International Workshop on Grid Computing for Complex Problems

Associate action to create national Grid initiative:
Making the Grid accessible for electronic science in Slovakia
(Sprístupnenie Gridu pre elektronickú vedu na Slovensku)

November 29 – December 2, 2005
Bratislava, Slovakia

The workshop is organized by
Institute of Informatics, Slovak Academy of Sciences
Faculty of Electrical Engineering and Informatics, Technical University of Košice

The workshop is supported by
EU 6FP RI project - Enabling Grids for E-sciencE
(2004-2006) INFSO-RI-508833

Program Committee

Ladislav Hluchý	Institute of Informatics, Slovak Academy of Sciences (General Chair)
Ján Paralič	Technical University of Košice

Organizing Committee

Ladislav Hluchý
Miroslav Dobrucký
Giang Nguyen
Viera Šipková
Oľga Schusterová
Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
E-mail: {hluchy.ui, dobrucky.ui, giang.ui, sipkova.ui, sekr.ui}@savba.sk

Proceeding Editors

Ladislav Hluchý
Giang Nguyen
Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
E-mail: {hluchy.ui, giang.ui}@savba.sk

ISBN 80-969202-1-9

VEDA - the publishing house of the Slovak Academy of Sciences
Graphical adjustment by Peter Repka.

Preface

Welcome to the International Workshop on Grid Computing for Complex Problems - GCCP2005. The workshop is three-day combined event for grid users: workshop with invited lectures, plenary discussions, accompanied by induction and grid application developer course, which is in the scope of EGEE project - Enabling Grids for E-science 2004-2006, INFSO-RI-508833.

The topics of the workshop are:
- High performance applications
- Tools and services for grid computing
- Knowledge mechanisms applicable in grid computing

The next goal of the workshop is an associate action to create national Grid initiative "Sprístupnenie Gridu pre elektronickú vedu na Slovensku" (Making the Grid accessible for electronic science in Slovakia) which will help to improve the e-science in Slovakia through the creation of virtual organizations for individual science branches. The associate action aims to join Grid specialists with complex application users, to provide a medium for the exchange ideas between theoreticians and practitioners to address the important issues in computational performance and computational intelligence towards Grid computing.

The workshop on Grid Computing for Complex Problems - GCCP2005 has attracted papers and active participations from Austria, Czech Republic, Hungary, Poland, Slovakia and Ukraine that are presented in the proceeding.

Many people have assisted in the success of this workshop. I would like to thank all the member of the Programme and Organizing Committees, the workshop Secretariat for their work and assistance of the workshop. I would like to express my gratitude to all authors for contributing their research papers as well as their participations in the workshop that made our cooperation more fruitful and successful.

Ladislav Hluchý
December 2005
Bratislava, Slovakia

Table of contents

Invited section

Grid Computing and Knowledge Management in EU RTD Projects of IISAS

Ladislav Hluchy,
Ondrej Habala, Giang Nguyen, Branislav Simo, Viet Tran and Marian Babik

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
hluchy.ui@savba.sk

Abstract. The paper presents history of grid computing and experience management at the Institute of Informatics, Slovak Academy of Sciences. It also presents influenced and current projects, their activities and applications. The ANFAS project has developed the support decision system for flood prevention and protection, integrating the most advanced techniques in data processing and management. The CrossGrid project went to exploitation of new Grid components for interactive computation-and-data intensive applications. Project EGEE is aiming at creating and deploying Grid technologies to enable the widespread uptake of e-Science applications throughout the European Research Area. The Pellucid project is the ontology-based experience management with the aim to relieve and improve work effectively in public organizations. The K-Wf Grid project addresses the need for a better infrastructure for the future Grid environment with envisioned approaches supported by semantic Web and Grid communities. The MEDIGRID project aims at creating a distributed framework for multi-risk assessment of natural disasters; their models will be upgraded to web applications.

Introduction

The Department of Parallel and Distributed Computing of the Institute of Informatics, Slovak Academy of Sciences has a long and fruitful history of international projects in the area of computer science. The first of the series of collaborations, which led to its current research efforts, has been the ANFAS project. While being concerned mainly with parallelization of simulation codes and hydraulic simulations, it prepared solid ground for the following project CrossGrid. This project extended the work done in ANFAS by using its results in a complex simulation scenario for flood prediction, using a cascade of meteorological, hydrological and hydraulic simulations. Also for the first time the Computational Grid has been heavily used as the supporting infrastructure and workflows as a method of application management. In parallel with CrossGrid, another important track of IT research at the department has been represented in the Pellucid project. This project has developed a knowledge infrastructure, supporting collaboration tasks and team knowledge management.

With the rapid development of grid computing and information society technologies, the opportunity to connect the two research lines of the department - grid computing and knowledge management - has been grasped in the K-Wf Grid project. This project further extends the results of the CrossGrid project, by bringing an elaborate knowledge management suite based in the Pellucid project into the area of SOA-based workflows, grid services and virtual organizations.

Apart from the research in fields of grid computing and knowledge management, the department takes part in infrastructure projects and projects trying to apply modern IT technologies in other areas of science. The EGEE project is a vast EU-wide collaboration created for the task of creation, management, interconnection and extension of European grids, their hardware infrastructure and supporting middleware. The MEDIGRID project creates the tools and a distributed environment for applications from the area of natural disasters and multi-risk management.

EGEE – Enabling Grids for E-sciencE

The EGEE project aims to provide researchers in academia and industry with access to major computing resources, independent of their geographic location. The EGEE project will also focus on attracting a wide range of new users to the Grid.

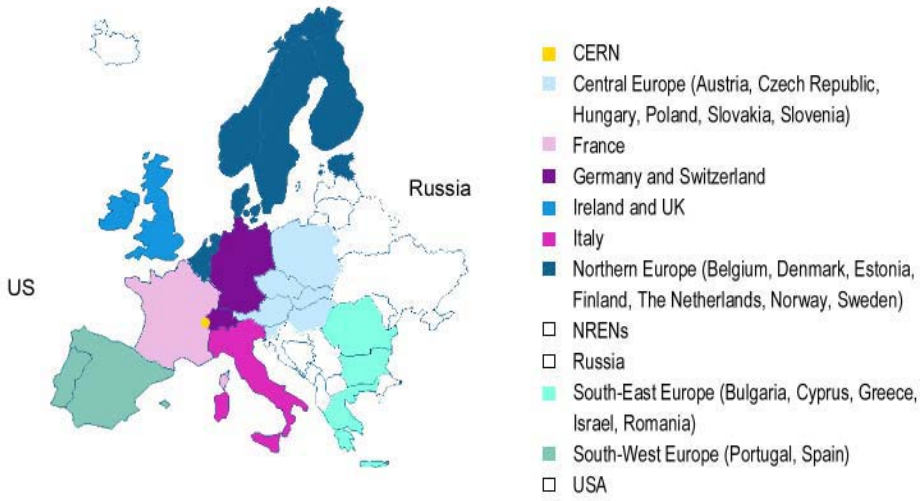


Fig. 1. EGEE participants

The project will primarily concentrate on three core areas:

– The first area is to build a consistent, robust and secure Grid network that will attract additional computing resources.

– The second area is to continuously improve and maintain the middleware in order to deliver a reliable service to users.
– The third area is to attract new users from industry as well as science and ensure they receive the high standard of training and support they need.
– The EGEE Grid will be built on the EU Research Network GÉANT and exploit Grid expertise generated by many EU, national and international Grid projects to date.

Funded by the European Commission, the EGEE project community has been divided into 12 partner federations, consisting of over 70 contractors and over 30 non-contributing participants covering a wide-range of both scientific and industrial applications.

Two pilot application domains were selected to guide the implementation and certify the performance and functionality of the evolving infrastructure. One is the Large Hadron Collider Computing Grid supporting physics experiments and the other is Biomedical Grids, where several communities are facing equally daunting challenges to cope with the flood of bioinformatics and healthcare data. The envisioned evolution of the European Grid infrastructure is involved from these two pilot applications to an infrastructure serving multiple scientific and technological communities, with enormous computer resources.

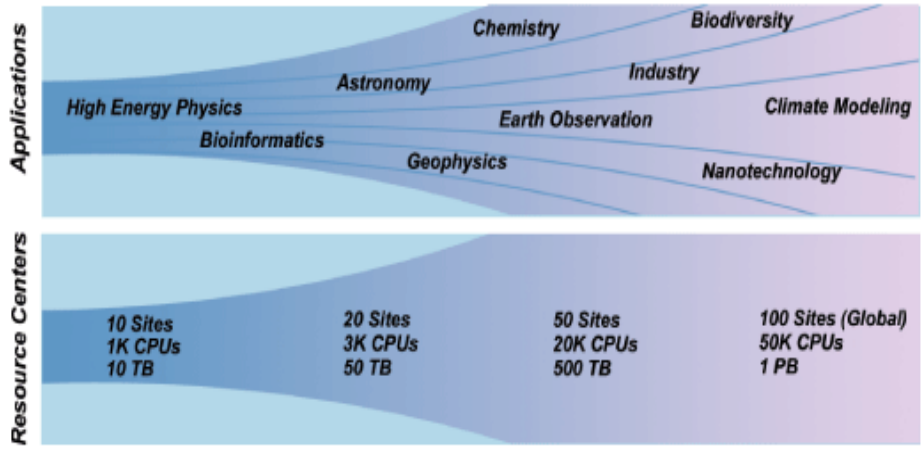


Fig. 2. Evolution of the EGEE

IISAS is participating in the EGEE activities SA1 (European Grid Support, Operation and Management) and NA3 (User Training and Induction). European Grid Support includes tasks such as grid monitoring and control and resource and user support. User Training and Induction aim at tasks such as organizing on-site training and producing training and course material. IISAS has been very successful in fulfilling the objectives of these tasks. This is supported by the fact, that IISAS has organized several induction and application developer courses and managed to assists in the integration of four Slovakian computing centers into the EGEE infrastructure.

ANFAS - Data Fusion for Flood Analysis and Decision Support

The ANFAS project is able to use data from the most advanced acquisition technology; in particular remote sensing imagery (optical, radar, interferometer radar) will be incorporated into a conventional Geographical Information System database. The project has strong end-users involvement through the definition of the objectives, the conception of the System, the evaluation of the simulation results; it ensures that the System answers to "terrain needs" and to be well adapted in practical use.

The ANFAS system is based on a 3-tier architecture:

- the ANFAS client,
- the ANFAS core server,
- the servers – or wrappers – allowing to “connect” possibly in remote access external features as the database, the models and possibly additional processes (e.g. computer vision ones).

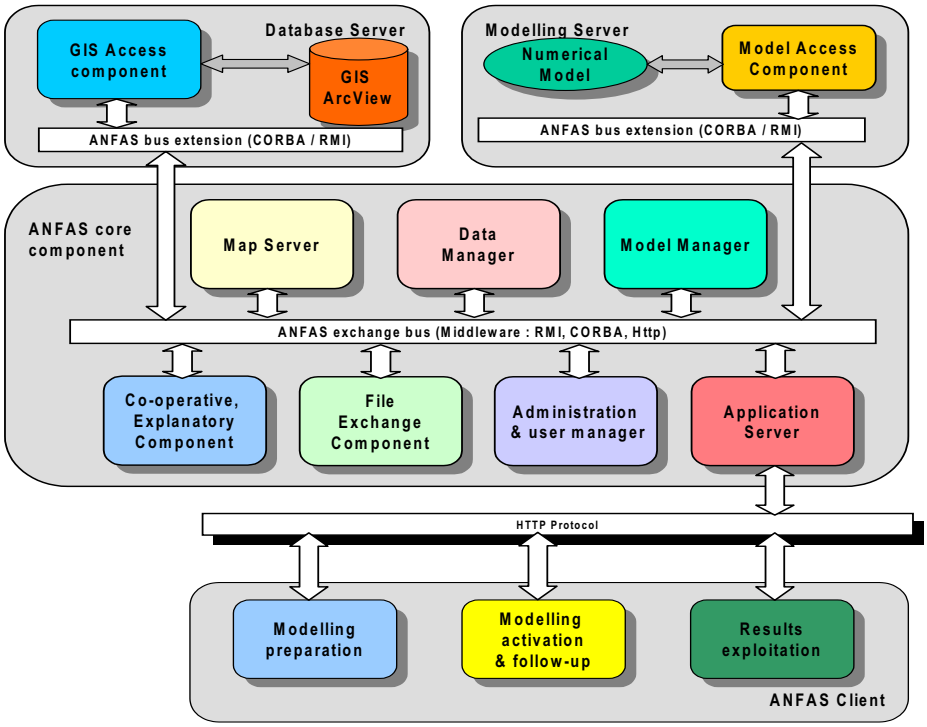


Fig. 3. ANFAS architecture

The main functionalities of ANFAS system are:

- to perform flood simulation, given a scenario, water flow propagation is simulated; the user can interact with the scene that models the real site in order to add/remove dikes or other human constructions;

- to assess flood damage, flood assessment can be done either using the simulation results or using the remote sensed images of a real flood event;
- at a prospective level, to analyze floodplain morphology including riverbed and subsurface properties changes due to repetitive floods for a given site

The main modeling module in ANFAS is FESWMS (Finite Element Surface-Water Modeling System) Flo2DH which is a 2D hydrodynamic, depth averaged, free surface, finite element model supported by SMS. Flo2DH computes water surface elevations and flow velocities for both super- and sub-critical flow at nodal points in a finite element mesh representing a body of water (such as a river, harbor, or estuary). Effects of bed friction, wind, turbulence, and the Earth's rotation can be taken into account. In addition, dynamic flow conditions caused by inflow hydrographs, tidal cycles, and storm surges can be accurately modeled. Since Flo2DH was initially developed to analyze water flow at highway crossings, it can model flow through bridges, culverts, gated openings, and drop inlet spillways, as well as over dams, weirs, and highway embankments. Flow through bridges and culverts, and over highway embankments can be modeled as either 1D or 2D flow. Flo2DH is ideal for modeling complex flow conditions at single- and multiple-bridge and culvert roadway crossings, such as the combination of pressurized flow, weir overflow, and submerged weir overflow, which are difficult to evaluate using conventional models. It is especially well-suited at analyzing bridge embankments and support structures to reduce and eliminate scour during flooding.

Modeling and simulation at Vah River pilot site

Vah river is the biggest river in Slovakia, therefore it was chosen as a pilot site for ANFAS project. Modeling and simulation of flood need following input data:

- Topographical data: There are several sources of input data for topographical data: orthophotomaps, river cross-sections, LIDAR (Light Detection and Ranging) data. GIS (Geographical Information System) is used to manage, combine these data and product a final map for simulation.
- Roughness conditions: These data cannot be directly obtained, but are derived from orthophotomaps.
- Hydrological data: These data are obtained from monitoring past flood events. These data can be used as boundary conditions as well as for calibration and validation of models.

Creating and calibrating models is a challenging process that can be done only by experts in hydrology. SMS (Surface-water Modeling System) provides a convenient graphical interface for pre- and post-processing that can help the experts to speedup the modeling process.

CROSSGRID - Development of Grid Environment for Interactive Applications

The aim of the IST CrossGrid project was to develop, implement and exploit new Grid components for interactive compute and data intensive applications like simulation and visualization for surgical procedures, flooding crisis team decision support systems, distributed data analysis in high-energy physics, air pollution combined with weather forecasting. It involved 21 partners from Poland, Netherlands, Austria, Germany, Cyprus, Italy, Greece, Ireland, Spain, Portugal and Slovakia. II SAS was the only partner from Slovakia. II SAS was responsible for the flood forecasting application, its integration into the grid environment and with the tools developed by other project's partners.

The flood forecasting application consisted of several simulation models (meteorological, hydrological and hydraulics) and appropriate post-processing tools connected together, thus constituting a workflow. The models have been calibrated in cooperation with Slovak Hydrometeorological Institute.

The workflow could be processed by the grid resources with a single click by exploiting the workflow grid service, implemented by II SAS, which managed its whole execution. Results of the whole simulation and each of its steps could be registered to the replica manager and described by metadata stored in the metadata catalog service (also implemented by II SAS). The metadata catalog could be browsed and searched later on in order to retrieve the registered files.

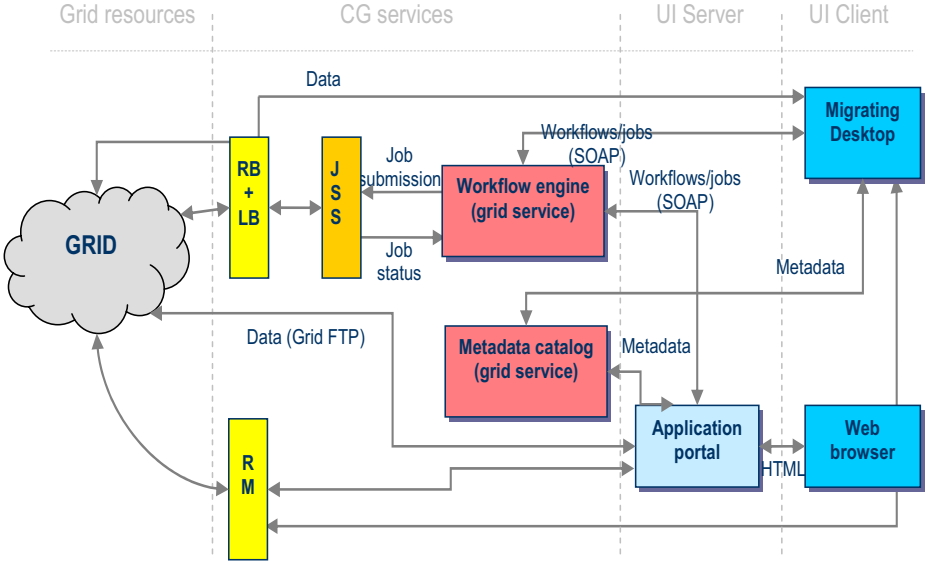


Fig. 4. Flood application architecture

Above-mentioned functionality could be accessed using two user interfaces: a web based application portal and a Migrating Desktop

The flood portal portlets have also been incorporated into the OGCE collaborative environment giving user a possibility to work with simulations and collaborate with other users in the same environment.

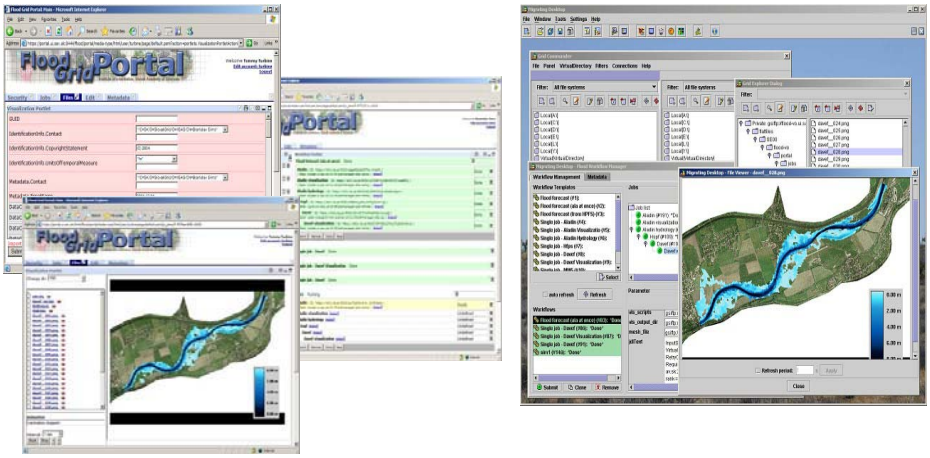


Fig. 5. Application portal (left) and Migrating Desktop (right) running flood application

II SAS has been specifically responsible for following activities in the project:

- Adaptation of simulation models for grid environment
- Development the workflow and metadata services for flood application
- Development of application portal
- Implementation of visualization tools for results of hydraulic simulation (2D, 3D)
- Administration of local PC cluster that was a part of the project grid testbed

The CrossGrid project has been evaluated as one of the most successful IST projects. Its dissemination page can be reached at http://www.crossgrid.org.

Pellucid system: Experience Management in Public Organizations

The purpose of Pellucid is to support and enhance employees' performance by providing them with the knowledge required by the activity they are performing at the time they are actually performing the activity. To do so, it is included the concept of Active Hint (AH), a representation of experience within the organization. Experience can be seen as knowing what to do in particular circumstances. The circumstances correspond to the context and the "know what to do" is characterized by the action and resources needed in that action.

Important background of the Pellucid design is its ontology, which defines structure and relationships among experience entities. The ontology is the main mechanism used for the representation of information and knowledge, definition of the meaning of the terms used in the content language and the relation among these terms. Ontology is essential to the whole work of the Pellucid platform. Agents

communicate using ontology structure, store and retrieve data organized and described by ontology to/from OM. The analysis and design of the Pellucid ontology includes a contextual and conceptual modeling, where aspects of a generic public organization are modeled in fluency with the CommonKADS methodology [CommonKADS]. The Web Ontology Language [OWL] describes hierarchical relationships, property relationships, equivalent/disjoint concepts, cardinality constraints, etc.

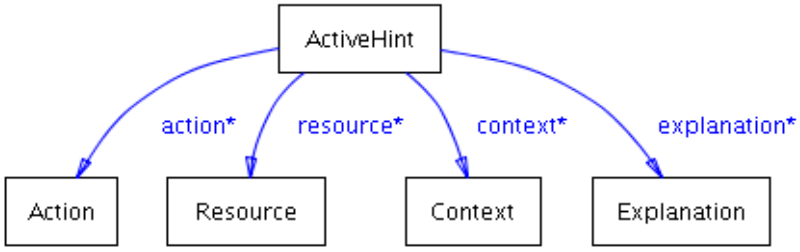


Fig. 6. Active Hint concept

The Pellucid experience management model comprises three phases: Capture and Store, Analysis and Presentation, and Experience Evolution:

- The Capture and Store phase is concerned with observing and storing experience in a particular context. There are 3 ways of capturing: analyzing employees' actions and workflow events; analyzing documents entered into the sys-tem; and by direct input from workers. Capturing experience from working actions and events is particularly beneficial in repetitive tasks; they are used to create common patterns that can be retrieved in the future in order to assist other employees. Documents constitute an important asset in an organization. The direct capture of experience from employees is carried out through free-text notes written by the employees themselves. This constitutes a good source of knowledge, particularly in the transmission of experience from experienced employees to novices.
- The Analysis and Presentation phase is concerned with providing such knowledge. To do so, the concept of an active hint is introduced, a representation of experience within the organization. An active hint is triggered in a context and includes an action, a knowledge resource and a justification for the hint. The context is determined by the particular activity that is carried out by the employee at that time in a workflow sys-tem. An action corresponds to an atomic act on a knowledge resource, for example use a document template, read a document or a note, or consider a contact list. The justification gives to the employee a reason for the hint
- The aim of Experience Evolution phase is updating the available experience. The Pellucid system will include a set of methods and semi-automatic tools to allow knowledge engineers and expert users to update the experience stored in the Organizational Memory.

