: equal in one comparing process and different in the other. An interesting result of these tests is that the kind of available resources varies very fluently. Moreover, the results of analyzing equal and similar resources are aging very fast. This underlines the necessity of the AssessGrid information retrieval component.

On the Use of Call Option Contracts on Grid Resources

Anton Bossenbroek, Alfredo Tirado-Ramos, Derek Groen, Dick van Albada
Faculty of Sciences, Section Computational Science University of Amsterdam, The Netherlands

In order to provide uniform views and access to the resources, Grids aggregate them in a highly controlled fashion [1]. In Grids, access is controlled by local policies translated in sharing rules, which clearly defines under which conditions the resources can be used. Resource providers and consumers with the same sharing rules form a Virtual Organization or VO [2]. Examples of VOs are loosely-coupled organizations formed by cycle providers, storage service providers, hospitals and scientific institutes which have to manage and process digital images, and so on. VOs therefore can be considered equal to a real administrative organization, but may also span multiple administrative domains across geographical boundaries. Many issues have to be addressed to effectively share resources in a Grid; such issues differ per type and purpose of the infrastructure to be built.
To effectively serve as an extension for parallel computers, computational Grids have to provide at least the

same service level in regard to optimal resource access as such traditional systems. The most straightforward method to provide such a service level is to let the user choose which resources to use, though depending on the amount of resources needed, such method might be tedious and a chosen allocation scheme might be not optimal for either the user nor the Grid system in total. Complex resource discovery and scheduling services such as [3] may help to circumvent some of these problems. Furthermore, under some conditions [4][5], an optimal allocation scheme where users receive the Grid resources they demand can be found by synthesizing an economic market. A programmed economic market would allow users and Grid resource providers to discover information and voluntarily participate in the trade of resources [6]. Indeed, since economic markets are sociological processes, mathematical models such as the general equilibrium theory of Walras [7], as formalized by Arrow and Debreu [8], may be used to recreate such mechanism in a Grid model where a specific type of contract can be used to allocate Grid computational resources.

We propose a model that builds on existing Grid resource management models such as described in [9]. Our model consists only of the essential components deemed necessary to investigate the use of options in a Grid environment, where resources are allocated using economic models. In this work, we only consider implementations of economic markets in Grid environments where the demand for a Grid resource, determined by demand functions, is driven by a monetary value, which we refer to as Grid Bucks (G$). We propose a model for Grids where resources are considered as commodities that can be traded between resource providers and users, but not among job submission services, in order to avoid arbitrage if different markets exist in the same VO.

By introducing call options, we offer users the possibility to buy call options with the same at exercise time, therefore guaranteeing that a job that needs multiple Grid resources simultaneously can indeed be processed. This way, users or job submission services can use call options to protect the user against the possible negative effects resulting from Grid resource price fluctuations. We describe the rationale for call options to be of interest in Grids where resources are allocated using economic market models. As our work concerns the perceived quality of the Grid in terms job completion, processing time and costs, we introduce three metrics to measures effectiveness and performance. Finally, we present our results, which in this context do not refer to the computing performance of the Grid but rather to the amount of time needed and the budget consumed to complete a job. We implement a discrete event simulation in order to analyze and validate our model.

We assume the price of Grid resources to behave like in a similar way as the price of electricity in electrical Grids, as opposed to Brownian motion. In our model, users can request an option writer to create a call option on a resource. In contrast to common economic research, our model uses the economic theory not to explain a phenomenon, but to induce an economic process.

References

[1] Foster, I., Kesselman, C., Tuecke, S., The anatomy of the grid: Enabling scalable virtual organizations, International journal of high performance computing applications, 15 (3): 200-222, 2001.

[2] Foster, I., Kesselman, C., et al., The grid 2: Blueprint for a new computing infrastructure, Morgan Kaufmann, San Francisco, USA, 2003.

[3] Gao, Y., Rong, H., Huang, J.Z., Adaptive grid job scheduling with genetic algorithms, Future Generation Computer Systems 21 (1), pp. 151-161, 2005.

[4] Pareto, V., Manuel d'economie politique, Paris: Griard et Briere, 1906.

[5] Edgeworth, F.Y., Mathematical physics, London, Kegan Paul, 1881.

[6] Abramson, D., Buyya, R., Giddy, J., A computational economy for grid computing and its implementation in the Nimrod-G resource broker, Future Generation Computer Systems, 18 (8), pp. 1061-1074, 2002.

[7] Walras, L., Elements d d'economie politique pure; ou, Theorie de la richesse sociale, Lausanne: L. Corbaz, 1874.

[8] Arrow, K., Debreu, G., Existence of an Equilibrium for a Competitive Economy, Econometrica, 22(3), 265-290, 1954.

[9] Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., et al., A Resource Management Architecture for Metacomputing Systems, The 4th Workshop on Job Scheduling Strategies for Parallel Processing, 1998.

Online Steering of HEP Grid Applications

Daniel Lorenz, Peter Buchholz, Christian Uebing, Wolfgang Walkowiak, Roland Wismüller
University of Siegen, Siegen, Germany

Online steering and visualization of scientific applications is a well-established method for accelerating research and saving resources. However, in Grid environments no secure online steering tools exist. As a part of the HEP-CG (High Energy Physics Community Grid) project, which is part of the German D-Grid initiative, we are developing a online steering system for Grid applications, which is specifically targeted towards HEP applications used for the ATLAS experiment at CERN.

The steering system is integrated into the ATLAS experiment software by providing special algorithms and services for use with the Athena Framework. Thus, the main functionality of the steering system can be used without changing (or instrumenting) the source code of the Athena framework; only a proper

configuration of the job is necessary. More advanced and/or customized functionality, however, requires a source code modification or additional components.

On the visualization side, our system is integrated into the commonly used visualization toolkit ROOT. Again, no code modification is required, since ROOT provides an extension mechanism based on dynamic class loading. From within ROOT, access to steered data is possible via the ROOT command line for scripting and automated data evaluation as well as via a graphical user interface. We support the monitoring of the most common intermediate results in the ATLAS experiment and the steering of job parameters and job execution.

