Papers accepted for CGW'05 /in alphabetical order by titles/

A Case Study of Interoperability in a Distributed Tools Environment

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

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

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

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

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

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

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

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

Bibliography:

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

A "Plug and Play" CA for Grids

Martin Hoch (1), Bernd Schuller (2), Roger Menday (2)
(1) University of Applied Sciences, Wuerzburg
(2) Central Institute for Advanced Mathematics, Research Centre Juelich, Germany

To deploy and manage a Grid based on current middleware such as UNICORE, a solution for user registration, certification, and certificate management is needed.
During our experience in project and production use of Grid middleware, we have repeatedly noted that most difficulties in deploying and running a Grid solution are caused by the security infrastructure, Solutions based on standard toolkits such as OpenSSL are widely used. However, these are typically hard to configure, and lack a convenient front end where the user can register and submit his certification request.

Therefore, we have designed and implemented a easy-to-use and easy-to-manage software consisting of a registration authority web frontend (RA) and a certificate authority (CA). The software is written in pure Java, and can be operated in true "Plug and Play" fashion with minimal extra configuration.

Other deployment scenarios such as running the CA on a separate machine (even without network connection) are also possible.

AAB - Automatic Service Selection Empowered by Knowledge

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

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

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

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

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

Adapting the Development Model of the Grid Anatomy to Meet the Needs of Various Application Domains

Soha Maad, Brian Coghlan, Gabrielle Pierantoni, Eamonn Kenny, John Ryan, Ronan Watson
Trinity College Dublin, Ireland

The grid is penetrating various application domains. Two implications arise: the grid is reshaping our programming models for various application domains; and the development and deployment of grid middleware is reshaped by the needs of various application domains. Whereas the implementation of the grid anatomy (Foster et al, 2001) (Foster et al, 2002) succeeded so far, to some extent, in realizing the Virtual Organization (VO) concept concerning the security, the workload management, and the discovery and monitory of grid resources (Czajkowski et al, 2004) (Foster et al, 2005), work remains to be done in adapting the implementation of the grid anatomy to meet the needs of various application domains. The key issue addressed is the software engineering practice adopted in grid middleware development. Firstly, the paper surveys the penetration of the grid in nine application domains (physics, medicine, astronomy, environment, engineering, media, chemistry, finance, and government). This survey highlights, for each application domain: 1) the challenges faced; 2) the technology needed to address the challenge; 3) the limitations of the existing solutions; 4) the grid solution; and 5) the current limitations of the grid solution. Based on this survey, the paper, secondly, discusses the impact of the application domain on the development of grid middleware. The latter is being extensively and continuously reshaped by the needs of various application domains. This takes the form of adding new middleware services (Polak et al, 2004), developing new middleware kernels (Kranzlmüller et al, 2003) (Benedyczak et al, 2004), introducing new metagrid engines (Graschew et al, 2005), developing resource brokers for the discovery and selection of suitable grid resources, developing grid service APIs (Ordas et al, 2005), developing grid application toolkits (Schuller et al, 2005), raising the grid awareness of business concepts (Dolenc et al, 2004), advising semantic-aware solutions (Maad et al, 2005), or declaring the grid middleware as a special purpose one meeting specific application domain needs (Tatebe et al, 2004) (Yamamoto et al, 2004). This continuous demand for change in existing grid middleware is attributed to the adoption of a bottom-up approach for development (where individual parts of the system are specified in detail and the parts are then composed to form larger components, which are in turn composed until a complete system is formed). For instance, the Globus Alliance "Ecosystem" of Grid Components framed the exercise of building a Grid system or application as a software integration problem, hopefully, leveraging existing grid components to reduce the development cost (Globus-web). However, our survey of the grid penetration in various application domains reveals that this cost is considerable. By adopting a bottom-up approach, most of grid middleware development has evolved in isolation of application domain needs. This motivates re-orienting grid middleware development by taking into consideration a top-down approach, starting from modeling an application domain and working down to a middleware. On the other hand, relying solely on a top-down approach for grid middleware development may lead to narrow solutions and tend to be less generic. Based on this discussion, the paper, thirdly, draws general guidelines for adapting the implementation of the grid anatomy to meet the needs of various application domains. Marrying a top-down approach with a bottom-up approach for middleware development is proposed. A concrete example is given.

An API for Building New Clients for UNICORE

Roger Menday, Lidia Kirtchakova, Bernd Schuller
Division "Distributed Systems and Grid Computing", Central Institute for Applied Mathematics, Research Centre Jülich GmbH, Germany

In this paper we shall describe a high-level API for building Grid agents, portals, frameworks, clients (command line, rich internet applications, 'thick' installed applications, etc). The initial target of the API is the UNICORE[1] Grid middleware.

For many years only one real client has existed for the UNICORE middleware. Perhaps the reason for this is the absence of a sufficiently high-level API for accessing a UNICORE Grid. The Arcon library[1] for UNICORE provides some assistance for building clients but it is quite low level, and thus does not make client development easy. Moreover the existing UNICORE client is coded in such a way that the Grid access functionality is not easily separated from operation of the GUI, and thus this avenue for re-use is only useful as a reference.

Developed within the DEISA[2] project, Roctopus is an API that a programmer can use to interact with the UNICORE Grid. Rather than attempting an programming-language neutral design, like for example the SAGA API[3], this work focuses squarely on a API for Java.

The model is as follows: a Grid consists of a set of links to the Sites for that particular Grid configuration. One can submit a Job to be executed, retrieve a list of executing jobs and query the Attributes (description of the resources) at each Site. A site contains links to a number of Storages, which consequently contain Files, which a programmer can manage and import/export files with.

The API also includes flexible support for the job submission and monitoring. Currently UNICORE uses a polling model for checking the status of jobs, transfers, etc. Roctopus presents this as a clean asynchronous interface (currently implemented internally by repeated polling), where interested parties are able to register for notifications. Operations for cleanup and staging are non-blocking - concurrency issues are taken care of by the API.

In addition, Roctopus provides tidy, uncluttered access to the raw information from a Unicore Grid - information that can combined, re-processed and presented in interesting new ways. Indeed, new and unexpected UNICORE client applications have already been developed.

The public API consists almost entirely of interface classes. In the style of many other APIs in java, for example specifications derived via the JCP[4], multiple implementations are to be expected. Currently, the current implementation is directed at the existing production version of UNICORE, version 4.x. The Web services architecture of Unicore/GS is the target for a second implementation. Moreover, the carefully considered design does not preclude implementations for other (non-UNICORE) distributed infrastructures at some point in the future.

The paper completes with a short overview of current prototype work exploiting this new API for UNICORE, which point to a more diverse and interesting future for multi-client access to UNICORE.

[1} http://unicore.sf.net
[2] http://www.deisa.org
[3] https://forge.gridforum.org/projects/saga-rg
[4] http://www.jcp.org

An Architecture based on a Social-Economic Approach for Flexible Grid Resource Allocation

Gabriele Pierantoni, Eamonn Kenny, Brian Coghlan
Department of Computer Science, Trinity College Dublin, Ireland

One of the most challenging aspects of Grid Computing is the design and implementation of effective

resource management and allocation systems. This is due to the fact that, in Grids, resources belong to different domains, are subject to different policies and are geographically distributed.
Therefore it is quite difficult even to decide what the concept of optimality is, and thus what criteria should drive the allocation of resource. To cope with these difficulties we use a social and economic paradigm that tries to mimic, in the Grid's environment, known social and economic behaviors. This paper illustrates an architecture to implement in the Grid a resource management and allocation system based on social and economic paradigms.

The design of a system capable of finding the "optimal" resource to perform a given task must take into proper account a number of factors: firstly, the different ownerships of resources and jobs and the different policies that these resources are subject to. Secondly the system must be able to define what "optimal" resource selection means in regard to which actor (Resource Owners, Job's Owners, Global Welfare of the Grid, Optimal Occupation of the resources, etc.) it is working for. Consequently, this implies different solutions for allocation and optimization.
The system should also have the ability to cope with different scenarios encompassing forms of cooperation and competition.

To design an architecture capable of implementing these functionalities we used a social paradigm to describe how the various actors, which we termed "Social Grid Agents", interact among each other and gather in arbitrarily complex structures called "Grid Societies". We described the flow, dependencies and interactions of grid operations using the economic paradigm of production and composition of services and we described the allocation and sharing of resources with the paradigms of economic exchange markets or social forms of co-operation.
The economic approach allows the investigation of the use of economic theory to analyze and optimize the allocation of resources in competitive scenarios. This conceptual architecture is then completed with the creation of a list of "Meta-Actors", a concept similar to a pattern in Object Oriented languages. Meta-Actors describe a topology of "Social Grid Agents" and "Grid Societies" that can be used to implement different complex actors. The Meta Actors, via specialization and implementation, are used to create real actors to model the behaviour of Grid Agents such as Resource Owners, Resource Brokers, Job Owners, etc.

Based on this conceptual architecture we are evaluating the existing Grid Systems to decide what services and what Grid Systems will be used to implement the Social Grid Agents. Particular attention has been given to main Grid Middlewares such as the Large Hadron Collider Grid (LCG2) and the Globus Toolkit 4 (GT4). This evaluation process also encompasses Grid Middlewares and schedulers that offer economic-oriented functionalities such as GridBus and Libra.
Prototypes, currently under development, are used to evaluate the soundness and feasibility of the solutions proposed in the architecture.

Autobuilding Multiple Ports of Computing Nodes for Grid Computing

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

Last year Grid-Ireland reported on its work to port the Scientific Linux 3 version of the Large Hadron Collider Grid (LCG2) software to Red Hat 9, Fedora Core 2, Mac OS X, AIX and Irix. In the past year the work has been extended to include Fedora Core 4, Mac OS X Panther and Tiger, SuSE 9.3, CentOS 4.1 and Solaris 10. 64-bit versions of AIX, Mac OS X, SuSE and CentOS ports are planned for the near future.

The short-term goal for Grid-Ireland in 2005 is to provide a CentOS 4.1 32-bit port for some of its current infrastructure and a SuSE 9.3 64-bit port for the new Irish supercomputing centre.

