ORAL PRESETATIONS
(ABSTRACTS)

Tuesday, October 16, 2007

Session C1 **11:45 – 13:00, aula**

InteliGrid Project: Lessons Learned and Future Work

Matevz Dolenc (1), Ziga Turk (1), Krzysztof Kurowski (2) and Peter Katranuschkov (3)
(1) University of Ljubljana, FGG, Ljubljana, Slovenia
(2) Poznan Supercomputing and Networking Center, Poznan, Poland
(3) Technical University of Dresden, Dresden, Germany

A challenge for collaboration infrastructures is to support dynamic virtual organisations that collaborate on the design, production and maintenance of products that are described in complex structured product model databases. Such virtual organisations are typical for industries with long supply chains like automotive, shipbuilding and aerospace. Perhaps the most complex dynamically changing virtual organisations are in architecture, engineering and construction sector.

The InteliGrid project addressed the above challenge by successfully combining and extending state-of-the-art research and development in three key technology areas: (a) semantic interoperability, (b) virtual organisations, and (c) grid technology. The developed platform provides standards-based collection of ontology based services and grid middleware in support of dynamic virtual organisations as well as grid enabled domain specific engineering applications. The platform services include interoperability services, business domain specific client and server side applications, and extended existing grid middleware where a clear focus was on providing a secure access and integration of diverse data resources.

In this paper a general overview of the InteliGrid results is presented together with the lessons learned in development and deployment of an engineering grid-based collaboration platform. Related future research and development work is also discussed.

Towards Greater Grid Universal Accessibility: Initial Benchmarks and the Road Ahead

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

To establish a global grid infrastructure meeting the needs of a wider user community and catering for the EU FP7 challenge of "Access for all: inclusion and elderly access", greater accessibility to the grid will increase in importance in grid research agendas. This is fueled by cultural differences in user communities, the varying spectrum of requirements, and varying levels of qualifications and abilities of human resources using the grid worldwide. Despite the availability of various grid portals none of them address the wider agenda of universal accessibility to a global grid. This motivates our work on the development of preliminary benchmarks to assess the universal accessibility of existing grid portals. Based on our results, a research agenda is outlined for greater universal accessibility to the grid.

This paper is divided into five sections. The first section overviews and classifies existing grid portals in five categories and two development frameworks. Categories of grid portals include: 1) portals providing single access point for user support; 2) portals providing a user-friendly access to services of a single grid; 3) portals providing access to services of multiple grids; 4) portals supporting grid enabling applications; and 5) portals supporting workflow. Portal development frameworks include: 1) frameworks for building grid portals; and 2) frameworks supporting grid accessibility via various media delivery channels. Our overview of grid portals covers the developers, the objectives, the implementations, and the features.

The second section establishes comparative benchmarks for universal accessibility to the grid. The criteria for universal accessibility to the grid are identified and classified by order of importance. The third section assesses the universal accessibility to the grid according to various criteria including multi-lingual support, patterns of interaction, accessibility levels, accessibility through various media delivery channels, disabled access, elderly people access, and rich media content. Our results reveal a low level of accessibility to existing grid portals, mainly due to the absence of any concept of universal access at design stage. Existing grid portals should be reengineered for greater accessibility.

The fourth section considers use case scenarios that motivate a research agenda for greater universal accessibility to the grid. These use case scenarios include: greater accessibility for grid security management; grid accessibility for commercial 3G mobile applications; accessibility for the disabled; edutaining accessibility; and better accessibility for collaborations. The paper concludes with an agenda for the road ahead towards greater universal accessibility to the grid.

Dynamic Workflow Composition for Grid Applications

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

Since the Grid infrastructure is becoming more and more powerful each year, the Grid applications also grow in size and complexity. The computation of the applications usually does not consist of a single task but many tasks connected together by data dependences. Workflow management became one of the main focuses of research and developments in Grid computing.

Although many workflow management tools are available, they have a common weakness. The users must prepare the workflow in advance using some text or graphical editors, and then submit the workflow to Grid for executions. If the users want to modify the workflows after submitting them, they have to cancel the running workflow, use editor to modify the workflow description and submit it again. That makes great difficulties for applications with dynamic workflows, e.g. decision supports, where the workflows may change according to output from previous jobs.

In this paper we will present a concept for dynamic workflow composition, where the users may change the workflows at any time. The workflow engine is divided into two part: workflow composition service, which will

automatically composed the workflows according to input/output data of jobs; and job execution service, where jobs, which are ready for executions (jobs with all their predecessors executed) will be submitted for executions into Grid.

Jobs in workflow are described in the same format like GridWay (http://www.gridway.org), which is compliant to the OGF standard DRMAA, and can be easily changed to other formats. Jobs of a workflow are submitted to the workflow composition service at any time, even when the workflow is being executed. The workflow composition service will analyze the input/output data of the jobs and add them to the workflow accordingly. By using algorithms in Data Driven Graph, the overhead of the workflow composition is proved to be constant. When a job has all its predecessors executed, the workflow composition service submits it to job execution service.

Job execution service can be implemented by many ways: as a simple service wrapper for some middleware (e.g. WSRF service wrapper of gLite CLI or GridWay) or more complex agent-based scheduling tool with queuing systems and worker jobs. As all jobs, which are ready for executions, have no data dependence among them, they can be executed independently from each other.

This approach has two advantages over classical workflow engines: flexibility and simplicity: The users have only to describe jobs, not the dependence or parallelism among them, so that can simplify the applications with workflows. Workflows can be modified during their execution: the users can add more jobs to workflow or delete some waiting jobs. Splitting execution mechanism to a service make workflow engine independent from middleware and can be easily adapted to new infrastructures.

BIS-Grid: Business Workflows for the Grid

Felix Heine (1), André Höing (1), Stefan Gudenkauf (2), Guido Scherp (2)
(1) Technische Universitaet Berlin, Complex and Distributed IT Systems, Germany
(2) OFFIS Institute for Information Technology, Germany

Enterprise Application Integration (EAI) has become a well-established way to integrate heterogeneous business information systems. This provides a basis to map business processes considered as workflows to the technical system level. Often, this is accomplished via service orchestration in service-oriented architectures (SOA). By

doing so, Web Services provide means to enable service orchestration as well as to hide the underlying infrastructure.

Modern Grid technologies such as the Grid middleware UNICORE 6 and Globus Toolkit 4 are based on the Web Service Resource Framework (WSRF) which is a standard to allow stateless Web Services to become stateful. Stateful WSRF-based Web Services, also called Grid Services, provide the basis to build SOA using Grid technologies. Thus, Grid technologies and EAI have much in common since both technologies focus on integration problems within a heterogeneous environment - Grid technologies on resource level and EAI on application level.

In BIS-Grid, a BMBF-funded project in the context of the German D-Grid initiative, we intend to realise a horizontal Service Grid in the application domain of business information systems. The goal is to enable Grid technologies to be used for the integration of decentralised business information systems. The project especially addresses small and medium enterprises, which normally are not able to run own computing and storage centres used in Grid infrastructures. BIS-Grid will enable these enterprises to design and run workflows realised as Grid Service orchestrations to develop and provide dynamic solutions for arising business challenges.

One technical goal of the project is to provide a WS-BPEL-based workflow engine capable of integrating Grid Services. This engine is a composition of the WS-BPEL engine ActiveBPEL and UNICORE 6. Thereby, ActiveBPEL is solely used for workflow execution based on WS-BPEL. Upon ActiveBPEL, a Grid layer based on UNICORE 6 enables workflows to use Grid Services and to be offered as Grid Services as well. The technical issues relevant to Grid Services, in particular security, are completely managed by the Grid layer and are transparent to ActiveBPEL. Thus, our extensions will only affect UNICORE 6 so that the underlying WS-BPEL engine is replaceable with any other WS-BPEL engine.

This paper proposes the general architecture of the planned Grid-enabled workflow engine that extends WSBPEL-based service orchestration to allow the orchestration of stateful WSRF-based Grid Services. We point out the technical challenges we have to cope with and present our solution approach.

Services in Fraunhofer Enterprise Grids

Julian Bart, Anette Weisbecker
Fraunhofer IAO, Stuttgart, Germany

Starting with the interest in Grid Computing, it is necessary to check which applications are appropriate for the integration in a Grid infrastructure. How can the application be run on remote sites? What about necessary licenses? How big is the effort and how big is the benefit? How will the users use the infrastructure? The project Fraunhofer Enterprise Grids is considering these questions and has integrated a set of applications in a Grid infrastructure, the Fraunhofer Resource Grid. The Fraunhofer Resource Grid is used not only for a variety of computing problems in the Fraunhofer Gesellschaft, but also to demonstrate the possibilities of grid computing toall kinds of users. The project is delivering answers to the necessary technologies concerning application integration, Grid infrastructure, storage management and all required middleware components for a solid working Grid. An important factor of the success of a Grid infrastructure is the acceptance of the users. One way to use the Grid is the installation of a Grid middleware on the user's site. Together with this installation, the user needs to be registered on all computing and storage systems he is allowed to use. This might be a solution for an expert user, who is frequently using the Grid from one site, but for first-time, test or sporadic users, this is far too complex. Fraunhofer Enterprise Grids (EPG) is answering this challenge with a portal, which allows the user to easily access the applications, storages and computing resources. The portal is integrated in the Fraunhofer ResourceGrid.The portal was and will be developed and tested with the methods and processes of User Centred Design (UCD)and the Portal Analysis and Design Method (PADEM). This keeps the requirements of the users in the centre of attention and is based on the design of interaction between users and the system, referred to as interaction design.