The steering system is based on a layered architecture with well defined, application independent interfaces between each layer, consisting of a communication layer, a data consistency layer, a data preparation layer and the application layer. The communication layer establishes a secure, interactive connection between the visualization tool and the remote Grid job. Since the worker node, where the job runs on, is generally not known to the steering tool, connection establishment must be based on the job identifier. Furthermore, the interactivity must not compromise the site's security, but has to deal with e.g. firewalls and private IP networks in a suitable way. Therefore a naming service and an additional relay component, installed either on the site's compute element or on some external machine, is needed. The data consistency layer handles the higher-level data exchange between the job and the visualization tool. The data preparation layer is provided to support automated pre-evaluation of the job's data, either in the steering tool or in the job itself. Planned uses of this layer include, e.g., automated checking of intermediate results and error reporting. Finally, the application layer consists all application specific functionality, esp. data acquisition and visualization.

A first prototype of the system (with a simplified communication layer) is already available and is currently evaluated by our HEP group.

Operation and Management Issues in the EGEE/SWE Grid Infrastructure

G. Barreira(1), G. Borges(1), M. David(1), N. Dias(1), J. Gomes(1), J.P. Martins(1), C. Borrego(2), M. Delfino(2), K. Neuffer(2), G. Merino(2), A. Pacheco(2), F. Bernabé(3), J. Fontán(3), J. Lopez(3), R. Marco(4), J. Palacios(5), P. Rey(3)
(1) LIP: Laboratório de Instrumentaçao em Física Experimental de Partículas, Portugal
(2) PIC: Port d'Informació Científica, Spain
(3) CESGA: Fundación Centro Tecnológico de Supercomputación de Galicia, Spain
(4) IFCA/CSIC: Instituto de Física de Cantabria / Consejo Superior de Investigaciones Científicas, Spain
(5) IFIC/CSIC: Instituto de Física Corpuscular / Consejo Superior de Investigaciones Científicas, Spain

The EGEE South-West federation, covering Portugal and Spain, is one of the twelve regions taking part of the LCG (LCH Computing Grid) and EGEE (Enabling Grids for E-SciencE in Europe) grid infrastructures. This paper presents the most important federation operation activities as well as the local and central grid services deployed inside the South-West region. The coordination of all operation issues is handled by a shared ROC (Regional Operations Centre) where the different participating sites take full responsibility for general management tasks, such as the monitoring of grid resources and the resolution of grid problems from the sites and users point of view. Such shared hierarchy is not usually found inside grid infrastructures and enables a faster and more efficient response to the specific needs of users and grid managers. Among the different responsibilities shared between participating sites, a very special emphasis is devoted to the accounting issues, authentication protocols and support mechanisms developed inside the South-West production infrastructure as well as to the new middleware validation tests handled in the South-West pre-production test-bed.
In conclusion, this paper contributes to a better understanding of the LCG and EGEE projects, their fundamental organization and to acknowledge how the different resources work together to deliver high quality services to the users.

Paravirtualization and Operating System Level Virtualization Techniques in Dynamic Grid Environments

M. Pawlik, J. Kwiatkowski
Institute of Applied Informatics, Wroclaw University of Technology, Poland

Incorporation of virtualization techniques into Grid environments can raise maintability, provide better security and make the installation more flexible. In dynamic grid environments the need for checkpointing, process migration and possibility of creation per-user environments make system virtualization even more important. Most of the deployed solutions in this area seems to be currently focused on utilization of

paravirtualizers. Although this technique is indisputably powerful, in many situations lighter virtualization methods are worth considering.

In the Institute of Applied Informatics, Wroclaw University of Technology the need for virtualized grid environment appeared during the deployment of the Cumulus Grid. The Grid creation was motivated by the need for an effective tool that can be utilized for research and educational purposes. Due to the budget limits, the standard approach of creating a typical cluster environment composed from dedicated machines had to be replaced with another, more cost-effective solution. The decision was made to utilize existing computers located in the university laboratories. The dynamic cluster scheme was chosen offering an opportunity to deploy a computational environment at a fraction of costs needed for building a typical installation of dedicated machines. The nature of this solution creates a new level of possibilities but also requires the system flexibility that can be addressed by incorporation of virtualization techniques into the Cumulus environment.

When the Linux operating system is considered, the first solution that usually comes into mind in the context of virtualization is the Xen virtual machine monitor. It highers the security through resource isolation, makes it possible to run multiple operating systems on a single machine, supports checkpointing, migration and live migration to name only a few basic yet powerful of its features.

Unfortunately these possibilities do not come without cost.

Creation of the Xen paravirtualized environment requires serious modifications of the underlying operating system. The Xen incorporation into the official kernel tree seems to be postponed in favor of lightest and more universal mechanisms. Xen architecture limits the number of device drivers available in the virtual machines and introduction of additional software layers between application and the underlying hardware results in the latency growth and limit the number of "guest systems" that can be concurrently run. Strict isolation of operating system instances highers the overall security but at the same time makes the installation less flexible and harder to modify and control.

Because of these drawbacks alternative methods of virtualization gained our attention. Among the mature ones based on the operating system level virtualization, OpenVZ turned out to be the best candidate for the further evaluation.

In the paper an overview of virtualization techniques is presented together with the discussion of the possibilities of their utilization in dynamic Grid environments. The comparison of the most popular approaches makes it possible to select the one that best suits the needs of a particular Grid installation.

"River Socha Project" - Visualization of Massive Amount of Data with Grid Based Engine

Marjan Sterk, Ivan Leben, Gregor Pipan
XLAB Research, Slovenia

Handling and visualizing a large amount of mesh data has always been a challenging field to research and an exciting task to complete. In the past and recent years some very efficient algorithms have been discovered to handle such amount of input while still preserving the necessary speed at computing and visualizing the whole scene. However, every algorithm, even if the complexity is not high, has its limits to how much data it can handle efficiently.

We present here a visualization engine that uses Grid based engine in order to enable smooth moving over a large terrain consisting of millions of triangles. The methods we used are mainly well known and efficient algorithms with logarithmic complexity, but even logarithmic scale starts to raise pretty fast when we come to millions points of input data. That's why we decided to use a Grid in order to solve described problem on a sub-sections of data and then gather the results on the visualization workstation.