It was found that a complicated auto-building architecture was required so that the dependencies required by the LCG2 middleware could be packaged effectively and ported to so many platform types. If this architecture is not in place, a large amount of effort is required to begin a port on a new platform. The EGEE gLite nightly building software was expected to deal with this issue but was found to require a complete rebuilding of dependency modules for specific architectures to provide new ports. Grid-Ireland have concluded that a dependency build machine with a local CVS repository must be set up to build all the software required by the LCG2 middleware software. This is a major undertaking, but once in place eases not only the LCG2 middleware porting burden but also autobuilding of other dependent middlewares. The details of this auto-building architecture will be provided in this paper.

The porting of the middleware to four non-reference Linux platforms plus the inclusion of local reference ports for SL3 and Red Hat 7.3 results in a requirement for six Linux build machines and at least six test worker nodes on which to test job submission, replication and LCG2 site functional tests (SFTs). Because of the increasing number of resources required to provide these ports, a set of virtual machines (using the Xen architecture) was created to support the porting, the details of which are explained in this paper.

The porting issues under different architecture are varied but stem for the most part from the following set of issues: (a) POSIX compliance, (b) Linux build platforms (c) RPM packaging assumptions, (d) newer compiler failings, (e) dependency classes, (f) inflexibility of the build system, and (g) difficulties in rolling back to previous version. All of these issues will be described in some detail with solutions presented where required.

The current status of the autobuild procedure will be reported in this paper, plus a tabulated set of SFT results.

Building Grid Services Using Portlets and GPE Middleware

J. Jurkiewicz (1) , K. Benedyczak (1,2) , P. Ba³a (1,2)
(1) ICM, Warsaw University, Poland
(2) Faculty of Mathematics and Computer Science, N. Copernicus University, Toruñ, Poland

Grid middleware is designed to provide access to remote high performance resources. The most important paradigm is seamless access to the resources. The user obtains number of tools ranging from the set of scripts to the graphical application launched on the client workstation such as UNICORE Client [1].

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

Recently an effort has been undertake to adopt UNICORE middleware to the web services environment. The UNICORE has evolved to the Grid Programming Environment [2] which is fully web serviced based solution. The GPE provides user tools to create and implement various services, ranging from file transfer to job execution. An important part of the GPE is user interface which replaces UNICORE Client. The GPE Application client provides user with the GUI for job preparation, submission, control, as well as certificate management. The UNICORE plugins has been replaced by the GridBeans which are stored on the server and are dynamically loaded according according to the users needs.

The GPE provides different types of the clients: Application Client dedicated for the single GridBean execution, Expert Client which allows for workflow management, and JSR168 portlet which can be placed in the portal environment.

The GPE portlet is however limited by the JSR168 standard, which allows only for HTML type interface. Unfortunately, this solution does not allow for development of interactive graphical interfaces.

In this work we present developed by us new type of the GPE Client which can be integrated with the portal and provides user with the full featured interactive interface to the application. As result the GPE application client functionality has been achieved.

The developed technology allows for easy development of the application interfaces which can be used in the different types of the GPE clients, including JSR168 portlets. The solution is based on the GPE middleware which provides fully interoperable grid framework.

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

[1] http://www.unicore.org

Capability Languages in C-GMA

Ondrej Krajicek, Ales Krenek, Ludek Matyska, Miroslav Ruda, Jiri Sitera
CESNET, z.s.p.o., Prague, Czech Republic

The recently proposed Capability-based Grid Monitoring Architecture (C-GMA) targets certain shortcomings of the general Grid Monitoring Architecture (GMA). The C-GMA main goal is to define an interoperable Grid monitoring framework that will allow co-existence and co-operation of existing and future Grid monitoring systems. The C-GMA uses a duality of data attributes and component capabilities to provide a general approach towards different requirements which are behind different Grid monitoring systems and their implementations.

This paper elaborates on two metadata layers used in C-GMA that work with different meta-data information. One layer, called capability and attribute layer, is used to describe properties and requirements of components (the capabilities) and also properties and requirements of data (the attributes). Another layer, called data-definition layer, describes data types of published and requested data, and also includes data specification.

The capability and attribute layer can unify several different GMA systems or implementations. As the same capability language is used to describe capabilities of the components and attributes of the data provided, these different GMA implementations can easily co-exist within one layer. The systems can remain independent---they use different data-definition languages and completely disjoint sets of capabilities and attributes and their components will never be matched---but co-existence within one capability and attribute layer opens a way to define bridges between any two (or more) GMA implementations (worlds). The bridge is defined as a Grid monitoring component with multiple interfaces, each using different data-definition language, capable to translate data (events) between the implementations. The uniformity at the capability and attribute layer ensures that the bridges are correctly discovered and deployed by the C-GMA system.

The paper will present two capability and attribute languages. The one will be an XML based language, that describes capabilities and attributes as XML documents and uses XPaths for expressing compatibility conditions. The second capability language is based on the ClassAd language and is using the standard ClassAd matchmaking for compatibility checking. A prototype implementation and evaluation of the ClassAd language based approach will be presented. The description of two capability and attribute languages will be also used to derive some general requirements on the capability languages.

The presentation will be concluded with a C-GMA implementation that includes two different GMA implementations on the data-definition level. Subsets of the Mercury Grid monitoring system (developed as part of the GridLab project) and Logging and Bookkeeping service (developed as part of the DataGrid and EGEE projects) will be described using the same capability and attribute language and their interoperability will be demonstrated.

Centralised Fabric Management for a National Grid Infrastructure

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

The Grid-Ireland infrastructure is unusual among national grids in that all Grid service nodes at our eighteen sites are managed from a central Operations Centre (OpsCentre) which controls the configuration of the Grid middleware and the standard OS components.

The Grid middleware is validated at the OpsCentre before being deployed to the remote sites: any troublesome issues can be resolved before the new software is rolled out. As all aspects of the OS configuration are controlled from a central location, the OpsCentre can ensure that all the infrastructure machines are up-to-date with security patches and that their configuration is consistent.

It is clear that such an architecture requires a well-thought out management system: ad-hoc methods that may suffice for a single site do not scale to an entire national Grid comprising many sites. A comprehensive fabric management system driven from a central configuration database is essential.

In the first generation of our deployment architecture we used LCFGng to manage the configuration of the Grid service nodes running RedHat 7.3. After a period of evaluation, we recently migrated our entire infrastructure to use the Quattor fabric management system. We now present the results of this evaluation and our experiences with the transition.

The goal of Grid computing is to create a highly robust and stable infrastructure that enables scientists to access globally distributed computing and storage resources in a straightforward way. The Grid will have succeeded when it becomes a largely invisible network service that researchers can take for granted as they do Internet connectivity. We believe that a centrally managed architecture like ours enables a realistic roadmap to creating this infrastructure.

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

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

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

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

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

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

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

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

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

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

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

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

Cumulus - Dynamic Cluster Available under Clusterix

Jan Kwiatkowski (1), Marcin Pawlik (1), Roman Wyrzykowski (2), Konrad Karczewski (2)
(1) Institute of Applied Informatics, Wroclaw University of Technology, Poland
(2) Institute of Computer and Information Sciences, Czestochowa University of Technology, Poland

CLUSTERIX grid environment is a production cluster grid consisting of local PC-clusters based on 64-bit Linux machines. Local clusters are placed accross
Poland in independent centers connected by the Polish Optical Network PIONIER. Currently the basic infrastructure of CLUSTERIX comprises 250+ Itanium2 CPUs located in 12 sites. At the same time, the computing power of the CLUSTERIX environment can be increased dramatically by connecting additional clusters to the basic infrastructure dynamically.

The paper presents the method how to convert a set of independent computers (e.g. computer laboratory, personal machines at the desks of employees, etc.) to a fully operational, Linux-based Beowulf cluster. This cluster is ready to be utilized as a dynamic subsystem of the CLUSTERIX grid. In the second part of the paper, the way how to connect such a cluster to the core infrastructure of CLUSTERIX is presented. This opens access to an extremely powerful grid installation. The only requirement is availability of communication channels between the machines. The only additional element needed is a single computer acting as a cluster server/firewall. The machines stay fully operational, without any degradation of their previous functionality. If they are network-boot ready, no modifications to their software is needed, while machines without that feature need only a network-boot capable bootloader.

The connection of dynamic local clusters, like the Cumulus cluster deployed at Wroclaw University of Technology, to the CLUSTERIX backbone, opens possibilities to access a shared environment with the extraordinary computational power, and dedicated applications designed to take advantage of the power delivered by the computational grid.

The initial Cumulus configuration consists of 80 computers available at two computer laboratories of the Faculty of Computer Science and Management, Wroclaw University of Technology, located at two different buildings. Presently this cluster is used for education purposes (some extra students jobs) as well as for research.

One of the most important design requirements is that the deployment of the dynamic cluster can not interfere with a typical role of the computers used for university courses, so this role of the computer nodes should be preserved. During lesson hours the cluster is turned off and all computers are normally utilised. When all classes are finished, the computers are automatically rebooted or, if turned-off, started. Then they begin to form the computational cluster. Every computer connects to the server located at each laboratory. The nodes load their configuration files, initial ramdisks, and operating system kernel. They mount the remote root filesystem composed of shared and individual parts, and become operational. Flexibility of the configuration makes it possible to use either diskless nodes or diskfull ones, with dedicated scratch and swap space, as well as deploy fat, diskfull nodes operating from the root filesystem on their own hard disks.

During Cumulus initialisation the cluster server connects to the CLUSTERIX firewall node, and lets it modify settings needed to access the Custerix core system. The dedicated ssh tunnel is opened, and the necessary components of the CLUSTERIX middleawre are started, including the monitoring system and resource broker. As a result, the cluster becomes a part of the CLUSTERIX environment. When the cluster is taken down, a special procedure is applied to inform the monitoring system that it should initiate performing appropriate actions, e.g. migration of unfinished jobs.

Data Management and the InteliGrid User Scenarios

Matevz Dolenc (1), Erik Balaton (2), Krzysztof Kurowski (3)
(1) University of Ljubljana, Faculty of Civil and Geodetic engineering, Ljubljana, Slovenia
(2) EPM Technology AS, Oslo, Norway
(3) Poznan Supercomputing and Networking Center, Poland

A challenge for grid collaboration infrastructures is to support dynamic virtual organisations 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. With the increasing pressure of reducing costs and time-to-market, industrial end-users have turned their focus from traditional RDMS and file based product data storage to product model servers. Product models are based on an object-oriented paradigm that includes complex object models and object relationships in several object layers as well as elaborate class extension mechanism for inclusion of dynamically generated objects.