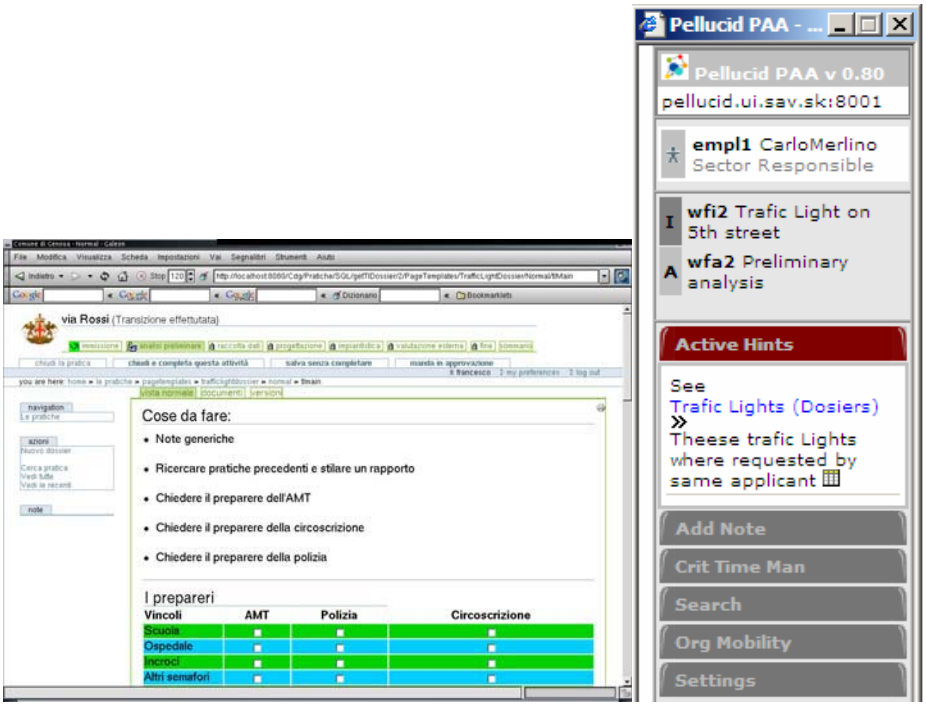


Fig. 7. Pellucid platform working with underlying WfMS/WfTS (Italy pilot site application)

The Pellucid platform consists of four main layers/components: Organizational Memory (OM), Process Layer, Interaction Layer and Information and Search Access (ISA) Layer. The Pellucid system provides a generic solution of supporting system that works as an extension to underlying WfMS/WfTS. Agent-based components in the Process Layer cooperate with other Layers and Organizational Memory in order to provide the best assistance to employees. The Pellucid system find and generate useful active hints for each employee in every their working context. It processes events reported from external systems, stores and analyzes them and in the future returns such experiences in the form of active hints appreciated for current situations. The recommendation from the Pellucid system is presented to end-users through the Interaction Layer with user-friendly and customizable GUI.

MEDIGRID - Mediterranean Grid of Multi-Risk Data and Models

The aim of the project is to create a distributed framework for multi-risk assessment of natural disasters that will integrate models, which have been developed in previous projects funded by European Commission. These include models for simulation of forest fire behavior and effects, flood modeling and forecasting, landslides and soil erosion simulations. Also, a distributed repository with earth observation data,

combined with field measurements is being created, which will provide data to all models using data format conversions when necessary. The entire system of models and data will be shaped further as a multi-risk assessment and decision support information platform.

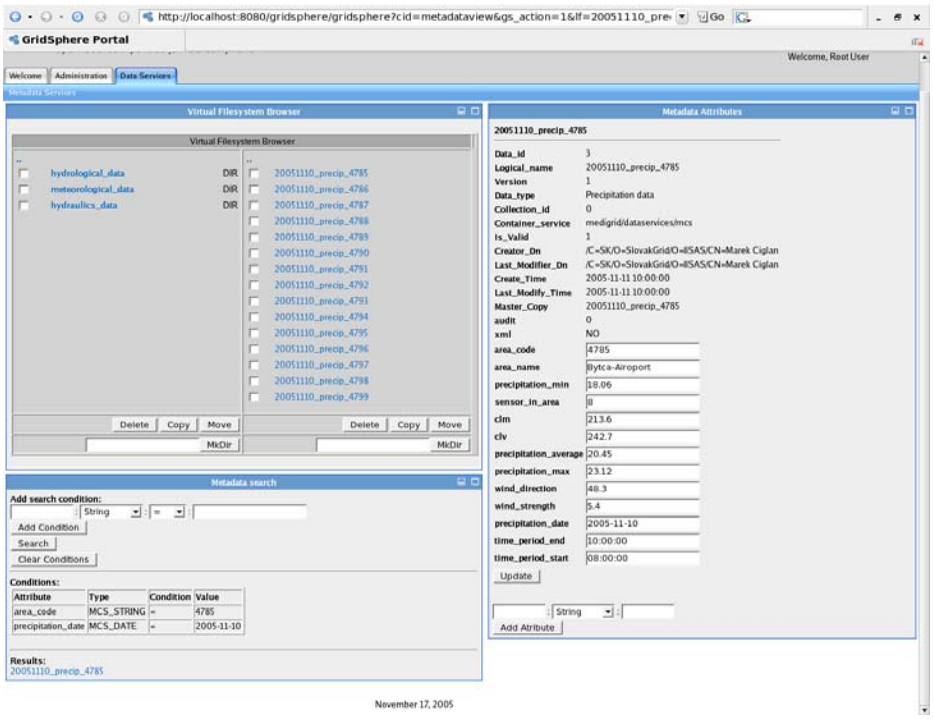


Fig. 8. Metadata browser in MEDIGRID portal

There are 6 partners in the project from Greece, Portugal, France, Spain, United Kingdom and Slovakia.

In order to create a virtual organization (VO) for multi-risk assessment of natural disasters a grid middleware had to be chosen to be used on computing resources. Because each of the partners provides some of the services on his own resources that run both Linux and Windows, the Java implementation of the WSRF specification by the Globus alliance has been chosen, as it runs on both platforms. It is an implementation of core web (grid) services with security, notifications and other features. Each of the system components – simulation models, data providers, information services or other supporting services – will be exposed as a web service. We use WSRF as a basic technology that both serves as an implementation framework for individual services and also glues the individual components together. The whole system will be accessible via a web portal. We have chosen GridSphere portal framework for its support of portlet specification. Application specific portlets

will allow users to invoke all the simulation services plugged into the system in application specific manner; for example using maps for selection of a target area or an ignition points for forest fire simulations. There will be portlets for browsing results, metadata describing those results, testbed monitoring and others.
So far, two services have been implemented on top of the WSRF: Data Transfer service and Job Submission service.

Data Transfer service (DTS) serves as a replacement for widely used GridFTP tools. The main disadvantage of GridFTP is that implementations are available just for the UNIX platforms. In Medigrid, Windows is a platform of several models and porting them to UNIX world was not an option for developers. DTS provides data access policies definition and enforcement in terms of access control lists (ACLs) defined for each data resource – a named directory serving as a root directory for given directory tree accessible via the service. It has been also integrated with central catalog services: Replica Location Service (RLS) and Metadata Catalog Service (MCS).

Job Submission service provides the ability to run the executable associated to it with parameters provided with job submission request. Currently, jobs are started locally using the "fork" mechanism on both Linux and Windows. Requests are queued by the service and run in the "first come first served" manner in order not to overload the computer. In near future we plan to add job submission forwarding from the service to a Linux cluster and later on to a classical grid.

A base of the project's portal has been set up based on the Gridsphere portal framework. Portlets have been developed for browsing the contents of the metadata catalog service and a portlet for generic job submission.

K-Wf Grid - Knowledge-based Workflow System for Grid Applications

The K-Wf Grid project employs modern technologies, like web services, the WS-Resource and related standards, knowledge management and workflows in order to make grid-based scientific computations simpler and more appealing to scientists. The project's main goal is to develop a framework, able to aid users in creation and execution of workflows composed of web and grid services. This framework uses knowledge management techniques, ontologies and reasoning and it is able to gather experience and reuse it in the optimization of future tasks.

The project is a collaboration of 6 partners from the EU. Being led by the Fraunhofer Institute FIRST from Berlin, partners from II SAS (Bratislava, Slovakia), CYFRONET (Cracow, Poland), University of Innsbruck (Austria), LogicDIS, s.a. (Greece) and Softeco, s.a. (Genoa, Italy) work together in the frame of the established architecture (see Fig. 1). The final system is supposed to enable user to create, execute, monitor, evaluate and annotate workflows of web and grid services, working

through a web portal and seamlessly using several highly specialized knowledge management tools.

The user accesses the world of grid hidden behind K-Wf Grid system through a portal interface. He/she logs in, and immediately may start the process of the creation of his/her application workflow. For this purposes the system incorporates the User Assistant Agent (UAA), through which the user may state his/her problem in the form of a simple sentence. This sentence is analyzed, its context is established, and based on this context the user is given choice of several known solutions. When the user picks one of these solution, UAA contacts the Grid Workflow Execution Service (GWES), which creates a new abstract workflow, and this is displayed in the Grid Workflow User Interface (GWUI). Through GWUI the user may view the workflow, commence its decomposition into a set of real service invocations, and finally execute it in order to obtain the needed results. The interconnection of the right services, data management and the process of service invocation is handled by the K-Wf Grid system, without the user needing to understand how it works.

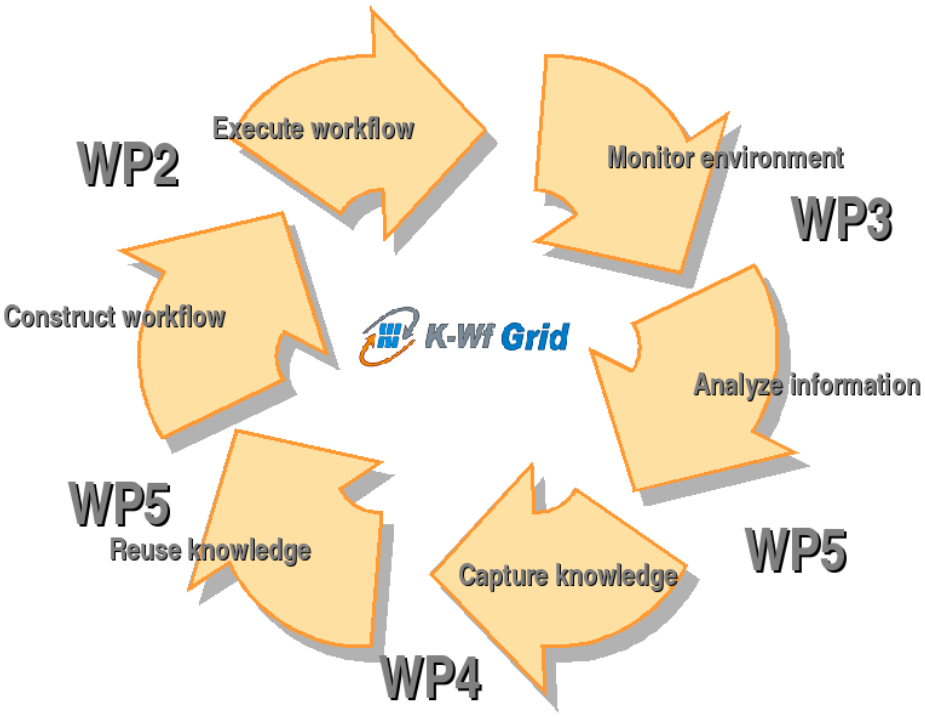


Fig. 9. Knowledge management in K-Wf Grid

The system employs also a system of distributed monitoring sensors, performance monitoring and analysis services, which gathers new data about the executed workflows, employed services and grid environment conditions. This data is then

analyzed by the Knowledge Assimilation Agent (KAA), new knowledge is extracted from it and stored in the Grid Organization Memory (GOM) for later reuse in new application workflows.

The project develops also 3 pilot applications, dealing with the problems of flood prediction based on a cascade of meteorological and hydraulic simulations, enterprise resource planning and coordinated traffic management in the city of Genoa.
The project's results will enable more scientists from scientific areas outside of IT to use the power of grid computing, virtualized resources, and collaborative work and knowledge management systems at their daily work.

Acknowledgements

This paper is supported by the project: EGEE - EU 6FP RI (III) project: Enabling Grids for E-sciencE (2004-2006) INFSO-RI-508833; K-Wf Grid - EU 6FP RTD IST project: Knowledge-based Workflow System for Grid Applications (2004-2006) FP6-511385; Pellucid - "A Platform for Organizationally Mobile Public Employees", EU 5FP RTD IST-2001-34519; MEDIGRID - EU 6FP RTD: Mediterranean Grid of Multi-Risk Data and Models (2004-2006) FP6-004044; CROSSGRID - EU 5FP IST RTD project: Development of Grid Environment for Interactive Applications (2002-05) IST-2001-32243; ANFAS - EU 5FP IST RTD project: datA fusioN for Flood Analysis and decision Support (2000-03) IST-1999-11676; NATO project: Flood Forecasting on Grid Infrastructures (2004-2006) EST.EAP.CLG 981032, VEGA No. 2/3132/23, VEGA No. 2/6103/6; APVT-51-024604 and SPVV 1025/2004.

References

1. [IISAS projects] http://ups.savba.sk/parcom/projects.html
2. [CommonKADS] Schreiber August Th., et al.: Knowledge Engineering and Management: The CommonKADS methodology, ISBN 0-262-19300-0, The MIT Press, 2002.
3. [EM] Ralph Bergmann: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, Lecture Notes in Artificial Intelligence, ISBN 3540441913, Springer-Verlag Telos, 2002

Interactive Result Visualization on the Grid

Paul Heinzlreiter

GUP Linz, Johannes Kepler University Linz
Altenbergerstraße 69, A-4040 Linz, Austria
heinzlreiter@gup.jku.at

Considering the nowadays important role of grid computing within the domain of computational science and related applications the visualization of results produced on the grid becomes a crucial task. Since grid-based simulations commonly produce result data of signi□cant size visualization is in most cases required to support the users understanding. On the other hand the size of the datasets to be visualized turns the visualization itself into a suitable grid application.

However the present status of visualization on the grid is not really satisfying. In contrast to other grid services, such as batch processing, data transfer, and resource scheduling, visualization is still commonly utilized in the traditional point-to-point fashion, with applications integrating visualization as subroutine calls or even post-mortem. This situation is addressed by the Grid Visualization Kernel GVK, which proposes a fully grid-enabled approach to interactive scienti□c visualization. The infrastructure of GVK enables visualization and interactive steering of a remote simulation while the actual processing of the data within the visualization pipeline is transparently performed on the available grid resources.

For delivering visualization services GVK relies on glogin which offers interactive direct connections to grid resources. Within GVK-based applications glogin serves as data transportation layer and additionally supports interactive application steering over secure connections. Another important building block of GVK are grid-enabled modules based on Visualization Toolkit (VTK) classes. For performing the visualization task which is represented as a pipeline it is decomposed into multiple data conversion stages which can be executed on different resources within a grid testbed.

The GVK approach has been used within several different application domains in the scope of the EU Crossgrid and Austrian Grid projects. We have realized GVK-based prototypes for disaster management, medical applications and astronomical data visualization.

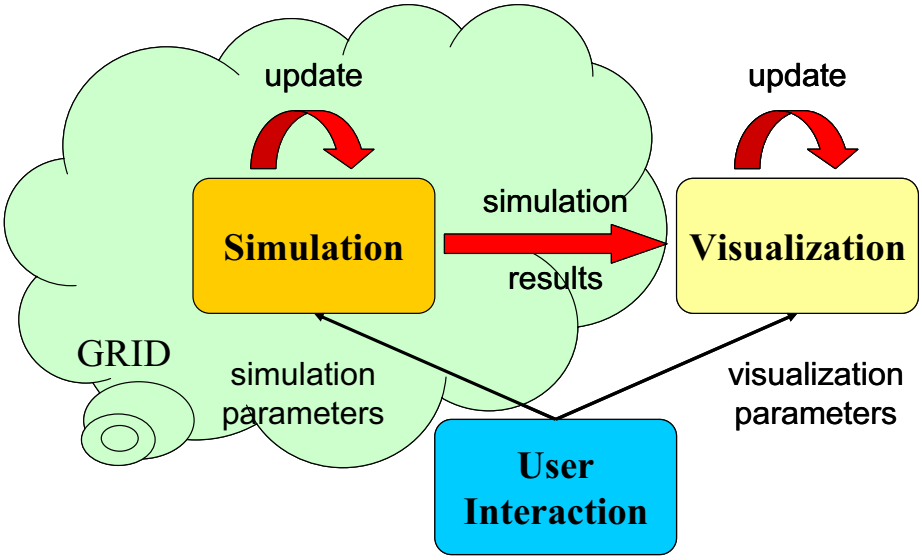


Fig. 1. Interactivity - „Putting the user into the loop“

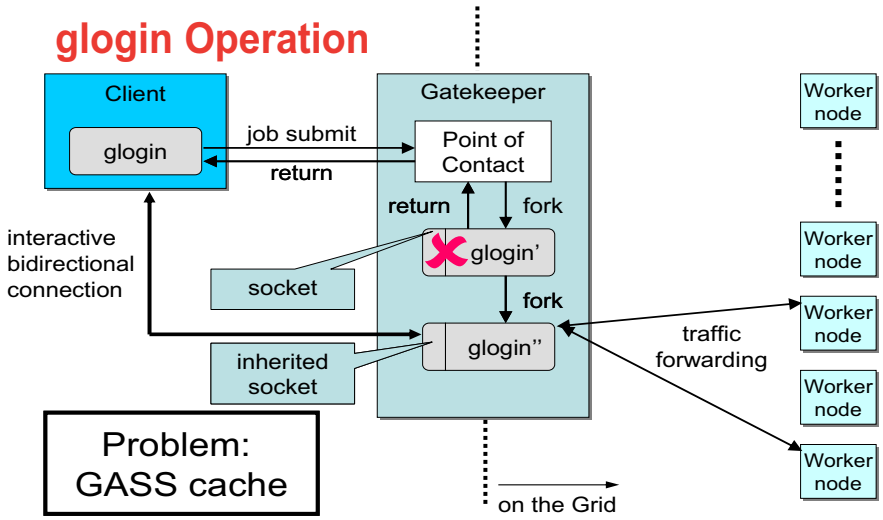


Fig. 2. glogin Operation

P-GRADE Portal: Towards a User-friendly Grid Environment

Peter Kacsuk and Miklos Kozlovszky

MTA SZTAKI, Budapest, H-1528 P.O.Box 63, Hungary
{kacsuk, m.kozlovszky}@sztaki.hu

Abstract. The rapid evolution of Grid systems in the last ten years has made the life of their users quite difficult. As each new system is deployed, users have to adapt to a whole new set of operating methods. Users working with the worlds various Grid systems have to learn new sets of commands for each system they utilize. Typically these commands are low-level, tedious to memorize and error-prone to use. Learning these commands demands significant effort and time from the user before they even get to use the system. P-GRADE, P-GRADE Grid portal and Mercury Grid monitor are a family of tools developed by MTA SZTAKI to specifically combat these problems.

Introduction

P-GRADE breaks down the barriers between incompatible Grid systems, helping the user to develop parallel code that can be used on both supercomputers, clusters and in various Grids. P-GRADE hides the complexity of the Grid from the user, making it transparent. Users no longer have to worry about which service or resource they are accessing, as they will use the same high-level graphical environment and tools.

The P-GRADE Grid Portal

The P-GRADE Grid Portal is a workflow-oriented Grid portal that enables the creation, execution and monitoring workflows in grid environments through high-level, graphical Web interfaces [1]. Components of the workflows can be sequential and parallel (MPI, PVM) jobs. The P-GRADE Grid Portal hides the low-level details of Grid access mechanisms by providing a high-level Grid user interface that can be used for any Grid.

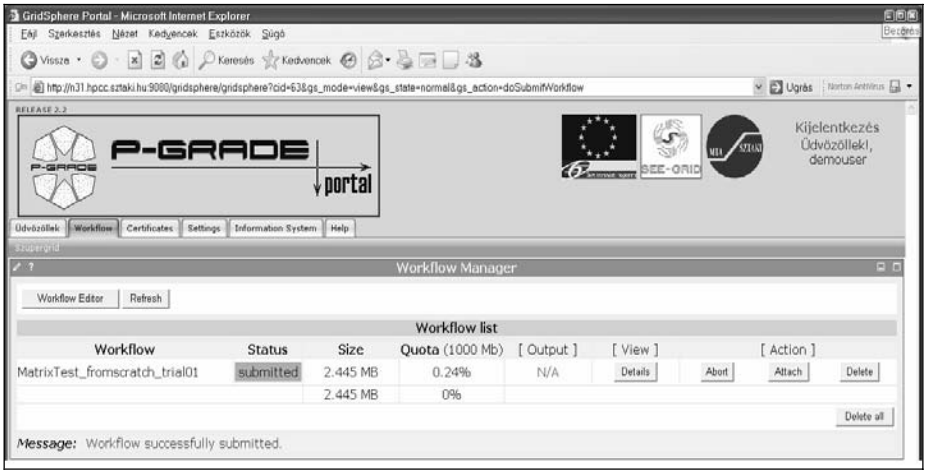


Fig. 1. User interface of the P-GRADE Portal

Workflows developed in the P-GRADE portal are portable between different Grids without any re-learning of the features of the new Grid system. In this way P-GRADE Grid Portal helps the user to cope with the large variety of Grid systems. More than that P-GRADE portal can be configured to access several Grids and if a user has valid certificate for several Grids, then she can exploit all those Grids simultaneously during the execution of her workflow [2]. Accessing Grid resources with the P-GRADE Grid Portal is as easy as never before, thus the P-GRADE Grid Portal is a perfect tool to anyone who would like to effectively solve computational intensive problems and avoid complicated grid protocols and commands at the same time.