Grid support for A Toroidal LHC ApparatuS (ATLAS), a part of the Large Hardon Collider (LHC)

Jan Pieczykolan (2), Łukasz Dutka (2), Krzysztof Korcyl (3), Tomir Kryza (1), and Jacek Kitowski (1,2)
(1) Institute of Computer Science AGH University of Science and Technology, Krakow, Poland
(2) Academic Computer Centre CYFRONET AGH, Krakow, Poland
(3) Institute of Nuclear Physics, PAN, Krakow, Poland

Industry solutions that are based on grid architecture or just use a production grid installation are still a rarity. Despite efforts undertaken by academia, international cooperation and EU support, the industry fails to take full advantage of the potential of distributed computation using resources combined in grid as a virtual organization. This is partly due to the fact that computational power is cheap nowadays and even a medium-size organization can build a powerful computation environment on its own. However, another reason comes from the existing grid implementations and mechanisms, which are most often batch-based and do not meet the industry requirements of soft real-time processing, data stream processing and interactivity, posed by modern systems and users. An example of such a system is the ATLAS solution, developed at CERN to support high energy physics calculations.

The software built for the ATLAS project is assigned to handling streams of data from sensors installed at the Large Hadron Collider accelerator (LHC) located in CERN. It is characterized by high frequency of incoming data and strict time regime on their analysis. Meeting the requirements posed by this software requires large amount of computational power which is achieved at present with a cluster farm with nodes efficiently connected.

The goal of the paper is to present results of grid architecture application to processing data from sensors, acquired from the ATLAS detector, as a part of LHC [1].

Since the available grid middleware is not adapted for online processing required by algorithms of ATLAS software, a dedicated solution – Real-Time Dispatcher – is built. The aim of this solution is to deliver an abstraction layer over the grid that uses job approach, allowing for online processing of data streams with strict QoS requirements. In fact, the solution implements the master-worker paradigm in the classic computational grid environment [2]. The master is a slightly modified component of the ATLAS software, that retrieves data from ATLAS detectors and delegates data to worker processes that are responsible for performing computations using algorithms that are a part of ATHENA framework. The worker itself is a process that runs on Grid as a job.

The article describes the integration method of ATLAS with the Real-Time Dispatcher and the Grid. Moreover, evaluation of efficiency tests is presented and discussed. Thereby a method of adopting a kind of legacy applications for the Grid is shown.

Acknowledgements. This research has been done in the framework of EU IST 031857 int.eu.grid project, AGHUST grant is also acknowledged.

References

1. J. Pieczykolan, L. Dutka, B. Kryza, K. Korcyl and J. Kitowski: Data Dispatcher for Real Time applications in Grid environment, ICCS 2007, poster not in proceedings.
2. J. Pieczykolan, K. Korcyl, L. Dutka and J. Kitowski: Implementation of master-worker paradigm for high throughput applications in grid environment, e-Science 2007, Bangalore, India (under review).

ATLAS Distributed Analysis

S. González de la Hoz (1), L. March (1), D. Liko (2)
(1) IFIC-Valencia (Instituto de Física Corpuscular)
(2) CERN (European Organization for Nuclear Research)

The ATLAS production system has been successfully used to run production of simulation data at an unprecedented scale. Up to 10000 jobs were processed by the system in one day. The experiences obtained operating the system on several grid flavours was essential to perform a user analysis using grid resources. First tests of the distributed analysis system were then performed. In the preparation phase data was registered in the new LHC File Catalog (LFC) and replicated in external sites. For the main test, few resources were used in nine sites.

All these tests are only a first step towards the validation of the computing model. Reality tests should involve many physicists working in concurrent mode. The ATLAS management computing board decided to integrate the collaboration efforts in distributed analysis in only one project, Ganga. The goal is to test the ATLAS reconstruction and Analysis software in a large scale Data production using LCG (LHC Computing Grid) grid flavor in several sites. Ganga allows trivial switching between running test jobs on a local batch system and running large-scale analyses on the Grid, hiding Grid technicalities; it provides job splitting and merging, and includes automated job monitoring and output retrieval.

Interactive grid access using Matlab

Marcus Hardt
Forschungszentrum Karlsruhe GmbH (FZK), Institute of Scientific Computing (IWR)

Matlab is a widely used problem solving environment. It is mainly being used in the fields of science and engineering for prototyping and development of algorithms. Often, a lot of computing power is required to run these algorithms. However, Matlab is usually bound to running on only computer. This leads to a considerable amount of time that is required to solve problems. This puts a major constraint on the size of problems that can be solved using matlab.

One way to overcome this problem is the commercially available distributed computing toolbox. This toolbox is expensive and is not available for todays grid infrastructure.

This contribution will show one possibility way to access the grid from within matlab. We have developed an integrated solution that integrates matlab with the gLite middleware. We will show how we enable remote procedure calls (RPC) to resources that are allocated via gLite. We will furthermore show, how to make local matlab functions available remotely.

As part of the presentation a life demonstration of a simple test application will be given.

CancerGrid: Enterprise Desktopgrid Solution with Workflow Support for Anti-cancer Drug Design

Zoltan Farkas, Robert Lovas, Peter Kacsuk
MTA SZTAKI, Budapest, Hungary

Background

In the three years of this multidisciplinary EU FP6 research project (www.cancergrid.eu), the 10-member, SME-led consortium develops and refines methods for the enrichment of molecular libraries to facilitate discovery of potential anti-cancer agents. Using grid-aided IT solutions, the likelihood of finding anti-cancer novel leads will substantially increase the translation of basic knowledge to application stage. In particular, through the interaction with novel grid technologies and biology, the R&D consortium aims at

developing focused libraries with a high content of anti-cancer leads and also building models in order to describe/predict the relationship between the molecular structures and the biological, chemical, or physical properties

developing a workflow-oriented and desktopgrid-based enterprise solution, which helps to accelerate and automate the in-silico design of libraries for drug discovery processes.

In this paper, we focus on the scientific and technical aspects of the second aim exceeding the functionalities provided by the two preceding grid projects, OpenMolGRID (FP5 IST project) and ADMeTOXGrid (Hungarian national grant), served as a base point to the work.

Desktopgrid with workflow support

Complex systems always need high-level user interfaces, which are sophisticated enough to perform all activities but remains easy to use. Grid portals are nowadays developed for all kinds of Grid infrastructures, and one of the most feature-rich portals is the P-GRADE Portal (http://portal.p-grade.hu) with the ability to define workflow applications graphically and to execute them e.g. on the EGEE or on UK NGS production-level Grids. Desktop Grids currently lack such portals. Moreover, the project defines new requirements, because for the calculation of molecular descriptors and ADME parameters the existing portal does not provide sufficient high-level and scalable solution with its workflow description language. Support for Monte Carlo methods, parameter study applications, and automatic data splitting at both graphical level and workflow manager level will be designed and implemented in the frame of the project. The integration of the portal with SZTAKI Desktop Grid

(http://www.desktopgrid.hu) needs a new workflow manager with the Desktop Grid execution mechanism instead of the Globus/gLite-based and other similar job-execution mechanism used in other Grids. For the specific purpose of the project, this task will also give on-line visualisation support for model building (extending the portal by new application specific portlets), and will provide an adequate high-level support for database access. The developed portal will be used by the project partners intensively since they will have the access to the computational resources through this portal.

In this paper, we will focus particularly on the integration issues of the desktopgrid and the workflow-oriented portal including scheduling the workunits of the workflows in the desktopgrid.

Summary

The high-throughput CancerGrid system will be based on the SZTAKI Desktop Grid, making the integration of computational/storage/networking facilities possible, and meeting the strict safety requirements of the pharmaceutical industry at the same time. The system will be used to exploit the unused capacity of PCs, and will be further developed to integrate clusters and databases as resources into the new grid system. A Cancergrid portal will be developed based on the P-GRADE Portal, that will facilitate and accelerate the complex model development of the drug design with a new, enhanced workflow language and with CancerGrid specific functionalities.

The new workflows in the CancerGrid system will be able to make these calculations starting from the 2 dimensional structure of a compound, converting it to a 3 dimensional one, and finally geometrically optimising it. This enables the calculation of information-rich molecular descriptors, which – to date – were impossible to be determined within a reasonable time for large compound sets commonly used in the pharmaceutical industry nowadays. The arsenal of 2 and 3 dimensional molecular descriptors available in the system provides a possibility for building models that are much more accurate than the models presently available in the pharmaceutical field.

Grid Solving a Bioinformatics Challenge: a First Step to Anchoring the Nucleosome

Christophe Blanchet, Alexis Michon, Krystyna Zakrzewska, Richard Lavery
CNRS, France

Institut de Biologie et Chimie des Protéines (IBCP UMR 5086);CNRS; Univ. Lyon1; IFR128 BioSciences Lyon-Gerland; 7, passage du Vercors, 69367 Lyon cedex 07, France Bioinformatics is today challenging the grid concept and its implementations. As an example, the sequence analysis of genes and proteins are both very data-intensive. Their main goal is to identify a single character among millions of sequences, as in the case of the EMBL and GenBank databases, which respectively contain 70 and 57 million sequences. The algorithms used for sequence analysis can require scoring profiles or matrices defined with studies such as that described here.