The input data consist of several sets. First data set contain points, which are described by X,Y, and Z. These points are provided in very high resolution of up to 100 points on a square meter, which ware acquired by a LiDAR (laser scanning). Second set consist of orto-photo material, which is mapped on the data provided by LiDAR. This data enables us to display more accurate representation of the region visualized. The points we got from the LiDAR scan is stored as a scattered cloud of vertices with no information to connect them, so the first step was to construct an initial mesh out of all the data, which was later saved in geographically region files. Region files are later distributed to the Grid engine, which is able to process request from the visualization workstation, by loading the data in the memory, and run data abstraction algorithms described later, in order to minimize processing and RAM requirements on the visualization side. In addition to the computational power a Grid is used as data storage for various other data to be displayed on top of the landscape.

In the paper we will present two aspects of the application, which are of the concern for Grid deployment. First being the visualization engine, which consist of many different approaches to render the landscape, which have to be implemented in a way to enable distribution, and second being the data distribution algorithm, which has to be capable of distributing the data through the Grid in a way to minimize the maximum load on an individual node.

In the future we plan to provide an interface for a simulation and display of various data, which could be of importance, with emphasis on environmental data (i.e. flooding scenarios).

RMI-Tool - Visualization of Monitoring Distributed Application Based on RMI

Wlodzimierz Funika, Jan Marszalek, Tomasz Kowalczewski
Institute of Computer Science AGH, Krakow, Poland

For the past few years programs written in Java gained great popularity. Together with distributed approach, many applications are exploiting the Remote Method Invocation mechanism (RMI). Since it is sometimes essential to know the exact time of each phase of method invocation, J-OCM - Java oriented monitoring infrastructure was developed. It is a command-line program, up to now it did not provide a GUI, thus it can be hard to understand by people who are not experts in J-OCM messages. This motivated the development of RMI-Tool - a visualizer plug-in module for J-OCM, which can be used to visualize activities within other applications as well.

One of the most important tasks of the tool under discussion is to provide performance visualization displays rather than raw monitoring data coming from J-OCM. Since J-OCM focuses on capturing important RMI-relevant events within the application execution, our task is concentrated on proper processing these events and displaying results of RMI phases. One may wish to launch several different remote methods at one time and monitor them, so the tool should handle this case as well.

In this paper we focus on the issues of monitoring the remote method invocation and designing its performance displays. Raw data, which are captured by J-OCM are not sufficient to understand performance problems of the application, so it was necessary to develop a kind of intuitive user-friendly interface. We use J-OCM services to access RMI-bound monitoring data. It is also possible to connect the tool to other sources of data (i.e. files) or even another monitoring application that uses the OMIS specification.

Acknowledgments. The research is partly supported by the EU IST K-Wf Grid project.

References:
[1] W. Funika, M. Bubak, M.Smetek, and R. Wismuller. 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.

Secure Remote Execution

Katrin Shechtman, Maxim Vainstein, Emanuel Hachamov and Michel Bercovier
School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel

This paper introduces Secure Remote Execution (SRE), a platform for GRID based remote tasks execution. SRE implements device driver architecture, its design allows easy and secure remote execution using existing GRID based computation platforms based on Windows OS. The implementation is based on kernel device driver programming using C/C++, and the modules include process management, network authorizations, management of file access rights as well as temporary files removal, registry protection, network and other system objects filtering. The aim is to provide a robust, platform independent, easy-to-use and easy-to-integrate security platform that does not perturb or restrict scheduled task execution and at the same time to keep a server machine secure as well.

SRE is based on a virtualization concept, where all remote tasks execution are done in separate execution environment that simulates unrestricted access to the system resources and at the same time protects the remote machine from corruption by any data or events that live inside the execution environment. More specifically, SRE concept is based on redirection, inheritance and component independence. Redirection principle in our case consists of taking both objects and output destined for one location and send it to another totally transparent to appropriate request generator. A location, where things are directed to, may be configurable parameter of the system. As to inheritance SRE expends set of characteristics that can be inherited from prior defined objects by dividing them into 'collections'. Each collection may have different properties and goals. Once some object is in a specific collection and an inheritance relation on that

collection is defined, all objects that inherit from it, will be in this collection as well. For example, if a user runs some process within SRE, all its child processes can be configured to inherit SRE settings applied to this process. This way, all those processes will run in SRE with no need of explicit request. Component independence is a fundamental and vital principle for systems such as SRE. The aim is to isolate different OS components, and treat each one of them separately. On one hand, it adds a great flexibility to the system, from the other hand each component can be implemented efficiently avoiding unnecessary overhead due to generic attitude. As for the user he is provided with a unified view of the system and as such is completely unaware of the process isolations.
SRE extends existing security concepts of running processes safely on some machine. SRE plus is to provide users with the capacity to run something on his/her computer without compromising its security, moreover it also acts as some sort of barrier between running processes and the underlined machine by way of protecting process data and activities. From SRE point of view, security challenges are symmetric. Once some object does not belong to SRE related collection, it can nor view nor modify any data and cannot affect the behavior of objects coming from this collection.
Generally SRE consists of two subsystems, one resides in kernel mode and the second one is user mode based. The first subsystem involves device driver mentioned above, the second includes a set of utilities for SRE management and maintenance. The device driver is fully managed by SRE manager, which has an ability to set different SRE parameters and to configure SRE system. SRE has a set of interesting state-of-art features such as dynamic configuration. All configuration parameters can be changed dynamically during SRE processing. It includes collection definition, filtering related parameters and logging activity. .SRE is able to log all suspicious actions taken by processes running within SRE and/or underlined machine actions intervening in SRE.