Grid Infrastructures served by P-GRADE Portal

The P-GRADE Portal has been already demonstrated successfully at different forums, with many complex application scenarios (ultra-short range weather forecasting [3], drug discovery, etc). This workflow-oriented Multi-Grid portal solution is already the entry point of the following Grid systems:

- SEE-Grid
  SEE-Grid is the South-East European Grid infrastructure, and P-GRADE Portal is the official portal of the SEE-GRID Grid infrastructure.
- HunGrid
- HunGrid is the Hungarian Virtual Organization (VO), and P-GRADE Portal is the official portal of the Hungarian Grid infrastructure.

Besides that the P-GRADE Portal is available as service for the following different Grid systems as well:

- VOCE - Central European Virtual Organization (VO) of the EGEE Grid infrastructure

– UK NGS - UK National Grid Service
– Croatian Grid
– EGrid (Italy)

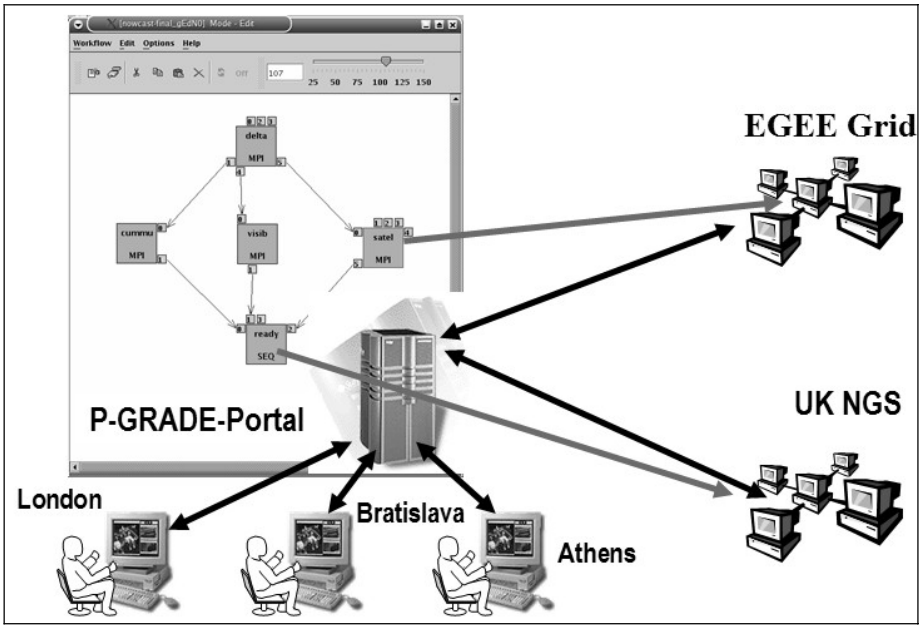


Fig. 2. Multi-Grid usage scenario with P-GRADE Portal

Due the Multi-Grid capability of the P-GRADE Portal, the users are able to launch workflows on different Grid infrastructures, and different jobs can be executed in different Grids.

References

1. P-GRADE Graphical Parallel Program Development Environment: http://www.lpds.sztaki.hu/projects/pgrade
2. P. Kacsuk, G. Dózsa, J. Kovács, R. Lovas, N. Podhorszki, Z. Balaton, G. Gombás: P-GRADE: A Grid Programming Environment. In: Journal of Grid Computing, Vol. 1. No. 2, pp. 171-197. 2004.
3. R. Lovas, et al., "Application of P-GRADE Development Environment in Meteorology", Proc. of DAPSYS'2002, Linz, pp. 30-37, 2002

Virtual Laboratory – closer the eScience

Marcin Lawenda, Damian Kaliszan, Piotr Mierzyński, Marcin Okoń,
Dominik Stokłosa, Tomasz Rajtar, Norbert Meyer and Maciej Stroiński

Poznań Supercomputing and Networking Center
vlab@man.poznan.pl
http://vlab.psnc.pl

An advanced scientific research becomes more and more expensive nowadays. It concerns especially experimental science, where the major cost of the research and development belongs to the experiment. Additionally the laboratory equipments that are used during the experimental work are very expensive and therefore unique as well, i.e. spectroscopes, radio telescopes or CAT scanners.

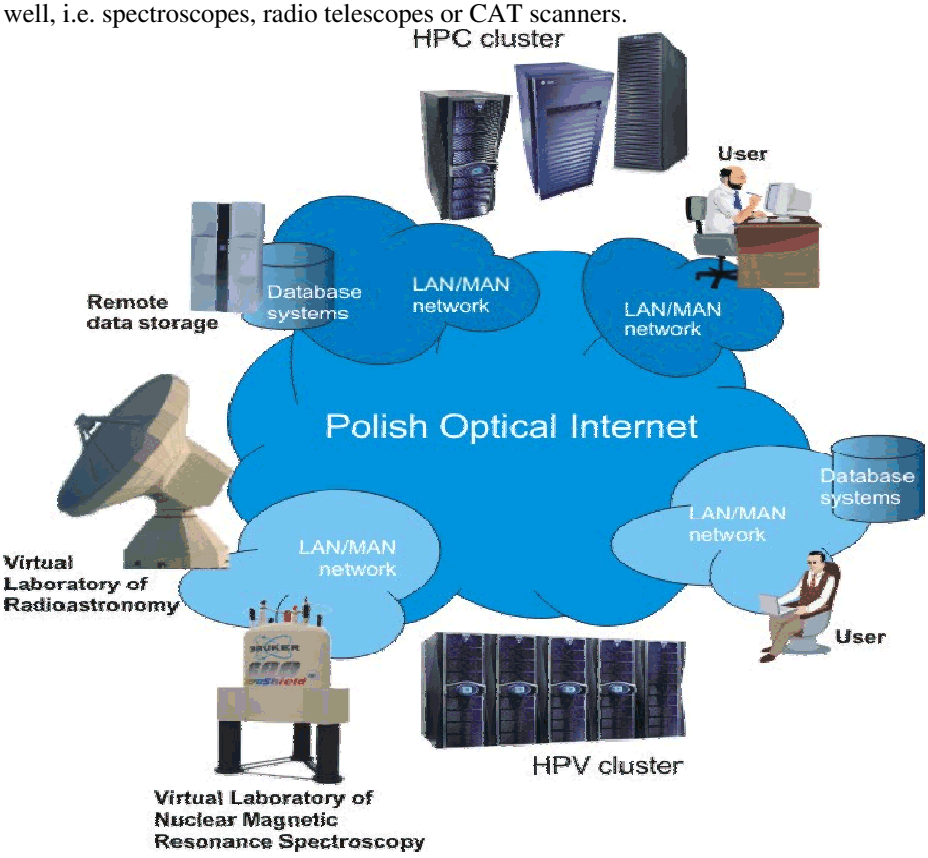


Fig. 1. The infrastructure of Virtual Laboratory in Poland

Only big regional or national centers can afford to purchase and use it, but on a very limited scale. That is a real problem that disqualifies all other research groups

not having direct access to these instruments. On the other hand we notice a lot of activities in European Union related eInfrastructure and equal chances for the entire scientific community.

The Virtual Laboratory (VLAB) project deployed a general architecture framework, which plays a crucial role in equalizing the chances of all research groups. It enables a remote usage of many different scientific instrumentations and focuses its activity on embedding labor equipments in grid environments (handling HPC and visualization). In general the issues concern the standardization of the labor equipment definition to treat it as a simple grid resource, supporting the end user under the term of the workflow definition and prioritizing jobs, which follow experiments on the equipments.

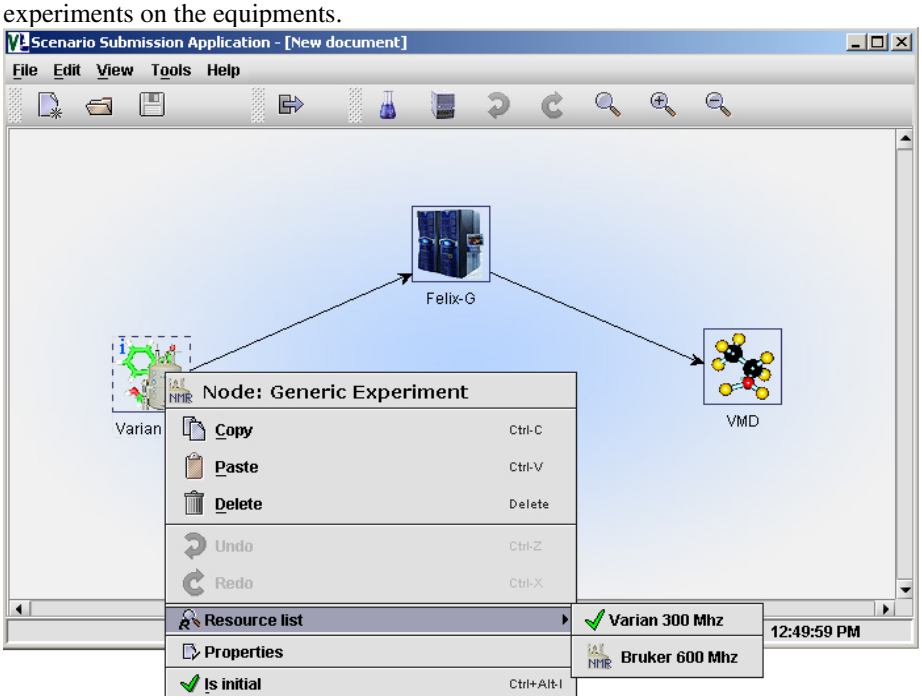


Fig. 2. A simple workflow management for the Varian spectroscope

The VLAB in its current state is operational in a distributed computational environment. The physical layer includes the Polish national optical network PIONIER and the HPC and visualization infrastructure. The application, in fact the framework, can be used in all experimental disciplines, where access to physical equipments is crucial, e.g., chemistry (spectrometer), radio astronomy (radio telescope), or medicine (CAT scanner).

The scientific community will benefit from this project for several reasons. It will make scientific work easier. The infrastructure will provide remote access to laboratory devices. The users from smaller science institutions can use devices unavailable for them because of the cost or distance from the laboratory, which

possesses it. VLAB delivers also additional functionality, like digital libraries (training courses for young researchers, database of results with advanced visualization), tools for collaborative work and workflow systems, making the experiment as easy as possible.

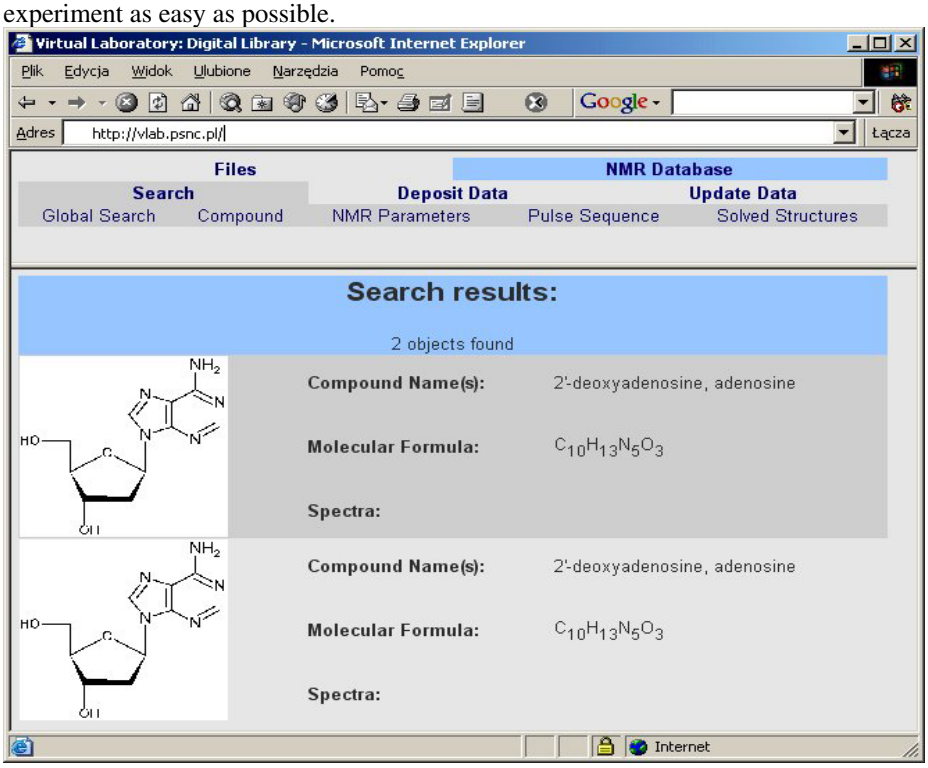


Fig. 3. Search engines in digital library

On the other hand there are still open issues in the terms of computer science (e.g. resource management, accounting, advance resource reservation, QoS – quality of service) and scientific disciplines, where the facilities will come (how to share the devices, how to decrease the necessary knowledge about the equipment to focus on the real experiment and scientific problems)

The demonstration will show how we deployed the concept in chemistry and radio astronomy, supporting these disciplines with grid environment and embedding the Bruker Avance 600MHz and Varian spectrometers, and radio telescope.

Acknowledgement

The work were done in co-operation with the Institute of Bioorganic Chemistry Polish Academy of Sciences and the Department of Radioastronomy of the Nicolaus Copernicus University (project no. 6 T11 0052 2002 C/05836, 4 T11 F 010 24)

EGEE middleware for grid application developers

Miroslav Ruda

Masaryk University and CESNET
Zikova 4, 160 00 Praha 6, Czech Republic
ruda@ics.muni.cz

The EU EGEE projects aims to build the largest general purpose Grid in a world. While primarily driven by the requirements from High energy physics community, the EGEE middleware -- gLite -- provides a solid basic Grid working environment. The lecture will provide a short gLite middleware overview, followed by more specific information about the key features of the gLite workload management and data management concepts.

More detailed explanation will be given to services developed by the CESNET (Czech Republic) groups that target the difficult task of job tracking in a distributed Grid environment. The overview and details of use of the Logging and Bookkeeping service, which keeps up to date information about live jobs on the Grid, will provide enough details for its advanced use and for understanding of information that it provides. The newly introduced Job Provenance is a job information repository, that keeps data about all the jobs run on an EGEE Grid (currently provided they were submitted via the gLite workload management system). This data could be used for any statistical purposes (including the evaluation of Grid performance), but they can also be used to re-run jobs in the same or modified conditions.

The EGEE users are organized in Virtual Organizations (VOs), that are also responsible for allocation of resources and VO specific user support. Most EGEE VOs are application oriented, we will present a VOCE, the Virtual Organization for Central Europe. This VO is application neutral, providing international Grid environment for people with serious grid needs but not (yet) associated with any application oriented VO. The way how to register and use VOCE, and how to access information related to it will be presented.

In the last part, high level API for grid applications (GAT), outcome of a GridLab project, will be introduced. This API, which is currently further, developed within the Simple Application Grid API (SAGA) activity in GGF, targets application developers' needs for a simple uniform development environment for otherwise complex and heterogeneous Grids.

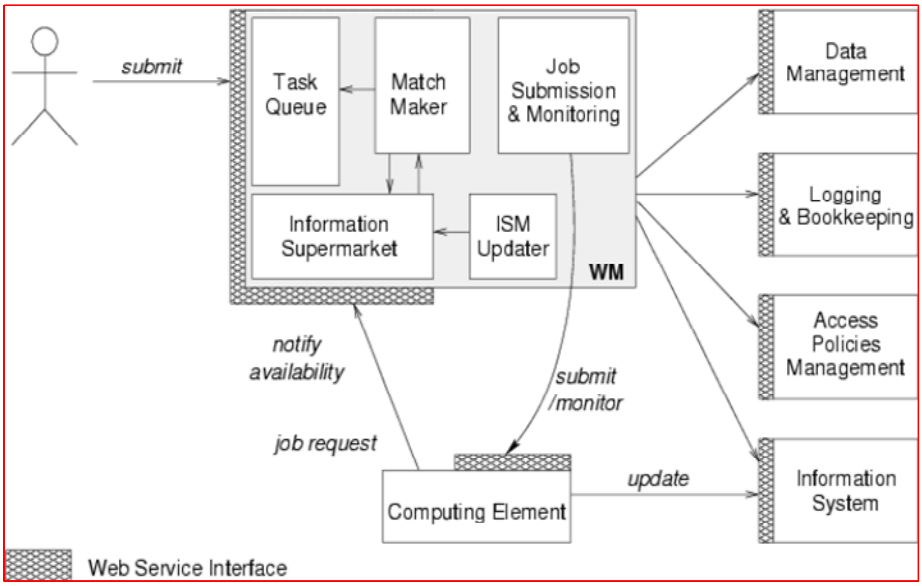


Fig. 1. WMS in gLite

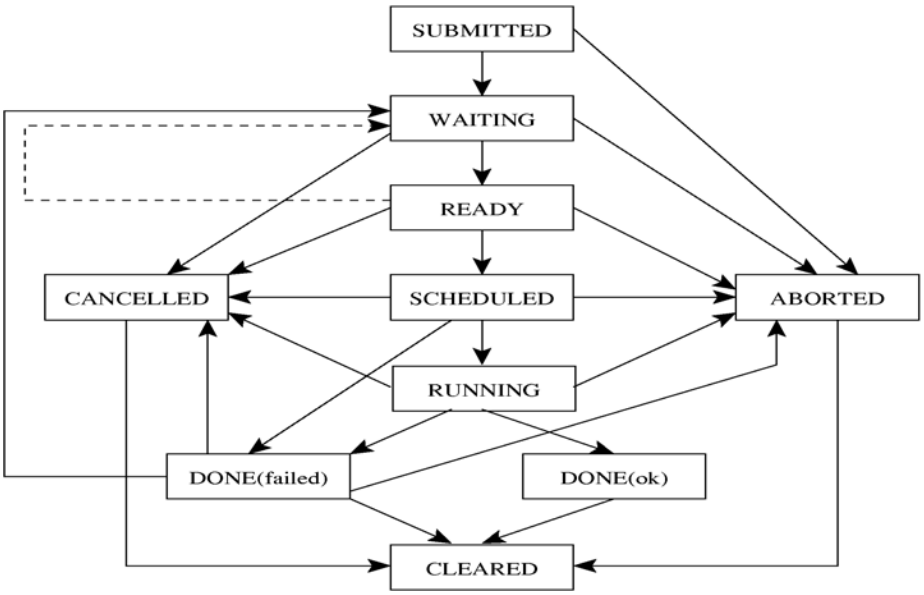


Fig. 2. Possible job states

To Performance Evaluation of DPA in NOW

Peter Hanuliak, Ivan Hanuliak

University of Zilina, Faculty of Control and Informatics, Slovakia
hanuliak@frtk.utc.sk

Abstract. The article is devoted to the problematic of performance evaluation of distributed parallel algorithms (DPA) on the network of workstations (NOW). For this type of parallel computer this article in an illustrative way describes the development of real parallel algorithms for Jacobi iteration. On this individual practical example it is demonstrated the influence of decomposition strategies for performance evaluation of parallel Jacobi iteration and discussed the ways for their parallel implementations.

Introduction

For the contemporary technical and programmable level of the reachable computer means (personal computers, minicomputers, supercomputer etc.) is dominant the using of various forms of the basic principles of the parallel activity. For example in the district of the technical equipment the continuous speeding up of the individual processor performance is achievable mainly through the parallel activity of the pipeline execution in combination with blowing up the capacity and number of various buffer memories (caches).

In the field of programming equipment the parallel support is also in two levels. The one level forms the district of the operation systems and in universal the system supporting programming tools. The other level creates the user developing programming environments, which support the development of the modular application programs as the basic condition to their potential parallel activity. This parallel support goes in this time up to the level of the elementary program elements in the form of the objects (OOP - object oriented programming).

The architectures of the parallel systems

From the point of system classification we can divide all to this time realized parallel systems to the two mutual very different groups [1, 6, 14, 15, 20]:

- Synchronous parallel architectures. To this group we can give practically all known parallel architectures except the computer networks. The basic system properties are given through the existence of some kind of the common shared memory, which in substantial measure simplify their application programming using.

– Asynchronous parallel architectures. This group covers the field of various forms of computer networks. Their basic property is the mutual interconnection as in the remote form with using of the existed telecommunication lines (WAN networks) as in the local form in reaching range of the used fixed lines (LAN networks). There is, in contrast to the first discussed groups, no form of common shared memory in the connected system.

Load balancing, inter-processor communication, and transport protocol for such machines are being widely studied [3, 5, 7, 12, 13, 17 and 18]. With the availability of cheap personal computers, workstations and networking devises, the recent trend is to connect a number of such workstations to solve computation-intensive tasks in parallel on such clusters. To exploit the parallel processing capability of a NOW, the application program must be paralleled. The effective way how to do it for a concrete application problem (decomposition strategy) belongs to a most important step in developing a effective parallel algorithm [5, 7, 15, 16].

The development of the parallel network algorithm includes the following activities:

– Decomposition - the division of the application into a set of parallel processes and data
– Mapping - the way how processes and data are distributed among the nodes
– Inter-process communication - the way of corresponding and synchronization among individual processes
– Tuning - alternation of the working application to improve performance (performance optimization)

When designing a parallel program the description of the high-level algorithm must include in addition to design a sequential program the method you intend to use to break the application into processes and distribute data to different nodes - the decomposition strategy. The chosen decomposition method drives the rest of program development. This is true is in case of developing new application as porting serial code. The decomposition method tells you how to structure the code and data and defines the communication topology.

To choose the best decomposition method for these applications, it is necessary to understand the concrete application problem, the data domain, the used algorithm and the flow of control in given application. Therefore we used according the concrete character of given task the following decomposition models:

– perfectly parallel decomposition
– domain decomposition
– control decomposition
– object-oriented programming OOP

The role of performance

Quantitative evaluation and modeling of hardware and software components of parallel systems are critical for the delivery of high performance. Performance studies apply to initial design phases as well as to procurement, tuning, and capacity planning

analysis. As performance cannot be expressed by quantities independent of the system workload, the quantitative characterization of resource demands of application and of their behavior is an important part of any performance evaluation study. Among the goals of parallel systems performance analysis are to asses the performance of a system or a system component or an application, to investigate the match between requirements and system architecture characteristics, to identify the features that have a significant impact on the application execution time, to predict the performance of a particular application on a given parallel system, to evaluate different structures of parallel applications. To the performance evaluation we briefly review the techniques most commonly adopted for the evaluation of parallel systems and its metrics.

Performance evaluation methods

To the performance evaluation we can use following methods:

- analytical methods
  - Application of queuing theory results [4, 7, 8, 9, 10, 11]
  - Petri nets [7, 15]
- simulation methods [2, 4]
  - Experimental measurement [7, 15, 16]
  - Benchmarks [7, 15]
- direct measuring of concrete developed parallel application [8, 10, 19]

In order to extend the applicability of analytical techniques to the parallel processing domain, various enhancements have been introduced to model phenomena such as simultaneous resource possession, fork and join mechanism, blocking and synchronization. Hybrid modeling techniques allow to model contention both at hardware and software levels by combining approximate solutions and analytical methods. However, the complexity of parallel systems and algorithms limit the applicability of these techniques. Therefore, in spite of its computation and time requirements, simulation is extensively used as it imposes no constraints on modeling.