How proteins find their targets amongst millions (or more) of competing sites is still largely an unsolved problem. Understanding this process in detail is however central to understanding the mechanisms underlying gene expression. A better understanding of site-specific targeting is also a vital step towards rational re-engineering of proteins for therapeutic purposes. The problem becomes even harder when a complex of several proteins bind to DNA, an in the case of the nucleosome core particle. The nucleosome involves an eight protein complex (histones) binding to 147 bp of DNA. Simulating a nucleosome core bound to a single DNA sequence would require roughly 250,000 atoms (including solvating water molecules) and many months of computer time. However, to understand selective binding we need to compare many potential binding sequences. Given that any of the four nucleic acid bases can occupy each position within the bound DNA, there are roughly 10power86 potential sequences to test. We have been able to reduce this task by dividing the DNA into overlapping fragments containing four nucleotide pairs. Each pair can have 44=256 sequences. By minimizing each sequence in turn for each fragment, and then moving one step along the nucleosome bound DNA, we can reconstruct the binding energies of all possible sequences with approximately 36,000 optimizations. Each optimization uses the JUMNA program developed in our team and takes, on average, one hour. This implies the whole task would require roughly four years on a single processor.

A grid platform is very useful to distribute this large number of minimizations as we can run these independent and quite short jobs in parallel. We have used the production grid set up by the EGEE-II project, which brings together 35,000 CPUs and 5 PB of storage amongst 200 sites world-wide. We have submitted each of 35,840 energy minimizations as an individual job on the grid. This means that each job had gone through the submission processes, and thus paid the overhead inherent to the grid architecture and internal processes: from the submission through the user interface (UI), via the scheduling step on the resource broker (RB) to the execution on the computing element (CE), a cluster with several worker nodes (WN).

The whole computing task was launched through 12 RBs, which have scheduled all the jobs on 23 CEs. The total cumulated computing time has been of 1,275 days and 6 hours, with an average job duration of 51 minutes. The task was completed after 4 days and 16 hours, yielding a crunching factor of 271. This is considerably less than the maximum number of recruited CPUs, because we obtained up to 1039 jobs running at the same time. Using the EGEE grid to obtain a first indication of the binding specificity of the nucleosome turned out to be rather efficient. Although the crunching factor is considerably less than the maximum number of recruited CPUs, this is mostly due to the mean submission time that was quite long, almost three seconds per job. Our perspectives on the grid aspect for the a more refined study of the nucleosome, will be to decrease the overheads, in particular at the submission step, by grouping several jobs together, or by using another scheduler such as the new experimental gLite component or the GridWay metascheduler. However from the point of view of structural bioinformatics, the results have been very fruitful, and have demonstrated the sustainable status of the EGEE grid for large scale experiments in a true laboratory workflow. We are planning to continue our study with an improved model that will require 140,000 energy minimizations, corresponding to roughly 16 years of sequential CPU time. This work is supported by the CNRS, by the French Agence Nationale de la Recherche through the projects HIPCAL (ANR-06-CIS6-005) and HUGOREP (NT05-3_41825) and by the sixth European Framework Program through the project Enabling Grid for E-science II (EGEE, EU-FP6 INFSO-RI-031688).

Dynamic Runtime Environments for Grid Computing

Daniel Bayer (1), Tashfeen Bhimdi (2), Balázs Kónya (3), Frederik Orellana (4), Anders Wäänänen (4), Steffen Möller (1)
(1) University of Lübeck, Germany,
(2) George Mason University, United States,
(3) Lund University, Sweden
(4) Copenhagen University, Denmark

In a grid context, the execution of jobs on remote machines of collaborating institutes often requires the provision of more than mere CPU time. Besides those libraries and utilities that are distributed with the operating system in question, further software is required by the jobs and may hence need to be installed prior to the jobs execution. Grid computing aggravates this problem due to decreased personal contacts among the participants.

Currently, the various Virtual Organizations of the existing production grids typically agree on a set of Runtime Environments that participating sites install manually. Grid jobs requesting a particular runtime environment then seamlessly have the corresponding software available. The challenge is to further automate this process, thus easing the burden of site maintainers and allowing grid-wide software updates with immediate effect. This paper presents a schema for the description of runtime environments and the implementation of a service for their automated and dynamic installation.

The implementation introduces a catalog of runtime environments using the Resource Description Framework (RDF). Site maintainers specify constraints on the basis of these formal descriptions upon which installations are performed without further manual supervision. The concept aims at easing the collaboration between heterogeneous user groups to thus foster the adoption of grid computing technologies. The increased dynamics are also anticipated to help graying the distinction between the very diverse but singular web services and the ubiquitous while homogeneous grid computing.

Automatic Deployment of an Application Software on a Grid *)

Clovis D. Jiogo, Sébastien Noël, Pierre Manneback
Dept of Computer Science, Information Technologies Unit, Faculte Polytechnique de Mons, Belgium

Many application software are requiring substantial computing power, which can be in particular provided by Grid environments. The deployment of these software on a Grid must deal with many constraints like user and application requirements (QoS), software licenses, resource heterogeneity or Service Level Agreement. We propose to present in this contribution how to automatically deploy some specific application softwares on Grid environment.

We will propose a general deployment architecture. This is based on a Deployment Manager and a Grid Broker. Deployment Manager is responsible for the load and extraction of the application data, their transfer and execution on the end nodes. It is composed of deployment descriptors storing system parameters and several script files using these parameters. The Grid Broker gathers the application and user requirements and the static and dynamic resource properties. Following them and using an efficient strategy based on multicriteria cost estimation and resource load and capability, it proposes an allocation, a deployment plan and a task scheduling We have implemented these deployment and brokering modules on a Grid composed of multiple distant clusters. The chosen platform is made of components of Globus Toolkit namely MDS4 for resource discovery, GridFTP for data transfers and GRAM for resource allocation and management.

We have plugged it Ganglia as monitoring tool and Sun Grid Engine as local resource scheduler. This framework has been tested for the deployment of a professional 3D real-time rendering software (Reality Server c from Mental Images Gbmh). This software is supposed to be used concurrently by many users, e.g. architects, on a Grid environment. For this purpose, we have prototyped a Grid Broker which proposes suitable resources and then derive a deployment plan , according to the application (3D scene complexity, image resolution, number of available license, etc.), user requirements (frame rate, latency) and resource properties (cpu load, memory capacity, bandwidth). This plan is materialized by file descriptors which specify a batch of information (either static or dynamic) required by the deployment process: software parameters, resources localization, job and license management. The plan is then executed and monitored by the deployment module, leading to a Grid-enabled application.

Keywords: Grid Deployment , Grid Broker, Grid Resource Management, Globus, 3D rendering.

*) This work has been carried out within the framework of the European Integrated Project Business Experiments in Grid (BeinGrid), BE03 workpackage Visualization and virtual reality, programme FP6-IST-2005-2.5.4, contract no 034702. We acknowledge the contribution of our partners Art&Build, CETIC and Mental Images in BE03.

Extensions to the ETICS Build System Client Allowing Porting to Multiple Platforms at Local Sites

Eamonn Kenny, Brian Coghlan, John Walsh, Stephen Childs
David Trinity College Dublin

The ETICS build system is now a well developed and reliable build system with support for building the gLite middleware stack in EGEE II. When used in conjunction with Metronome and Condor, it performs a remote build of a prescribed project in the software stack. However, Grid-Ireland sites such as the one at Trinity College Dublin have build nodes requiring proxy support behind multiple firewalls that inhibit the use of remote building for porting purposes. Currently the only way to obtain a workable solution to porting is to employ local distributed building.

Although this will change in the future, the certification and corroboration of remote/local building at other sites in differing environments will always remain useful. The injection of patches into a centrally managed build system is not always immediate. The build system and version control repository should be kept under the strict control of the developer, but the software should remain open enough to give a best-effort approach to porting of the middleware to alternative platforms.

Grid-Ireland is working toward porting the gLite middleware to CentOS 4.5, SuSE 9.3, CentOS 5.0 and MacOS X for 32-bit and 64-bit architectures. Patching multiple platforms requires a persistent structure across new releases of the middleware. Usually patches break when the software is changed significantly. Therefore, a more dynamic patching system is proposed. In addition, the porter must be able to set up site dependent configurations or create subset builds of a complete middleware stack in a versatile way. For this reason an XML schema has been developed to describe a local build configuration. The build configuration is called by a Python module wrapped around the ETICS build client. The goal is to allow the porter to build any number of modules from a software stack, in any order, with a number of user-defined dependencies, on any porting platform with a discrete set of software patches, posting the results-set to a prescribed repository machine in the local network. Nightly continuous build systems tend to build a specific branch of the middleware. The advantage of a XML description of a subset build is that it produces a dynamic approach to building software that essentially results in a comprehensive rollback procedure, something that most build systems don't implement well.

This paper describes the XML structure of the build configurations, the structure of the virtual machine testbed at Trinity College Dublin, the patching system and the reporting mechanism for both the local site results and applied software patches.