Key words: GRID, Parallel, Distributed, Security, Windows

Security and User Interactions in Integrating Legacy Experiment Environment to Scientific Workflows

Zhiming Zhao, Dmitry A. Vasunin, Adianto Wibisono, Adam Belloum, Cees de Laat
Faculty of Science, University of Amsterdam, the Netherlands

Grid environments enable collaborations involve large scale resources and people, and make data and computing intensive application feasible. Using Grid infrastructure, Scientific Workflow Management Systems (SWMS) are an important guise of Problem Solving Environment (PSE). A SWMS explicitly models the dependencies between experiment processes, and orchestrates the runtime behaviour of involved resources according to a flow description. Data flow, control flow or Petri net based mechanisms have been used to model scientific workflows. Wrapping and integrating different types of software components, in particular legacy environments, in scientific workflows is a crucial to realise an effective SWMS. This paper is about integrating legacy domain specific experimental environment in workflow systems, and is carried out in the context of Dutch Virtual Laboratory for e-Science project.

VL-e is an e-Science project driven by applications; it aims to realise a Grid enabled generic framework via which scientists from different domains\footnote{Currently six applications are included in VL-e: food informatics, medical diagnosis and imaging, bio-diversity, bio-informatics, high energy physics, and tele-science.} can share their knowledge and resources, and perform domain specific research. Life science is an important field in the VL-e project; it currently has three sub programs(SPs) in bio-informatics, bio-medical and bio-diversity respectively. The bio-informatics SP focuses on semantic integration among different data sources. One of the typical use cases is to integrate genome distribution (chromosome locations) of histone and transcription factor and to unravel the genetic background of special phenomena, e.g., diseases. Supporting large scale data integration and enabling interactively data annotation and discovery are highly demanded in the workflows in this SP. The bio-medical SP is interested in automating the routines in analyzing complex medical images. Scheduling massive tasks and steering the computation are a main concern in the experiments. Moreover, storing and accessing large volume and distributed medical data with different levels of security and authorization control are also important issues. In the bio-diversity SP, tuning models for different eco- phenomena and using the models for forecasting is a main experiment scenario. Farming the massive computing tasks and optimizing the parameter space are the important requirements. Computing facilitates for statistics and visualisation are highly demanded in a number of VL-e application. One of the tools being heavily used in daily experiments is R.

As an important experimental environment in bio-sciences, R realises rich functionality of data statistics and visualisation. Including R in scientific workflow enables distributed computing of R scripts on Grid resources, moreover scientists can coordinate the execution of legacy R code together with the other software components in a larger size experiment. An R environment has been included in a number of SWMS, such as Kepler and Taverna, in which a specialised R component is provided and an R engine is integrated with a scientific workflow at run time via a conventional TCP Socket or Web service based interface. In such integrations, managing security issues in the different sessions of R execution is highly demanded by the

both application scientists and underlying Grid platform administration polices. Interactive visualisation of the intermediate results is crucial to realise the inclusion of human expertises in the workflow execution; managing remote X display of the R environment has to take the execution of the workflow into account. These issues are important for realising an effective workflow, however current SWMSs have not explicitly provided solutions. In this paper, we will extensively discuss these issues and present our solutions.

Acknowledgment. This work was carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl). Part of this project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC$\&$W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).The authors of this paper would like to thank all the members in the VL-e SP2.5.

Semantic Binding Specifications in S-OGSA

Oscar Corcho, Pinar Alper, Paolo Missier, Ioannis Kotsiopoulos, Ian Dunlop, Sean Bechhofer, Carole Goble
School of Computer Science, University of Manchester, Germany

Semantic-OGSA (S-OGSA) [1] is an architecture that extends OGSA to support the explicit handling of metadata in Grids. The S-OGSA model introduces a novel type of Grid resource called Semantic Binding, which contains explicit metadata about one or several Grid Entities (resources or services) and relates that metadata to one or several Knowledge Entities (normally ontologies). S-OGSA also identifies a set of associated knowledge services that support a spectrum of service capabilities, including ontology services, reasoning services, metadata services and annotation services. Finally, it proposes a set of mechanisms to handle and deliver the explicit metadata available in the form of Semantic Bindings.

Among the aforementioned services, the metadata service plays a central role in the architecture, as it is responsible for storing and allowing access to explicit metadata, that is, Semantic Bindings. S-OGSA defines different layers of functionalities for this service, including:

- Basic functionalities for the creation, access and querying of explicit metadata related to one or to several Semantic Bindings. Since Semantic Bindings are related to Knowledge Entities, querying them may require reasoning involving not only the metadata itself but also a combination of the metadata and the content of those Knowledge Entities.

- Advanced functionalities related to the state and lifetime management of Semantic Bindings. The lifetime of a Semantic Binding is defined as the time interval between its instantiation and its destruction. While the existing specifications about resource lifetime focus mainly on the means by which resources can be destroyed, in S-OGSA we also describe the means by which a Semantic Binding can be updated and destroyed, as well as the means by which the lifetime of a Semantic Binding can be monitored.

All these functionalities have been implemented, in the context of the EU OntoGrid project, using Globus Toolkit 4. The implementations extend, reuse and realise the WS Resource Framework specifications, including WS-ResourceProperties (WSRF-RP), WS-ResourceLifetime (WSRF-RL), WS-ServiceGroup (WSRF-SG), and WS-BaseFaults (WSRF-BF). In this paper we will describe the current status of these implementations, together with the assumptions and design decisions made for them, and will show how they are being used in several of the OntoGrid use cases.

[1] An overview of S-OGSA: a Reference Semantic Grid Architecture. Corcho O, Alper P, Kotsiopoulos I, Missier P, Bechhofer S, Goble C. Journal of Web Semantics 4(2):102-115. June 2006

Social MetaGrid Agents: an Approach for Flexible Resource Allocation in Heterogeneous Grid Middlewares

Gabriele Pierantoni, Brian Coghlan, David O'Callaghan, Oliver Lyttleton, Eamonn Kenny, Soha Maad, Geoff Quigley
Department of Computer Science, Trinity College Dublin

Among the challenges that the Grid Community is facing in these days, two are particularly hard to tackle. The first is posed by the existence of many different Grid middlewares, while the second is posed by the many different ways in which actors and resources can be arranged in a Grid. The first challenge is dealt with by a field of research called Grid interoperability, which tries to design systems capable of allowing different Grid middlewares to interoperate with each other and also to be harnessed in heterogenous groups. The second challenge is tackled by a field of research focusing on the design of systems for the allocation and sharing of resources.