Evaluating system performance via experimental measurements is a very useful alternative for parallel systems and algorithms. Measurements can be gathered on existing systems by means of benchmark applications that aim at stressing specific aspects of the parallel systems and algorithms. Even though benchmarks can be used in all types of performance studies, their main field of application is competitive procurement and performance assessment of existing systems and algorithms. Parallel benchmarks extend the traditional sequential ones by providing a wider a wider set of suites that exercise each system component targeted workload. The Parkbench suite especially oriented to message passing architectures and the SPLASH suite for shared memory architectures are among the most commonly used benchmarks [7, 15].

Performance evaluation metrics

For evaluating parallel algorithms, there have been developed several fundamental concepts. Tradeoffs among these performance factors are often encountered in real-life applications.

Performance concepts

Let O(s, p) be the total number of unit operations performed by p-processor system for size s of the computational problem and T(s, p) be the execution time in unit time steps. In general, T(s, p) < O(s, p) if more than one operation is performed by p processors per unit time, where $p \geq 2$. Assume T(s, 1)=O(s, 1) in a single-processor system (sequential system). The speedup factor is defined as:

$$S(s,p) = \frac{T(s,1)}{T(s,p)} \tag{1}$$

It is a measure of the speedup factor obtained by given algorithm when p processors are available for the given problem size s. Ideally, since $S(s, p) \leq p$, we would like to design algorithms that achieve $S(s, p) \approx p$.The system efficiency for an p-processor system is defined by:

$$E(s,p) = \frac{S(s,p)}{p} = \frac{T(s,1)}{p\,T(s,p)} \tag{2}$$

A value of E(s, p) approximately equal to 1, for some p, indicates that such a parallel algorithm, using p processors, runs approximately p times faster than it does with one processor (sequential algorithm).

The isoefficiency concept

The workload w of an algorithm often grows in the order O(s), where s is the problem size. Thus, we denote the workload w = w(s) as a function of s. In parallel computing is very useful to define an isoefficiency function relating workload to machine size p needed to obtain a fixed efficiency E when implementing a parallel algorithm on a parallel system. Let h be the total communication overhead involved in the algorithm implementation. This overhead is usually a function of both machine size and problem size, thus denoted h = h(s, p).

The efficiency of a parallel algorithm implemented on a given parallel computer is thus defined as:

$$E(s,p) = \frac{w(s)}{w(s) + h(s,p)} \tag{3}$$

The workload w(s) corresponds to useful computations while the overhead

h(s, n) are useless times attributed to synchronization and data communication delays. In general, the overhead increases with respect to both increasing values of s and p. Thus, the efficiency is always less than 1. The question is hinged on relative growth rates between w(s) and h(s, p).

With a fixed problem size (fixed workload), the efficiency decreases as p increase. The reason is that the overhead h(s, p) increases with p. With a fixed machine size, the overload h grows slower than the workload w. Thus the efficiency increases with increasing problem size for a fixed-size machine. Therefore, one can expect to maintain a constant efficiency if the workload w is allowed to grow properly with increasing machine size.

For a given algorithm, the workload w might need to grow polynomial or exponentially with respect to p in order to maintain a fixed efficiency. Different algorithms may require different workload growth rates to keep the efficiency from dropping, as p is increased. The isoefficiency functions of common parallel algorithms are polynomial functions of p; i. e., they are O(pk) for some $k \geq 1$. The smaller a power of p in the isoefficiency function is, the more scalable the parallel system. Here, the system includes the algorithm and architecture combination.

We can rewrite equation for efficiency E(s, p) as E(s, p)=1/(1=h(s, p)/w(s)). In order to maintain a constant E, the workload w(s) should grow in proportion to the overhead h(s, p). This leads to the following relation:

$$w(s) = \frac{E}{1-E} h(s,p) \tag{4}$$

The factor C = E/1-E is a constant for a fixed efficiency E. thus we can define the isoefficiency function as follows: fE(p)=C. h(s, p). If the workload grows as fast as fE(p) then a constant efficiency can be maintained for a given algorithm-architecture combination.

Grid computation

Partial differential equations (PDE) are used to model a variety of different kinds of physical systems: weather, airflow over a wing, turbulence in fluids, and so one. Some simple PDE´s can be solved directly, but in general it is necessary to approximate the solution at a finite number of points using iterative numerical methods. Here we show how to solve parallel one specific PDE – Laplace´s equation in two dimensions – by means of a grid computation method that employs what is called a finite- difference method. Although we focus on this specific problem, the same programming techniques are used in grid computations for solving other PDE´s and in other applications such as image processing etc.

Laplace´s Equation

Laplace´ equation is a practical example of Jacobi iteration application. The equation for two dimensions is the following:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \quad (5)$$

Function Φ represents some unknown potential, such as heat or stress.

Given a spatial region and values for points on the boundaries of the region, the goal is to approximate the steady – state solutions for points in the interior. We can do this by covering the region with an evenly spaced grid of points, as shown in Fig. Each interior point is initialized to some value. The steady – state values of the interior points are then calculated by repeated iterations. On each iteration the new value of a point is set to a combination of the old/or new values of neighboring points. The computation terminates either after a given number of iterations or when every new value is within some acceptable difference EPSILON of every old value.

There are several iterative methods for solving Laplace´s equation, including Jacobi iteration, Gauss – Seidel, successive over-relaxation (SOR), and multigrid. In this paper we show how parallel implementation of Jacobi iteration using message passing interface (MPI). The algorithms for other methods converge more rapidly but are somewhat more complex than Jacobi iteration. Their parallel implementations have similar communication and synchronization patters and these aspects are the most important.

Given a spatial region and values for points on the boundaries of the region, the goal is to approximate the steady – state solutions for points in the interior. We can do this by covering the region with an evenly spaced grid of points, as shown in Fig. Each interior point is initialised to some value. The steady – state values of the interior points are then calculated by repeated iterations. On each iteration the new value of a point is set to a combination of the old/or new values of neighbouring points. The computation terminates either after a given number of iterations or when every new value is within some acceptable difference EPSILON of every old value.

There are several iterative methods for solving Laplace´s equation, including Jacobi iteration, Gauss – Seidel, successive over-relaxation (SOR), and multigrid. In this paper we show how parallel implementation of Jacobi iteration using message passing interface (MPI). The algorithms for other methods converge more rapidly but are somewhat more complex than Jacobi iteration. Their parallel implementations have similar communication and synchronization patters and these aspects are the most important.

Jacobi iteration

We applied Jacobi point iterative method to the grid (1, 0) x (1, 0) for which the boundary conditions are known. N is the number of interior grid points.

$U_{i-1,j} + U_{i,j-1} - 4*U_{i,j} + U_{i+1,j} + U_{i,j+1} = 0;$ $\quad i, j = 1, 2, \ldots N;$

$U_{0,j} = - y_{j2} ;$ $\quad U_{N+1,j} = 1 - y_{j2} ;$ $\quad j = 1, 2, \ldots N;$

$U_{i,0} = - x_{i2} ;$ $\quad U_{i,N+1} = 1 - x_{i2} ;$ $\quad i = 1, 2, \ldots N;$

where $x_i = i / (N+1);$ $\quad y_j = j / (N+1;)$

In this case the exact solution is known:

$$U_{i,j} = x_{i2} - y_{j2}$$

Conclusion and results

To performance evaluation we used two decomposition strategies in order to analyze their influence:

- Decomposition of Jacobi iteration according to Fig. 4 (domain decomposition 1). In this strategy there were giving each computation node a vertical strip of U_{ij}. After each iteration „boundary conditions“ have to be shared with neighboring nodes (Fig. 1.)|
- Decomposition according Fig. 2 (domain decomposition 2). In this strategy we used twice much computation nodes as in the first case.

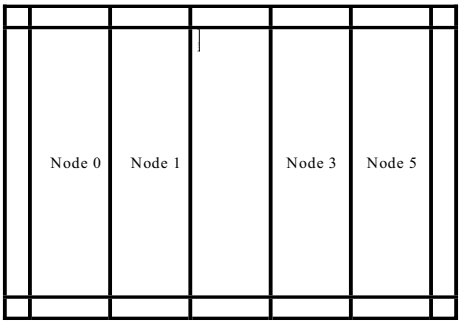


Fig. 1. Domain decomposition 1

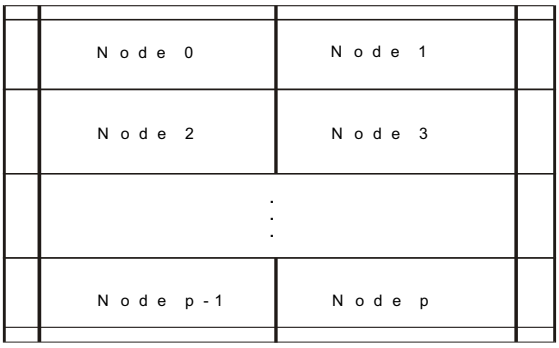


Fig. 2. Domain decomposition 2

On the Fig. 4 we illustrated the performance comparison of both used decomposition strategies. We can see that in using more computation processors we did not come to increased performance. The causes are in increasing the overheads more (control, communication, synchronization) than the speed-up of higher computation nodes. The realized experiments were done on parallel system at EPCC Edinburgh (parallel system Cray T3E).

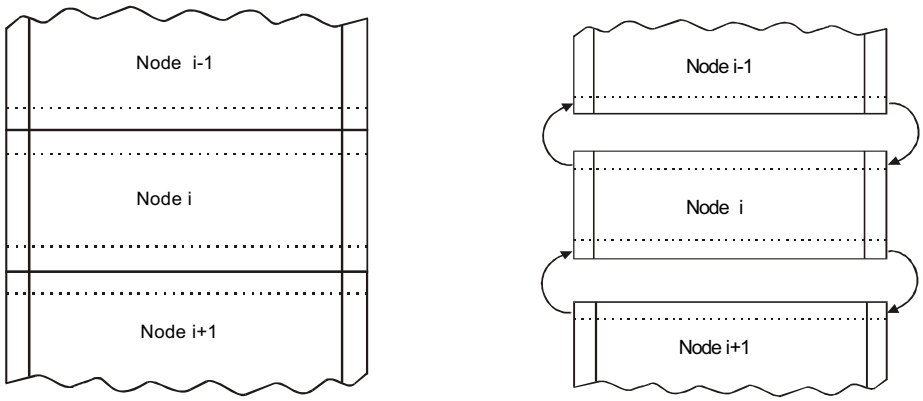


Fig. 3. The shared points and the way of their exchange

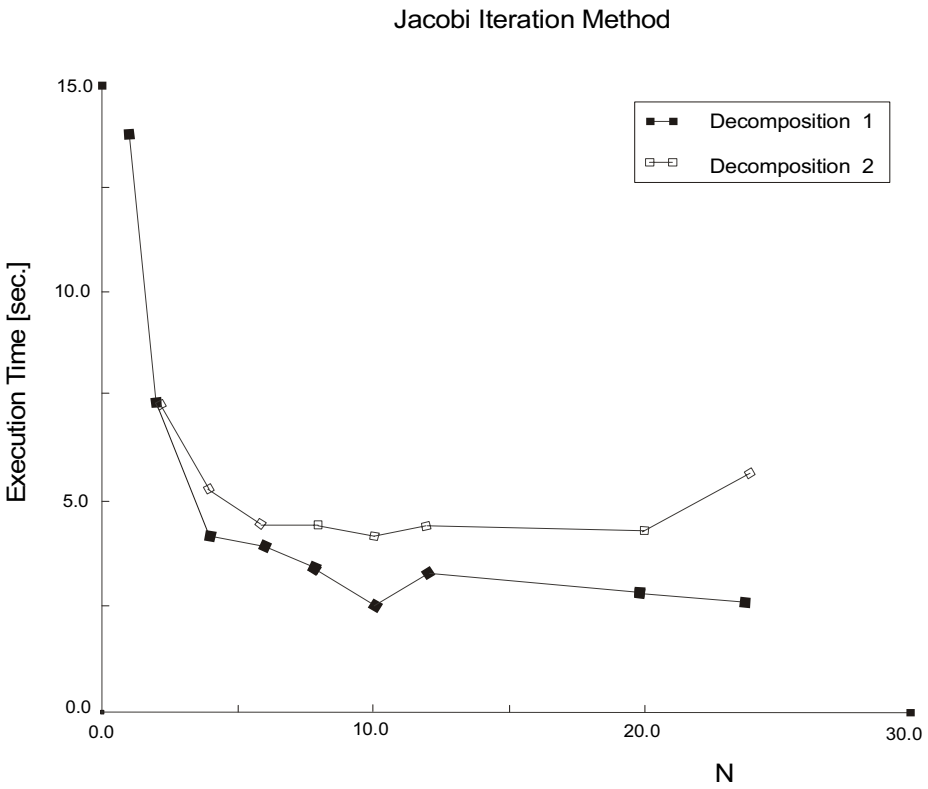


Fig. 4. The comparison of used decomposition methods

References

1. Andrews G. R., Foundations of Multithreaded, Parallel, and Distributed Programming, Addison Wesley Longman, Inc., 664 pp., USA, (2000)
2. Banks J., Dai J. G.: Simulation studies of multiclass queueing networks, IEEE Transactions, Volume 29, 1997, (1997) 213 – 219
3. Basoglu Ch., Lee W., Kim Y.: An efficient FFT algorithm for Super-scalar and VLIW Processor Architectures, Real Time Imaging 3, USA (1997) 441-453
4. Fodor G., Blaabjerg S., Andersen A.: Modelling and simulation of mixed queueing and loss systems, Wireless Personal Communication, N. 8, (1998) 253 - 276
5. Greenberg D. S., Park J. K., Schvabe E. J.: The cost of complex communication on simple networks, Journal of Parallel and Distributed Computing 35, (1996) 133-141
6. Hanuliak I., Parallel architectures – multiprocessors, computer networks (in Slovak), 187 pp., Book centre, Zilina (1997)
7. Hanuliak I.: Parallel computers and algorithms (in Slovak), Košice, ELFA Press, 327 pp. , (1999)

8. Hanuliak I., Hanuliak J., To performance evaluation of distributed parallel algorithms, Kybernetes, The International Journal of Systems & Cybernetics, West Yorkshire, Volume 34, No. 9/10, United Kingdom, (2005) 1633-1650
9. Hanuliak J., Hanuliak M., Analytical modelling of distributed computer systems, NOW, In Proc.: TRANSCOM 2005 (6-th Europ. Conf. in Transport and Telecommun.), , Zilina, (2005), 103-10
10. Hanuliak J., To performance evaluation of parallel algorithms in NOW, Communications, 4/2003, The University of Zilina, Slovakia, (2003), 83 – 88
11. Hanuliak M., Analytical models of distributed computer based communication networks, in Proceedings of 8th International Conference ", May 24-26, TÂRGU - JIU, Romania, (2002), 1 - 6
12. Harrison P. G., Patel N.: Performance modelling of communication networks and computer architectures, Addison - Wesley Publishers, 480 pp., (1993)
13. Hsu W. T., Pen-Chung Y.: Performance Evaluation of Wire-Limited Hierarchical Networks, Parallel and Distributed Computing 41, pp. 156 - 172 , (1997)
14. Hesham EL-Rewini, Ted. G. Lewis, Distributed and parallel computing, 467 pp., Manning Publications Co., USA, (1997)
15. Hwang K., Xu Z., Scalable Parallel Computing: Technology, Architecture, Programming, Mc Graw-Hill Companies, 802 pp., USA, (1998)
16. Kumar V., Grama A., Gupta A., Karypis G., Introduction to Parallel Computing (Second Edition), Addison Wesley, 856 pp., (2001)_
17. Marinescu D. C., Rice J. R.: On the scalability of asynchronous parallel computations, Parallel and Distributed Computing 31, (1995) 88-97
18. Nancy A. L., Distributed Algorithms, 872 pp., Morgan Kaufmann Publishers, Inc., USA, (1996)
19. Varsa P., Hanuliak P., Grid implementation of parallel algorithm for Laplacean equation computation by Jacobi iteration method, Communications 4/2003, The University of Zilina, pp. 36 – 41, (2003)
20. Williams R., Computer Systems Architecture – A networking approach, Addison Wesley, 660 pp., England, (2001)

Paper contributions

Supercomputers applications in quantum chemistry

Martin Breza

Department of Physical Chemistry, Slovak Technical University,
SK-81237 Bratislava, Slovakia
breza@cvt.stuba.sk

Abstract. Quantum-chemical calculations are based on solving a great number of integro-differential equations. They demand multiple operations with huge matrices and additional space for the storage of huge number of the integrals evaluated in each step of the iterative SCF procedure to obtain the energy and electronic structure for the given geometry and electronic state of the molecular system under study. In the next steps, geometry optimization (based on energy gradients) and/or physico-chemical properties evaluation follows. Author's experience with large molecular systems calculated using various quantum-chemical software packages on some supercomputers is presented.

1 Introduction

The fundamental postulate of quantum chemistry is Schrodinger equation

$$H\Psi = E\Psi \tag{6}$$

where H is Hamilton operator *(*composed of operators of kinetic energy of nuclei, T_N, and electrons, T_e, electron-nuclear attraction, V_{eN}, and mutual repulsion of nuclei, V_{NN}, and electrons, V_{ee}) and E is the energy of the state described by the wave function Ψ (with nuclear end electronic space and spin coordinates as parameters).
In Born-Oppenheimer approximation the nuclear positions are fixed and electronic and nuclear wavefunctions may be treated separately. Electronic wavefunction for *n* electrons is used in the form of Slater determinant composed of one-electron functions (spinorbitals).

$$\Psi(1,2,3,...n) = \frac{1}{\sqrt{n!}} \begin{Vmatrix} \Psi_1(1) & \Psi_1(2) & \cdots & \Psi_1(n) \\ \Psi_2(1) & \Psi_2(2) & \cdots & \Psi_2(n) \\ \vdots & \vdots & & \vdots \\ \Psi_n(1) & \Psi_n(2) & & \Psi_n(n) \end{Vmatrix} \tag{2}$$

Electronic structure of molecules is described by molecular orbitals φ_i (basis functions) as linear combination of atomic orbitals (LCAO approximation) which are treated as the sum of several Slater (~ $\exp(-\alpha r)$, r being the interelectronic distance) or Gauss (~ $\exp-(\alpha r^2)$) functions χ_μ.

$$\varphi_i = \sum_{v}^{m} c_{iv}\chi_v. \tag{3}$$

In Hartree-Fock treatment, the values of LCAO coefficients $c_{i\mu}$ and molecular orbital energies ε_i are obtained from Roothaan equations (4) – (10) that are solved iteratively by self-consistent field procedure (SCF)

$$FC = SCE \tag{4}$$

where F (Fock matrix), C (LCAO coefficients matrix) and S (overlap matrix) are square matrices with dimensions given by the number of molecular orbitals m

$$F_{\mu\nu} = H_{\mu\nu} + G_{\mu\nu} \tag{5}$$

$$H_{\mu\nu} = \int \chi_\mu(1) h(1) \chi_\nu d\nu_1 \tag{6}$$

$$G_{\mu\nu} = \sum_{\lambda\rho} P_{\lambda\rho}\left[\langle\mu\nu|\lambda\rho\rangle - \frac{1}{2}\langle\mu\lambda|\nu\rho\rangle\right] \tag{7}$$

$$P_{\lambda\rho} = 2\sum_{i=1}^{n} c_{\lambda i} c_{\rho i} \tag{8}$$

$\langle\mu\nu|\lambda\rho\rangle$ denotes two-electron integrals (their number is proportional to m^4)

$$\langle\mu\nu|\lambda\rho\rangle = \iint \chi_\mu(1)\chi_\nu(1)\frac{e^2}{r_{12}}\chi_\lambda(2)\chi_\rho(2) d\nu_1 d\nu_2 \tag{9}$$

$$S_{\mu\nu} = \int \chi_\mu(1)\chi_\nu(1) d\nu_1 \tag{10}$$

The above mentioned Hartree-Fock treatment using single Slater determinant functions (2) is insufficient for more accurate calculations and configuration interaction (CI) using several Slater determinants must be used. The resulting formulas are very complicated and the related calculations are very time-consuming. This implies the use of simplified CI treatments based on perturbation theory (such as MP2) or modified Hamiltonian (such as density functional theory, DFT).

Energy hypersurface is a parametric function of nuclear coordinates and electronic state of a molecular system. Its minima correspond to stable conformations. Geometry optimization is usually performed by various gradient methods what demands additional evaluation of integrals (9) derivatives. Second Cartesian derivatives of energy (Hessian matrix) are necessary for the energy minima verification as well as for vibration spectra calculations. Electron spectra calculations are based on transitions between the electronic ground and excited states. Other physico-chemical properties of molecular systems may be evaluated as well.

In molecular dynamics calculations, the chemical reactions are modeled by molecules/ions in motion and their collisions are statistically evaluated for large number of starting geometries and velocities. These type calculations are used for small molecular systems or simplified methods of quantum chemistry.

2 Calculations

There are various program packages for quantum-chemical calculations. They differ mainly by the operation memory and integrals storage management, in the details of SCF procedure and geometry optimization, in the extent of physico-chemical properties evaluated etc. The author's experience with several program packages at some supercomputers of Table 1 is summarized in the next section.

Table 1. The list of supercomputers and software used

Computer	Location	Software package	Ref.
SGI Origin 2000	CCR of SUNY at Buffalo, USA	ADF2002	[2]
NEC SX-6	HLRS Stuttgart, Germany	Gaussian98	[3]
IBM p690-Cluster Jump	NIC Juelich, Germany	Gaussian03	[4]

3 Results

The size of the matrices (4) is given by the number of basis functions (molecular orbitals) and depends on the number and type of atoms in the molecular system as well as on the quality of atomic basis sets (i.e. the number of components of individual atomic orbital – primitive functions). Small basis sets and/or oversimplified quantum-chemical methods produce incorrect results. Large matrices (4), high (negative) charges, transition metal atoms, higher spin states and inadequate starts for geometry optimization may cause significant problems in SCF procedure. This implies the necessity of SCF damping and additional CPU time for calculations. Additional problems arise due to incomplete code parallelization. Some operation systems check the use of the allocated processors and in the case of their ineffective use the calculations are stopped. The large molecular systems in Table 2 have been optimized by MP2 or less exact DFT methods. On the other hand, DFT treatments enable use of larger basis sets and are less time-consuming. The systems in Table 2 demand 1-3 GB operation memory and 2-20 GB external space for integrals storage. Larger operation memory enables lower external space demands and significantly accelerates the calculations.