Run-time Fault Diagnosis for the Grid

Jan Ploski (1), Wilhelm Hasselbring (2)
(1) OFFIS e. V., Oldenburg, Germany
(2) Software Engineering Group, University of Oldenburg, Germany

Run-time fault diagnosis means determining the cause of a failure after its occurrence in an operational system. This aspect of systems management consumes significant resources and Grid-based systems are no exception. For this reason, the stakeholders (developers, administrators and end-users) should take care to optimize their diagnostic processes. We note that the main reason why fault diagnosis is difficult in practice is that due to the considerable size and relative novelty of Grid-based systems the people who perform it must rely on imperfect system models. Based on the experience that faults reoccur across time and/or across system installations, it is reasonable to say that preserving fault-related information in order to support future diagnoses has potential benefits. We assume that the current forms of describing faults and the associated repairs can be improved upon by adopting a more systematic approach – the event-based run-time fault diagnosis.

Our approach can be summarized as follows:

1. Grid components whose state may contribute to faults are identified and their observability is increased through an appropriate instrumentation. In particular, the occurrence of events that can be a priori classified as "exceptions" at the level of the operating system or of a programming language is made observable through uniform logging.

2. A first-time manual diagnosis consists of increasing log levels to capture the event traces and then analyzing the traces to identify the undesired events as well as the repair actions necessary to prevent their occurrence (fault removal). If necessary, further events that should have been captured, are also identified and fed back into the instrumentation phase. A diagnostic case which relates the observed events with the required repair actions is stored in a database, shared among system installations.
3. A repeated diagnosis takes advantage of the diagnostic database to capture event traces that might contain fault-related events, matching them against the prior diagnostic cases, and recommending repair actions.

The empirical evaluation of our approach happens through its implementation in the research project WISENT, which utilizes resources of the German Grid (D-Grid). The evaluation's goal is to validate

1. whether the terminology imposed by the approach is sufficient to describe the system behaviors that must be reasoned about during diagnosis (based on representative cases collected through the project),
2. whether the approach can speed up repeated diagnoses by supporting the accumulation and evaluation of the relevant fault-related data,
3. to what extent the approach can support automated fault diagnosis.

Transparent Cross-Border Migration of Parallel Multi Node Applications

Dominic Battré (1), Matthias Hovestadt (1), Odej Kao (1), Axel Keller (2), Kerstin Voss (2)
1) Technical University of Berlin, Germany
2) Paderborn Center for Parallel Computing, University of Paderborn, Germany

The Grid is on the verge of attracting commercial customers. In this context a Service Level Agreement (SLA) is a business contract like instrument for defining all obligations and expectations within the business relationship between the provider and the service customer. The EC-funded project HPC4U (Highly Predictable Cluster for Internet Grids) developed a Grid fabric that provides not only SLA-awareness but also a software-only based system for fault tolerance that allows job completion and SLA-compliance even in case of resource outages. The software stack consists of a resource management system(RMS) and dedicated subsystems for fault tolerance and checkpointing of processes, network, and storage. Checkpoints are generated transparentlyto the user in the background (i.e. the applications do not need to be modified in any way) and allow restarting and continuing the execution of an application after the checkpoint is migrated to healthy resources.

Of course the availability of spare resources can be a limiting factor for jobs with deadlines that are affected by a resource outage. Only if such spare resources are available, the job can be restarted from the checkpoint and completed in time. To increase the number of potential spare resources, the HPC4U system is not only able to restart jobs on the same cluster resource, it may migrate the job to other clusters within the same administrative domain or even over the Grid to arbitrary resources, too.

In both cases the migration process consists of these major steps:

1. search of migration target,
2. transfer of checkpoint dataset,
3. remote execution, and
4. transfer of result dataset back to the user.

Transferring the checkpoint dataset impacts the scheduling on the target system and implies negotiation with the remote RMS by using provisional reservations. Of course questions on security, authorization, authentication, and accounting come into play when moving to the inter-domain cross-border migration case. Fortunately, Grid systems like the Globus Toolkit offer mechanisms for these questions. However, the central RMS has to provide interfaces for driving a negotiation process over the Grid. In joined work with the EC-funded project AssessGrid, both an implementation of the WS-Agreement protocol and an interface to the RMS have been realized.

In this paper, we describe the migration mechanisms within the HPC4U system both for the intra-domain case as well as for the migration over the Grid. We will in particular focus on the architecture and the differences between intra-domain and Grid migration. Moreover, we will present initial results, open problems, and a brief outlook on already started work.

Design of a Sequential Tentative Hold Protocol for Efficient Grid Coordination

Kunyi Luo, Yongjian Wang
Sino-German Joint Software Institute of BeiHang University, Beijing, China

Background

Grids provide resource sharing and resource virtualization mechanism to end-users, allowing for resources to be accessed as a utility. As grid users strive to automate the interaction among grid applications efficiently, the relationships among grid applications become more and more complex which causes increasing resource conflicts. So coordinating the access of computing or data resources is essential for an efficient grid system.

Drug Discovery Grid (DDGrid) which has been undergoing for more than 3 years is a key project supported by China National Grid (CNGrid). DDGrid utilizes the resources including the clusters and personal computers that scatter over the Internet for new drug screening service.

All the computing resources in DDGrid are organized in mater/worker pattern. Master node is in charge of job split and dispatch like RB/WMS modules in gLite, and workers will actually process the jobs. As many worker instances can be processing various kinds of jobs simultaneously, and resource locking order varies from jobs, it's difficult to predict underlying resource usage. So how to coordinate the locking of underlying resources, utilize resource efficiently and avoid the common problems such as dead lock, priority inversion has become a big challenge in DDGrid.

Proposal and Improvement

Tentative Hold Protocol (THP) is an open framework that facilitates the coordination of resource access. It allows multiple DDGrid workers to reserve the same resource item by placing tentative holds on it and verify availability before actually consuming the resource. When one worker locks this item, the others will receive notifications that their holds are no longer available.

In THP, when a worker receives all the confirm messages of hold requests, it locks all the resources needed, and the resources won't be released until the whole process of this worker is completed. This policy caused the resources being locked for too long, thus reduces the resource utilizations. Another shortage is that it may cause priority inversions. Furthermore, the resources are reserved randomly, in spite of the logic order in which they will be accessed.

We proposed Sequential Tentative Hold Protocol (S-THP) to overcome these shortcomings. The coordination mechanism of S-THP is different from THP not only on the aspect of resource reservation, but also on the aspect of hold expiration during the execution of a worker.

S-THP acts as follows for resource reservation: from the side of resource owner, when the number of reservation holds reaches the overhold size of a resource, further requests will be abandoned, otherwise the requesting worker will be added to a queue that records the reservation order of the workers. From the side of resource requester, which is a worker in DDGrid, the difference is that in THP, the worker just requests holds randomly, while in S-THP; the worker will maintain the resource access sequence until the process is completed. The other difference is that S-THP doesn't immediately lock all the resources when worker starts to run, instead, it locks the corresponding resources during its execution when the item is to be consumed. This solution, May cause a resource conflict if the over- hold size is larger than the total number of the resource.

Conflicts on a resource result in expirations of its tentative holds. In THP, when conflicts happen, the latest worker will be notified that the hold has been expired, and it has to execute rollback or compensation immediately, but in S-THP, we suggest the worker to wait for a certain amount of time. And if there aren't any items of this resource released during this period, the worker should look up the resource reservation sequence to find all the resources behind the conflicted resource and cancel the holds for them, then execute rollback or compensation.

S-THP brings the following benefits: First, improvement of resource utilization: the resource is only locked when it's really in use and thus can be better used to satisfy more transactions. Second, more real-time resource status: S-THP provides workers with completely up-to-date data to base their decisions upon. Third, improvement of flexibility: S-THP can be adapted to various grid applications by adjusting the over-hold size and time duration when conflicts happen appropriately.

Experiment

In order to evaluate the performance of the S-THP, we consider a scenario in DDGrid. We simulate DDGrid workers that require certain kinds of resources in different order. The results show that S-THP performs significantly better both in resource utilization and efficiency of workers. And by appropriately adjusting some parameters, S-THP obtains distinctly better performance in certain cases.

A novel Portal Architecture for Real-Time Online Interactive Applications on the Grid

Christoph Anthes, Roland Landertshamer, Roland Hopferwieser and Jens Volkert
Joh. Kepler University Linz, Austria

The edutain@grid project [1] offers an architecture for a novel class of applications on the Grid [2]. Edutain@grid provides a three tier middleware in order to support Real-Time Online Interactive Applications (ROIAs). Two categories of applications have been identified as online games and education applications. They both have the requirement to support a high amount of concurrent users and act under very tight real-time constraints. A high consistency has to be kept and security is a crucial issue. In order to support these needs a three tier middleware is being developed. This middleware consists of a business layer, offering a variety of business models for the difference roles of users, accounts can be created, the needs of hosters and distributer can be matched and other business related operations can be performed in this layer. This top layer communicates via web services with a management layer.

The management layer is set up on a hosters site, where different data centres are able to host ROIAs. Sessions over multiple data centres are possible as well. The layer below, the real-time framework (RTF) [3], allows the distribution of a game world or the data of an e-learning application over several servers in order to provide scalable applications. A portal in the edutain@grid project acts as a user interface to the edutain@grid middleware.