In this paper we propose a combined approach to both problems which is based on the marriage of two concepts: a MetaGrid concept that tackles the issue of Grid interoperability, and a concept of Social MetaGrid Agents that tackles the issue of resource sharing and allocation.

A MetaGrid is to Grid middlewares what Grid middlewares are to Operating Systems and applications. It offers a platform where different Grid middlewares can coexist and interact and where a user can utilize multiple different Grid middlewares without being bothered by all the underlying technicalities. MetaGrid is the result of an ongoing research that started with the design and implementation of a system capable of using three different Grid middlewares: LCG2, GT4 and WebCom. After a first prototype was developed and tested, a more comprehensive approach led to the current design. The principal components of the MetaGrid are workflow engines, Grids, a transport layer, native MetaGrid services and value-added services.

On the other hand, Social MetaGrid Agents are components that use the MetaGrid services to offer the user a flexible way to find optimal bundles of resources to run their jobs. Social MetaGrid Agents also allow different users to interact with each other in a variety of ways in relationships that range from competitive to co-operative. Social MetaGrid Agents try to mimic (in the Grid world) parts of the human behaviour that are the basis of competitive and co-operative relationships in societies.

We present the treatment of MetaGrid components as Social MetaGrid Agents. In particular, two native MetaGrid services, a Job Exchange and a Security Exchange, are presented in this way. The paper also offers a mathematical approach for the description of the different MetaGrid services. This formalism can be used by the Social MetaGrid Agents to represent the set of resources available. Finally, first results obtained by such system are summarized.

SPARQLing UNICORE

R. Menday, A. Streit
Central Institute for Applied Mathematics, Research Centre Jülich, Germany

DEISA is a consortium of leading national supercomputing centres that deploys and operates a persistent, production quality, distributed supercomputing environment of tera-scale performance and with continental scope. DEISA is a heterogeneous environment, and uses UNICORE as a uniform interface over these resources. The graphical UNICORE client, is complimented by a number of alternative client options, such as command line environments and web interfaces. This paper considers a web interface to UNICORE resources, applying techniques from the Semantic Web. Such a web interface for DEISA leads to a HTTP façade for both machine as well as human consumers, offering many new interaction possibilities for users.

Following the REST architectural guidelines using the HTTP protocol, we model the underlying Grid as a HTTP URI-space of resources with a rich system of links between them. Strong emphasis is placed on this careful organisation of resources at a site, and assigning significant resources unique URIs. Furthermore, the links between the URI resources can be semantically enriched using Resource Description Framework (RDF) to describe the nature of the relationship.

UNICORE provides a normalised, abstract view of resource description using terms defined by the Abstract Job Object (AJO). Therefore some translation is necessary for this to be published as a RDF knowledge base. However, the translation process is relatively straightforward since the UNICORE AJO is naturally amenable to the graph structure of RDF and the Web Ontology Language (OWL). The resulting Ontology defines a vocabulary for the Grid for describing resources and the relationships between them. The HTTP facade maintains this RDF model of the information from the underlying UNICORE Grid, providing a rich source of data to be queried.

We consider gradients of information exposure, dependent on the client. For example, the grid `window shopper' looking for new machines to buy time on requires something like a Grid search engine. Conversely for authenticated users we provide more detailed information, including running jobs, etc. This source of information can be queried, and we offer a SPARQL endpoint for querying this model. It is possible to serialise the result of a SPARQL query in the Javascript object notation (JSON), which is straightforward to manipulate within the Javascript environment of the client web browser. Using common web techniques, we demonstrate the process of styling and re-using this information to build customised web pages and information dashboards for human clients. This includes, for example, the integration with the mapping capabilities of online mapping services such as Google Maps.

We briefly consider how the semantically enriched HTTP facade assists the user with her resource and data management tasks. Finally, at the end of the paper we preview future work where we examine the possibilities of applying the `web style' architectural approach to the actual middleware running the Grid.

StoRM: A SRM Solution on Disk Based Storage System

E.Corso, S.Cozzini, A.Forti, A.Ghiselli, L.Magnoni, A.Messina, A.Nobile, A.Terpin, V.Vagnoni, R.Zappi
INFN-CNAF, Bologna, Italy
EGRID Project, The Abdus Salam ICTP, Trieste, Italy

From the beginning of the Grid one of the main purposes was to provide a way to share geographically distributed heterogeneous resources, giving the illusion to the user to work on a local system. In a data Grid, in which more emphasis is given on data intensive application that produce and read large volumes of data, the need to provide an efficient management of the storage resources become fundamental.

A Storage resource can be composed by different storage systems: disk only systems, tape archiving systems, or by a combination of both. The basic logical entities of a storage resource are space and file. Space must be allocated when a new file have to be stored into a storage resource, and files could be dynamically removed to create the necessary space. This is the main goal of Storage Resource Managers (SRMs), middleware services whose function is to provide dynamic space allocation and file management of shared storage components. SRMs services agree on a standard interface to hide storage dependent characteristics and to allow interoperability between different storage systems.

In HEP community high performance storage resources based on disk storage solutions using parallel file systems are becoming increasingly important to provide reliability and high speed I/O operations needed by HEP analysis farms.

In this article we describe the StoRM project, an implementation of the Storage Resource Manager interface version 2 for disk based storage solutions. StoRM is designed to take advantage from native parallel file systems, but also standard POSIX file systems are supported. StoRM provides space reservation capabilities and uses native high performing POSIX I/O calls for file access. Besides standard Grid protocols, also the file protocol is supported. This allows applications to perform a standard POSIX operation without interacting with any external service that emulates data access, improving performance when the underlying file system is efficient. Therefore, an application executed on a grid system can be adapted without any change in data access pattern.

Another important driving feature of StoRM is security. Security concerns user authentication, request authorization and the permission enforcement on the storage resource. In StoRM authentication is based on VOMS certificates, different authorization models are supported (using plug-ins) and file system ACLs (Access Control Lists) are used to enforce permission. This authorization model is also designed to cater for the interests of Economics and Finace as represented by the EGRID Project, given that security is an important driving requirement.