Another problem is connected with the administrative rules for users. Despite the quantum-chemical programs enable restarting the interrupted calculations; there are some minimal time demands on the program run. Large molecular systems demand sometimes several weeks (or months) of CPU time and thus the minimal CPU time demanded may be 1-2 days. Thus if the maximal CPU time for external users is restricted to several hours, the computer is unusable for our purposes. Moreover, multiple restarts decrease the effectiveness of the calculations.

Finally it may be concluded that high-level quantum-chemical calculations of large molecular systems demand large operation memory, large temporary files for integral storage and very long CPU time despite multiple processors use. Powerful

supercomputers use is necessary for precise quantum-chemical study of large molecular systems of chemical interest.

Table 2. Characteristics of the molecular systems calculated

Molecular system[a)]	Method	Basis functions	Primitive gaussians	Ref.
$^1[C_{28}H_{22}CuN_2S_2]^+$	DFT	651	-	[5]
$^2[C_{28}H_{22}CuN_2S_2]^{2+}$	DFT	651	-	[5]
$^1[C_{28}H_{22}CuN_2S_2]^+$	MP2	527	1126	
$^2[C_{28}H_{22}CuN_2S_2]^{2+}$	MP2	527	1126	
$^1[C_{13}H_{13}CuN_3O_3]^-$	DFT	347	688	[6]
$^2[C_{13}H_{13}CuN_3O_3]^0$	DFT	347	688	[6]
$^1[C_{13}H_{13}CuN_3O_3]^+$	DFT	347	688	[6]
$^3[C_{13}H_{13}CuN_3O_3]^+$	DFT	347	688	[6]
$^1[C_{13}H_{14}CuN_3O_3]^0$	DFT	349	692	[7]
$^1[C_{20}H_{28}Cl_4Ru_2]^0$	DFT	610	1164	
$^1[C_{16}H_{19}Cl_2Ru]^-$	DFT	414	785	
$^1[C_{21}H_{27}ClORu]^0$	DFT	520	957	
$^1[C_{28}H_{39}O_2Ru]^+$	DFT	674	1213	
$^1[C_{27}H_{37}O_2Ru]^+$	DFT	650	1171	
$^1[C_{27}H_{35}ORu]^+$	DFT	626	1129	
$^1[C_{20}H_{11}F_5N_2]^0$	MP2	579	952	[8]

Remarks:
[a)] left upper indices denote spin multiplicity, right upper indices are total charges

Acknowledgements

This work was supported by National Science Foundation, USA (project No. NSF-CHE9981864), EC-funded project HPC-Europa (contract No. 506079), NIC Juelich (project ESK00), Science and Technology Assistance Agency (contract No. APVT-20-004504) and Slovak Grant Agency VEGA (contract No. 1/0052/03).

References

1. Remko, M.: Molekulové modelovanie. Princípy a aplikácie. Slovak Academic Press, Bratislava (2000)
2. (a) te Velde, G., Bickelhaupt, F.M., van Gisbergen, S.J.A., Fonseca Guerra, C., Baerends, E.J., Snijders, J.G., Ziegler, T.,: J. Comput. Chem. 22 (2001) 931;
   (b) Program ADF2002.01, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, http://www.scm.com
3. Frisch, M. J. et al: Gaussian 98, Revision A.9, Gaussian, Inc., Pittsburgh PA (1998)
4. Frisch, M.J. et al: Gaussian 03, Revision C.1, Gaussian, Inc., Pittsburgh PA (2003)
5. (a) Rapta, P., Kožíšek, J., Breza, M., Gembický, M., Dunsch, L.: J. Electroanalyt. Chem. 566 (2004) 123;
   (b) Breza, M., Kožíšek, J.: Polyhedron, in press
6. Breza, M., Biskupič, S.: J. Mol. Struct. (THEOCHEM), in press
7. Breza, M., Biskupič, S.: Polyhedron, in press
8. Breza, M. : J. Mol. Struct. (THEOCHEM), in press

Simulation of Data Flow Architecture in Parallel environment

Marek Straka, Branislav Sobota

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics, Technical University of Košice
Letná 9, 040 01 Košice, The Slovak Republic,
Marek.Straka@tuke.sk, Branislav.Sobota@tuke.sk

Abstract. The paper describes simulation of the parallel data flow architecture DF KPI. This is designed on the department of computers and informatics FEI TU Košice. This contribution is divided to three parts. In the first part is described architecture of DF KPI. Next part describes some problems with programs writing (DFG) for data flow architecture. Third part describes simulation tool, which was developed for simulation of parallel architecture DF KPI. Additionally in this part is also described migration the simulator to the grid or cluster architecture.

1 Introduction

Data flow architecture is parallel architecture, where computation is controlled by flow of data. In a true data-flow implementation, all modules are pure functions (i.e., their outputs are fully defined by their inputs). Hence, processes are stateless with no side-effects. These properties enable detection of parallelism on the lowest level (fine – grained parallelism). On the department of Computers and informatics was developed Data Flow architecture DF KPI. DF architectures are usually joined with functional languages [2]. Functional languages are declarative languages and use declarative statements in program. A functional program is a single expression executed by evaluating the expression. Anyone who has used a spreadsheet has experience of functional programming. In a spreadsheet, one specifies the value of each cell in terms of the values of other cells. The focus is on what is to be computed, not how it should be computed. The same case is in depth algorithm in computer graphics for visibility problem solving [6]. It is necessary to compute and compare z-distance of every object (faces of object) from camera, but the final output frame isn't depended from the order of computation objects.

1.1 Architecture DF KPI

Structural model of DF KPI architecture is on Fig.1 Architecture is build from these components:

CP – *Coordinating Processors* are used for control, coordination and execution of instructions of DF program. Internal structure of coordinating processor is designed like dynamic multifunctional system from five segments:
– LOAD,
– FETCH,
– OPERATE,
– MATCHING,
– COPY.
DQU – *Data Queue Unit* is unit for storing data token (DT) which represent waiting operands in computation process of running program.
IS – *Instruction Store* is memory of instructions (DFI) of DF program like data flow graph (DFG),
FS – *Frame Store* is memory of matching vectors, which are used by CP for execution of operators which have more that one input operand and for storing of structural data.
IN – *Interconnection Network* is used for connection between all CPs.

Support components of DF system are necessary for preparation of real computationally environment. Support components are:
HOST – Main computer for support of standard functions of computers system
IT – *Information Technology* unit for special application (virtual reality, diagnostic, e-learning)
S I/O – *Special I/O* processors for quick direct input and output (standard I/O support main computer).

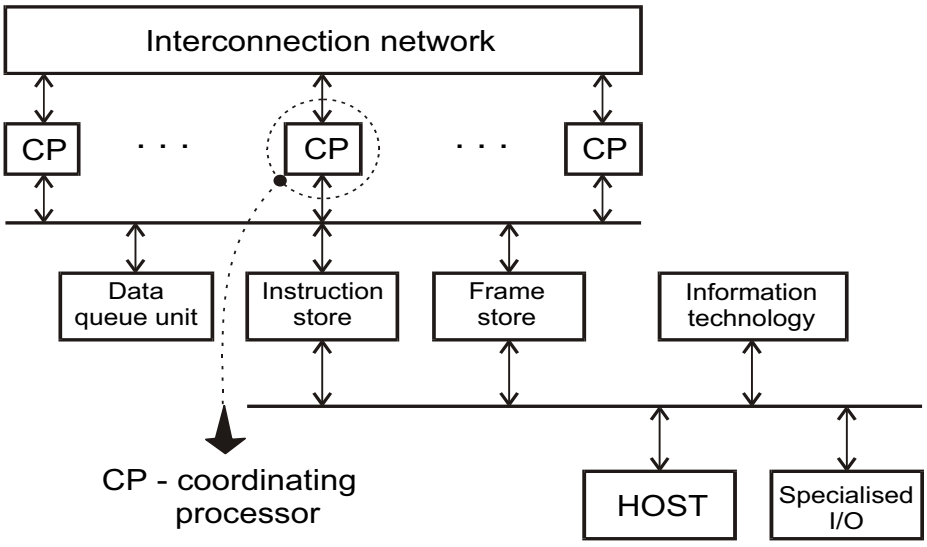


Fig. 1. Structure of DF KPI architecture

2 Programming of DF KPI Architecture

Program writing for DF KPI architecture is process of data flow graph (DFG) building [5]. One node in DFG represents operator which can have one or more input operands (input data tokens) and generates one or more copies of output data tokens. It depends on the type of operator. More details about DFG in [1]. Direct writing of DFG is hard task therefore is used some standard functional language for description of program and then is functional program convert to DFG graph. Functional language Haskell [3] [4], which is used all over the world, is used like intermediate language between description of the problem and the final DFG. Simple example of calculation program for normalization of 3D vector in Haskell is bellow and also equivalent data flow graph to this program is on Fig.2

```
normalize :: Vector -> Vector
normalize v =  ( Vector x y z)
where
          dist = lenght v
          x =  vector_X v /dist
          y =  vector_Y v /dist
            z =  vector_Z v /dist
```

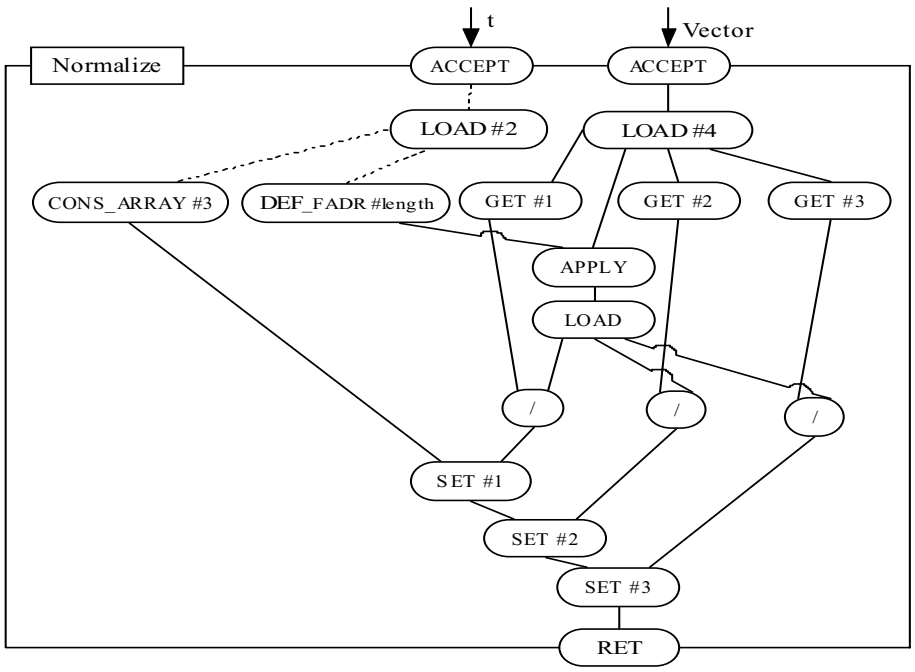


Fig. 2. Data flow graph of function normalize

3 Simulation of DF KPI architecture

It is very expensive solution to create real prototype of DF KPI architecture. It is better to create simulator for testing of designed architecture and application programs. Simulator of DF KPI architecture has to simulate all components, which were described in chapter 1.1. The simplest solution is simulation on single PC. This simulator was developed on the department of computers and informatics. It simulates unbounded count of coordinating processors in data flow architecture. Single PC version doesn't need some special hardware besides the ordinary PC. The part of this simulator is also development environment for DFG programs design. Developed programs can be saved in DF KPI architecture format. Important part of program is compiler with syntax and semantic control. Created program is possible to execute in environment of this simulator. Disadvantage of single computer version is in semi parallel simulation of execution and communication between coordinating processors. Example of execution environment is shown on Fig.3.

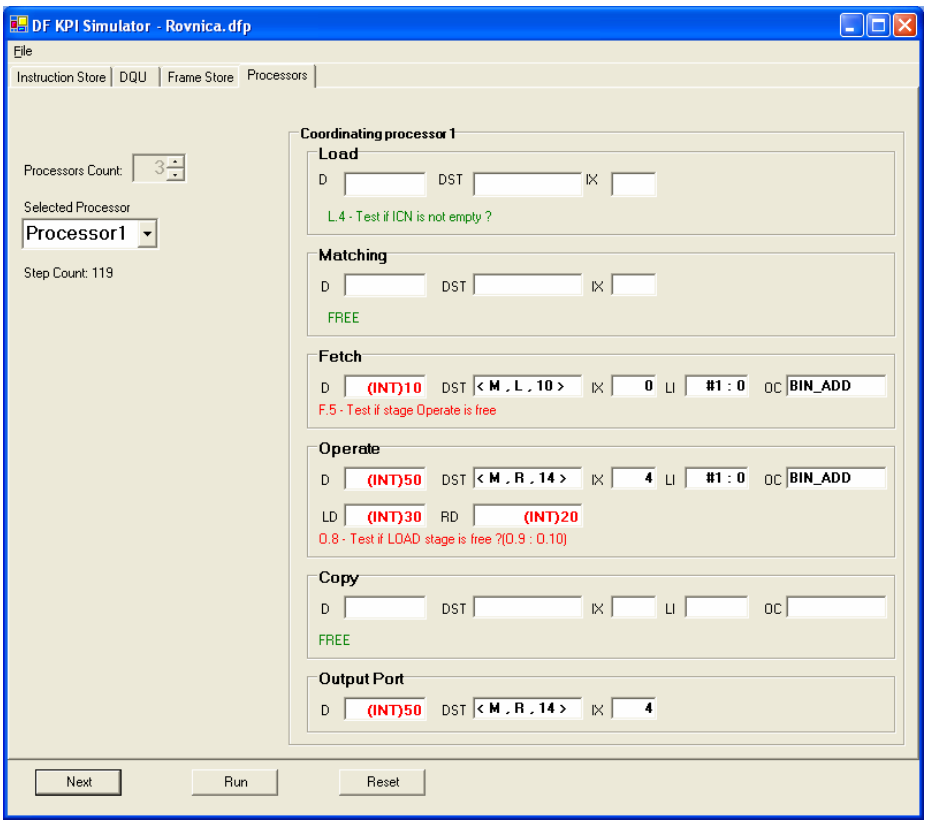


Fig. 3. Screenshot of DF KPI simulator

3.1 Migration to parallel environment

The precise simulator it is possible to obtain by using:
- Grid solution
- Cluster solution
- Tightly coupled multiprocessor computer

Used environment depends on the requirement from the simulation. If it is necessary to measure only effectiveness of used coordinating processors it can be used the grid architecture for simulation. If it is necessary to measure real time execution, than better solution is cluster of PCs or multiprocessor computer using. Based on Fig.1 migration to the parallel environment can be divided to two levels:
1. migration of simulation module of *Coordinating Processor* (CP)
2. development of simulation module of *Interconnection Network* (IN)

Coordinating processor can by migrate to grid architecture with minimal changes. When CP obtains data token (DT) on the input port, the token must cross all necessary segments the same way like in single PC version. Output data token can be send:
- again on the input of the same coordinating processor, if is free
- through the interconnected network to the neighbor coordinating processor, if is free
- through the interconnected network to the data queue unit (DQU)

Interconnected network is represented in grid solution by Ethernet network. Ethernet network doesn't allow real time simulation, but it can be used for communication cost measuring of tested DF application and effectiveness of used coordinating processors. Through the running simulation simulator can gathering statistic information about weak places in the DF KPI architecture (Fig. 4). Statistic information's contains relative time spend in current segment of current coordinating processor.

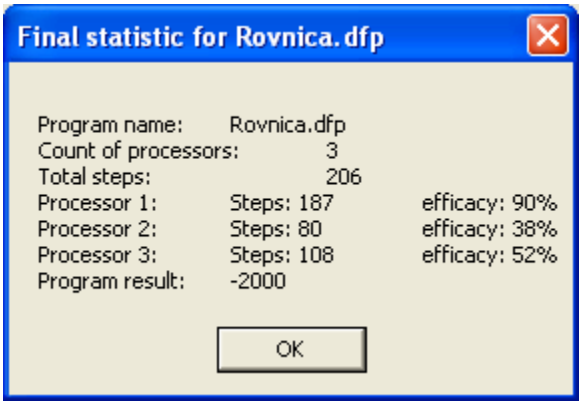


Fig. 4. Example of final statistic dialog

4 Conclusions

Grid solution enables much precise measure of communication traffic between coordinating processors and with other components of DF KPI architecture. It is possible to compare time spend in communication component and in computation parts. First results of simulation argue, that communication is slowest part of architecture and it is important to design interconnection network carefully to whole architecture. This contribution was written under the project 1/2174/05 Research of high-performance parallel environments to solve computational processes in specialized application areas: theory, models, simulation, assessment and applications.

References

1. Jelšina, M. a kol.: Architektúra dataflow KPI, Výskumná správa, TU Košice, 2003.
2. Hankin Ch.: An Introduction to Lambda Calculi for Computer Scientists, King's College Publications, 2004, ISBN 0-9543006-5-3, Vol.2
3. Doets K.,Eijck J.: The Haskell Road to Logic , Maths and Programming, King's College Publications, 2004, ISBN 0-9543006-9-6, Vol.4
4. Hudak P., Peterson J.: A Gentle Introduction to Haskell 98, University of California, 1999
5. Davis, A.I.,Keller,R.M.:Data Flow Program Graphs, IEEE Computer, Vol.15, 1982, pp.26-41.
6. Sobota Branislav, Straka Marek: Visualising Kernel of Virtual Reality System and its Rendering Part, Proceedings of the Fifth International Scientific Conference „Electronics Computers and Informatics´2002", Košice - Herľany, 10.-11.10.2002, Košice, KPI FEI TU Košice, 2002, pp. 323-327, ISBN 80-7099-879-2 http://hornad.fei.tuke.sk:8180/epc/lists/htmlabstract.jsp?id_publication=170&language=sk

Job management in the LCG-type Grid

Miroslav Dobrucký

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
miroslav.dobrucky@savba.sk

Abstract. Secure co-existence of many users' jobs in the Grid requires usage of certificates that allows also assigning different roles and priorities. Firstly, certificate management applicable in most of present-day Gird testbeds is introduced, and then a job preparation is described, with secure file transfer, job submission specifications and usage of MyProxy technique. Finally, a demo will show our FloodGrid application developed within the CrossGrid project using LCG middleware, now being ported into EGEE testbedthat is operating in LCG but now starting to use its own gLite middleware.
Keywords: grid, LCG middleware, X.509 certificates, MyProxy

Introduction

In the recent years, Grid computing has been rapidly growing and many scientists from different areas start using the huge computational power provided by Grid computing. According to Ian Foster's original definition, *"a computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities"*. The idea was that people could access computational power, content, and other computer services in an easy way, just like using electricity by plugging a device into a wall socket. Realizing the idea is a challenging work. Grid consists of many computational resources from different organizations, geographically distributed and dynamically linked via the Internet. Problems like security, reliability, performance need to be thoroughly solved in order to make grid computing usable. Today, grid computing is "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations" where multiple organizations collaborate with each other, sharing their resources to solve common problems.

There have been many international projects in Grid computing like Data-Grid, GridLab, CrossGrid [1], EGEE [2] etc. Grid computing is also funded on national level like HellasGrid, UK National e-science etc. In Slovakia, we are trying to establish such national grid initiative [4].

How do I login on the Grid?

To maintain many users in large grid testbeds with distribution of resources, AAA (*Authentification and Authorization*) should be of higher level for secure access and therefore X.509 certification system (PKI) can be used:
- Secure communication (SSL, SSH keys, Kerberos)
- Security across organizational boundaries (PKI, RSA)
- Single 'sing-on' for users of the Grid (proxy certificates)

Two basic concepts are used:
- Authentication = "*Who am I*?" is equivalent to ID card, passport and is realized by Certificates, and
- Authorization: "*What can I do*?" means certain permissions, duties, etc. and is realized through Virtual organizations membership and by systems like ACL (access control list) or VOMS (virtual organization membership service).

Certificates

Each user must have a valid X.509 certificate issued by a recognized Certification Authority (CA). Before doing any Grid operation, user must log in User Interface (UI) machine where his personal private key is securely stored and here he/she creates a proxy certificate. A proxy certificate is a delegated user credential that authenticates the user in every secure interaction, and has a limited lifetime: in fact, it prevents having to use one's own certificate, which could compromise its safety.

Creating the key/request pair

The user enters "*grid-cert-request*" via command line on the User Interface (UI) machine by entering his full name and choosing a passphrase for private key. Other values in user's certificate requests are pre-filled done by configuration of UI machine.

```
%[miro@cluster2 miro]$ grid-cert-request
%Enter your name, e.g., John Smith: Miroslav Dobrucky
%A certificate request and private key is being
created.
%You will be asked to enter a PEM pass phrase.
%This pass phrase is akin to your account password,
%and is used to protect your key file.
%If you forget your pass phrase, you will need to
%obtain a new certificate.
%Using configuration from /etc/grid-security/globus-
user-ssl.conf
%Generating a 1024 bit RSA private key
%.....................++++++
```

```
%.........................++++++
%writing new private key to
/home/miro/.globus/userkey.pem'
%Enter PEM pass phrase: ****************
```

Getting certificate

The user should securely (on diskette, CD or USB Flash memory stick) deliver his/her request to the relevant Registration or Certification Authority (RA or CA) and personally authenticate by his/her ID card, passport or similar official document. The user must accept CP&CPS of CA [3], for example there is minimal mandatory length of the key passphrase and user is obliged to keep his private key in secure place and report immediately in case of security breach or steal accident.

```
%[miro@cluster2 miro]$ cat
home/miro/.globus/usercert_request.pem \
% | mail ca.ui@savba.sk
```

Creating a proxy

The user generates a proxy certificate - his/her delegated credential that authenticates the user in every secure interaction, and has a limited lifetime:

```
[miro@cluster2 miro]$ grid-proxy-init
Your identity: /C=SK/O=SlovakGrid/O=IISAS/CN=Miroslav
Dobrucky
Enter GRID pass phrase for this identity:
Creating proxy ....................................
Done
Your proxy is valid until: Fri Nov 25 12:37:05 2005
```

This proxy has both private and public parts of a key and therefore should be maintained in secure environment - for large grid infrastructure there is potential high risk for violation and therefore the lifetime of proxy is kept very short, e.g. 12 hours or less. Other useful commands for managing proxies:

- grid-proxy-info
- grid-proxy-destroy

Job preparation and submission

Before a user submits his/her job to Resource Broker (RB), a user has to create a file describing the submitting job in Job Description Language (JDL), in which he/she specifies:

- what program (will be submitted into insecure world)
- what data files (or replicas will be used)

– what are the requirements on OS, libraries, ...