The OGSA-DAI middleware was identified as the enabling grid technology for providing uniform access to divers data sources within engineering virtual organisations supported by the InteliGrid project. To enable access to engineering data sources the OGSA-DAI middleware was extended with support for the WebDAV protocol and product model servers.

This paper presents the engineering end-user requirements and scenarios identified within the InteliGrid project that led to development of OGSA-DAI extensions as well as the implementation and use of those extensions.

DG-ADAJ: a Java Computing Platform for Desktop Grid

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

During these two last decades, the inter-connected resources of treatment knew an exponential increase. This evolution is accompanied upstream by continuous improvement of treatment resources and downstream by an unprecedented increase in the complexity of calculations. The fast workstation network evolution gives rise to a new architectural alternative for the treatment parallel and makes it possible to answer this request, with new paradigms of programming and new concepts and software system. Desktop GRID computing which exploits unused resources in the Intranet environments can deliver tremendous computing power to solve many complex problems.

In that context, we propose a 100 % Java compliant system built around the object environment in particular around the object technology. The principal characteristic of the heterogeneous applications is to be irregular and unforeseeable. To allow an effective execution of that kind of application we propose DG-ADAJ, a middleware platform which has mechanisms ensuring the automatic adaptation of the elements of the application, in response to the computing evolutions and to the modifications of the resource availability. We control the granularity of the treatment and the distribution of the application on the platform of execution. Thus, programmers don't have to be concerned with this distribution. ADAJ gives an programming environment for distributed applications in Java and it gives also an execution environment which optimizes the dynamic placement of the application objects on a of computers. This distribution is based on a new mechanism of observation of the activity of the objects and relations between them.
Initially, ADAJ was built for cluster computing and our current work is to re-examine the mechanisms to extend them for large scale distributed computing. The optimization of the initial distribution of the application object on the GRID and redistribution mechanisms are also studied. In that purpose, we work on evaluation of distributed application computing built with software components (project CCADAJ). The French GRILLE 5000 (which belongs to European CoreGrid) platform will be used as soon as it will be installed. We hope to observe in the Desktop grid computing environment, the same efficiency of the monitoring and object distribution mechanisms as in the cluster computing case.

DIDA, a Distributed Discovery Architecture for Grid Environments

Carlos de Alfonso, Miguel Caballer, Vicente Hernandez
Departamento de Sistemas Informaticos y Computacion, Universidad Politecnica de Valencia, Spain

The convergence of Grid Computing and Web Services (WS) has happened because of the evolution of Grids and its appliance out of the scientific environments. The Open Grid Services Architecture (OGSA) defines a Grid architecture based up on services, and which needs in implementation are mostly satisfied using WS. Nevertheless, OGSA defines some concepts, such as state fullness in services or notifications, which are not covered by WS. In this scope appeared WS Resource Framework (WSRF) which extends the WS specification, and introduces the WS-Resource for covering these misbehaviours.
So in WS as in Grid environments, the efficient and effective discovery of services (and resources) is one of the main problems that the architecture may deal with. In both "Universal Description, Discovery, and Integration" (UDDI) and "Monitoring and Discovery Service" (MDS), the discovery facilities are mainly oriented to directories which index the services. These architectures provide with several kind of nodes organization for coordinating the management of the directories, grouping them, etc. Nevertheless, these architectures provide multiple entry points which have partial views of the indexed services. The whole directory can be only acceded from few points (i.e. only the root MDS).
This paper proposes a Distributed Discovery Architecture (DIDA) for Grid environments, with multiple entry points to the whole directory of services (or resources). The system is built up on the Globus Toolkit 4 (GT4) Index Services (IS), using Federated Advanced Directory Architecture (FADA) technology for avoiding cascade node organizations.
In the proposed architecture, the services are deployed in local IS, as usual. In an upper level, the IS are registered into a FADA infrastructure. The nodes which form the FADA framework are distributed among the system, so as each node only knows about their neighbours.
The entry points are created by deploying Distributed Discovery Services (DDS) which are acceded as WSRF services, using XPath querying language (as in IS). The DDS are aware of the existence of FADA nodes, and utilizes them for discovering the IS in the system. Later, each IS is queried using XPath and the results are gathered for the client of the DDS.
DIDA is highly scalable, as any new IS can be immediately integrated into the system by registering it into any of the FADA nodes. On the other hand, any FADA node can be added to the architecture by notifying its existence to any other FADA node. The architecture enables the mobility of services, as any of them may be registered into any other IS and thus being immediately discoverable. Also the system is fault tolerant as when any of the FADA nodes or the IS fail, the rest of the network is not affected because it is totally independent and the FADA network will be self rearranged.
DIDA architecture has been developed in the framework of the gCitizen project, which is financially supported by the Spanish "Ministerio de Industria, Turismo y Comercio", with the reference FIT-350101-2004-54.

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

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

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

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

Pawel Swierszcz, Wlodzimierz Funika, Marcin Smetek

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

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

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

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

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

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

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

Early Experiences on Application Checkpointing for Realizing QoS Provision

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

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

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

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

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

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

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

Experiments with Distributed Training of Neural Networks on the Grid

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

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

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

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

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

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

Exploring OGSA Interoperability with LCG-based Production Grids for Biomedical Applications

Alfredo Tirado-Ramos, Derek Groen, Peter Sloot
University of Amsterdam, The Netherlands

Inter-Grid interoperability is widely expected to become a crucial issue in computational science once real production Grids start to go mainstream and interconnect. Interoperable services that seamlessly communicate, transfer data and computation among different Grid infrastructures will allow scientists to create something more than the current set of non-interoperable distributed systems, which is one of the reasons that hinder production-type Problem Solving Environmment (PSE) developers from collaborate and integrate seamlessly across large Grids.

On one hand, the Open Grid Services Architecture (OGSA), one of the main results of a global effort to advance Grid interoperability, extends the Web Services paradigm to include Grid concepts, and to manage the creation and termination of resources as real services. The Globus Toolkit 4 (GT4), one of the most popular implementations of the OGSA architecture, is currently being deployed widespread among new Grid developers, though it provides a somewhat immature technology for running production-type Grids.

On the other hand, more mature frameworks such as the CERN's Large Hadron Collider Computing Grid (LCG) has been successfully deployed by production-type Grid infrastuctures like the EU IST CrossGrid project, providing scientists with large testbeds that run efficiently. The testbed includes resources across 16 European sites, which range from relatively small computing facilities in universities to large research computing centers, offering an ideal mixture to test the possibilities of an experimental Grid framework. The testbed also includes a set of tools and services such as monitoring tools, development tools, a remote access server, portals and a prototype of the parallel resource broker. The CrossGrid focused on the development of Grid middleware components, tools and applications with a special focus on parallel and interactive applications, extending the Grid to interactive applications. Nevertheless, the CrossGrid testbed largely inherits from the European Data Grid (EDG) experience on testbed setup and LCG middleware distributions for services, a technology that is expected to evolve with the development of gLite (based on a service oriented architecture), and probably will need to interoperate with GT4 services in the near to midterm future. This is a real concern within our Biomedical PSE project at the University of Amsterdam, where we have been working to extend the current testbed and expect to interoperate with new and unforseeable partners in the future.

In this paper we present our results after performing a series of data transfer and computation bootstrapping of an application within our biomedical PSE, using GT4 OGSA services interoperating with our current extended LCG production-level infrastructure. We study resource availability and performance overhead associated with a hybrid approach to biomedical data access, and present our conclusions.

Keywords: Computational Grids, LCG, OGSA, Interoperability, Biomedical

Fault Tolerance and Data Synchronization in Grid Registry for Workflow Applications

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

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

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

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

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

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

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

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

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

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

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

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

References

[1] OCM-G - Grid-enabled OMIS-Compliant Monitoring system,
http://www.icsr.agh.edu.pl/ocmg

[2] Clovis Seragiotto, Jr., Hong-Linh Truong, Bernd Mohr, Thomas Fahringer, Michael Gerndt, Tianchao Li, 'Standardized Intermediate Representation for Fortran, Java, C and C++ Programs', APART Technical Report Workpackage 1, http://www.fz-juelich.de/apart

[3] B. Balis, M. Bubak, J. Dziwisz, H.-L. Truong, and T. Fahringer. Integrated Monitoring Framework for Grid Infrastructure and Applications. In P. Cunningham and M. Cunningham, editors, Innovation and the Knowledge Economy. Issues, Applications, Case Studies, pages 269-276, Ljubljana, Slovenia, October 2005. IOS Press.

[4] OMIS - On-line Monitoring Interface Specification. Version 2.0. Lehrstuhl fur Rechnertechnik und Rechnerorganisation Institut fur Informatik (LRR-TUM), Technische Universitat Munchen.
http://wwwbode.informatik.tu-muenchen.de/~omis

GEMINI A Universal Monitoring Framework: Concept, Design, and First Prototype

Bartosz Balis (1), Marian Bubak (1,2), Jakub Dziwisz (2),
Kuba Rozkwitalski (2), S³awomir Hurnik (1)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Academic Computer Centre CYFRONET, Krakow, Poland

Monitoring available resources, running applications, middleware services, etc., is necessary in every distributed system, especially in Grid systems which are large, dynamic and maintained in different administrative domains. One of
the trends in today's Grid development is evolution towards service-oriented, semantic-rich systems where applications are realized in a workflow model. One example of such an approach is the K-Wf Grid Project [1], in which semi-automatic workflow composition as well as intelligent workflow execution are addressed. Worklow composition and workflow execution require knowledge which is captured during the system's lifetime. To capture this knowledge, combined information from various sources is needed, for example, from executing applications, grid middleware services, or grid infrastructure elements. To obtain this information, we need a monitoring infrastructure which is (1) generic in terms of data types supported and monitoring domains addressed, (2) distributed and decentralized to span potentially the whole grid system, but at the same time efficient and scalable, (3) suitable for operation in a service-oriented environment, and (4) standardized in terms of interfaces, languages, semantic-rich data representation, and certain procedures, in order to enable its use by different tools and services in the Grid. Current monitoring solutions do not address all those issues. Only few of them are meant to be generic, for example R-GMA or Mercury, however, they are limited with respect to supported data model, service-oriented operation, efficiency, or compliance to standards.