Here we present a solution based on GPFS file system from IBM. It is a parallel file system highly scalable in number of nodes and disks, that offers POSIX semantics, allows for large volumes creation and provides extensive fault tolerance and recovery possibility.

SZTAKI Desktop Grid - a Hierarchical Desktop Grid System

P. Kacsuk, A. Marosi, J. Kovacs, Z. Balaton, G. Gombas, G. Vida, A. Kornafeld
MTA SZTAKI, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary

Originally, the aim of the researchers in the field of Grid was that anyone could offer resources for a Grid system, and anyone can claim resources dynamically, according to the actual needs, in order to solve a computationally intensive task. This twofold aim has been, however, not fully achieved. Currently, we can observe two different trends in the development of Grid systems, according to these aims.

Researchers and developers in the first trend are creating a Grid service, which can be accessed by lots of users. A resource can become part of the Grid by installing a predefined software set (middleware). The middleware is, however, so complex that it needs a lot of effort to maintain. Therefore it is natural, that single persons do not offer their resources but all resources are maintained by institutions, where professional

system administrators take care of the hardware/middleware/software environment and ensure the high-availability of the Grid. Examples of such Grid infrastructures are the largest European Grid, the EGEE (Enabling Grids for E-SciencE) and its Hungarian affiliate virtual organisation, the HunGrid, or the NGS (National Grid Service) in the UK. The original aim of enabling anyone to join the Grid with one's resources has not been fulfilled. Nevertheless, anyone who is registered at the Certificate Authority of such a Grid and has a valid certificate can access the Grid and use the resources.

A complementary trend can also be observed for the other part of the original aim. Here, anyone can bring resources into the Grid system, offering them for the common goal of that Grid. Nonetheless, only some people can use those resources for computation. The most well-know example, or better to say, the original distributed computing facility example of such Grids is the SETI@home [1]. In Grids, similar to the concepts of SETI@home, personal computers owned by individuals are connected to some servers to form a large computing infrastructure. Such systems are called with the terms: Internet-based distributed computing, public Internet computing or desktop grid; we use the term desktop grid (DG) from now on. A PC owner should just install one program package, register herself on the web page of the Grid system and configure the program by simply giving the address of the central server. Afterwards, the local software runs in background (e.g. as a screensaver) and the owner does not need to take care of the Grid activity of her computer. In a desktop grid, applications can be performed in the well-known master-worker paradigm. The application is split up into many small subtasks (e.g. splitting input data into smaller, independent data units) that can be processed independently. Subtasks are processed by the individual PCs, running the same executable but processing different input data. The central server of the Grid runs the master program, which creates the subtasks and processes the incoming sub-results.

The main advantage of a desktop grid is its simplicity thus, allowing anyone to join. The main disadvantage is that currently only problems computable by the master-worker paradigm can be implemented on such a system. Desktop grids have already been used at world-wide scales to solve very large computational tasks in cancer research [2], in search for the sign of extraterrestrial intelligence [1], climate predictions [3] and so on.

Desktop grids can be used efficiently and conveniently in smaller scales as well. We believe that small scale desktop grids can be the building blocks of a larger Grid. This is a new concept that can bring closer the two directions of Grid developments. It is easy to deploy desktop grids in small scale organisations and to connect individual PCs into it therefore we get a grid system that can spread much faster then heavy-weight grid implementations. On the other hand, if such desktop grids can share the resources and their owners can use others' desktop grid resources, the many user concept of the other trend is also realised. A major step towards the collaboration of desktop grids is the support of hierarchy of desktop grids within a large organization or community.

SZTAKI Desktop Grid (SZDG) is designed and implemented to realize this idea. SZDG is based on BOINC but significantly extends the client concept of BOINC in order to enable the creation of hierarchical desktop Grids.

Imagine a university where the needs of departments can be satisfied by using the basic SZTAKI Desktop Grid. All PCs of a department can be connected into one local (department level) DG system and distributed projects can use all these resources. It is natural to ask, what if there are several departments using their own resources independently but there is an important project at a higher organisational level (e.g. at a school or campus level of a university). Having the previous set-up in the departments, only one of the departments can be selected to run the project. Of course, the ideal would be to use all departments' resources for that project. Besides again developing something new component (e.g. a broker) to control over the different desktop grids, there is the possibility to build a hierarchy of desktop grids. In such a hierarchy, desktop grids on the lower level can ask for work from higher level (pull mode), or vice versa, desktop grids on the higher level can send work to the lower levels (push mode).

SZTAKI Desktop Grid supports the pull mode, as this is the original way how desktop grids work. The control of important work on the higher level can be realised with priority handling on the lower level. A basic SZTAKI Desktop Grid can be configured to participate in a hierarchy, that is, to connect to a higher-level instance of SZTAKI Desktop Grid (parent node in the tree of the hierarchy). When the child node (a stand-alone desktop grid) has less work than resources available, it asks for work from the parent. The parent node can see the child as one powerful client.
Of course, the BOINC-based server has to be extended to ask for work from somewhere else (i.e., behave similarly as a client) when there is not enough work locally. Fortunately, this can be done separately in the case of BOINC. Work units are generated by the running applications and they are put into a database of the BOINC server. Whether a work unit arrives from outside or from a local application, it does not matter. Therefore, it is enough to create a new daemon on the server machine that observes the status of the desktop grid. When client machines' requests for work are rejected – or when the daemon predicts that this will happen soon – the daemon can turn to the parent desktop grid and ask for work units. The daemon behaves towards the parent as a BOINC client, asking for work and reporting results. However, it puts all those work units into the database of the local server thus, client machines will process them and give the

results. The daemon should also wait and look for the incoming results and send them back to the parent.

The paper will describe in detail the principles and implementation of the hierarchical desktop Grid.