There are following job types supported:
– simple sequential,
– parallel (MPICH P4, MPICH-G2),
– interactive (StdOut StdErr on-line delivered, X-windows or other -MD, VNC) and
– checkpointable (planned in the future).

Job is submitted by command *edg-job-submit some_file.jdl* and a job identifier is returned, for example *https://rb01.lip.pt:9000/vrTuD4Tm-rQZvtdMma0a7w*, that can be written in a file (using *option -o*) to be later used in querying the job status by command *edg-job-status -i job_id.file*.

When the status is "Done", the job's outputs can be transferred (through sandboxes technique) by command *edg-job-get-output -i job_id.file*. Output files which names were specified in JDL file will be found in the folder specified in this command's output text.

Instead of submitting a job, the user may wish to know available resources that will be used for his job listing suitable CEs by command *edg-job-list-match some_file.jdl*.

The user can bypass the resource broker for jobs, whose binary executable is present on target machine, for example "*ls*" command: *globus-job-run cluster.somewhere.com /bin/ls*

Other possible specifications in JDL:
– Environment variables
– Input data (PFN and/or LFN)
– What data access protocol
– Output data (to which SE, registering)
– Rank (preference: freeCPU, response time, random)
– Requirements (maxCPUtime, maxWallClock, *dom\'ena,
– testbed, estimated response time, freeCPU$>$x, totalCPU$>$x, ...) …

An example of the simple JDL File:

```
miro@cluster2 miro]$ cat hostname.jdl
Executable = "/bin/hostname";
StdOutput = "std.out";
StdError = "std.err";
OutputSandbox = {"std.out","std.err"};
```

Secure file transfer

In LCG middleware, GridFTP service can be used for transferring input/output files (data and executable binary) in secure mode using proxy certificates. Third party transfer can be also used. Full path to the file, together with accessing mode should be specified: *globus-url-copy command*

```
[miro@cluster2 miro]$ globus-url-copy \
gsiftp://storage.ui.sav.sk/home/miro/grid-
data/Malpasset.2dm \
file://$PWD/Malpasset.2dm
```

Other useful commands:
- edg-gridftp-ls
- edg-gridftp-exist
- edg-gridftp-mkdir
- edg-gridftp-rmdir
- edg-gridftp-rm
- edg-gridftp-rename

MyProxy technique

Myproxy server is useful for:
- Very long jobs (ordinary proxy would expire), proxy is automatically renewed in the job runtime.
- Getting proxy on other machines than UI (typical for portals) or working in less-secure environment (internet cafe).

Proxies with short lifetime will be issued by *MyProxy* server, to which a user stores his "*myproxy*" delegate with medium lifetime, for example for one week. This "*myproxy*" is secured by another passphrase, other than user's own personal certificate passphrase:

```
[miro@cluster2 testgrid]$ myproxy-init -s
myproxy.server.com
Your identity: /C=SK/O=SlovakGrid/O=IISAS/CN=Miroslav
Dobrucky
Enter GRID pass phrase for this identity:
Creating proxy ................ Done Proxy Verify OK
Your proxy is valid until: Fri Nov 25 15:25:40 2005
Enter MyProxy pass phrase:
Verifying password - Enter MyProxy pass phrase:
A proxy valid for 168 hours
```

Then user or LCG middleware can get the short lifetime proxy issuing the command:

```
[miro@cluster miro]$ myproxy-get-delegation -s
myproxy.server.com
Enter MyProxy pass phrase:
A proxy has been received for user miro in
/tmp/x509up_u1001
```

Some other useful commands:
- grid-proxy-info

- grid-proxy-destroy
- edg-job-cancel
- edg-job-get-logging-info
- myproxy-info
- myproxy-destroy
- globus-job-submit
- globus-job-status
- globus-job-cancel
- globus-job-get-output

Conclusion

Basic techniques and routines how to submit a job into the Grid was described. Besides command-line interface users can manage their jobs also via high level interfaces like portals, which are more user friendly and one of them described in [4] will be demonstrated at this workshop [5].

Acknowledgements: This work is supported by EU 6FP EGEE INFSO-RI-508833 project and VEGA No. 2/3132/23, VEGA No. 2/6103/6.

References

1. CrossGrid project: http://www.eu-crossgrid.org
2. EGEE project: http://public.eu-egee.org/
3. SlovakGrid Certification Authority: http://ups.savba.sk/ca/
4. L. Hluchy, O. Habala, V.D. Tran, B. Simo, J Astalos, M. Dobrucky, Infrastructure for Grid-based Virtual Organizations., Proc.of International Conference on Computational Science, Part III, LNCS 3038, Springer-Verlag, 2004, pp. 124{131.
5. M. Dobrucky and IISAS CrossGrid team: FloodGrid demo (CrossGrid project), Proc.of GCCP2005 workshop with international participation, [local reference].

Grid technológie v modelovaní dopravnej problematiky

Peter Kvasnica

Trenčianská univerzita A. Dubčeka v Trenčíne, Fakulta Mechatroniky,
Študentská č. 2, 911 50 Trenčín, Slovak Republic
kvasnica@tnuni.sk

Abstract. The content of this article summarizes the authors‘ look from design the grid computing information systems in the transort. This is accomplished by some computers, which are able to create a grid computer system for solving. The grid computing system is needed for fulfiiel estimation and control the large transport problems. Communication and modeling of this system is made by communcation computer of the type knot. An important issue is the manner of methods using modeling stateful resources with web services in simulation and analytic approache, sophisticated compute of simulating models. By modeling the real transport problems on the grid computers is supported the time saving.
Kľúčové slová: Informačné technológie, grid computing, portlety, simulačné modely, modelovanie dopravných systémov

Úvod

Pre výpočtové modelovanie je primárny interaktívny výpočet, v ktorom riešiteľ môže podľa svojich predstáv výpočet kontrolovať, korigovať, modifikovať a riadiť. Z pohľadu matematických modelov dopravných komplexov je dôraz kladený na počítačový model s jeho riešením na danom technickom a programovom vybavení.
Budúcnosť vyžaduje, aby si firmy v širokom merítku zadovažovali nové technické a programové vybavenie. Väčšina inštalovaných zariadení IT v podnikoch a iných inštitúciách je dimenzována na špičkové výkony, ktoré sú však potrebné len málokedy, a v inom čase sú výpočtové kapacity nevyužité. To má byť odstránené GRID computingom, ktorý virtualizuje počítačovú kapacitu v sieti a sprístupňuje ju pre viac užívateľov.

Čo sú gridové technológie

Budúcnosť pre rozsiahle simulácie je v grid computingu, ktorý predstavuje technológiu umožňujúcu zdielať na diaľku zdroje IT rôznych organizácií a tak vytvárať štruktúru, ktorá sa javí používateľovi ako jeden výkonný počítač. Grid computing sa považuje za formu distribuovaných výpočtov, je orientovaný na veľké

množstvo pripojených prostriedkov, ktorými sú výpočtové prostriedky, systémy na ukladanie údajov, výkonné zobrazovacie systémy a pod..

Podstata je v tom, že veľký výkon jednotlivých počítačov sa dá vďaka špeciálnemu sieťovému (grid) programovému vybaveniu použiť na diaľku prostredníctvom Internetu. Výpočtový výkon takého systému je cez sieť „dostupný“ užívateľom, ktorí by posielaním svojich údajov požadovali ich spracovanie (Application Testbed for European GRID computing, 2005).

Jednou z bŕzd rýchlejšieho rozvoja grid computingu je dôvera resp. nedôvera. Správcovia výpočtových systémov IT, manažéri riadia a administrátori musia definovať, že ich počítače zapojené do gridu budú využívané zmysluplne a bezpečne. Myšlienka je v tom, „vezmi keď potrebuješ“ a „daj keď môžes“, ktorá umožňuje veľké úvahy o využití grid computingu.

Technická definícia Grid-u

Jedna z najpoužívanejších definícií grid computingu pochádza z príručky (Foster, 2002), kde je nasledujúce znenie:

- grid koordinuje zdroje, ktoré nie sú podriadené centralizovanej kontrole
- grid používa štandardné, otvorené a všeobecne definované protokoly a rozhrania
- grid poskytuje zaručenú kvalitu služieb.

PC klastre sa používajú od druhej polovice 90. rokov a tešia sa záujmu komerčnej aj akademickej sféry vďaka výhodnému pomeru výkon/cena. PC klaster sa skladá z uzlov – PC. Podľa spôsobu využitia uzlov rozoznávame dva základné spôsoby práce:

1. load balancing – na jednom uzle môže byť riešených viac užívateľských úloh súčasne.
2. batch mode – na každom uzle sa rieši iba jedna užívateľská úloha.

Grid: v súčasnosti poznáme dve základné štruktúry prepojenia uzlov s črtami globálneho počítača:

1. P2P (peer to peer) – dočasná sieť počítačov, zväčša s rovnakým programom, vhodná na riešenie špecifických problémov. Výhodou je veľký počet pripojených PC v krátkom čase v závislosti od atraktívnosti riešeného problému (100-ky či tisíce CPU).
2. GRID – trvalá sieť obslužných uzlov, založená a budovaná na princípe spolupráce medzi virtuálnymi organizáciami.

Očakávaný je nový systém, založený na výhodných vlastnostiach predchádzajúcich dvoch riešení. Napríklad obsahuje domáceho užívateľa v USA, PC klaster v Slovinsku a diskové pole v Číne. Systém nie je centrálne riadený, každý uzol má svoju vlastnú politiku prístupu a ochrany zdrojov. Inštalované programové vybavenie môže byť Globus Toolkit, t. j. súčasný štandard. Naopak, siete P2P sú distribuované, poskytujú nadštandardné služby, ale programové vybavenie je veľmi špecifické, závislé od aplikácie, nepoužívajú voľne dostupné štandardy pre distribuované spracovanie úloh.

Portál gridu všeobecne podporuje :

- Úplne implementovaný API portletov
- Vývoj portletov je podporovaný použitím štandardu Java Server Faces (JSF)
- Rozšírenie implementácie Portlet API úplne kompatibilnej s IBM's WebSphere® 4.2.
- Model vysokej úrovne pre vytváranie komplexov portletov použitím vizuálnych vlastností a knižnice doplnkov GridSphere User Interface (UI).
- Základný flexibilný XML portál prezentácií opisu môže byť jednoducho modifikovaný vytvorením užívateľských návrhov portálov.
- Vytvorená podpora pre Role Based Access Control (RBAC) oddelených užívateľov ako guests, users, admins and super users.
- GridSphere jadro portletov.
- Lokalizácia s podporou v portlet API implementácií a GridSphere jadre portletov s podporou Francúštiny, Angličtiny, Nemčiny, Čestiny, Polštiny, Taliančiny a ďalších
- Open-source a 100% voľný (Welcome to the GridSpehe Project , 2005)!
- Vo vývoji gridu sa rozlišujú tri generácie: 1G – záujem o paralelné a distribuované počítanie (60. roky) a špecializované užívateľské programové vybavenie bez spoločných modulov (80. roky). 2G – široká ponuka programových modulov, zatiaľ nekompatibilných. Globus Toolkit 2.0 je prvá verzia pre prevádzkové gridy (apríl 2002). 3G je charakteristická snahou vytvoriť štandard s využitím webových služieb Global Grid Forum OGSA/OGSI štandard (jún 2003) a ohlásením návrhu nového štandardu WS-Resource Framework (január 2004).

Takmer všetky veľké IT spoločnosti sa upísali k podpore nových štandardov spojených s grid computingom. Zloženie aliancie naznačuje, že snaha pokryť všetky oblasti, ktoré majú na úspešnom presadení gridu vplyv. Za týmto účelom sa v 90-ych rokoch kreovalo globálne fórum pre grid (The Global Grid Forum – GGF). Toto je štandardizačný orgán, ktorý vytvoril celý rad dokumentov popisujúcich kľúčové zavádzanie gridu, predovšetkým ide o bezpečnosť, výkonnosť a plánovanie procesov (Access to GRID, 2005).

Simulácia a modelovanie v doprave

Pre modelovanie sú veľmi významné pomerne jednoduché objektovo orientované techniky a interaktívne programové prostredia s integrovanou grafickou podporou.

K problematike modelovania dopravných komplexov patrí spôsob financovania výstavby dopravných trás (pozemných trás, železničných či vodných trás, komunikačných trás), vplyv na plynulosť dopravy, bezpečnosť, ekonomičnosť s dostupnosťou v danom regióne, ako aj s jej vplyvom na životné prostredie či ekológiu. Rozhodovacie procesy v každej oblasti majú svoje špecifiká, ktoré na základe modelového riešenia je možné na vstupoch meniť a sledovať ich vplyv na modelovú situáciu.

Pre modelovanie zložitých dopravných komplexov je vhodné preto použiť technológiu grid, ktorá umožňuje z geograficky distribuovaných výpočtových a pamäťových zdrojov vytvoriť univerzálny výpočtový systém s extrémne veľkým výkonom a pamäťou. Výhodou gridu je vysoká efektívnosť využitia združených technických kapacít a tvorivého potenciálu užívateľov.

Táto architektúra nájde výhodné použitie v náročných úlohách modelovania a simulácie dopravných systémov. Spôsob jej realizácie je potrebné podriadiť i analýze IS modelov opisovaných komplexov a podsystémov s následným budovaním programovej aplikácie vo vybranom programovacom jazyku a operačnom prostredí. Štandardným postupom môže byť z programu napísaného v jazyku C/C++ získaný binárny kód. Pomocou jazyka RSL (Specification Language z projektu Globus) a jeho rozšírenej verzie pre NorduGrid xRSL vytvoriť úlohu pre úspešne spustenie v gride. Syntax xRSL je pomerne jednoduchá, slúži na detailný opis úlohy a jej vzťahu so vstupno - výstupnými súbormi alebo zadanie špecifických atribútov na riešenie danej úlohy. Pri použití rôznych nástrojov aplikačný programátor určuje efektívnosť využitia uzlov (Software, 2004).

Modelovanie dopravných úloh s podporou Grid-u

Modelovanie dopravných systémov s podporou webových služieb sa bude pravdepodobne realizovať metódou ilustrovanou na obrázku č.1, kde sú uvedené tieto podstatné časti : definovanie zdrojov, základných chýb, obnovenia odkazov, skupiny obsluhy, pridelenia a úlohy požiadaviek zdrojov. Takýto systém je opísaný dopravnými uzlami.

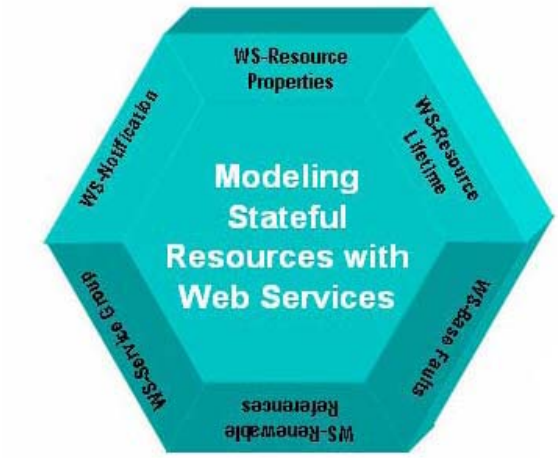


Fig. 1. Modelovanie zdrojov webovými službami

Dopravný uzol je miesto malé z pohľadu územia, o ktorom uvažujeme v dopravnom systéme s týmito funkciami :
- tvorenie, resp. rušenie dopravných kompletov,
- vstup elementov do systému, resp. z výstup z neho
- zhromažďovanie dopravných elementov

Použitím uvedenej metódy môžeme napríklad modelovať :
- simuláciu dopravných uzlov (železničných, automobilových, cestných, leteckých, telekomunikačných a pod.),
- riadenie a simulácia prepravných procesov na mikroskopickej a makroskopickej úrovni,
- riadenie a simulácia procesov prenosových dráh (linky, uzle, média),
- inteligentné dopravné systémy a kvalita dopravných služieb a pod.
- bezpečnosť a spoľahlivosť dopravných systémov.

Pre prevádzkovateľa dopravných distribučných systémov je potrebné mať presný obraz o správaní sa celého systému. Pre tento účel sa budujú dispečerské strediská vybavené modernými prenosovými zariadeniami, ktoré umožňujú v reálnom čase sledovať a meniť charakteristiky vo vybraných bodoch siete z centrálneho miesta.

Monitorujú sa vstupy a výstupy z podsystému, hodnoty parametrov, množstvá jednotiek a iné miesta súvisiace s prevádzkou objektov. Vhodným doplnením takýchto systémov sa stali matematické simulačné modely. Správne zostavený a odladený model siete dokáže simulovať charakteristiky dopravného systému v každom jeho uzle a v rôznych meniacich sa podmienkach. Uzol je definovaný kapacitou a priepustnosťou.

Ak dokáže model zachytiť tieto zmeny pomerov v čase, hovoríme o simulačných modeloch. Štandardným postupom pre matematické modelovanie je:
- zadávanie vstupných dát (schéma siete, typológia siete daná súradnicovým systémom uzlov, miesta vstupov a výstupov a pod.),
- kalibrácia matematického modelu,
- verifikácia matematického modelu.

Distribuovaný simulačný systém je založený dnes na dostupných WWW štandardoch SOAP a XML (Developing new grid components, 2005). Projekt je zameraný na simulovanie problematiky dopravných systémov v extrémnych prevádzkových podmienkach: vysoké nároky na stabilitu, vysoké prepravné kapacity a rýchlosť spojená so spoľahlivosťou. Dôležité pritom je upriamenie pozornosti na efektívne združenie použitých programových modulov, s podporou každého programového modulu webovou službou.

Záver

Cez počiatočný záujem a nadšenie sa dnes účastníci odborných konferencií začínajú pýtať na používanie gridu. Odpovede sú však rozpračité.

Napriek pôvodným optimistickým predpokladom, že užívateľ jednoducho zadá svoju požiadavku formou zdrojového programu do gridu a dostane výsledok, ukázal na ich súčasnú neodôvodnenosť v plnej miere. Tento dočasný nedostatok však neuberá na atraktívnosti gridu, skôr motivuje k originálnym riešeniam. Existencia otvorených produkčných projektov otvára dvere študentom a vedcom z krajín s menej vyspelými IT k novej technológii a účasti na zaujímavých projektoch.

Literatúra

1. Zabawa, J., Zasady rachunku kosztów działań na przykladzie symulacyjnego modelu systemu, Zborník z XXVI Medzinárodného zimného kolokvia, 2004, ISBN 80–86840-03-4, s 245-250.
2. Christopher Watkins, Stephen R. Marenka: Virtual Reality ExCursions, with programs in C, AP Professional: Boston, 1995, s.63, ISBN 0-12-737-865-0.
3. Foster, X.: The Grid , In Blueprint for a New computing Infrastucture, 1992.
4. Welcome to the GridSpehe Project, http://www.gridspehere.org (2005-05-27)
5. Access to GRID, Developing new grid components, http://www.crossgrid.org (2005-05-05)
6. Application Testbed for European GRID computing, http://www.crossgrid.org (2005-05-05)
7. Software, http://www.crossgrid.org (2004-12-06)

A Real-Coded Genetic Algorithm for the Determination of Liquids Refraction Index

Jana Pigošová

Institute of Measurement Science, Slovak Academy of Sciences,
Dúbravská cesta 9, 841 04 Bratislava, Slovak Republic
jana.pigosova@savba.sk

Abstract. Double exposure holographic method has been widely used to determine parameters of objects such as their refractive indices or sizes. However, there is a great disadvantage of this method – the refractive index of the used immerse liquid has to be known very accurate. This work schemes out a technique how to estimate the refractive index of immerse liquids for interferometric purposes and how to eliminate its refractive index change in dependence on the change of temperature and on shakes.

1 Introduction

The Genetic Algorithm (GA) has been developed by John Holland at the University of Michigan in 1975. Fundamentals of GA are specified by universal principles of evolution. The evolution in the context of numerical methods is represented by the evolution of the solutions. The fittest part from their population is allowed to survive and reproduce the next generation with better solutions – Fig. 1 [1]. The reproduction process consists of the natural selection, crossover and mutation of the genes of the species. The result of the great number of generations is an optimal solution. Due to this fact, the evolutionary algorithms can be used to optimalize function parameters.

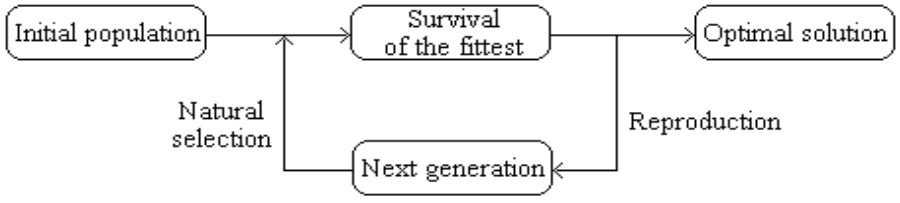


Fig. 1. Evolution in nature by Darwin

In the case of binary string encoding of real variables and the normal values of mutation probability (P_{mut} = 0.001-0.01), theory of the GA is based on the so-called hyperplane sampling and Schema Theorem [2], by which the GA is able to render the global minimum with the probability equal to one after time $t \to \infty$. However, the

GA with binary string encoding is not able to overcome so-called Hamming's barrier that is caused by the fact that for two chromosomes representing two "neighboring" numbers the Hamming length, i.e., the number of different bit variables, can be very large. The Hamming's barrier can be eliminated using nonbinary alphabets [3-4], an inverse operator, the Gray's encoding, or messy chromosomes [2]. However, the simplest manner, how to do it, is the use of nonbinary (virtual) alphabets.

In this paper, the two-beam interference fringe method is applied for step-index optical fiber preforms to obtain refractive index of the used immerse liquid. The Genetic Algorithm with the real string encoding of variables is applied to its evaluation and its principle can be found, for example, in [3-4].

The appearance of the interferograms made using a double exposure holographic method [5] is the consequence of the optical path difference between two expositions. The record of the cuvette with the object corresponds to the 1st exposition and the record of the cuvette without object to the 2nd one. The wedge prism was used to rotate the direction of the reference beam between the expositions to enable the finite width fringe method [6]. Two step-index optical fiber preforms from Institute of Physics, SAS were used as caliber. In both cases the value of numerical aperture was 0.22 and the refractive index of the preform on the axis of the core is $n_1 = 1.45718$.