We have designed and developed a first prototype of a generic monitoring infrastructure, or framework, called GEMINI, which is aimed to satisfy the mentioned requirements. The framework provides storage and delivery of diverse monitoring information about a variety of distributed entities, including Grid infrastructure, applications, and middleware. The requests for data are issued via a Web Service interface, while the data transfer itself may be realized either through outputs of WS operations or an efficient stream-based communication channel, to address both interoperability and efificiency requirements. The supported data types of monitoring information are standardized and described in OWL ontology language. An important feature of GEMINI is the support for monitoring of applications, which includes a specification of an instrumentation service, and development of application sensors which enable efficient and easy instrumentation of applications.

In this paper, we present the current development status of GEMINI and discuss our approaches to infrastructure and application monitoring. We describe how, using the standardized generic sensor libraries of GEMINI, one can develop arbitrary GEMINI-compliant sensors, either stand-alone or embedded in application's code. We also show how, using the same features, one can adapt existing "legacy" monitoring systems to work as data sources for GEMINI. Currently we have adapted Ganglia [2] for monitoring of infrastructure and are working on adaptation of the OCM-G [3] for monitoring of legacy MPI applications.

Acknowledgements. This research was partially funded by the K-WfGrid EU IST Project and by the K-fGrid

SPUB-M.

References
[1] K-Wf Grid Project Homepage: http://www.kwfgrid.net
[2] Ganglia homepage: http://ganglia.sourceforge.net
[3] OCM-G -- Grid-enabled OMIS-Compliant Monitoring system, http://www.icsr.agh.edu.pl/ocmg

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

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

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

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

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

Grid-based Simulations of MAMMALIAN Visual System

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

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

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

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

Integrating Dynamic Clusters in CLUSTERIX Environment

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

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

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

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

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

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

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

Inter-application Control Through Global States Monitoring on a Grid

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

This paper presents methods for organizing execution control of Grid applications by means of predicates computed on globally monitored application states. This new Grid control paradigm [1] has been embedded in a graphical Grid-application design environment (Grid PS-GRADE) [2], which is currently under implementation. It enables designing graphically inter applications control on a Grid in a much more flexible way. Grid-level application state monitoring is done by a special kind of structural application elements called synchronizers. Grid-level synchronizers collect state reports from applications, construct consistent Grid-level global states, evaluate predicates on them and send control signals to applications. Control signals can modify applications behaviour in asynchronous manner. A similar control method based on monitoring of global process states by application-level synchronizers has been also embedded internally inside Grid applications [3]. In this way a flexible user programmable control infrastructure has been created, which enables asynchronous control at any desired granularity level. In this infrastructure application-level synchronizers co-operate with Grid-level synchronizers thus providing new structurized methods for conveying intra and inter application control decisions.
In this paper the following co-operation schemes included into the proposed Grid environment will be discussed:
1. Simple workflow. A selected application starts executing after a set of selected applications is completed. Example: complicated scientific computations performed stage by stage in different computer networks.

2. Alternative workflow. One of several applications is selected for execution depending on the results (state) of former applications. Example: one of two available program packages is run depending on computation results performed so far.
3. Supporting workflow: A set of curently executed applications require activation of auxiliary applications, which will provide useful results. Example: In a coarse grain simulation of a system of moving objects a collision that appears, stimulates a change in the Grid application configuration. An application which models the collision in a detailed way (with a fine granularity of events) is activated. After the results of the detailed simulation of the collision are known, the coarse grain simulation continues.
4. Partial canceling of workflow: Applications that become superfluous from the global grid-level point of view are stopped. Example: The exhaustive parallel search on a Grid for the optimal solution in a solution space is stopped or restricted when the search provides a satisfactory solution.
5. Workflow coupling: A global status of many applications is monitored and control directives are distributed to particular applications as needed. Applications compute parameters that are subject to mutual exchange. Some parameters in meta-level applications are updated with the use of results of some auxiliary computations. Example: An exhaustive search run on a Grid using Branch & Bound method is supported by heuristic search algorithms, results good for search bounding that come from the heuristic algorithms are introduced to the B&B application.
The paper shows how the described application co-operation schemes can be implemented in the proposed Grid system, which supports the control based on global computation state monitoring using an infrastructure provided by application and Grid level synchronizers. Resulting efficiency improvement is evaluated.

Bibliography

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

K-WfGrid - Knowledge-based Workflow System for Grid Applictions

K-WfGrid Consortium (www.kwfgrid.net)

LCG-API: LCG-2 and GridSphere 2.0 Integration

Jakub Dziwisz, Pawel Plaszczak and GridwiseTech Team
GridwiseTech

For non-experts Grid portals are the only reasonable way to experience the Grid. They can provide users with the entire Grid system functionality without forcing them to hack the scripts and command line interfaces. And the more popular Grid computing is, the more important those means of access are -- with portals usage of cutting-edge technologies can be as easy as sending an e-mail.

LHC Computing Project (LCG) is said to be the first worldwide deployed Grid production system. It is used not only at the universities and not only by the technical gurus. Thus many of the potential users are not accustomed with terminal applications, but virtually everyone has sent a few e-mails.

Therefore it is no wonder that there is a growing demand for Grid portals compliant with LCG. To our amazement it is difficult to find a product suitable for a corporate user. Taking everything into account GridwiseTech has decided to develop their solution.

In this paper we will share our early expressions on integrating LCG-2 with GridSphere Portal, one of the most popular JSR-168 portlet containers. We will substantiate our choice of the portal solution, analyze other available LCG compliant Grid portals and examine alternative ways of integration. Moreover we will investigate how the solution fits in GridSphere's general PortletService paradigm. We will also discuss the strengths and weaknesses of our solution, demonstrate how to "LCG-enable" simple Grid portlet and discuss the differences between "regular Grid portlet" and "LCG portlet".

Learning Reliability Models of Grid Resource Supplying

Cyril Briquet, P.-A. de Marneffe
Department of Electrical Engineering and Computer Science, University of Liege, Belgium

Transient exchange of resources inside Virtual Organizations is a key feature of Grid computing. Resource exchange between Grid participants can enable them to face an unstable request environment.
As Grid participants obviously require incentives to exchange resources with one another, methods of resource exchange are based on some kind of trading [1], which may be market-based or decentralized and

moneyless, like distributed bartering.

Grid participants must also face an unstable resource environment due to the partial and intermittent nature of the exchanged resources. With a passive stance, when a resource does not deliver the contracted computing power, not much is done to avoid contracting with this unreliable resource in the near future. With a proactive stance, it is the most reliable resource that would be selected from a group of equally priced or equally performing resources. This selection is performed so as to actually avoid as much as possible dealing with unreliable resources. However, dealing with failure of resource supplying is currently often passive, rather than proactive, for these methods do not yet systematically take the reliability of resource suppliers into account.

A recent distributed bartering method is the Network of Favors [2], where each Grid participant act as both a supplier and a consumer of resources, maintaining its own, private bookkeeping of exchanged resources (doing a favor is providing access to a resource in the expectation that it will be returned).
We extend this method with learning capacities. Machine Learning algorithms are used to automatically learn models of resource suppliers reliability. The expected benefit is to cut down the mean job completion time (i.e. utility) of each Grid participant.

[1] R. Buyya, Economic-based Distributed Resource Management and Scheduling for Grid Computing, PhD Thesis, Monash University, Melbourne, Australia, 2002.
[2] R. Santos, A. Andrade, F. Brasileiro, W. Cirne and N. Andrade, Accurate Autonomous Accounting in Peer-to-Peer Grids, Proc. 3rd Int. Workshop on Middleware for Grid Computing, Grenoble, France, 2005.

LSF and SGE Interoperability Using Globus Toolkit 4

Kuba Rozkwitalski, Chris Wilk, Pawel Plaszczak
GridwiseTech

In distributed, heterogeneous environment the interoperability between DRM solutions from different vendors is essential. Currently, the de-facto standard, Globus Toolkit 4 (GT4) is able to submit jobs to various job schedulers through its WS-GRAM component. In the sametime the GGF's DRMAA working group has been developing universal API for job management.

One of the job schedulers integrated with GT4 is Platform LSF. With GridwiseTech's compliance suite for Sun Grid Engine it is possible to submit jobs to the SGE through WS-GRAM interface.

In this paper we want to share our experiences in building truly interoperable environment for job submission to various job managers. We will present results of this integration, focusing on SGE and LSF. This will be followed by discussion on interoperability issues in job schedulers, as well as possible future directions of such integration.

Making the Best of Your Data - Offloading Visualization Tasks onto the Grid

Peter Praxmarer (1), Paul Heinzlreiter (1), Dieter Kranzlmueller (1), Wolfgang Kapferer (2), Sabine Schindler (2), Jens Volkert (1)
(1) GUP, Johannes Kepler University Linz, Austria
(2) Institute of Astrophysics, University Innsbruck, Austria

Visualization of data in general, and rendering of volume data in particular, is a common task in computational science and engineering. The application presented in this paper demonstrates our work for visualizing the density distribution of gas within a galaxy cluster resulting from an astrophysical simulation.

An important aspect in obtaining visual output from data is the identification of a suitable transfer function, which is applied for mapping the input data onto color and opacity values, which are then used for rendering. In case of the amorphic structure of the gas distribution only the human user can really judge the validity and usefulness of a visualization and the transfer function.

In our approach, and to enable an iterative improvement of the transfer function, a genetic algorithm is applied. Within this algorithm the user judges the validity of the resulting visualization and selects the best fit. Obviously, the approach can be quite time and resource consuming, since it requires the rendering of multiple images. Therefore, we perform a parameter study on the grid such that the rendering of a population is done in parallel. This is well-suited for grid environments due to its characteristic requirements for computational resources.

A comparable approach to the technique described is demonstrated in [1]. Our method extends this earlier solution by automatically rendering the images on the grid as a parameter study.

In case we're accepted we'll present the refinement process and the rendering on the grid including performance measures.