References
[1] D.P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, D. Werthimer: SETI@home: An Experiment in Public-Resource Computing, Communications of the ACM, Vol. 45 No. 11, November 2002, pp. 56-61
[2] United Devices Cancer Research Project: http://www.grid.org/projects/cancer
[3] D. A. Stainforth et al.: Uncertainty in the predictions of the climate response to rising levels of greenhouse gases, Nature, 27 January 2005, vol 433.

The IGOR File System for Efficient Data Distribution in the GRID

Kendy Kutzner and Thomas Fuhrmann
System Architecture Group, University of Karlsruhe, Germany

This work is funded under EU grant number 511438 ("SIMDAT").

Many GRID applications such as drug discovery in the pharmaceutical industry or simulations in meteorology and generally in the earth sciences rely on large data bases. Historically, these data bases are flat files on the order of several hundred megabytes each. Today, sites often need to download dozens or hundreds of such files before they can start a simulation or analysis run, even if the respective application accesses only small fractions of the respective files.

The IGOR file system ("IGOR-FS"), which has been developed within the EU FP6 SIMDAT project, addresses the need for an easy and efficient way to access large files across the Internet. IGOR-FS is especially suited for (potentially globally) distributed sites that read or modify only small portions of the files. IGOR-FS provides fine grained versioning and backup capabilities; and it is built on strong cryptography to protect confidential data both in the network and on the local sites' storage systems.

IGOR-FS is based on the IGOR overlay network, a Chord [1] variant which combines peer-to-peer technology and service orientation. With IGOR, all participating machines automatically organize into a structured virtual overlay network. Resources and services are addressed via hashed application keys. Furthermore, IGOR is proximity and quality of service aware, that is, it prefers a responsive or local resource over an unresponsive or remote resource, both when building the overlay network ("proximity neighbour selection") and when routing requests ("proximity route selection") [2].

The IGOR file system uses the FUSE technology [3] to be mounted into the Linux file system name space. Thereby, it converts all file system structures such as folders, links and the files itself on the fly into variable size blocks, which can be easily sent across the IGOR network. If a mounting site knows both the block identifier and the decryption key of a file block or a folder it has read access. If it authenticates as source site for that file or folder, it has write access, too. Reading and writing may affect only a small fraction of the entire file system. Hence the network traffic is greatly reduced. Furthermore, IGOR-FS can store several versions of such blocks and thereby efficiently reproduce several consistent versions of the entire file system. Moreover, sites may cache the individual blocks to speed up further access to the same part of the data. This eliminates bottleneck problems that are typical for centralized data distribution approaches.

IGOR-FS is available for experimental use. It has been successfully employed with large bioinformatics data bases within the SIMDAT project.

[1] Stoica et al. "Chord", SIGCOMM 2001
[2] Gummadi et al. "On the impact …", SIGCOMM 2003
[3] http://fuse.sourceforge.net/

Toward Cost-Effective, Enterprise-Class Grid Storage Services

Maciej Brzezniak (1), Tomasz Makiela (1), Norbert Meyer (1), Angelos Bilas (2)
(1) Poznan Supercomputing and Networking Centre, Poland
(2) Institute of Computer Science (ICS), Foundations for Research and Technology - Hellas (FORTH), Heraklion, Greece

Cost-effective storage systems that can scale to large capacities and high-performance are required for supporting an increasing number of data-intensive applications and services in data centres. Traditionally, high-end, scalable storage systems have relied on custom, storage technologies, such as Fibre Channel. However, recently, new storage technologies have started to emerge that rely extensively on commodity interconnect, disk, cpu, and memory technologies, however requiring substantial systems software support.

FC-based systems are composed of custom storage devices i.e. FC disk matrices with FC interfaces and FC or (S)ATA disks as well as FC network components: FC switches, FC Gigabit-speed links and FC interfaces in the client hosts. FC matrix controllers and FC switches are managed by custom software. Client hosts do not need any special software to access the disk volumes except FC HBA drivers. On the contrary, commodity-based approaches rely on the hard disk drives connected to PCs that act as controllers. Controllers run regular operating systems (e.g. Linux) and special system software (e.g. kernel-level drivers) that provides storage services. Client hosts use another piece of the system software in order to exploit these services. Controllers and clients are interconnected by popular network technology (e.g. Gigabit Ethernet).

This paper contrasts the FC-based and commodity, PC-based approaches to building scalable storage systems. We examine how the traditional, aggressive approach is currently used in typical installations and the features that make it successful. Then, we contrast it with the commodity-based approach and we examine how features of aggressive systems may be provided on top of this new architecture. Our goal is to categorize and discuss the perceived gap between aggressive and commodity-based systems and to expose the issues that need to be addressed before commodity-based storage systems can be used in a wider range of applications in data centres and Grids.

We discuss the general architecture of Fibre Channel-based and commodity systems as well as their typical applications. In particular, we focus on contrasting the two approaches in terms of: (a) Capacity and performance issues from the perspective of their most important resources: disks, CPUs, and memory. (b) Logical and physical scalability of the whole system as dictated by the interconnect used in each architecture. (c) Availability and reliability, taking into account durability and redundancy features of storage nodes and devices. (d) Security considerations and specifically, traffic separation, mutual authentication and authorisation of network elements, as well as access control on the storage nodes and in storage clients. (e) Finally, we briefly mention, practical challenges related to management and thermal, electrical, and spatial density features since they arise in both types of storage systems and they influence the total cost of the infrastructure.

Virtual Organization Approach for Running HEP Applications in Grid Environment

L. Skital (1), L. Dutka (1), R. Slota (2), K. Korcyl (3), M. Janusz (2), J. Kitowski (1,2)
(1) ACK Cyfronet AGH, Cracow, Poland
(2) Institute of Computer Science, AGH University of Science and Technology, Cracow, Poland
(3) Institute of Nuclear Physics PAN, Cracow, Poland

Interactive and real-time applications put requirements, which are difficult to fulfill by most of current production grid middleware. Typically, the only interactivity, which is offered by the middleware is ability to submit a job and get output or cancel the job. The job submission process takes itself minutes to start the job, which is unacceptable for most real-time applications. Also any communication between the user and the job is difficult due to local security polices of service providers. The problem of interactivity using grid environment has been already addressed elsewhere. One of the good examples is the EU IST Crossgrid project [1]; more critical requirements are now formulated for some applications of the EU IST int.eu.grid project [2]. One of its pilot real-time applications concerns High Energy Physics (HEP). The application is based on ATLAS system, which is designed to be highly available, making use of monitoring, calibration and filtering system for the LHC experiment. The application is empoyed as the third level of event filtering and has to be capable to process about 3500 events per second, ~1.5MB each. Originally, the ATLAS system is designed to use local computer farms which run Processing Tasks (PT). During int.eu.grid project the application will be extended to take advantage of PT running on grid resources.