Thus, considering straightforward transition of the light beam with the wavelength across the step-index fiber, the coordinates x and y of the finite width interference fringes corresponding to the core of optical fiber are described by

$$f: y = y_0 + 2K(n_1\text{-}n_2)\sqrt{a^2 - (x - x_0)^2} + 2K(n_2 - n_G)\sqrt{b^2 - (x - x_0)^2} \qquad (7)$$

where x_0 is the coordinate of the center of optical fiber, $K = h/$ is a constant determined by the angle of the rotation of the wedge prism, h is the interfringe spacing, a and b are the radii and n_1 a n_2 refractive indices of the fiber core and cladding, n_G is a refractive index of the used immerse liquid (glycerine). In the case of graded-index optical fibers, Eq. (1) is more complicated.

2 Computer Processing and Data Acquisition

The sequence of steps for interferometric records processing is schematically shown in Fig. 2. The colored image is transformed to gray scale one. The black and white image is obtained using the algorithm for two-level image thresholding based on Bayesian formulation and the maximum entropy principle developed by Chang, Fu, Yan and Zhao [7]. Their method belongs to the global thresholding methods in which the entire image is thresholded with a single threshold value. This value is determined under the condition that the most information given by the original image is retained after it was thresholded. We used following parameters (in both cases): error $\varepsilon = 0.01$ and linear shape of the conditional probability function.

After thresholding of an image, it is not difficult to obtain its dark or bright points. However, in the case of holographic interferograms, number of these points is very large, i.e., the fringes are thick. To narrow them, there is possible to use either edge detecting methods or fringe thinning algorithms.

Edges within an image correspond to intensity changes in them. A variety of edge detectors has been proposed. They are applied at a single resolution or at a multiresolution scale. For example, Canny formulated edge detection as an optimization problem. He defined a comprehensive set of performance criteria to compute edge points: maximizing SNR, good edge localization, and only one response to a single edge. Canny's edge detector ranks among the single resolution detection methods. Siddique and Barner [8] used multiresolution gray-level stacking edge map pyramids. Hou and Koh [9] proposed edge detection by which the edge structure is first detected using robust statistics, and then localized by a robust contrast test. Toivanen et al. [10] presented edge detection methods in multispectral images based on the use of the self-organizing map, Peano scan, and gray-scale edge detectors.

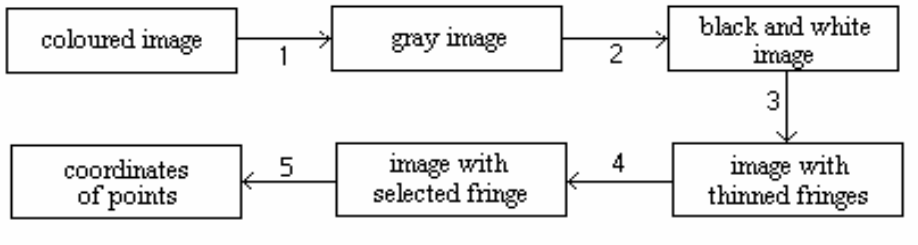


Fig. 2. Sequence of steps for data acquisition

An interferogram can be described essentially by a 2-D continuous sinusoidal distribution of gray scale [11]. After its two-level thresholding, the information generated by the interferometer remains available in the form of interference fringes. The midpoints of the fringes before and after thresholding remain the same. Thus, we need not locate the intensity maxima or minima in the bright or dark fringe bands, it is sufficient to trace the midpoints in them. In practice, a greater noise level in the high-intensity regions, possibly owing to device saturation in the recording medium, can be observed. From this reason, the fringe thinning method is related to the location of intensity minima in the dark fringe bands. The Gaussian profile of the laser light output was not taken into account because of small interferogram area and collimation of the laser beam. Then the two-level thresholded image is traced and its midpoints are acquired. The fringe selecting method is based on the polygon drawing around the fringe.

3 Implementation of the GA

The values of parameters K, n_1, n_2, a, b, x_0 in Eq. (1) can be considered to be known, or computable from the numerical aperture, respectively. Thus, our genome is consisting only of two components: y_0 and n_G saved in type *record*. The parameters of the optical fibers from the interferograms are listed in Table 1.

The evaluation function has the following form

$$M : (y_0, n_2) \rightarrow \sum_{i=1}^{n} \left| y_i - f(x_i) \right|^2 \tag{2}$$

where n is a number of points in the selected fringe and the function f is defined by Eq. (1). The phenotype of the genome is represented by the value M. In this case, all coordinates x are not different in several orders, the relation for the fitness function need not be defined. Instead of it, we used the evaluation function.

4 Results and Discussion

Parameters of the GA were set as follows:
- number of generations = 20
- number of genomes in the population = 20,000
- number of genomes allowed to reproduce = 2,000
- the crossover probability = 0.5
- the mutation probability = 0.2.

The Table 2 gives the values of refractive index of glycerin for 10 runs of the GA. Considering Gaussian distribution of the refractive index of the immerse liquid, its extended uncertainty for 95% confidence interval is 6.8×10^{-5} for the thicker fiber and 4.5×10^{-5} for the thinner fiber.

Concerning the same perform for the both fibers; we can suppose they have the same parameters. In addition, the conditions during the construction of both interferograms can be considered to be constant (interferograms were made step-by-step). Because of that, the real value of the refractive index of glycerine will be in the conjunction of the confidence intervals of both fibers, i.e., the refractive index of the glycerine belongs to the interval [1.45471; 1.45475].

The essential influence to the overall error of the proposed method for the determination of the refractive index of liquids has the setting of the genetic algorithm parameters (the total number of the genomes in the population, number of genomes in elite population, probability of crossover and mutation, etc.).

Table 1. Parameters of the corresponding interferograms of the optical fibers.

Parameters	Interferogram #1	Interferogram #2
x_0 [pix]	181	152
b [pix.]	178	138
a [pix] = 0.9 * b [pix.]	160.2	124.2
h [pix.]	103	156
b [mm]	0,825	0,55
n_1	1.45718	1.45718
n_2	1.44048	1.44048

In the case of the use of the binary alphabets, the number of genomes in the population is usually set to be equal $10^{\text{number of genes in genome}}$ and the ratio of the number of genomes in the elite population to the total population equal to 1/10 [1]. In the case of the use of nonbinary alphabets, it is difficult to determine necessary number of genomes in the population because of the increase of the number of hyperplanes. From this reason, the number of genomes in one generation was set to be 20,000.

We used uniform arithmetical crossover that was done by using the mathematical formulas. The elite population was held like as some "intermediate generation" that was modified only in the case when the new solution was better than its worst one. For this reason the mutation probability value was selected larger than it is used usually. The value of the number of the generations was determined on the knowledge base obtained using GA for the evaluation of the interferograms of the optical fibers [3].

Table 2. Refractive index of glycerine determined using Genetic Algorithm.

Number of the attempt	Fiber #1	Fiber #2
1	1.45464	1.45474
2	1.45470	1.45472
3	1.45474	1.45475
4	1.45471	1.45473
5	1.45471	1.45478
6	1.45466	1.45477
7	1.45466	1.45476
8	1.45468	1.45477
9	1.45467	1.45478
10	1.45466	1.45469
Average value	1.45468	1.45476
Standard uncertainty	3.10^{-5}	2.10^{-5}

5 Conclusions

The proposed method for interferometric measurement of immerse liquids refractive index can determine its value without its replacement. This is necessary from the reason that the properties of the immerse liquids (density and so refractive index) depend strongly on the temperature gradient and shaking during the dragging. It is not a big problem after or before the production of interferograms to do another one (calibrating) from which we can discover the value of refractive index of the used immerse liquid. For the purposes of the optical fibers interferogram processing can be showed that one must know the value of the refractive index of the immerse liquids with the accurate of 4 decimal places (for comparison, the Abbe's refractometer provides the ability of its determination only with the accuracy of 3 decimal places).

The proposed method was tested on two preforms of the same type with different radii. In both cases we acquired similar results from which we determined the refractive index of glycerine during the expositions of holographic interferograms. Because of that, we can consider the method acceptable and advisable.

This work was supported by grants VEGA 2/4091/04 and APVT-51-022702.

References

1. Harrison, P.: Computational Methods in Physics, Chemistry and Biology: An Introduction. John Wiley&Sons. Ltd., Chichester New York Weinheim Brisbane Singapore Toronto (2001)
2. Kvasnička, V., Pospíchal, J., Tiňo, P.: Evolučné algoritmy. Vydavateľstvo STU, Bratislava (2000) 1-13, 31-57, 100-120
3. Pigošová, J.: On Solving Optical Problems Using Genetic Algorithms. In: Štrba, A., Senderáková, D., Hrabovský, M. (eds.): 14th Slovak-Czech-Polish Optical Conference on Wave and Quantum Aspects of Contemporary Optics. Proceedings of SPIE, Vol. 5945 (2005) 445-453
4. Blanco, A., Delgado, M., Pegalajar, M.C.: A real-coded genetic algorithm for training recurrent neural networks. Neural Networks 14 (2001) 93-105
5. Balaš, J., Szabó, V.: Holografická interferometria v experimentálnej mechanike. VEDA, Bratislava (1986)
6. Sokkar, T. Z. N., Mabrouk, M. A., El-Bawab, H. F.: Refractive index profile of GRIN optical fibre considering the area under the interference fringe shift: II. The mismatching case. J. Opt. A: Pure Appl. Opt. 1 (1999) 64-72
7. Chang, Y., Fu, A. M. N., Yan, H., Zhao, M.: Efficient two-level image thresholding method based on Bayesian formulation and the maximum entropy principle. Opt. Eng. 41 (2002) 2487–2498
8. Siddique, J. I., Barner, K. E.: Nonlinear image decomposition for multiresolution edge detection using gray-level edge maps. Opt. Eng. 41 (2002) 2749–2763
9. Hou, Z., Koh, T. S.: Robust edge detection. Pattern Recognition 36 (2003) 2083-2091
10. Toivanen, P.J., Ansamäki, J., Parkkinen, J. P. S., Mielikäinen, J.: Edge detection in multispectral images using the self-organizing map. Pattern Recognition Letters 24 (2003) 2987–2994
11. Wei, Y., Yan, D., Cha, S. S.: Comparison of results of interferometric phase extraction algorithms for three-dimensional flow-field tomography. Optics and Lasers in Engineering 32 (1999) 147–155

Ordered Fuzzy Decision Tree Building in Parallel

Vitaly Levashenko, Penka Martincová, Karol Matiaško

Faculty of Management and Informatics,
University of Žilina, Slovakia
levashen@yahoo.com, Penka.Martincova@fri.utc.sk, Karol.Matiasko@fri.utc.sk

Abstract. Fuzzy decision tree (FDT) induction is an important problem of Machine learning and Fuzzy logic. There are several methods for Fuzzy Decision Trees (FDT) induction. One of the key points of these methods is choice an expanded attributes which associated with FDT node. Now we propose some heuristics for select such expanded attributes with differing costs. The basis of these heuristics is cumulative information estimation. Usage these heuristics allow us to build FDT with different properties: unordered, ordered FDT, etc.

Introduction

In nowadays with an enormous growth of data stored in databases and data warehouses, it is particularly important to develop tools for analysis of such data. Data mining is a process of discovering useful knowledge, which consists of methods that discover interesting, non-trivial, and useful patterns, hidden in the data. A general purpose of data mining is to process information, embedded in data so as to develop better ways to handle data and to support future decision-making.

Decision tree induction has been widely used in extracting knowledge from feature-based examples for classification and decision-making. An induction algorithm is used to learn a classifier, which maps the space of feature values into the set of class values [11]. One approach for solving classification task based on decision tree induction follows [14]. The generalizing of decision tree algorithm for fuzzy sets resulted into Fuzzy ID3 algorithm and its known variants and can be found in [1, 2, 5, 12, 13, 15-17].

Detecting every feature value can be obtained by diagnostic tests of input attributes that have associated with integrated (financial and temporal) costs. An interesting problem here is to introduce such a method that would search for an optimal (or sub-optimal) sequence of tests to be undertaken when recognize a new subject in order to minimize the cost of diagnostics.

In this paper, we present our approach, which deals with fuzzy defined data. These data are more accurate in reflecting a real around world. Our approach also should be able to analyze the order in which different diagnostic tests should be performed in order to minimize the diagnostics costs and to guarantee a desired predefined level of accuracy.

For these purposes, we use a technique to compute cumulative information estimations of fuzzy sets [7]. The application of such estimations allows inducing minimum cost decision trees based on new optimality criteria. We obtain new type of fuzzy decision tree: ordered tree. Ordered tree differs from unordered fuzzy decision tree in the way of testing attributes. In ordered trees the order of attribute tests is independent from the results of previous tests, so we can test next attributes in parallel. This leads to decreasing of expenditures for test attributes.

The paper is structured as follows. Section 2 contains brief information about classification task. Section 3 shortly describes algorithms for fuzzy decision trees induction using simple example. Results of experimental investigation are presented in section 4.

Classification Task

People express their subjective feelings, background knowledge and short-time memory, rather than any frequency criteria, to distinguish different data. Fuzzy logic is a popular approach to capture this vagueness of information [18]. The basic idea is to come from the "crisp" 1 and 0 values to a degree of truth or confidence in the interval [0,1].

Analyzing the corresponding values of a membership function performs the fuzzification of the initial data. Here, each attribute value can be seen as likely estimate. In this paper we analyze a particular case when the sum of membership values of all linguistic terms for an attribute equals to 1.

A typical classification problem can be described as follows [17]. A universe of objects U={u} is described by N training examples and n input attributes $A=\{A_1,...,An\}$. Each attribute A_i ($1 \le i \le n$) measures some important feature and is presented by a group of discrete linguistic terms. We assume that each group is a set of mi ($mi \ge 2$) values of fuzzy subsets $\{Ai_{,1},\ldots,A_{i,j},\ldots,A_{i,mi}\}$.

The cost of an attribute Ai denoted as Costi is an integrated measure that accounts financial and temporal costs that are required to define the value of the Ai for a certain subject. We will suggest that each object u in the universe is classified by a set of classes $\{B_1,...,B_{mb}\}$. This set is described by output attribute B.

Let us consider the following example, which will be used in this paper.

Example. An object is presented by four input attributes: $A=\{A_1,A_2,A_3,A_4\}$ and one output attribute B. Each attribute has values: $A_1=\{A_{1,1}, A_{1,2}, A_{1,3}\}$, $A_2=\{A_{2,1}, A_{2,2}, A_{2,3}\}$, $A_3=\{A_{3,1}, A_{3,2}\}$, $A_4 = \{A_{4,1}, A_{4,2}\}$ and $B = \{B_1, B_2, B_3\}$.

The membership of these attributes and costs are presented in Table 1.

$Cost_i$ is a cost for accomplishing the required tests and procedures.

In this paper we introduce an approach for a sub-optimal sequence of expanded attributes testing, i.e. determination of input attributes' values $\{A_{1i},...,Ani\}$ of a new subject, that allows to accomplish the correct diagnostics. It is obvious that the problem is that the sub-optimal sequence should guarantee correct diagnostics with in advance defined level of accuracy while minimum cost for accomplishing the required tests and procedures is reached.

Table 1: A small training set (adopted from [15])

No	A_1			A_2			A_3		A_4		B		
	$A_{1,1}$	$A_{1,2}$	$A_{1,3}$	$A_{2,1}$	$A_{2,2}$	$A_{2,3}$	$A_{3,1}$	$A_{3,2}$	$A_{4,1}$	$A_{4,2}$	B_1	B_2	B_3
1.	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2.	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	0.6	0.4	0.0
3.	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8	0.3	0.6	0.1
4.	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5.	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6.	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7.	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8.	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
9.	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
10.	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
11.	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.3	0.7	0.0
12.	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13.	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14.	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15.	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0
16.	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.5	0.5	0.0
Σ	**6,6**	**5,8**	**3,6**	**6,1**	**5,6**	**4,3**	**5,6**	**10,4**	**6,5**	**9,5**	**4,4**	**4,4**	**7,2**
***Cost*$_i$**	**2,5**			**1,7**			**2,0**		**1,8**				

Fuzzy Decision Trees Induction

We propose a new interpretation of Fuzzy ID3, which is based on cumulative information estimate [6, 7].

Apart from the selection of expanded attributes, the determination of the leaf node is another important issue for fuzzy decision trees induction. The key points of a proposed algorithm for induction fuzzy decision trees are (a) a heuristic for selecting expanded attributes and (b) a rules for transform nodes into leaves. Expanded attributes are such attributes that according to values of attribute trees are expanded at the nodes considered.

The induction of an ordered fuzzy tree has less complexity, while it does not require information estimates calculations for each branch of a tree. Choosing an expanded attribute num is sufficient enough to maximize the increment of information about the attribute at minimum of costs. Usage of cumulative information estimations allows forming like a criterion for ordered fuzzy decision tree induction:

$$num = \arg\max I(B; U_{q-1}, Ai_q) / Cost(Ai_q), \quad (8)$$

Where operator argmax V_i defines index i belong to maximum value of V_i.

Algorithm for ordered fuzzy decision tree induced shown in Figure 1.

Input: (9)

The train set {CS,CV,AS,AV,C,} (see table 1):

CS– set of input attributes CS={A_1,…,A_n};

CV–values of input attributes CV={$A_{i,j}$} (i=1,…,n;j=1,…,mi);

AS–output attribute AS ={B};

AV–values of output attribute AV={B_1,B_2,…,B_{mb}};

C–verification cost of initial attributes C={Cost(A_i)};
P– set of N instances
minimal authenticity decision - β and
maximal frequency of input attributes combination - α.
In future we suppose that β=0,75 and α=0,25.

Initial conditions:
Attributes = CS; // Potential expanded attributes
q = 0 // initial level of FDT
$U_q = \varnothing$; // Number of selected attributes from root to actual node

Output: Ordered FDT with minimum cost

OrderedFDT = buildTree (U_q, Attributes) {
1. Calculate cumulative information I(□;U_q,A_i) for each A_i from Attributes;
2. Select A_{num} among $\forall A_i \in$ Attributes:
   num = argmax I(□; U_q,A_{num})/Cost(A_{num})
3. Fix expanded attribute A_{num} as node FDT:
   OrderedFDT ← node(A_{num})
4. Delete attribute Anum from set Attributes:
   Attributes = Attributes \A_{num}
5. Check leaves and continue:
   q++; $U_q = U_{q-1} \cup A_{num}$
   for($\forall A_{num,j}$),j=1,…,m_{num}; //For each branch of A_{num}
      if($A_{num,j}$ is leaf) OrderedFDT ← leaf($A_{num,j}$)
      if($\exists A_{num,j}$ is not leaf) OrderedFDT =buildTree($U_{q,Attributes}$)
}

Fig. 1. Algorithm for ordered FDT induction

Ordered fuzzy decision tree induced from the data presented in Table 1 can be seen in Fig. 2.

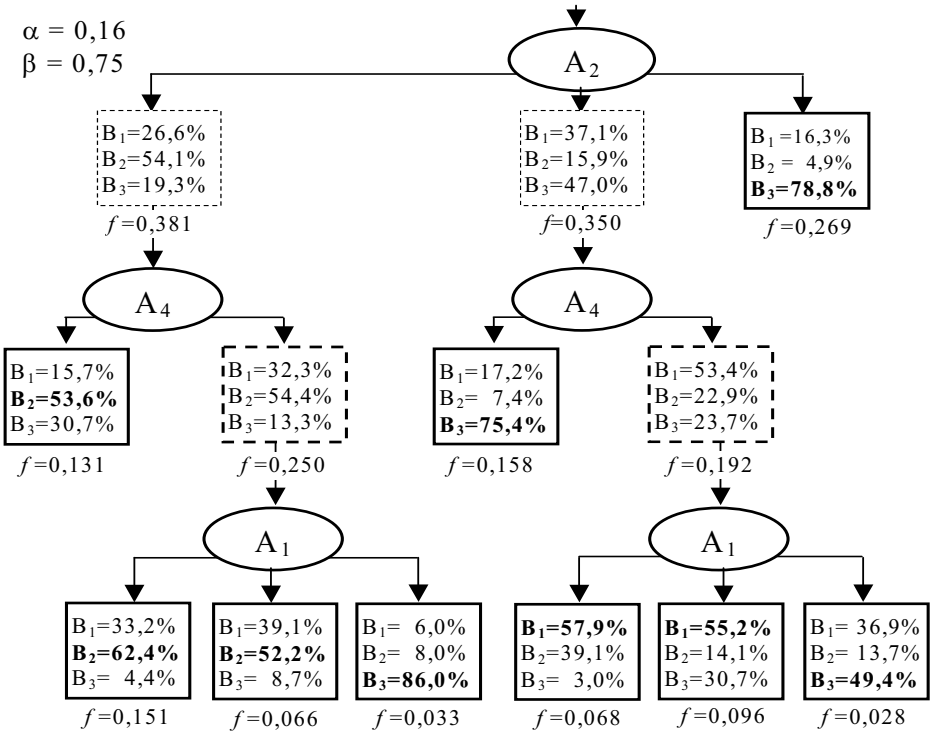


Fig. 2. Ordered fuzzy decision tree induction by rule (1)

Induction of an ordered decision tree allows reducing the classification time due to the possibility of parallel checking of attribute values at several levels of the decision tree. Actually, when classifying an instance at level s, using the unordered decision tree (s=2,…) we need to define Ais attribute's value and it is impossible to know in advance which attribute value we need to evaluate at the next level (s+1). On the contrary, when using an ordered decision tree, one attribute is associated to all branches of the level. That is because we know in advance the sequence of attributes to be evaluated for each branch. Obviously, the construction of an ordered decision tree implies additional costs, but its use can be beneficial in the situations when time factor is critical and there is a possibility to implement the check of several attributes simultaneously. Parallel processing is possible because the choice of next attribute does not depend on values of the preceding attributes.

The order of attribute tests is independent of a situation and only the amount of attribute tests depends on it. This feature allows us to test next attribute even if we do not know result from preceding attributes testing. The testing can be executed using more than one processor, which will lead to decreasing of the total executing time.