Bibliography:
[1] T. He, L. Hong, A. Kaufman, H. Pfister: Generation of Transfer Functions with Stochastic Search Techniques, Proc. IEEE Visualization'96, San Francisco, CA, USA, pp. 227-234, October 1996

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

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

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

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

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

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

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

Bibliography
[1] http://www.kwfgrid.net/
[2] Krawczyk, K., Slota, R., Majewska, M., Kryza, B., Kitowski, J., Grid Organization Memory for Knowledge Management for Grid Environment, in: Bubak, M., Turala, M., Wiatr, K. (Eds.), Proceedings of Cracow Grid Workshop - CGW'04, December 13-15 2004, ACC-Cyfronet UST, 2005, Kraków, pp. 109-115.
[3] Kryza, B., Majewska, M., Krawczyk, K., Slota, R., Kitowski, J., Unifying Grid Metadata Representation through Ontologies, PPAM Conference, LNCS, Springer Accepted, 2005.

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

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

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

Finally, we successfully verify results of the numerical simulations with the observational findings.

Monitoring the Distributedness of Java Applications

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

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

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

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

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

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

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

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

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

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

Monitoring Web Services with the J-OCM Monitoring System

Lechos³aw Trebacz, Wlodzimierz Funika, Piotr Handzlik, Marcin Smetek

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

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

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

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

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

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

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

Multi-Grid and Multi-VO Job Submission based on a Unified Computational Model

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

The Grid concept gives rise to many different implementations. These may be data-centric, computational-centric or hybrid with each having particular strengths and weaknesses. This opens a wide and interesting field of research which aims to harness the different capabilities of each of these Grid implementations.
To use heterogeneous Grids within a single environment two main issues have to be addressed; there must be a bridge infrastructure able to communicate with different Grid middlewares and there must be a unified model capable of expressing the workflow of operations to be executed by the different Grid Solutions.

The Unified Computational Model known as Condensed Graphs is ideal for expressing such workflows and its implementation (WebCom) allows its execution with eager or lazy evaluation policies.
At the same time stateful Web Services, such as those provided by the Globus Toolkit 4 (GT4), offer a promising technology for the implementation of the bridge infrastructure.
In this paper we describe an architecture that uses the power of the Condensed Graphs Unified Computational Model to express complex workflows of operations that are executed in an environment encompassing WebCom, the Large Hadron Collider Grid (LCG2) and GT4.

To achieve this goal, the architecture must support the execution of a workflow of operations on heterogeneous Grids on behalf of users belonging to different Virtual Organizations in a resilient and scalable way.
Firstly, the design of the interoperability system has to cope with a fundamental difference of architecture between WebCom and the conventional Grid Solutions. The current release of WebCom is based on a highly dynamic topology in which different instances of WebCom are instantiated on different machines and connect to each other in a tree-like structure while conventional Grids are based on rarely changing topologies where services run for most of the time on the same machines. The next release of WebCom will support a peer-to-peer topology. Secondly, WebCom currently has a basic file management behaviour that had to be improved by an additional file staging mechanism. Finally, security interoperability issues had to be addressed for each of the actors involved (WebCom, LCG2 and GT4).

To implement this solution a design concept of internal and border regions is used: An internal region is a set of machines that run either WebCom, or a homogenous set of Grid Services (either LCG2 or GT4.0 Services) while a border region is a set of machines where WebCom instances and Grid services coexist and interact. Thus the border region effectively extends the internal sets such that they overlap at the border set.This approach permits the merging of the dynamic topology of WebCom with the rarely changing topologies of conventional Grid solutions.

To design and implement this system two process were used in parallel: A top-down approach led to Analysis and Design documentation while a bottom-up approach produced a set of prototypes to evaluate the soundness and feasibility of the solutions proposed by the analysis and design process.

The interoperability infrastructure has been successfully implemented and is now undergoing as set of stress tests to evaluate its capacity to cope with computationally demanding workflows.

Multiple HLA Federate Processes in Grid Environment

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

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

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

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

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

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

On-line Performance Tunning of Grid Applications - A Case Study

Tomasz Szepieniec (2), Marian Bubak (1,2)
(1) Institute of Computer Science, AGH, Kraków, Poland
(2) Academic Computer Centre CYFRONET AGH, Kraków, Poland

From application users point of view the grid environment is very promising while it enables access to large scale of resources. However, the initial costs of addapting applications to the grid as well as a lost of efficiency due to heterogenous environment and unpredictable variations of almost all its ingredients limits the usability of this platform. While a decrease of reliability and stability is inseparable from the environment's size growth, the only solution to deal with this obstacles is to compensate them by dynamical adaption of the applicaton behavior to the actual grid resources status.

The main scope of our research is to provide the possibility to apply various adaptation methods to the application running on the grid. The spectrum of possible adaptations range from coarse-grained methods like checkpointg and migration of the whole application to another location [1], to more fine-grained methods like redistribution of work within application. The proposed system should be able to detect possible actions, then choose the most accurate adaptation scenario that would cause the maximum increase of performance taking into account the cost of adaptation, and finally be able to execute choosen tunning actions.

As we mentioned before, the important factor of usability of the grid systems is limitation of effort made by developers to enable dynamic adaptation in their application. So, in our system we should not only follow the separation of concerns principle, but also exclude as much functionality from application as possible.

The above requirements lead us to a concept of a Grid Application Performance Supervisor (GAPS) - a framework that could remotely control and steer a running grid application in order to tune its performance. The advantage of this solution is that our adaptation decision can be supported by knowledge from different components of application as well as from the grid infrastructure. Another advantages is possibility of reuse of existing monitoring
systems [2] such as OCM-G [3].

In this papers, we consider using GAPS for parallel, domain decomposed application in the batch oriented grid enviroment. Analysis of possible adaptation scenarios and methods to implement it will be given to prove that presented concept is feasible in contemporary grid systems.

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

References
[1] G.D. van Albada, J. Clinckemaillie, A.H.L. Emmen, J. Gehring, O. Heinz, F. van der Linden, B.J. Overeinder, A. Reinefeld, and P.M.A. Sloot: Dynamite - Blasting Obstacles to Parallel Cluster Computing. In A.G. Hoekstra, L.O. Hertzberger, P.M.A. Sloot, M. Bubak (Eds), High-Performance Computing and Networking (HPCN Europe'99), LNCS volume 1593, Springer-Verlag, 1999.

[2] M. Gerndt et al: Performance Tools for the Grid: State of the Art and Future - APART White Paper, Research Report Series, Vol. 30, LRR, Technische Universitat Muenchen, Shaker Verlag, 2004.

[3] B. Balis, M. Bubak, W. Funika, T. Szepieniec, R. Wismüller, and M. Radecki: Monitoring Grid Applications with Grid-enabled OMIS Monitor, In F. Fernandez Rivera, M. Bubak, A. Gomez Tato and R. Doallo (Eds), Grid Computing, First European Across Grids Conference,Santiago de Compostela, Spain, February 13-14, 2003, Revised Papers, LNCS volume 2970. Springer, 2004.

Ontology Alignment and Ontology Similarity for Extension of Service Ontology in Grid Environment

Joanna Zieba {1}, Renata Slota {1}, Marta Majewska {1}, Bartosz Kryza {2}, Jacek Kitowski {1,2}
{1} Institute of Computer Science AGH University of Science and Technology, Poland
{2} Academic Computer Center CYFRONET AGH, Poland

Grid systems and Semantic Web are rapidly developing technologies in the IT field nowadays. K-WfGrid project [1] uses findings from both of them to make the use of Grid services as 'invisible' as possible. Its main goal is to create a knowledge-based framework which enables automated workflow composition, execution and monitoring in Grid environment. Such a system can capture knowledge in the course of its work and reuse it in order to provide more efficient workflow creation.

Complex large-scale Grid systems require the possibility of organizing people, institution and their resources into groups called Virtual Organizations (VO). A VO is considered as logically centralized, physically distributed community that pursues common goals and objectives. In order to allow disparate and usually heterogeneous organizations to cooperate within some ad-hoc VOs a need exists to share a higher level conceptual model of the domain and its relation with all kinds of Grid entities like services, data or hardware resources. Due to the complex nature of Grid and heterogeneity of organizations the VO approach needs to make use of knowledge represented by ontologies and semantic description.

The main K-WfGrid component designed for managing knowledge is called Grid Organizational Memory [2]. It is a set of ontologies and registries which contain semantic descriptions of existing data, resources, applications, workflows and services (accordingly, Data Ontology, Resources Ontology, Application Ontology, Workflow Ontology and Service Ontology). It is very important that services, which are basic blocks for workflow composition, should be described in a complete and consistent way (OWL-S language is used for that). On the other hand, there is a need for flexibility as the framework gathers a lot of knowledge from monitoring facilities, its own reasoning engines, and users' input. This knowledge should be integrated into GOM's Service Ontology without losing its consistency.

The paper focuses on situations, where there is a need to register a new service in the Grid Organizational Memory. As the latter has to be 'aware' of the new service's characteristics, its description should be added to Service Ontology. We identified following problems here: compliancy check of the ontology and the service's description, measuring similarity between the new service and any of the existing ones, choosing a place where the new description fits, finding mappings between existing and new concepts and choosing a suitable strategy for extending the Services Ontology.

What we are working toward, is development of an algorithm of extending the Service Ontology in a semi-automatic way, i.e, the algorithm which checks the new service's compliancy with Services Ontology, finds a suitable place for the service in the ontology and performs necessary merges. Such a procedure interactively receives feedback from the user, who can choose the place for the service in the ontology from one or more propositions. If the process of registration fails, the user is kept informed about it and receives suggestions why the service does not fit in.

The paper also discusses and compares various approaches to ontology searching, ontology mapping, merging and aligning for the purpose of accomplishing this task.