To provide access to the different layers and to support a variety of actors different types of portals are needed. The edutain@grid architecture will provide three different types of portals. The first one is the client portal which allows the users of applications to connect to running ROIA sessions. It offers functions for finding running sessions, for connecting to a ROIA session and for communication with other users. The client portal is implemented as a C++ API in order to be integrated directly into applications running on edutain@grid. Additionally this portal has been designed to be implemented as an external tool the client manager, which is responsible for logging and starting of legacy applications. It uses Axis2 [4] to interconnect with the business layer of edutain@grid. The second portal is the business portal. This portal is located in the business layer and it acts as the central access point for all participating actors in the edutain@grid system. It allows application users to register to the system and to manage their account data like the user's nickname or the payment method for applications which are liable to charges. This portal is also used by coordinators to deploy and undeploy ROIA sessions on one or more hoster sites and by hosters to provide server machines for the edutain@grid system. The last portal is the management portal. This portal is located on the datacentre manager machine of each datacentre. It is used by the hoster to visualize the state of his machines connected to the edutain@grid system and to add machines to or remove machines from the system. The business and the hoster portals are implemented as classic web portals using Gridsphere [5] as an API.

References

1. Thomas Fahringer, Christoph Anthes, Alexis Arragon, Arton Lipaj, Jens Müller-Iden, Christopher Rawlings, Radu Prodan and Mike Surridge. The edutain@grid Project. In International Workshop on Grid Economics and Business Models (Gecon), Rennes, France, August 2007
2. Ian Foster, Carl Kesselman and Steven Tuecke, The Anatomy of the Grid Enabling Scalable Virtual Organizations International Journal of High Performance Computing Applications, pp 200-222, 15(3), 2001
3. Jens Mueller and Sergeij Gorlatch Rokkatan: scaling an rts game design to the massively multiplayer realm. Computers in Entertainment 4(3) (2006) 11
4. Apache Axis2, http://ws.apache.org/axis2/
5. Gridsphere, http://www.gridsphere.org/

New Approach to Design UI for Grid Applications

Daniel Pasztuhov, Imre Szeberenyi
BME, Budapest, Hungary

Grid is one of the most promising research area of our time. As often, products created by a strongly researched area lacks the convenience and ease-of-use user interface (UI).

In most cases, the real users of the Grid are not very familiar with these interfaces, they require an easy-to-use interface. To satisfy the demand of this type is always the task of application developer. In most cases, if the developer has to implement a user interface, he needs to create similar things, "reinvent the wheel", again and again. It would be more cost-effective, if he had a comfortable framework to provide the new user interface.

Facing the requirements of the user and the requirements of the developer, to create a convenient user interface to submit grid jobs as rapid as we can, we developed the Conflet Framework (CONFigurable portLET). It is created to submit grid jobs, but because of its flexible architecture, legacy, command-line programs , even if they are on a remote host, can also be started by Conflet. Our system is based on GridSphere Portal Framework, but is flexible enough to use not only with portal systems. Conflet uses some configuration files to provide the look and feel, and the behaviour of a user interface which is to start a new grid job. The developed system is flexible enough to cooperate with different grid and cluster middlewares, and aims to hide the differences among middleware systems.

To avoid creating the configuration files by hand, we developed an Integrated Development Environment (IDE) based on Eclipse Framework, which helps the developer hold the creation of UI (called configuration) in hand. The Conflet IDE contains projects, text editors with syntax highlighting, specialized editors for different file types, automatic build function, and several views to help the developer.

Adaptive Grid Visualization

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

This paper presents a state of the art overview and evaluation of extant visualization approaches and techniques for distributed environments. Following a brief introduction of our background research work in the area of grid visualization, we describe our work in progress on the development of an adapted grid visualization immersive world and we highlight its added value over existing grid visualization solutions. The paper concludes with a future research agenda towards the development of a more universally accessible grid infrastructure.

Our overview of state of the art research in the area of visualization of distributed environments adopts a classification based on the type of the distributed environment, the aim of the visualization, and the medium of the visualization. A visualized distributed environment could be of three types: a simple physical network, a wireless network, or an infrastructure (such as the grid). The visualization of a distributed environment could be for learning purposes, for monitoring the distributed environment, for management, or for the user-friendly development of applications within this distributed environment. The medium of visualization could range from a simple 2D applet or desktop application to 3D/VR/AR and rich media immersive worlds. Our overview covers various tools and approaches including Etherape, VISUAL, IDtk, Migrating Desktop, AccessGrid, Grid Mapper among others.

Our overview reveals that the visualization of distributed environments is a highly challenging task. The main challenges to be addressed are complexity, scalability, and relevance to the user. The latter motivates a bigger research agenda of profiling the user (his expectations and taste) and adapting the visualization to the user profile.

Within the framework of the WebComG project (funded by Science Foundation Ireland) we ave developed an advanced grid visualization engine for two purposes: visualization on the grid (this involves the development of visual applications using our engine), and visualization of the grid (this involves the visualization of the grid architecture for learning, monitoring, management, and easy application development). In this paper we focus on the second objective, visualization of the grid, and we describe current work with the aim of addressing the challenges and limitations of existing visualization approaches.

Our visualization of the grid consists of developing an environment abstracting elements of the grid infrastructure using 3D animated metaphors coping with the problems of complexity, scalability and adaptability to the user profile. Using information gathered from LCG/gLite BDIIs our application creates a navigable world that contains representations of Compute Elements, Storage Elements, Worker Nodes, etc. Our application has been developed using OGRE.

Our aim is to take into account the user profile in adapting our navigable world of grid infrastructure components. At an initial stage we consider a very simple user profile consisting of the user taste in navigation, their expected level of complexity of the visualization, and their technical background. Users access our adapted navigable world after creating their own profiles. In response to the user profile, we offer varying patterns of interaction with the navigable world, different geometric metaphors and animations of grid elements suitable to the user taste, and a level of complexity adapted to the technical background of the user.

Our adaptive grid visualization offers an added value over existing approaches for Grid visualization. First, it addresses the problem of complexity by adopting the WebCom-G condensed graph model of condensation and evaporation of graph nodes. In case of our grid visualization, the level of complexity is decreased or hidden

through condensation of grid visual components, on the other hand complexity is increased through evaporation the grid visual components. This assumes that our grid visualization is mapped to a condensed graph (in the sense described within the framework of WebCom-G research) where grid visual components are nodes in the condensed graph. Second, it can potentially extend features of GridMapper beyond the geographical visualization of job execution and grid activity to respond to a user's interest in exploring for instance the notion of "bandwidth geography of the infrastructure" i.e. if some sites have a faster connection between each other then they are closer together. Thirdly, our work could be in the future a nice contribution to the virtual venue for collaboration in AccessGrid. Users wishing to collaborate together using AccessGrid may adapt our immersive grid visualization environment to suit the context of their collaboration. Fourthly, our research would add an adaptive dimension to the Poznan's Migrating Desktop. Adaptability would also take into account the target application domain. On our research agenda is to enrich our grid visualization environment with game techniques such as "multiplayer" sessions. This would raise the transparency of the grid and allow better control and monitoring of the grid activity and security.

Our research is still in its early stages and we outline a future research agenda to develop it further in different directions. Firstly to increase the number of infrastructure components that are represented graphically. Secondly to group all of these representations so that they can be adaptively displayed according to the user profile selected. Thirdly, to introduce new paradigms of interaction and access through various media delivery channels (e.g. rendered visualization on mobile devices), novel animations techniques, and varying level of complexity for the navigable world (e.g. a potential merge with Google Earth). Fourthly, to consider a richer user profile. And finally to reach a level where our work on adaptive grid visualization becomes an integrated part of the grid infrastructure and contributes to its universal accessibility.

Infogrid: a Relational Approach to Grid Middleware

Oliver Lyttleton, Brian Coghlan, Eamonn Kenny, Geoff Quigley, John Walsh
Trinity College Dublin, Ireland

Using the Grid for science or commerce can require the manipulation of large amounts of data. This data is in a file-based format for many Grid applications, and cannot be manipulated using standard database technologies such as SQL. In order to do so, the data has to be either parsed and inserted into an SQL database, or a custom application must be written that can be used to perform operations on the data. However this is technically beyond many users. Ideally, non-expert users should be able to perform operations on the data with a user-friendly GUI. Grid technology at present requires users to work with data at a file level, using an Operating System that many people are unfamiliar with (usually a flavour of Unix). This is not ideal, and presents a learning curve that will deter many from using Grid technology. It has often been stated by potential users of Grid technology that the technology must be easier to use than it is at present. There is a great deal of work to be done in making Grid technology more intuitive and easy to use.

We are developing a prototype application, Infogrid, which will use R-GMA as an interface to the grid, and will replace several components of the LCG/EGEE middleware. Users will submit data to the grid for processing by inserting data into R-GMA tables. Instead of specifying input files, output files, and executables using a job description language, rows are inserted into a table which represents a function. Some table columns represent inputs, and some columns represent outputs. A GUI can be used to perform these database operations. The benefits of this approach are an easier to use, more intuitive user interface. The ability to provide a federated view of this data using R-GMA and to enable easy, quick access to it is beneficial to users which need to access remote data. Data can be selected and used as if it were on the user's own machine, regardless of its location on the network. Many scientific and commercial applications are complex and the data they use is not stored centrally, but instead stored at various locations in the organisation, and sometimes even outside the organisation.