In the paper we present exemplary Virtual Organisation (HEP VO), which provides environment for pilot real-time HEP application. This VO will be build on top of the unmodified gLite middleware.

HEP VO is responsible for providing stable environment capable to process the desired event throughput. To perform this task we propose dynamic VO with two phase certification/monitoring process of each site. During the first certification phase sites are tested against ability to support HEP VO for a long period of time. They define Low-level Virtual Organisation. The second phase is run-time monitoring. Amount of resources in run-time environment is dynamically controlled by application, defining High-level Virtual Organisation. Resources can be excluded from the environment if they fail or if they are not needed. Failing resources are excluded until the problem is solved.

During both phases similar tests are performed. However, the sites which are not certified (i.e., failed in passing the first phase) cannot process the production data. HEP VO has to assure proper environment for the application. There should be up-to-date application software with calibration data on each participating site. There is also a need for verification mechanism to assure data security and computation

results correctness. To perform these tasks the HEP VO environment has to be equipped with both infrastructure monitoring and application monitoring services.

In the paper we present a detailed description of VO for interactive/real-time HEP application. The approach covers aspects like site certification and monitoring, software/database distribution, LRMS configuration and communication channels.

Acknowledgement.
The work has been supported by EU IST 031857 int.eu.grid project.

[1] The Crossgrid project http://www.crossgrid.org/
[2] The int.eu.grid project http://www.interactive-grid.eu/

Virtualized Access to the Grid Computational Resources

Michal Jankowski (1), Pawel Wolniewicz (1), Jiri Denemark (2), Norbert Meyer (1), Ludek Matyska (2)
(1) Poznan Supercomputing and Networking Center, Poland
(2) Faculty of Informatics, Masaryk University, Brno, Czech Republic

Controlled and secure access to grid computational resources requires authentication, authorization, an adequate level of job isolation, accounting and possibility of auditing user actions. These basic features should be realized with as little administrative effort as possible, though providing the administrators and Virtual Organization (VO) managers with enough control on their resources and users. In the grid environment, that comprises large number of users and resources in different administrative domains, these features are challenging. The requirements of the users and administrators are still more sophisticated: checkpointing and migration of jobs, detailed software requirements, quality of service, collaborative work and load balancing. From the user point of view, the whole Grid should be seen as a single computer with appropriate software, hiding all the technical details connected with physical locations, middleware, operating systems, etc. Virtualization, a quite old concept in computer science, appears also in the Grid computing. In the following presentation we compare two virtualization models that may help in resource management: virtual accounts and virtual workspaces. Both of them have different virtues and drawbacks and both are suitable for different purposes. We discuss Virtual Workspaces (VW) that implement the two models and provide webservice for managing them. The Virtual Workspaces require explicit workspace management: creation, setting time to live and destroying. This complicates integration with existing brokers or requires additional actions from the user. We postulate a higher level of virtualization - virtualized access to the resources that encompasses automatic, transparent workspace management. Moreover, support for accounting and audit must be provided. We propose a framework for resource access control that reuses Virtual Workspaces implementation and other already existing tools providing more transparent access to the resources and support for accounting and audit.

Workflow Interoperability in Grid-based Systems

Moustafa Ghanem(1), Nabeel Azam(1), Mike Boniface(2)
(1) InforSense Ltd, London, UK
(2) IT Innovation Centre, University of Southampton, UK

The use of workflow technology within grid-based computing has received a lot of attention recently. Informally, a workflow is an abstract description of the steps or tasks required for executing a particular real-world process, and the flow of information between these tasks. Work passes through the flow from start to finish and activities might be executed by people or by system functions. Within a grid computing environment, a workflow provides the mechanisms for constructing distributed end user applications through the composition of distributed data services and computational services. Workflow systems typically enable users to construct their workflows using a visual interface and to submit them for execution by an execution engine that controls the invocation and data transfer between the different services.

Within large collaborative projects, it is essential to admit the existence of multiple workflow systems that are already in use by different partners and the fact these tools are used for addressing different requirements in different organizations. A clear example can be seen in the EU-funded SIMDAT project which involves a consortium of 29 research and industrial partners. The project is developing grid-based technology for enabling large-scale industrial product design and focuses on four application areas: product design in the automotive, aerospace and pharmaceutical industry as well as service provision in meteorology.

Within SIMDAT, different industrial and academic partners have in place their own workflow solutions. These range from hard-coded workflows written using scripts to using one of three workflow engines

(Taverna/Freefluo, InforSense KDE and LMS Optimums). Although these workflow systems may look superficially similar, each has been designed and optimized for different requirements. Rather than attempting to enforce a single workflow system on all partners, we foster an approach that admits such heterogeneity and enabling run-time interoperability between these different workflows.

In this paper, based on our experience within the SIMDAT project, we discuss the main requirements for developing and using cross-origination grid-based workflows for engineering applications and product design. Our examples are based on the automotive, aerospace and pharmaceutical sectors and are used to provide comparison between the different workflow systems used within the project. We also describe how the different workflow systems used within the project have been modified to coordinate the execution of remote grid services in addition to remote web services. Our experience here is based on using GRIA as a middleware for supporting Grid service composition.

We then describe our experience in developing and using a generic approach for achieving interoperability between the workflow systems across-organizational boundaries. The conceptual approach is generic and enables us to wrap remote services as grid services and also to wrap remote workflow engines as services to which workflow descriptions can be submitted. Finally, we describe some of the higher-level issues and problems that exist when multiple workflow systems exist within the same environment and provide suggestions for addressing them.