Bibliography

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

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

Organization of the International Testbed of the CrossGrid Project

J. Gomes {1} and M. David {1} and J. Martins {1} and L. Bernardo {1} and A. Garcia {2} and M. Hardt {2} and H. Kornmayer {2} and J. Marco {3} and R. Marco {3} and D. Rodriguez {3} and I. Diaz {3} and D. Cano {3} and J. Salt 4} and S. Gonzalez 4} and J. Sanchez {4} and F. Fassi {4} and V. Lara {4} and P. Nyczyk {5} and P. Lason 5} and A. Ozieblo {5} and P. Wolniewicz {6} and M. Bluj {7} and K. Nawrocki {7} and A. Padee {8,9} and W. Wislicki {8} and C. Fernandez 10} and J. Fontan {10} and Y. Cotronis {11} and E. Floros {11} and G. Tsouloupas {12} and W. Xing {12} and M. Dikaiakos {12} and J. Astalos {13} and B. Coghlan {14} and E. Heymann {15} and M. Senar {15} and C. Kanellopoulos {16} and A. Tirado-Ramos {17}

1. Laboratório de Instrumentaç?o e Física Experimental de Partículas, Lisbon, Portugal
2. Forschungszentrum Karlsruhe GMBH, Germany
3. Instituto de Fisica de Cantabria (CSIC-University of Cantabria), Santander, Spain
4. Instituto de Fi'sica Corpuscular (CSIC-University of Valencia), Valencia, Spain
5. ACC CYFRONET, Krakow, Poland
6. Poznan Supercomputing and Networking Center, Poland
7. A.Soltan Institute for Nuclear Studies, Warsaw, Poland
8. Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw,Poland
9. Instytut Radioelektroniki PW, Warsaw, Poland
10. CESGA, Centro de Supercomputacion de Galicia, Santiago de Compostela, Spain
11. National Center for Scientific Research "Demokritos", National and Kapodistrian University of Athens, Greece
12. University of Cyprus, Cyprus
13. Ustav Informatiky Slovenska Akademia Vied, Bratislava, Slovakia
14. Trinity College Dublin, Ireland
15. Universitat Autonoma de Barcelona, Spain
16. Aristotle University of Thessaloniki, Greece
17. Universiteit van Amsterdam, Netherlands

The CrossGrid international distributed testbed shared resources across sixteen European sites, being one of the challenging points of the project. The sites ranged from relatively small computing facilities in universities to large computing centres, offering an ideal mixture to test the possibilities of a Grid framework to support interactive parallel applications.
In this paper a detailed description of the organization of this testbed is provided: installation of the resources, middleware evolution, certification authorities, integration effort and operation statistics. Finally the support offered for interactive applications demonstrations is described.

PeerThing: P2P-based Semantic Resource Discovery

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

In large, heterogeneous Grid systems, resource discovery is a main challenge. Future matchmakers have to fulfil two important requirements: On the one hand, they have to be semantically rich, in order to support a large variety of resource types and descriptions of dependencies and compatibilities. On the other hand, they have to be scalable to a huge number of resources and services. However, these requirements contradict to some extend. Semantically rich reasoning services, like DL reasoners, tend to be slow and not scalable.

Thus it is desirable to harness distributed processing powers for the reasoning process. The PeerThing system developed at the Paderborn Center for Parallel Computing follows this idea. Each pool of resources runs a so-called pool-head which knows about the local resources and maintains the static resource information. It uses a DL reasoner to structure and query this information, and it queries dynamic information like current load or available disk space directly from the resources or an accompanying monitoring system.

The pool-heads are connected by a gnutella-style p2p network. As query routing in unstructured p2p networks is done by flooding the network, the dissemination has to be restricted, typically with a time-to-live field. This means, that presumably only a part of the peers are reached. Thus, we extended the network by a mechanism to establish physical network locality in the p2p network. Thus the query always reaches local resources first (in terms of network locality), which are likely to be the best resources for the query, because they are best suited both to transfer the data and to do interactive communication.

The system is accompanied by a GUI client which uses the underlying resource ontology to guide the user through the possible queries, restricting selections to useful parameters. As it is else nearly impossible for untrained users to formulate a meaningful query, this is an important step towards the usability of complex grid systems.

Currently, the system is designed to discover desktop resources for the Paderborn campus Grid system (Piranha). Without having a central institution for managing the hardware and software installation of the numerous nodes at the campus (e.g. nodes in student pools or staff offices within the different faculties), the Piranha user is able to query for specific resources which may be available somewhere within the university. The user may also query for dynamic information, e.g. currently available main memory or storage capacity. However, as the architecture is flexible with respect to the kind of information which is stored and also to the underlying information sources, other use cases are possible.

Performance Aspects of Running Thermomechanical Applications in CLUSTERIX Environment

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

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

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

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

Practical Experiences with User Account Management in Clusterix

Michal Jankowski, Pawel Wolniewicz, Norbert Meyer
Poznan Supercomputing and Networking Center, Poland

In this paper we describe experiences with using Virtual User system in Clusterix- Polish national cluster of Linux systems. Virtual User System was designed to simplify the management of user accounts in a Globus based grids.
The system allows migration from the gridmap file -based to Virtual Organization -centric user management. Great part of the administrative burden is moved from the local administrator to the VO manager. The solution combines security policy of the VO and resource owner. The system assures proper isolation and logging of jobs, so it is possible to track user actions both for accounting and security reasons. The standard accounting data is gathered automatically and stored in a database. Except the mentioned management level features, a good user management system should be possibly invisible for the user, use as little resources as possible and produce minimal time overhead.

We tried to verify the mentioned requirements in practice. We point out virtues and drawbacks of the system basing on our experiences and in comparison with other existing solutions in the area. The analyze is done from the point of view of user, system administrator and manager of Virtual Organization. We also performed several tests in order to check performance of the system. The results and conclusions are presented in the paper.

Real Time Monitoring of Web Service Applications

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

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

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

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

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

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

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

References

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

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

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

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

Regional Customization of EGEE Infrastructure in Central Europe

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

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

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

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

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

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

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

Service Challenge Tests of LCG Grid Infrastructure for Large Hadron Collider Experiments at CERN

Andrzej Olszewski
Institute of Nuclear Physics PAN, Poland

The high energy physics experiments located at the new Large Hadron Collider at CERN are planning to perform their data processing tasks on the Grid. The LHC Grid should provide and maintain a data storage and analysis infrastructure for the entire high energy physics community that will use the LHC. To test the reliability and performance of the Grid services a series of Service Challenge tests has been prepared. Currently a third phase of these tests is running and the details of preparations and results from these tests will be presented.

Services for Tracking and Archival of Grid Job Information

F. Dvorak, D. Kouril, A. Krenek, L. Matyska, M. Mulac, J. Pospisil, M. Ruda. Z. Salvet, J. Sitera, J. Skrabal, M. Vocu
CESNET z.s.p.o., Praha, Czech Republic

P. Andretto, S. Borgia, A. Dorigo, A. Gianelle, M. Mordacchini, M. Sgaravatto, L. Zangrando
INFN Sezione di Padova, Italy

S. Andreozzi, V. Ciaschini, C. Di Giusto, F. Giacomini, V. Medici, E. Ronchieri
INFN CNAF, Bologna, Italy

G. Avellino, S. Beco, A. Maraschini, F. Pacini, A. Terracina
DATAMAT S.p.A., Roma, Italy

A. Guarise, G. Patania

INFN Sezione di Torino, Italy

M. Marchi, M. Mezzadri, F. Prelz, D. Rebatto
INFN Sezione di Milano, Italy

S. Monforte, M. Pappalardo
INFN Sezione di Catania, Italy

In this paper we present recent progress in development and deployment of two related job-tracking services developed in the EU EGEE project: Logging and Bookkeeping and Job Provenance.

The Logging and Bookkeeping (L&B) is a specialised distributed grid monitoring system which gathers information on a grid job from various sources and provides the end-user with aggregate view on her job state. The service was developed originally in the EU DataGrid project and it has been successfully deployed in this project as well as in others (LCG, CrossGrid). After a brief review of the core functionality and presenting statistics on the deployment scale we focus on new features (L&B proxy -- a lightweight component computing a partial view on grid job state based on locally gathered information only) and new foreseen applications -- support for computing grid usage statistics and Computing Element reputability ranking used as a feedback information for grid job scheduling.

We also present a longer term vision of extending the meaning of the term "grid job" to cover also e.g. data transfers or resource reservations, and adapting the L&B design into a modular system which would allow tracking life cycles of different kinds of such "jobs".

Unlike L&B, which is designed to proces live information on jobs, primary tasks of the newly developed Job Provenance (JP) service are long term archival of this information, data mining in these records, and their reusal for e.g. job resubmission. We describe its overall architecture, consisting of permanent Primary Storage servers used to gather and archive the job records, and volatile Index Servers which are created and populated semi-dynamically to serve particular user queries, and present typical usecases. We focus on an introduced flexible concept of generic job attributes, and the system of JP plugins which allows storing and handling virtually arbitrary data in various
formats.

The JP service was developed recently in the EU EGEE project and it is planned to be included in the upcoming release of the gLite middleware. Therefore we believe to be able to report the first deployment experience at the time of the conference.

Simulation Method for Estimation of Security Overhead of Grid Applications

Wojciech Rzasa (3), Marian Bubak (1,2), Bartosz Balis (1), Tomasz Szepieniec (2)
(1) Institute of Computer Science, AGH, Krakow, Poland
(2) Academic Computer Centre CYFRONET, Krakow, Poland
(3) Rzeszow University of Technology, Rzeszow, Poland

In past few years security of information transmitted over the computer networks become an important issue as network protocols used in Internet were not designed to meet high security demands of current applications (see eg. [2]). The security may be enhanced by using protocols based on symmetric and asymmetric cryptography to establish secure connections over insecure Internet links [3].

Security requirements of Grid applications are significantly complex thus could not be met by any standard solution. Therefore Grid Security Infrastructure [4] was worked out as a possible solution for the Grid.

Protocols based on cryptography may provide required security level. However, degradation of efficiency becomes and important disadvantage of this solution, particularly in case of frequent communication. Cryptography algorithms used in order to establish secure communication, ensure integrity and confidentiality of transmitted information, require additional computations and produce additional data that should be transmitted over the network. Results of experiments that confirm significant influence of security overhead on communication efficiency were presented in [1].

Security overhead resulting from application of protocols based on cryptography affects efficiency of whole distributed applications. Therefore we found useful to elaborate a method capable of estimating security overhead for Grid applications. The method should enable developers to estimate resource consumption and execution time in order to achieve a required security level for a given application, so elaboration of a high level intuitive model of the application and resources is necessary. It should also enable to easily model application that is still beeing developed. The model should use information concerning network communication between application processes, and information about security level. In order to obtain reliable results we decided to use Timed Coloured Petri Nets to create the executable model [5]. TCPN are designed to analyze concurrent processes with respect to time restrictions and TCPN models may be interactively simulated taking into consideration: concurency, synchronization, mutual exclusion, conflict and time. For these reasons it is an appropriate formalism to enable modelling of security overheads.