LCG/gLite middleware is implemented using many different applications, each of which perform different tasks, e.g. allowing users to submit a job, decide where a job should be executed, transfer a job to wherever it is to be executed, and remotely initiate the execution of that job. The Grid is non-trivial to install, configure, and maintain because of this proliferation of seperate applications that are used to implement it. We can implement much of the functionality of the applications using R-GMA. Infogrid uses the R-GMA technology to replace various components of the LCG Grid Middleware. Because there are substantially fewer applications involved, it is easier to install, configure and maintain.

Grid Simulator with Production Scheduling Algorithms

Miroslav Ruda (1,2) and Hana Rudova (1*)*
(1)(Masaryk University, Czech Republic
(2) Cesnet, Czech Republic

Grid simulators are often used for evaluation of new scheduling methods. These tools can help to comparison of different scheduling algorithms especially during the development phase. On the other hand, several drawbacks can be identified. First, new developed algorithms are sometimes compared to simplified algorithms not representing state of the art in production resource management systems. Second, workloads applied for evaluation of the methods are either synthetic or they are taken from Standard Workload Archive containing old workloads from several computing centers which are often obsolete in comparison with the current situation on clusters/grids.

As a complementary approach to our activity in grid simulators (Alea grid simulator) we have proposed and implemented a simulator running un-modified PBSPro installation. This was achieved with the help of virtual machines and our tool called Magrathea. About 150 nodes are represented as virtual machines which can run on a single 16 CPU machine. It allows us to process simulation of the workload representing substantial part of Czech Grid called MetaCenter. With this, simulation of the workload from the whole year can be computed within one day. Moreover, the accounting data from the real MetaCenter PBSPro were extracted from the last three years (2007-2005) and used as our simulation workloads.

We will describe the main components of the new simulator including the novel inclusion of virtual machines with Magrathea in our grid simulator. The work will concentrate on the representation of parallel jobs and on the inclusion of the job preemption. Let us emphasize that preemption of particular jobs can be achieved with the help of virtual machines and Magrathea. Magrathea allows an easy integration of PBSPro with virtual machines. Several virtual machines with diffrent execution environment can run on a single computer and Magrathea manages resources given to each virtual machines. Magrathea can preempt/suspend running jobs with their virtual machines which allows PBSPro to run another job with a different virtual machine on the same physical resource/computer.

We will present the first results of our simulations by comparing several setups of un-modified PBSPro installation including approaches with enabled preemption. Our goal lies in comparison of the approaches allowing preemption and potentials of its use in production systems. Results of the experimental comparison will be used for evaluation of PBSPro setup in MetaCenter and selection of the best performance algorithm for this production environment. In the future, we would like to use grid simulator for comparison of PBSPro algorithms with other scheduling algorithms we work on.

Prediction of the Jobs Execution on the Community Grid

Jakub Jurkiewicz (1,3), Krzysztof Nowiński (1), Piotr Bała (1,2)
(1) Interdisciplinary Center for Mathematical and Computational Modelling, Warsaw University, Poland
(2) Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland
(3) Faculty of Mathematics, Informatics and Mechanics, Warsaw University, Poland

Nowadays grid computing is well established and is used for a large scale simulations. The resources available on the grid vary from supercomputers to clusters and single PC's. Current technology allows user to utilize CPU cycles of different systems and makes this process easy and straightforward. The simplest programming model for such grid is based on the trivial parallelism. The job is composed of large number of independent tasks executed on separate processing units. The server which assigns work to the workers has to take care of their different computational speed or capacity to avoid unbalancing. One of parameters that has to been taken in to account is failure rate of computers. It is especially important in community grids – when every processing node is administrated by another person, and it could be down quite often.

In ICM we created simulator of grid that simulates computers, that has failure rates taken from defined distribution, but implements currently only this very simple job model. We have run our simulation on different number of tasks in job. Number of task where varying from 5 % to 120 % of amount of processing units. We have tested different scheduling strategies. We have chosen two strategies for reference: random scheduling strategy, and perfect scheduling – when job is assigned to computer that has left lifetime long enough for making it.

The best strategy for this simple case was (as expected) backup scheduler – submitting job to more then one processing node. We have compared backup strategy for running two and three copies of job. Relative quality of this two schedulers, at constant number of processing nodes, were depending on number of tasks in job. While we have submitting job just after starting system, with all computers down, all others schedulers were as bad as random scheduler.

The presented simulations confirms that job scheduling on the grid is not trivial. In the presented model we have investigated node failure and its influence on the total execution time of the job. We have been able to verify different scheduling strategies and predict their efficiency.

A Distributed Architecture for Multi-dimensional Indexing and Data Retrieval in Grid Environments

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

In this paper, we describe a service-oriented architecture of a generic middleware platform which provides the required services for efficient content search and retrieval in a distributed environment. Our design intends to introduce algorithms from the Peer-to-Peer computing in a grid infrastructure in order to establish advanced search and data management facilities among resources belonging to different Virtual Organizations. Peer-to-Peer systems have already been used widely for information sharing and discovery and they can offer effective solutions ensuring scalability, fault-tolerance and data availability despite nodes arrivals and departures. The proposed architecture can be applied in large-scale geographically distributed environments of heterogeneous systems capable of managing immense amounts of data based on the primary principles established by the Data Grid architecture [4]. The users of the system can access, store, transfer and search available data and share resources belonging to the Virtual Organizations they participate.

The search mechanism enables the discovery of stored data based on metadata descriptions, supporting both point and range queries. A multidimensional indexing technique based on Space Filling Curves [5, 6, 8] is adopted to meet the above requirements. Multiple attributes are indexed simultaneously and are considered to form a multidimensional space where each point represents a combination of metadata values corresponding to an existing data item. Each metadata file is stored in the node of a DHT-based overlay closest to its key. It has been taken into account that locality must be ensured due to the fact that the cost of a query execution is often proportional to the number of nodes that need to process the query, especially concerning range queries. However, in order to avoid uneven load distribution incurred mainly by heterogeneity of node capacities and available bandwidth we introduce a load balancing technique based on virtual servers.

The actual content is stored in available repositories and the deployed replication scheme reduces access latency, improves data locality, robustness and scalability. The existence of multiple replicas poses the requirement of an efficient mechanism to locate them. For this reason, a global distributed catalogue is used to resolve global names into locations for physical resources where data is stored. The Kademlia protocol [7] is enhanced in order to support the mutable operations in a distributed catalogue by virtue of its simpler routing table structures and the use of a consistent algorithm throughout the lookup procedure. A distributed Replica Location Service [2] is introduced based on the revised protocol for distributed hash tables that allows data to be change in a distributed and scalable fashion.

We also introduce GridTorrent [9], an innovative approach to data transfer directly from established GridFTP servers or other GridTorrent peers that are simultaneously requesting the same piece of data. GridTorrent is a modified implementation of BitTorrent [3] designed to interface and exploit well-defined and standardized Data Grid components and protocols. The peer-to-peer approach integrated in GridTorrent enables the aggregate data transfer throughput to escalate, even when numerous requests rely on a single data source, and achieve better utilization of the available Grid resources.

Our main innovation is an integrated architecture for search, storage and retrieval of annotated content in large scale distributed systems. The exploitation of P2P techniques contributes to system scalability and fault-tolerance, avoiding the use of centralized points subjected to crashes, DoS attacks and possibly unavailability consequent to regional network outages. The design of the system leads to an extensible architecture favoring the integration with other systems and the development of Grid applications.

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

References

1. M. Cai, A. Chervenak, and M. Frank. A peer-to-peer replica location service based on a distributed hash table. Proceedings of the SC2004 Conference (SC2004), 1:1, 2004.
2. Antony Chazapis, Antonis Zissimos, and Nectarios Koziris. A peer-to-peer replica management service for high-throughput grids. In 2005 International Conference on Parallel Processing (ICPP05), Oslo,Norway, 2005.
3. Bram Cohen. Incentives build robustness in bittorrent. In Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.
4. A.C.I. Foster, C.K.C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. Journal of Network and Computer Applications.

5. P. Ganesan, B. Yang, and H. Garcia-Molina. One torus to rule them all: multi-dimensional queries in p2p systems. Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004, pages 19-24, 2004.
6. J.K. Lawder and P.J.H. King. Querying multi-dimensional data indexed using the hilbert space-filling curve. ACM SIGMOD Record, 30:19-24, 2001.
7. P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS?02), 258:263, 2002.
8. C. Schmidt and M. Parashar. Enabling flexible queries with guarantees in p2p systems. Internet Computing, IEEE, 8:19-26, 2004.
9. Antonis Zissimos, Katerina Doka, Antony Chazapis, and Nectarios Koziris. Gridtorrent: Optimizing data transfers in the grid with collaborative sharing. In 11th Panhellenic Conference on Informatics (PCI2007), Patras,Greece, May 2007.