References
[1] Balis B., Bubak M., Rzasa W., Szepieniec T.: Efficiency of the GSI Secured Network Transmission, Proc. of Internationnal Conference on Computational Science, Krakow, Poland, June 2004, LNCS 3036, pp 107-115, 2004, Eds.: Marian Bubak, Geert Dick van Albada, Peter M. A. Sloot, and Jack J. Dongarra.
[2] Bellovin, S., Security Problems in the TCP/IP Protocol Suite, Published in Computer Communication Review vol. 19, no. 2 (April 1989) pages 32-48; http://www.research.att.com/~smb/papers/ipext.ps
[3] Dierks T., Allen C.: The TLS Protocol Version 1.0. January 1999, RFC-2246.
[4] Foster I., Kesselman C., Tsudik G., Tuecke S.: A Security Architecture for Computational Grids. Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.
[5] Jensen K.: Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. vol. 1-3. Berlin, Springer-Verlag 1995/96.

Steering and Interactive Visualization on the Grid Using the UNICORE Grid Middleware

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

The current grid tools allow for easy job submission and retrieval of the created data. User uses external applications to prepare input or perform advanced postprocessing. The results are stored in the files used as input for another job run in the grid environment which is used to transfer input and output files and to submit and control job.

Progress in the network bandwidth allows now for efficient transfer of large data files with the acceptable performance suitable for the interactive visualization which goes behind traditional file transfer schema. Recently, we have proved that latency and communication overhead caused by the grid middleware cam be reduced to the level which allows for on-line visualization in the grid environment.

In this paper we extend further capabilities of the UNICORE grid middleware [1] in order to provide mechanism for application steering using VISIT protocol [2]. The mechanism of the UNICORE gateway plugins has been used to stream visualization data from the computational server to the visualization workstation. The authorization and authentication as well as job control are performed using UNICORE tools providing standard grid capabilities. The important advantage of the presented solution is good security model and ability to work with the firewalls and local networks. No additional ports, except one used by the UNICORE gateway must be opened.

The developed technology for the on-line visualization allows for bi-directional communication which allows for application steering. The VISIT protocol has been used for sending steering requests from the visualization tools to the application.

The interactive steering and visualization tools were tested using high performance applications, proving ability to perform interactive visualization in the grid environment.

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

[1] http://unicore.sf.net
[2] http://www.fz-juelich.de/zam/visit/

Support for Automatic Workflow Composition Based on Ontologies in Semantic Grid Environment

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

The idea of an application based on workflow becomes more popular in the Grid community as it is an intuitive and natural method of functional decomposition of an application, where the important dependencies are expressed in set of connections of data flow and/or control flow [1].
While the scientific workflows are growing in size and complexity, a tool assisting an end user becomes a necessity [2].

In this paper we propose a novel approach to development of such a tool. Such an assistant tool should be able to analyze user's requirements about application results and should be able to work with information services (e.g. registries) that provide information on resources available in the Grid. The Workflow

Composition Tool is meant to provide the functionality of automatic workflow construction based on the process of semantic functionality discovery and matchmaking. Following a set of rules concerning similarity of data and side effects, the tool is based on the idea of IOPE-based comparison to find suitable elements that would build the future workflow. The composition takes place on higher level of abstractness so the subject of comparison are rather functional properties of interfaces instead of non-functional properties of certain realizations (like performance, failure rate etc.). The experience with our workflow composer and registry described in [3,4] was taken into account.

The paper presents details of the tool, including the design of composition algorithm and the method of ontological comparison used. A workflow composition is based on the Colored Petri Nets formalism where transitions denote operations, places denote data or control flow and different colors indicate different levels of abstraction of elements. Through analysis of input workflow the tool is able to identify a set of unmet dependencies that point out what parts of the workflow needs another, not yet found parts to work properly. Such a required elements are then discovered through querying mechanism to an external ontology store which contains semantical descriptions of software resources available to the user.
We present some recent results of using the tool in the K-WfGrid project [5]: to build workflows of applications from areas like Coordinated Traffic Managament or Flood Forecasting Simulations. Especially the second case has the level of internal complexity that makes the application of the workflow composition tool very interesting. The application has several main steps of simulation cascade and each step introduce a set of different tools that are similar in functionality yet differ in realization techniques. Additionally, between those main steps there are required some minor transformation activities.

We also provide some ideas how to follow that research towards the objective of truly invisible semantic Grid.

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

References
[1] C. Lee, D. Talia: Grid Programming Models: Current Tools, Issues, and Directions, Chapter 21, pp. 555-578 in: F. Berman, G.C. Fox, A.J.G. Hey: Grid Computing - Making a Global Infrastructure a Reality, John Wiley & Sons 2003.
[2] D.W. Walker, E. Houstis (guest editors): Complex Problem-Solving Environments for Grid Computing, FGCS Grid Computing vol.21, no 6, pp. 841-968, 2005.
[3] M. Bubak, T. Gubala, M. Kapalka, M. Malawski, K. Rycerz: Workflow Composer and Service Registry for Grid Applications. Future Generation Computer Systems 1(21), pp.79-86, 2005
[4] T. Gubala, M. Bubak, M. Malawski, K. Rycerz: Abstract Workflow Composition in K-WfGrid Project Environment, 4th Cracow Grid Workshop (CGW04)- Workshop Proceedings, ACC Cyfronet AGH, 2005, pp.85-92
[5] www.kwfgrid.net

Testing Grid Software: The Development of a Testing Framework for Collaborative Distributed Systems

Florian Urmetzer (1), Gareth J Lewis (1), Vassil N Alexandrov (1) and Peter Kacsuk (2)
(1) Advanced Computing and Emerging Technologies Centre, School of Systems Engineering, University of Reading, UK
(2) MTA SZTAKI, Laboratory of Parallel and Distributed Systems, Budapest, Hungary

Testing software is often seen as the hardest part of a software project. This perception may be true, because specific experiences in different specialist programming domain are rather hard to find. Testing collaborative software is certainly one of them. In this article the authors describe their experiences with the testing of a collaborative distributed P-Grade software tool and would like to offer a framework for testing. The setup is introduced and relevant issues are discussed, which have been occurring during the development of the collaborative version of the P-Grade portal. The outcome is a testing framework, which we hope is transferable to other Grid projects.

The German D-Grid Initiative

Harald Kornmayer (1) for the D-Grid initiative
(1) Forschungszentrum Karlsruhe, Germany

As Grid technology emerged fast over the last few years, many national initiatives started in different countries to establish an infrastructure to enable e-Science. In September 2005 the german national Grid initiative, called D-Grid (1), started to build a sustainable infrastructure for Grid-Computing and e-Science in Germany. Currently 6 community projects and one integration project were funded to provide Grid tools for scientists in different areas like medicine, climate research, high energy physics, engineering, astrophysics and social science.

In the paper the project goals will be presented as well as the organisaton of the projects. The main use cases of the different scientific areas will be described. As the D-Grid initiative aims from the beginning to build an integrated D-Grid platform for e-Science, the integration procedures will support different middleware (including GT4, UNICORE and gLite) and tools. All efforts in the project will be aligned to the provision of a sustainable platform to enable scientists to collaborate on the national and international level.

References:
(1) D-Grid homepage, http://www.d-grid.de

The Grid-embedded Interactive Jobs Invocation

Marcin Lawenda, Marcin Okon, Ariel Oleksiak, Bogdan Ludwiczak, Tomasz Piontek, Juliusz Pukacki, Norbert Meyer, Jaroslaw Nabrzyski, Dominik Stoklosa
Poznan Supercomputing and Networking Center, Poland

In this article its authors discuss the various aspects and problems connected with interactive jobs management in the grid environment. The specific nature of scheduling the interactive experiments is closely analyzed, revealing the complex approach that has to be taken into account in order to consider many new, scientific-domain scheduling factors. Diagrams for submission, launching, prolonging and ending the interactive sessions are proposed. The presented idea is generic and can be used in many grid-based systems. The grid broker e.g. Grid Resource Management System (GRMS) is required in our concept and we assume its presence in the environment.
There are two main kinds of jobs in the grid systems: regular (batch) jobs and interactive/visualization tasks (the ones performed in the real time, directly by the users - via the GUI). The main difference (and difficulty) between those types is that - in the interactive jobs - the time slot reserved for running a job on a computational machine must be synchronized with user preferences, considering specific work hours, daily schedule etc. Moreover, there are also maintenance periods, when a resource is unavailable.
Middleware systems for interactive jobs' invocation in the grid-based systems are in the initial state. It is very hard to find a general solution for the wide spectrum of applications due to different users and software requirements.
In the paper a general concept of interactive jobs invocation is presented. It is assumed that the Grid Resource Management System (GRMS) is present in the computing environment. The VNC Manager and the VNC server are installed on every machine. Moreover, the Interactive Applications Manager responsible for managing information about running interactive jobs is also given. All services rely on Globus.
The presented solution can be used by many client applications. The Virtual Laboratory (VLab) system can be an exemplary client of this solution. Most of applications used in virtual labs are interactive. They are used to perform experiments, control lab devices, analyze postprocessing data, etc.
The VLab is not a standalone system. It was designed to cooperate with many other grid systems, providing only the purpose-specific functionality and relaying on well known and reliable grid solutions. The most important system the VLab cooperates with is the Globus Toolkit - in the scope of scheduling computational tasks, software services and libraries for resource monitoring, discovery, and management. All computational tasks submitted in the VLab system are transferred to the Globus via the GRMS module. Among other external systems used by the Virtual Laboratory are: the VNC system, DMS (Data Management System), Authentication module and GAS authorization system.

Towards Multiprotocol and Multilanguage Interoperability: Experiments with Babel and RMIX

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

The component programming model is getting more popular for Grid computing, because of such advantages as code modularity, reusability, and facilitated mechanisms for creating components on distributed resources as well as remote connections between them. There are possible multiple protocols such as e.g. SOAP or JRMP, which may be used for intercomponent communication, each of them suited better for different occasions. On the other hand, the components themselves may be written in many programming languages, to name only Java, C, C++, Fortran 70 or Fortran 90. It is impossible, especially in Grid environment, to force the usage of single protocol or programming language, even within a single application. Therefore there is a need for providing solutions for interoperability between them.