Deployment of Interoperable Data Access Models in D-Grid

Gian Luca Volpato, Christian Grimm, Harald Schwier
RRZN – Regional Computing Centre for Lower Saxony, Leibniz Universität Hannover, Germany

The implementation of a firewall-friendly architecture is of paramount importance for any resource provider involved in the provisioning of a Grid infrastructure. Additional challenges are likely to appear when more than one middleware stack is installed at the same site. Since each implementation comes with its own specific features and shortcomings, the design of an appropriate deployment strategy cannot be easily decoupled from the choice of the middleware.

In this paper we present the approach endorsed by the German project D-Grid, which enables the coexistence of multiple Grid middlewares on the very same set of computing resources, in a way completely transparent for the users. In this project several academic and industrial partners scattered all over Germany have committed their resources being simultaneously and seamlessly accessible through Globus Toolkit, gLite and UNICORE. Such an approach achieves two significant goals with a minimum effort: firstly it maximizes the freedom of choice for the users, who can prepare, submit and monitor jobs with their preferred middleware. Secondly it capitalizes on the total usage of the resources, by making them available to a wider customer base.

Regardless of the multiple choices for job submission offered at the resource providers, a consistent solution for remote data access from the computing nodes should be provided. Grid jobs should be able to upload and download their input/output files with a procedure that ideally remains fully independent from the middleware. Several data access models are here presented and analyzed, taking into consideration different evaluation parameters, among others data transfer performance and impact on the configuration of local firewalls. It has however been detected that, because of technical limitations, not all middlewares are able to provide data access according to all the considered models. It was also possible to observe that the usage of special options of some data transfer tools can easily create conflicts with the functionality of the site firewalls. An alternative solution, that does not interfere with the firewalls and only slightly affects the transfer performance, is obtained by means of a different configuration of the TCP congestion control algorithm and it is documented with the results of an early deployment phase in the D-Grid core infrastructure.

Synchronizing Lustre File Systems

Dénes Nemeth, Janos Török, Imre Szeberényi
Budapest University, Hungary

Nowadays, asynchronous synchronization is considered as a solved problem for the commonly used file systems. There are many tools that reside above the file system layer watch for file modifications and copy the data in an efficient way. However, on a file system, where the amount of data is vast or the number of files/directories is large synchronization is not an easy task. Due to efficiency reasons it is impossible to walk through the whole directory structure to trace down which files were modified. Mainly, because it takes a lot of time, and it is not viable to issue global locks across multiple distributed Lustre systems in an order to synchronize a snapshot of a whole file system. To answer these problems an 'inotify' like distributed file system notification method would be the solution to trace the changes. In this scenario it is a key factor to sequence asynchronous events, which is demanding in a distributed environment.

To answer these problems we have developed a method for notifying file system changes on the distributed Lustre file system without observable system performance change. To achieve this we modified the kernel modules of the Lustre metadata server (MDS) to log all file catalog modifications on a separate kernel module, which is able to queue these events in kernel-space, while the event provider component persistently stores and

relays them to all event processors, which are part of the synchronization pool. In this pool every MDS server pair has a special Lustre client, which runs the event processor collecting events from all other event providers located on different MDS servers. As soon as all required events are available at the event processor it also caries out the actual synchronization tasks. To sequence and determine if the sequence of events is full (no missing events) we have created two scalable methods for distributing timestamps across different components of the system.

The first method uses external time synchronization (for example ntp) and considers time to be synchronized with minimum one second accuracy. It creates timestamps according to the standard UNIX time. If more events are created within one second an internal counter is used to distinguish these events. The problem with this is that minimal conflict-free interval has to be two seconds long. This means that all modifications on all MDS servers have to operate on separate entities within this interval to guarantee to global order.

The second method uses a central timestamp provider, which issues timestamps through the network to the timestamp requestor user-space application located on the MDS servers. This requires more communication, but can narrow down the conflicting events interval and allow a continuous counter for sequencing events. This is particularly useful since the event processor can know at which point has all events arrived from all event providers. The only problematic part is that the issuance of the timestamps requires dynamic measurement of the network and the queuing delay between the time provider and requestor in an order to ensure accurate time stamps. We have implemented the two different scenarios and measured the efficiency of the synchronization in the distributed Lustre file system. We will present the design of the system in Lustre, our measurements and show how it can be applied in almost any distributed file system, a finally demonstrate an on-line synchronization between two Scalable File Share systems.

Efficiency of Small Size Tasks Calculation in Grid Clusters Using Parallel Processing

Olgerts Belmanis, Janis Kulins
Riga Technical University, Latvia

Riga Technical University is one of Baltic Grid project partners. The article and presentation shows results of investigation obtained using RTU cluster connected to BG VO. Initially RTU cluster started with five servers AMD Opteron 146. Later was installed eight dual core AMD Opteron 2210 M2. Therefore now there are 21 working node with 1,8 TB common amount of memory. RTU cluster successfully completed many calculation tasks including LHCB virtual organization orders.

One of important tasks of cluster implementation is performance evaluation and optimal number of working nodes selected for different tasks. Important facility of cluster is possibility to use parallel algorithm for calculations instead serial. The speedup of using parallel processors on a problem, versus using only one serial processor was defined by G. Amdahl`s Law. Practically not full task is possible to parallelize. In every case there is some serial overhead. Percentage of overhead depends on amount of each task.

It is known that file transmission time between single node servers and multicore servers is different. With RTU cluster is possible to perform experiments to check this facility. For this reason three trials was performed. On first trial we choose 4 pairs of nodes from different multicore servers and exchange between them 1000000 bytes file. Overall average time during 100 iterations was 60.89 Mb/sec.

On second trial was selected three pairs of nodes. Overall average time during 100 iterations was 62.35 MB/sec. On the third trial was selected two pairs of nodes, where three cores belong to the same server. Overall average time during 100 iterations was 149.96 MB/sec.

Resume. File transmission time between cores of the same server is substantially less than between cores of separate servers. Transmission time reduced if number of nodes increased.

The next performed task is file processing with MPI on different number of CPU. Resume. For small size of tasks is not reasonable use in parallel more than 4-6 CPU.

The last experiment was to find parallel processing speed-up on different number of CPU. During the experiment each CPU perform multiplication of large size matrixes. Single CPU perform this task in 420 seconds. 6 CPU in parallel – in 126 sec.

Resume. Further increase of CPU number do not reduce processing time. It is necessary to underline that obtained figures are specific for used cluster.

Presentation will show architecture of cluster and connections between selected nodes. All results of experiments will be included in tables and graphs.

Performance Improvements to BDII - Grid Information Service in EGEE

Astalos Jan (1), Flis Lukasz (2), Radecki Marcin (2), Ziajka Wojciech (2)
(1) Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
(2) ACC CYFRONET AGH, Krakow, Poland

Global grid information system contains information about all grid resources and services that are present in the infrastructure. This kind of service is indispensable for performing any operation requiring data about services to use, like e.g. job brokering or data management. In EGEE grid infrastructure data about resources contains tens of megabytes and may be required at any time by any of 30000 users' jobs running concurrently. For that reason reliable information service is crucial for proper operation of the grid infrastructure and, in consequence, for providing scientists with robust environment for performing their work.

The service implementing functionality of information system in EGEE project calls a BDII [1] (Berkeley Database Information Index). It is based on LDAP protocol which make use of Berkeley DB transactional backend. Initially the only one instance of BDII served entire infrastructure but since the infrastructure growed to more than 200 grid sites it was clear that one instance will not handle the load. Due to performance reasons the top level BDII was split into several regional instances, but even though the regional instances experienced performance problems occasionally so further improvements were necessary.

This paper presents work at CE ROC to improve performance of regional instance of information system. Currently CE ROC BDII service runs three, geographically distributed instances registered in a DNS pool. Such a setup allowed to take advantage of load balancing, failover and transparent maintenance periods. For improved performance each instance use caching and database indices which speed up response time for most common queries. We also tested moving whole service directories to RAM disk to speed up response time. We measured impact of such changes by testing a service instance under high load conditions.

During the work on improvements we analyzed functionality and structure of the BDII. It appeared that currently, the service is fetching all the data from grid sites and rebuilding its database every 3-5 minutes. This approach is causing load peaks on BDII host and network. In this paper we describe how the design can be changed to significantly decrease both network and IO load on BDII host by use of persistent LDAP server and modification timestamps.

References

1. BDII service wiki page https://twiki.cern.ch/twiki//bin/view/EGEE/BDII

Bazaar of Resources Provides and Virtual Organizations

Tomasz Szepieniec and Anna Pagacz
ACK Cyfronet AGH, Nawojki 11, 30-950 Krakow, Poland

One of distinctive characteristic of grid environment is that resources, used by users grouped in virtual organizations (VO), are maintained by many providers, acting according independent policies. Therefore, management activities such as planning resource usage or negotiating contracts for computations, require more collaboration between providers and users. So, for operative infrastructure collaboration between parties should be good organized according to formalized procedures.

Unfortunately, for the last few years research related to resources management was focused on technical level mainly and was limited to resource brokering services.

Lesson learned from a real, multi-institutional infrastructure, like EGEE grid [1], shows that underestimation of organizational effort on that field or too complicated procedures can lead to serious problems with delivering required resources to the clients in reasonable time.

In this paper we described a proposal for a procedure and a collaboration tool which was designed to enable quick and positive response on users' requests for resources and make a cooperation between resources providers and virtual organizations more convenient.

The procedure is based on an assumption that resource centers are eligible to realize their own policy on resource allocation and also that they are motivated to offer their resources to users. In the process of negotiations a call for resources is published by a VO manager and resource provider can address it with an offer.

Next, a contract between the VO and the resource provider is negotiated directly. This gives VO manager an opportunity of choosing the sites which offer better conditions. The contract negotiation are provided using a collaboration tool developed specifically for this purpose - the Bazaar Portal. The portal provides a features to present the status of resources in terms of contracts, provide policy monitoring and contract execution tracking.

Direct negotiation between parties do not exclude the existence of a body that controls the grid environment, what is more the existence of Bazaar Portal enables better visibility of status of grid and operator's policy execution. Therefore, the proposed solution is suitable both for country-wide and global grids with different types of resources policy limitations. A prototype of Bazaar portal is currently tested. The first application of the framework is planned in PL-GRID Project [2].

References

1. EGEE Project, web page: http://www.eu-egee.org/
2. PL-GRID Project, web page: http://plgrid.cyfronet.pl/

Enabling Social and Economic Behaviour Based on Reliable Resource Metrics

Gabriele Pierantoni, Keith Rochford, Brian Coghlan, Eamonn Kenny
Trinity College Dublin, Ireland

Resource allocation and management in Grid Computing pose challenges of growing complexity; some of the solutions devised by the scientific community to cope with these challenges are based on economic and social paradigms. They attempt to apply to Grid Computing, the principles that form the base of economic exchange in the hope that the laws of the market will obtain efficiency and equilibrium in the Grid as they allegedly do in the real world.

This application of economic paradigms to Grid Computing introduces a number of complex issues regarding the process of price creation, arbitration of disputes and trust among actors. In addition, further complex challenges are presented by the need for reliable information and control systems, capable of interfacing the economic layer to the very fabric of the Grid.

In Trinity College Dublin, a prototype that enables economic and social transactions on the Grid is being developed under the name of "Social Grid Agents". A separate, yet complementary project - Grid4C, is developing a prototype for command and control of the grid.

The architecture of Social Grid Agents is based on two layers; one where Production Grid Agents compose various grid services as in a microeconomic supply chain and another layer where the Social Grid Agents (that own and control the agents in the lower layer) engage in social and economic exchange. Social Grid Agents allow different users to interact with each other in a variety of ways ranging from competitive to co-operative. They try to achieve this by mimicking (in the Grid world) parts of the behaviour that is the basis of competitive and co-operative relationships in human societies.

Grid4C aims to take the next step in the monitoring and management of Grid Computing systems by developing a distributed control plane for Grid resources and middleware components. It is intended to ease the burden on the members of the Grid Operations Centre and maximise resource availability. By introducing the ability to remotely manage grid resources, it has begun to close the loop on grid monitoring.

In addition to providing a monitoring and control system for use by human operators, the architecture of Grid4C is ideally suited to the application of machine control such as autonomic and agent-managed systems. Here, we explore this application and present an example of how the management endpoints provided by the system maybe exploited by other software components within the middleware. Grid4C management endpoints allow not only for more informed decisions to be made based on improved situational awareness, but also facilitate the provision of value added services through the ability to manage resources dynamically based on current requirements.

Grid4C's use of standards-based web service technologies not only facilitates interoperability and composition of management tools, but in this scenario, it also provides an abstraction layer in which we can define a uniform set of metrics, properties and operations to be made available to the clients.

During the developments of these prototypes the need for a system to determine the price and value of resource and the need for the social agents to exercise a degree of control over their resources suggested the merging and inter-operation of Social Grid Agents with Grid4C.

The solution described in this paper is based on the interoperability between Social Grid Agents and Grid4Cmanagement endpoints, allowing a two-way exchange of commands and information. Grid4C endpoints can be used by Social Grid Agents as both sensors and actuators on the Grid fabric. They provides Social Grid

Agents with information describing the production parameters of the various resources such as average waiting time, success rate and the like. The value and ultimately, the price of the resources can then be extrapolated by these values.

The flexibility of Social Grid Agents in implementing complex social topologies also allows for third, trusted parties to perform these measurements so that the price of the resources can be accepted by all those who trust the measurements actor. A first prototype that allows the managements of LGC2 resources and a controlled way for the determination of the price under the control of a third party is illustrated in the paper along with some preliminary results.

Combining Globus and JXTA for communication and collaboration of applications over heterogeneous network

Anatoly Doroshenko, Konstantin Rukhlis, Oleksandr Mokhnytsia
Institute of Software Systems of the National Academy of Sciences of Ukraine, Ukraine, Kiev

Various grid platforms are used to utilize constantly increasing computational power of the heterogeneous

resources. But the problem is incompatibility of the approaches they are based on. This fact makes effective collaboration of various solutions to be impossible, increases administration and management complexity, reduces benefits from the grid usage, raises TCO stats.

This paper presents AG-1.3 – the distributed platform for heterogeneous cluster resources management. Combining the power of Globus4 service-oriented architecture and peer-to-peer protocols of JXTA technology, it offers solution for both pure Globus and JXTA applications to communicate and collaborate over heterogeneous network. Also it provides means to access the homogeneous cluster resources. The platform utilizes JXTA to implement dynamic node discovery, clustering and messaging systems. It also applies agent-oriented approach for performance and status monitoring. Grid services were developed for implementing replication, network installation warehouse, cluster controlling and monitoring systems.

Use of the platform is demonstrated on interaction of several pure JXTA and Globus applications. Also, homogeneous cluster resources utilization is shown.

A Meta-Scheduling Algorithm for Load Balancing

Bin Wu, Yongjian Wang, Bo Li
Beihang University, Beijing, China

How to schedule tasks to proper computing element (Meta-scheduling) has becomes a big challenge in grid computing because of various QoS requirements. Almost all the grid middlewares have their own resource scheduling modules, but usually these modules are for general purposes such as RB/WMS in gLite, Matchmaker in Condor and so on, and can't satisfy the specific requirements of certain grid applications. Drug Discovery Grid (DDGrid) is a key grid application supported by CNGrid. The aim of developing the Drug Discovery Grid is to set up a grid environment for new drug screening service. Because of the poor resource scheduling policy used in DDGrid application, some computing elements are heavily loaded, while others are idle occasionally.

It has been proven that Meta-scheduling is NP-Complete problem and many heuristic algorithms have been proposed by researchers such as Min-min, Max-min and Genetic Algorithm (GA). Response Time and Load Balancing are the most important indexes in Meta-scheduling, but these algorithms either take only one of them as ultimate goal, or make a tradeoff between the two goals by a Weighted Average value of them which attend to one thing and lose another. In order to satisfy the specific requirements in DDGrid, we propose a QoS based LBMS algorithm.

The contribution of this paper including:

- Define a new Load Model to evaluate cluster load;
- Proposed a QoS based LBMS algorithm (Load Balancing Meta-Scheduling Algorithm).

1. Load Model

In Meta-scheduling circumstance, we schedule tasks among multiple clusters not within a single cluster. The previous load model which mainly employs a single machine's cpu or memory is not applicable in this circumstance. So we proposed a load model of cluster based on queue information of the batch system which is used to manage a cluster in this paper.

Like load models which employ utilization of cpu or memory in previous works, we take utilization of cluster which is based on the queue information of clusters as load model. In this paper, utilization of cluster is defined as Weighted Average value of utilization of clusters in the coming period and utilization currently, and the former is defined as the ratio between queuing task count and max-queuable task count and the latter is defined as the ration between running task count and max-runnable task count at a time. We also consider variance of all clusters' loads as indexes to evaluate load balancing of grid system. Load model we proposed in this paper is more applicable in meta-scheduling which schedules tasks among multiple clusters than the ones based on cpu or memory which is applicable in scheduling inside a single cluster.

2. QoS based LBMS algorithm

LBMS algorithm consists of two stages which are described in this chapter. Considering QoS of Deadline of tasks, the algorithm implements QoS based Min-min Heuristic to schedule tasks in the first stage in order to minimize the completion time of meta-task and adjust the task-resource mapping result generated from the first stage in order to avoid load imbalance. The procedure of the first stage is described as followed. At beginning, tasks which request Deadline QoS enter High-QoS queue, otherwise Low-QoS queue. It schedule tasks in High-QoS queue before ones in Low-QoS Queue. Next, it computes completion time for every task on every candidate cluster. The task with the overall minimum completion time is selected and assigned to the corresponding cluster (hence the name Min-min). Last, the newly mapped task is removed from High-QoS queue or Low-QoS queue, and the process repeats until all tasks are mapped.

In the second stage, the algorithm implements genetic algorithm (GA) which only has mutation operator to adjust the mapping result generated from the first stage. Some tasks might be moved from heavily loaded cluster to light loaded cluster in the stage. The goal of this stage is to avoid load imbalance among multiple clusters.

3. Experiments

We simulated part of the available computing resources in DDGrid application to evaluate the performance of the LBMS algorithm against QoS guided Min-min and Random algorithm. We simulated five clusters which consist of10,5,15,7,20 machines separately. The simulated experimental results show that load balancing is achieved by the proposed algorithm and does not increase extraordinary completion time of meta-task comparing to traditional heuristic algorithms, such as QoS guided Min-min.