Our approach to solve this interoperability problem is to investigate such software toolkits as RMIX [1], which is a multiprotocol library extending Java RMI, and Babel [2], which provides mechanisms of interoperability between multiple languages.
RMIX is an extensible RMI library, allowing for protocol negotiation and dynamic protocol switching at runtime. Protocols supported by RMIX include JRMP, SOAP and Sun RPC. It is also possible to switch the transport layer to use e.g. JXTA instead of TCP/IP. Babel is a software generating tool and runtime library for interoperability between Java, C, C++, Fortran 77, Fortran 90 nad Python. The interfaces of Babel objects are dwscribed in SIDL (Scientific Interface Description Language), and babel generates client and server bindings to these interfaces in all available languages. There is also experimental support for remote method invocations.

In this paper we are showing our experiences with using Babel RMI support and plugging in RMIX as the communication library. Such coupling of Babel and RMIX gives us the ability to interoperate of multilingual components and using multiple protocols, making the remote connections very flexible. Our further goal will be to plug the RMIX enabled Babel into MOCCA [3], which is a CCA-compliant distributed component framework, running on H2O resource sharing platform. This wll lead to a flexible and extensible component platform, well suited for Grid programming, where heterogeneity of languages and protocols is still unsolved problem.

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

References:

[1] D. Kurzyniec, T. Wrzosek, and V. S. Sunderam. Heterogeneous access to service-based distributed computing: The RMIX approach. In Heterogeneous Computing Workshop, in conjunction with IPDPS 2003, Nice, France, April 2003. IEEE Computer Society.
[2] S. R. Kohn et al. Divorcing Language Dependencies from a Scientific Software Library. In Proc. of the 10th SIAM Conf. on Parallel Processing for Sci. Comp., Portsmouth, USA, Mar. 2001. SIAM.
[3] M. Malawski, D. Kurzyniec, V. Sunderam, MOCCA - Towards a Distributed CCA Framework for Metacomputing, Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Joint Workshop on High-Performance Grid Computing & High-Level Parallel Programming Models - HIPS-HPGC, April 4-8, 2005, Denver, Colorado, USA, IEEE Computer Society Press, 2005, pp. 174a.

Trust Establishment in Pervasive Grid Environments

Syed Naqvi, Michel Riguidel
Computer Sciences and Networks Department, Ecole Nationale Supérieur des Télécommunications, Paris, France

Pervasive grids are characterized by their global mobility feature that enable them to hook anywhere at anytime to the computing resources. Therefore, providing a dynamic and adaptive security model that overlays a secure framework over an untrustworthy network is one of the biggest challenges in Pervasive Grids. Current grid security enforcement and policy maintenance models are generally based on the assumption of a stable, static, and long-term grid establishment with a small set of seldom-changing users. Hence, these frameworks cannot be directly applied to the pervasive grid framework. Ad-hoc and federated grids require an adaptive security model that incrementally builds a secure grid community based on the notion of trust. In this paper, we have outlined our approach to handle the challenging problem of establishing trust in the pervasive grid environments where there is no a priori trust among its entities and no mechanism to build some trust based on a history of previous interactions.

Using Ant Colony Optimization for Collaborative (Re)Search in Data Grids

Uros Jovanovic (1), Jaka Mocnik (1), Bostjan Slivnik (2)
(1) XLAB, Ljubljana, Slovenia
(2) University of Ljubljana, Slovenia

In recent years, leveraging the ubiquitous networked environments, organizations and individuals participating in various communities (scientific being only one amongst many) are forming virtual organizations. Such networked communities usually produce distributed data. This paper describes a framework for automated extraction and organization of structured data in a large-scale, dynamic grid environment, and subsequent efficient discovery of thus procured data.

In the scientific environment, data grids are often intended for storage and manipulation of vast data produced during experiments. But instead of limiting user queries to search through a predefined metainformation extracted from these data, the data grid should support more sophisticated queries on the original data as these queries require information about yet undefined relations between data and/or metadata.

Discovery of data in an unstructured network with no fixed topology is hard to solve efficiently. Most current techniques are based on flooding and random walks, resulting in poor performance. Improving the efficiency of such queries is necessary.

The proposed method for efficient extraction and discovery is based on self-organizing properties known from the complex adaptive systems and artificial life systems. The discovery process is facilitated by a self-organizing distributed index that is maintained by the queries themselves using a combination of techniques and algorithms from swarm intelligence theory.

The algorithms for the division of labour from ant colony optimization (ACO) methods provide the basis for distribution of programs for information extraction designed by individual researchers. The search results are returned to the task owner - different ants might extract different parts of the result. In order to organize the discovered data, algorithms for cemetery organization and brood sorting, also from ACO, are used. The data gathered at the cemeteries provide a preview of (successful) search results to other researchers.

The framework is based on the Globus Toolkit 4 infrastructure. The security is provided by GT4 authentication mechanism, preventing unauthorized access to data. The structured data are extracted from a variety of documents by special information retrieval programs and stored as XML document fragments in the GT4 WS-MDS Index.

Using MOCCA Component Environment for Modeling of Gold Clusters

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

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

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

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

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

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

References

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

Using the OCM-G for Performance Monitoring of GRID Superscalar Applications

Rosa M. Badia [3], Marian Bubak [1,2], Wlodzimierz Funika [1], Marcin Smetek [1]
[1] Institute of Computer Science, AGH, Krakow, Poland
[2] ACC CYFRONET AGH, Krakow, Poland
[3] Universitat Politecnica de Catalunya, Spain

An important role in any distributed system and especially in Grid environments is played by performance monitoring tools, due to the fact that performance and monitoring information is required not only by the user to get information about the infrastructure and the running applications, but also by most Grid facilities to enable correct resource allocation, job submission and many other activities.

The GRID superscalar (GS) [1], one of approaches to Grid computing, supports the development of applications, in a way transparent and convenient for the user. Its aim is to reduce the development complexity of Grid applications to the minimum, in such a way that writing an application for a computational Grid can be as easy as a sequential program. The idea assumes that a lot of applications is based on some repeating actions, e.g. in form of loops. The granularity of these actions is of the level of simulations or programs, and the data objects will be files. GS provides an underlying run-time environment capable of detecting the inherent parallelism of the sequential application and performs concurrent task submission. In addition to a data-dependence analysis based on these input/output task parameters which are files, techniques such as file renaming and file locality are applied to increase the application performance.

The above reasons motivated a design of a monitoring facility that supports development of applications to be run in the Grid environment using the GS system, to get on-line deeper insight into how an application behaves in such an environment, to help in its effective and fault-tolerant execution.

In this paper, we focus on the concept and implementation ideas of adapting the OCM-G monitoring system [2] to support GS applications. Its features allow to fit it well into the requirements of running an application on the Grid. In particular, we discuss what metrics are important to assess the performance of the application, these related to standard metrics like an operation time as well application specific metrics, expressed in a special language PMSL [3] allowing the user to define performance indicators most meaningfully giving the context-dependent features of the application and how to get them. Then we come to the general architecture of the functioning of OCM-G in the GS, and its implementation details.

Acknoledgements. This research was partially funded by the CoreGrid EU IST project and the CoreGrid SPUB-M.

References
[1] Rosa M. Badia, Jesus Labarta, Raul Sirvent, Josep M. Perez, Jose M. Cela, and Rogeli Grima: Programming Grid Applications with GRID Superscalar, Journal of Grid Computing, vol. 1, 2003, pp. 151-170.
[2] 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.
[3] B.Balis, M. Bubak, W.Funika, T.Szepieniec, R.Wismueller, and M.Radecki. Monitoring Grid Applications with Grid-enabled OMIS Monitor. In: F. Riviera, M. Bubak, A.Tato, and R. Doallo, editors, Proc. First European Across Grids Conference, pages 230-239, LNCS 2970, Springer, Feb. 2003.

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

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

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

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

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

Virtualisation for Grid-Computing

Marcus Hardt
FZK, Karlsruhe, Germany

Within the vast realm of computer science, one truism has held across all languages and environments: "there is no problem that cannot be solved by adding another layer of indirection."

One problem in distributed computing is bringing together application developers and resource providers. This is required to ensure that applications work well on the resources provided. Although Linux forms a common base on the majority of installations, various different Linux-distributions are available. Different resource provides prefer different linux distributions as do different application-developing groups.

A layer of abstraction between resources and applications can provide new possibilities in designing Grid solutions. Virtualisation provides such a layer of abstraction.

The virtualisation environment Xen (developed at the University of Cambridge / UK) is a novel approach to virtualising Operating Systems, including Linux, *BSD and Plan9. Xen uses the paravirtualisation approach in order to provide minimum performance loss in conjunction with minimal modification of the operating systems.

This contribution will explain the architecture of Xen, in particular the paravirtualisation. Furthermore we will present our approach to installing the recent EGEE middleware gLite. Experiences gained during a gLite installation course this fall, that was held on Xen-virtualised resources, will be published. Finally, preliminary performance measurements will be shown.

VOCE - A Grid Environment for Central Europe

J. Chudoba, L. Fiala, J. Kmunicek, J. Kosina, D. Kouril, M. Lokajicek, L. Matyska, M. Ruda, J. Svec

The grids provide a complex environment with several functionalities and immense power. Despite the number of possibilities offered by the grids, their potential is still not utilized to the maximum. The grid systems are quite complicated to get familiar with and require lot of training for the common users to be able to use the grids to solve their problems. Also for smaller groups of applications, it is unconvinient or even imposible to install and operate their own grid systems as such a task requires lot of time and experience of the administrators. Current grids are usualy utilized by large specialized groups of applications focusing at solving particular research problems. Attracting other applications is not easy but it is highly desirable for the grids to support multiple application areas and to provide a more general support for various kinds of problems.

This paper provides an overview of VOCE -- a VO for the Central Europe (CE) federation. VOCE was established as part of the EGEE project activities of the CE federation, which is composed of institutions from Austria, Czech Republic, Hungary, Poland, Slovakia and Slovenia. VOCE offers users from the Central Europe federation a complete set of grid services. The VOCE environment aims at supporting of grid novices that seek how to get an easy and fast access to grid technologies and provides them with an access to an enviroment that can be used to gain experience with the grids. Another goal of VOCE is to support of smaller application groups that cannot afford building their own grid infrastructure but would like to test the grid services and their usability for solving their particular problems.

The paper describes the environment and services provided by VOCE and also procedures necessary to get access to the VOCE resources.

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

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

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

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

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