Let us consider an example. If we test attributes sequentially, executing tests on one processor, the total time for testing (Figure 3a) is:

Cost of FDT1 = Cost(A_2)×1,0 + (10)
+ Cost(A_4)×(0,381+0,350) + Cost(A_1)×(0,250+0,193) = 4,121

If we use two processors (Figure 3b), the cost will be:

Cost of FDT_2 = (11)
=max (Cost (A_2), Cost (A_4)) ×1,0 + Cost (A_1)×(0,250 + 0,192) = 2,905

If we use three processors (Figure 3c), the total executing time will be:

Cost of FDT_3= max (Cost (A_2),Cost (A_4),Cost (A_1)) × 1,00 = 2,5 (12)

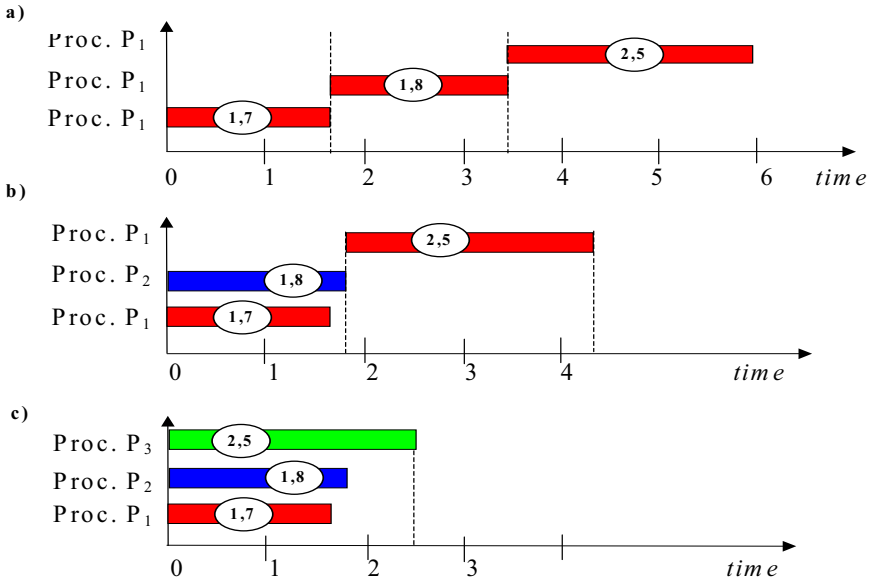


Fig. 3. Using parallel processing

This simple example shows, that exploiting the parallel processing can reduce total execution time for building the decision tree. The problem of mapping the processes to processors (in distributed system, or on SMP) is NP-hard task [9] and it is usually solved using heuristic algorithm.

Experimental Results

The algorithm is coded in C++ and the experimental results are obtained on a Pentium III with 256Mb of memory. The main purpose of our experimental study is to compare our algorithms with other methods.

The experiments have been carried out on Machine Learning benchmarks (dataset) each of which has at least one continuous variable. We had separated initial dataset into 2 parts. We used the first part (70% from initial dataset) for building classification models. The second part (30% from initial dataset) was used for

verification previous models. This process we had repeated 100 series, for the purpose average estimations.

A fragment of our results is shown in Table 2. Columns [Total Sets], [Input Variables] and [Number of classes] describes dataset. The column labelled [Errors] gives the count of error classification. It is calculated as the ratio of the number of misclassification combinations to the total number of combinations. The results in columns Naïve, kNN, uFDT and oFTD are according to Naïve Bayes, k-Nearest Neighbors, unordered and ordered Fuzzy Decision Trees Classification. The last column [Position] demonstrated our method's rate.

Dataset	Total Sets	Input variables	Number of classes	Errors				Position
				Naïve	kNN	uFDT	oFDT	
bupa	345	6	2	0.4414	0.3832	0.3915	0.4312	2
cmc	1473	9	3	0,5240	0,5816	0,5045	0,5223	1
glass	214	9	7	0,5347	0,3152	0,4028	0,4544	2
haberman	306	3	2	0,2595	0,3389	0,2942	0,3012	2
iris	150	4	3	0,0449	0,0473	0,0322	0,0420	1
pima	768	8	2	0,2491	0,2971	0,2563	0,2863	2

Fig. 4. Results on the UCI machine learning benchmark set

Conclusion

We have proposed the induction technique of new type of fuzzy decision tree, which is simple to understand and apply. The use of cumulative information estimations allows precisely estimating mutual influence of attributes. These evaluations are tool for analysis of training examples group. Our estimations are based on Shannon's entropy analogue. The use of such estimates allows inducing minimum cost decision trees based on different criteria. We introduced the cost of expanded attributes diagnostics into considered algorithms.

We have shown a possibility of reducing the execution time at the first stage of building a decision tree due to ordered processing of attributes.

Parallel processes were used for resolving time-consuming tasks in many scientific problems [3-5, 9]. The problem of parallelizing the process of building a fuzzy decision tree can be treated stage by stage. At the first step a fuzzification of the continuous data can be parallelized. If for each interval is created one process, then the level of parallelism will be equal to the number of intervals. The second improvement is possible through searching for maximum of the increment of information about the attribute at minimum of costs, which also can be done in parallel. Next possibility is to do parallel check of the attribute values, which was mentioned before in this article. Using parallel processes can also decrease the time for mining of association rules. Programming model of the processes [8, 10] must be carefully selected in order to reflect their real configuration.

Execution time improvements are not straightforward, because of the dependence of data localization, memory requirements, scalability and load balancing. Building the fuzzy decision tree is process, which requires communication between nodes and

also access to part of the data. Exact estimation of communication, memory requirements and scalability are important factors, which can influence resulting.

Acknowledgements

This research was supported by grants of Scientific Grant Agency of the Ministry of Education of Slovak Republic and the Slovak Academy of Sciences ZU/05VV_MVTS13 and (VEGA) No 1/1053/04.

References

[1] Chiang I-Jen, Jane Yung-jen, Hsu: Fuzzy Classification Trees for Data Analysis Fuzzy Sets and Systems, 130, pp. 87-99, 2002.
[2] Janikow C.Z.: Fuzzy Decision Trees: Issues and Methods, IEEE Trans. on Systems, Man, and Cybernetics — Part B: Vol. 28, 1, pp.1-14, 1998.
[3] Joshi M.V., Han E., Karypis G.: Parallel lgorithms in Data Mining, http://userpages.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm
[4] Kollár Ján: Metódy a prostriedky pre výkonné paralelné výpočty. Academic Press elfa, Košice, 2003, FEI TU v Košiciach, 110p., ISBN 80-89066-70-4
[5] Lee H.M., Chen C.M., Chen J.M. and Jou Y.L.: An efficient fuzzy classifier with feature selection based on fuzzy entropy, Proc. IEEE Trans. on Systems, Man and Cybernetics, Part B:, vol. 31, 3, pp. 426-432, 2001.
[6] Levashenko,V. Martincová,P: Fuzzy Decision Tree for parallel processing support ‚Journal of Information, Management and Control Systems, Faculty of Management Science and Informatics, University of Žilina, pp.47-54, ISSN 1336-1716, 2005.
[7] Levashenko V., E. Zaitseva: Usage of New Information Estimations for Induction of Fuzzy Decision Trees. Proc of the 3rd IEEE Int. Conf. on Intelligent Data Engineering and Automated Learning, 2002.
[8] Martincová P.: Software Environment for Cluster Computing, in Journal of Information, Control and Management Systems, Volume 1, No. 1, 2003, vyd.: University of Žilina, Faculty of Management Science and Informatics., Žilina, 2003, ISSN 1336-1716
[9] Martincová P., Kňaze M.: Using Parallel Processes for Optimization Tasks, Proc. of Mosmic 2003 (Modelling and Simulation in Management, Informatics and Control), Žilina University publisher, pp.139-144, ISBN 80-8070-139-3.
[10] Matiaško K.: Database Systems. EDIS Print, Zilina (2002) 303 p.

Authentication in Grid

Adrian Bagala and Ladislav Hudec

Faculty of Informatics and Information Technologies,
Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava 4, Slovak Republic
bagala@fiit.stuba.sk, lhudec@fiit.stuba.sk

Abstract. This document globally describes authentication methods and procedures, which are parts of the GSI component used in Grid environment. Document is specially addressed to authentication using standard X.509 certificates and out of it derived proxy certificates. In the last chapter, generation and usage of these certificates in Grid tool Globus Toolkit 4.0 are described. This paper contains concrete procedures of certificates generation, their security and usage during authentication.
This work was supported by Slovak Science Grant Agency grant no. VG 1/0157/03 entitled Methods and tools for development of the secure networked and distributed mobile computer systems and their management II.

1 Introduction

Grid is the technology, which generates an infrastructure composed of IT sources different organizations, institutions or individuals. This infrastructure represents a robust distributed system. For a user it looks like one high-performance parallel computational system with high-capacity data space. This system provides easy and transparent access for the user. The Grid consists of large array mutually connected resources of two types: computational systems and data storage systems. Individual nodes can be geographically widespread and connected via Internet.

The security plays an important role in these systems. Saying the grid system security we mean mainly privacy, integrity and authenticity. Security is provided with complex GSI (Grid Security Infrastructure). GSI fulfills GSS-API (Generic Security Service Application Programming Interface) standards [1], [2].

2 Authentication using X.509 certificates

GSI authentication in grid systems is based on certificates. Each entity in the grid environment is identified by X.509 certificates. Saying entity we mean any user (concrete person) accessing the grid resources or service, respectively. This service is understood as a grid environment service or any executable application respectively, which can access the grid system resources the same way as the user. A case can

occur when the service may run operations on behalf of user, who executed this application or process. In this case it is delegation of permissions, which authenticity is provided with short-term proxy certificates, which are derived from standard X.509 certificates. The process of permissions delegation and proxy certificates properties is detailed described in chapter 3.

Information contained in the X.509 certificate enables identification and authentication of user or service. An Internet format of certificate is described by RFC-3280 standard [3]. It consists of following essential parts:
- name and ID of subject
- public key of subject
- identifier of certifying authority, which signed this certificate; i.e. verification that public key and identifier belong to the subject
- digital signature of certifying authority

All certificates contain public key of the user or service. This certificate is signed by digitally signature of certifying authority. We consider this certifying authority to be trustful, what consecutively guarantees a trustfulness of the signed certificate.

As it was already mentioned, the identification and authentication of the user or the service in the grid environment is possible by using X.509 certificates. In this case we are talking about one-way authentication. Two-way (mutual) authentication is of course available in GSI. It means if both sides have certificates in the process of mutual authentication and both sides trust certifying authority, which signed both certificates, then both sides can mutually verified their identities. GSI uses SSL/TLS protocol for mutual authentication [8].

Before mutual authentication of both authenticating sides it is necessary that both sides trust the certifying authority, which digitally signed certificate of the second side. It means in practice, both sides must have the own copy of certifying authority certificate, which contains the public key of the certifying authority.

Detailed description of the authentication process and authentication protocols using X.509 is not included in this document.

3 Authentication, single sign-on and proxy certificates delegation

Proxy certificates are certificates derived from classical X.509 certificate, which was described in the previous chapter. An Internet format of proxy certificate is described by RFC-3820 standard [4]. As proxy certificates used the same format as X.509, they can in many cases use the same protocols and libraries equally as in case of X.509 certificates.

Proxy consists of new certificate and a private key. New certificate contains the user identity, which is slightly changed to indicate it is a proxy certificate.

Consecutively its owner signs this new certificate. It is easier and more practical than to be signed by certifying authority. After all, a time stamp is added to certificate. Just till this time the proxy certificate is valid. It means, they have short-term lifetime e.g. 12 hours.

The private proxy key should be protected, but because of the proxy has short-term lifetime, it is not necessary to protect it as much as the private key of an owner. Sufficient protection is to save it as non-encrypted file with the read-only permission for the owner only. By the proxy authentication, it is not necessary special entering pass phrase. Up on this reason, it is possible, apart from the possibility of permission delegation, to use the proxy also for single sign-on, which is described in following chapter.

3.1 Single sign-on

Usually the private key associated with long-term X.509 certificates is protected by several methods. These methods require manual authentication, it means during the authentication process is necessary to enter a pass phrase. In case of often access to resource these methods are rather restrictive and complicated.

Proxy certificate solves this problem by enabling single sign-on. This sign-on allows the user to authenticate one time only via X.509 certificate and with it related private key, at the moment of proxy certificate creation. The proxy certificate can by used repeatedly for authentication during definite time period.

The security of this authentication method lies just on this short-term lifetime of proxy certificate. The proxy certificate and the private key are saved in local file-system, which is protected by predefined local user permissions. It means in practice that private key of proxy certificate is allowed to be accessed only by its user (owner), who created it or corresponding user account in this system, respectively.

3.2 Delegation

Proxy can be understood as an appliance by means of which a certain person is authorized to execute some activities for the sake of other persons profit. In our certificate understanding this process is called a delegation of certain permissions. So the proxy certificates serve for permission delegation from one entity to another.

In practice, the user executes in the grid systems some processes or applications, which use services accessing to the grid resources. It means, these application or processes run certain tasks on behalf of the user. Because it is mainly net communication, every access to the grid resources via offered services must be every time authenticated because of security. This authentication is necessary also to guarantee, that every access request was demanded from the user with unique identifier, which belong to concrete person. Of course, this authentication could be handled using X.509 certificate of the user. But the private key is protected by password, what at first requires permanent entering this password by the user and then permanent presence of the user at certain running process or application. Also because

of these facts there was created the possibility of authorization delegacy from the user to the processes via proxy certificates.

So the proxy certificate, generated out of used X.509 certificate, is used by the applications, which were run by the user. These applications use it in every secure conversation for authentication of mentioned user. The next advantage of this solution is also reduction of conversation misuse risk. In the case of communication misuse an attacker has a possibility to misuse the proxy key pair only and not personal keys of the users. More over the short-term lifetime of proxy certificate supplies the security.

4 Generation and authentication by certificates in GT4.0

Globus Toolkit is an open source tool used for grid system setup. It is developed by Globus Alliance community. This chapter describes the authentication methods and procedures, using X.509 certificates and proxy certificates, in Globus Toolkit version 4.0.

4.1 Certificate authentication in GT4.0

By authentication using X.509 certificates and proxy certificates everything is generally, but also in GT4.0, about trust. Therefore, first of all, it is necessary to properly configure the GT tool, so that it trusts all certificates issued by certifying authority, which is of course trusted as well. To make it working, it is necessary that the certificate of trusted certifying authority itself is imported into GT4.0 (eventually certificates issued by all the other trusted certifying authorities) and a configuration file, where a signing policy of certificates (signed by this certifying authority) is described. After these settings, if the GT4.0 finds information about certifying authorities in mentioned files, it will trust to every certificate issued by these authorities. Thus following authentication will be correct and accepted (after this settings).

The user can authenticate himself in GT4.0 via his X.509 certificate and correspondingly generated proxy certificate. Using proxy certificate he can delegate permissions on his executed processes. These processes are essentially applications, which use grid services and access the grid resources in behalf of the user. They use the proxy certificate of the user for their authentication and this gives them possibility of single sign-on authentication with all corresponding delegated permissions. Authentication and delegation of permissions for remote access to the grid services and resources via public network communication is not mentioned in this document due to its extensive size and is detailed described in [5].

Security protocols used for authentication and secure communication in GT4.0 are described in more detail in [6].

4.2 Generation of X.509 certificate in GT4.0

GT4.0 provides a tool with a possibility of the key pair generation as well as the certifying request for subsequent signing by trusted certifying authority. Corresponding configuration necessary for creation and generation of correct requests, which will be accepted certain certifying authority, are described in administrator guide of GT4.0 [7].

After appropriate configuration for creation of key pairs it is available to generate these keys and certifying requests by "grid-cert-request" command. The private key generated and is saved in local file system. This private key is encrypted and is usable only if the pass phrase entering follows. Correspondingly to it is generated certifying request to be signed by certifying authority. Setting of local file system protects the private key with a read-only permission for his owner only.

As a mutual authentication of users and system resources is also used in GT4.0, it is possible to generate these keys for both entity types. GT4.0 tool will determine the entity type looking at the subject name of certifying request and later also by entity certificate itself. Subsequently this request is necessary to send to certifying authority, which after relevant identification of the person or the resource (according the subject name in request) will sign it and add other necessary information to it. A result of this process is public certificate of the entity, which generates relevant request and the private key corresponding to the certificate. This certificate is finally imported to the GT4.0 environment and the read-only permission for all system users is set in local file system.

4.3 Generation of X.509 proxy certificates in GT4.0

GT4.0 supports the permission delegation and single sign-on via proxy certificates. It contains as well tool for their generation. Generation of proxy certificate in GT4.0 is executed by "grid-proxy-init" command. The new key pair is generated and the proxy certificate containing the public key is signed by private key from X.509 certificate of the user, who generates this proxy. The user's private key is protected by pass phrase, which is to be entered. The subject name of the proxy certificate contains a full subject name of users X.509 certificate and the generated unique proxy certificate identifier extends this subject name. This proxy certificate is issued for the sake of user authentication and authentication of his executed processes with short-term validity (usually for 12 hours).

Practically, in GT4.0 one file contains public and private key of proxy certificate. The file more over contains the public key corresponding to private key of user's X.509 certificate, by which the proxy certificate is signed. The proxy certificate file is saved in temporary directory of local file system. The permissions are set to read and write for owner only.

Detailed procedures of configuration as well as the application settings of GT4.0 and certificate usage are documented in administrator guide [7].

Acknowledgement

This work was supported by Slovak Science Grant Agency grant no. VG 1/0157/03 entitled Methods and tools for development of the secure networked and distributed mobile computer systems and their management II.

References

1. RFC 2743: Generic Security Service Application Program Interface, Version 2. http://www.rfc-archive.org/getrfc.php?rfc=2078 (11/2005)
2. RFC 2743: Generic Security Service Application Program Interface, Version 2, Update 1. http://www.rfc-archive.org/getrfc.php?rfc=2743 (11/2005)
3. RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL). http://www.rfc-archive.org/getrfc.php?rfc=3280 (11/2005)
4. RFC 3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. http://www.rfc-archive.org/getrfc.php?rfc=3820 (11/2005)
5. Welch, V. – Foster, I. – Kesselman, C. – Mulmo, O.: X.509 Proxy Certificates for Dynamic Delegation. 3rd Annual PKI R&D Workshop, 2004
6. The Globus Security Team: Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective, Version 4 updated September 12, 2005. http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf (11/2005)
7. GT4 Admin Guide, April 2005. http://www.globus.org/toolkit/docs/4.0/admin/docbook/ (11/2005)
8. GT 4.0: Security. http://www.globus.org/toolkit/docs/4.0/security/key-index.html (11/2005)

Environmental assessment of climate-change driven risks in landscape-decision support tool

Rastislav Skalský, Juraj Balkovič and Erwin Schmid

Soil Science and Conservation Research Institute,
Gagarinova 10, 827 13 Bratislava, Slovakia.
University of Natural Resources and Applied Life Sciences,
Institute for Sustainable Economic Development,
Vienna, Austria
skalsky@vupu.sk

Abstract. The data-knowledge infrastructure for environmental modeling in field of agriculture and forestry has been established as a framework for generating the environmental variables needed in process of environmental assessment. Nowadays, the data-knowledge infrastructure prototype developed is being tested within the scope of EC project INSEA (Integrated Sink Enhancement Assessment, www.iiasa.ac.at/Research/FOR/INSEA/). The primary aim of environmental modeling within the project is to produce reliable environmental data (NUTS2 level) consecutively realized in socio-economical assessment models (EUFASOM) on the NUTS0 and EU25 level. The EPIC model (www.brc.tamus.edu/epic/) is employed to set the environmental variables (greenhouse gases emission/sequestration, drought stress, crop production, soil erosion, etc.) with respect to climatic scenarios and management/land use alternatives firstly predefined. The data-knowledge infrastructure defined like a modeling framework becomes substantial when it is related to geographical areas. Data origin (mostly geo-referenced data in raster data representations) and modeling time step/period assumed requires sufficient computation facilities and capabilities, when the environmental variables being modeled have to be set up on the regional or higher administrative level (national, EU). The number of individual modeling elements (individual cell number, homogenous units) as well as the number of possible combinations of input variables and time step/period defined is rapidly increasing the number of model runs and results in huge amount of data need to be post-processed.

Zameranie aplikácie

V príspevku je reprezentovaná aplikácia z oblasti životného prostredia. Aplikácia bola vyvinutá ako súčasť riešenia širšie koncipovanej úlohy zameranej na ekonomické hodnotenie potenciálu poľnohospodárstva a lesného hospodárstva pre znižovanie obsahu skleníkových plynov (GHG) v atmosfére. Hodnotený je najmä dlhodobý vplyv rôzneho spôsobu využívania krajiny na znižovanie emisií skleníkových plynov do atmosféry a vplyv využívania krajiny na mieru sequestrácie GHG v pôde a

biomase. Podobné aktivity majú základ v dokumentoch „Towards a Thematic Strategy for Soil Protection“ (EC 2002), Kyotskom protokole ku konvencii o klimatickej zmene (UNFCCC 1998) a dokumentácii k LULUCF (tzv. Land Use, Land-Use Changes, and Forestry, IPCC 2000).

Aplikácia vychádza z predpokladu, že dlhodobé zmeny v kvalite krajiny môžu byť sledované prostredníctvom zmien zvolených environmentálnych indikátorov. V prípade hodnotenia potenciálnej sekvestrácie GHG v rezervoároch poľnohospodárskej krajiny boli zvolené indikátory zmien v hromadení organickej hmoty v pôde a miera urýchlenej erózie pôdy. Tieto indikátory však nie je možné extrapolovať do budúcnosti s ohľadom na to, že predstavujú príliš komplexný jav v krajine. Z tohto dôvodu sa javí vhodné modelovať environmentálne indikátory prostredníctvom účelových modelov. V súčasnosti existuje niekoľko efektívnych programov, ktoré implementujú problematiku uhlíkového cyklu a erózie pôdy do komplexného hodnotenia systému hospodárenia v poľnohospodárskej krajine. Pre navrhnutú aplikáciu bol vybraný model EPIC (Williams et Renard 1985, Sharpley et Williams 1990a,b) – The Erosion Productivity Impact Calculator vyvinutý v USDA. Model EPIC vychádza z viacerých modelov, napr. CREMS (Knisel 1980), SWRRB (Williams et al. 1985), GLEAMS (Leonard et al. 1987) a CENTURY (Parton et al. 1994). Postupné rozširovanie aplikácií vyústil do zmeny názvu na „Environmental Policy Integrated Climate” (Williams et al. 1996) – www.brc.tamus.edu/epic.
Z hľadiska prezentovanej aplikácie bolo pre modelovanie environmentálnych indikátorov potrebné zabezpečiť výstupy (informácie), ktoré sú:
(i) geografické, t.j. informácie, ktoré majú jednoznačne definovanú väzbu na geografický priestor,
(ii) kvantitatívne, t.j. informácie, ktoré umožňujú jednoznačne kvantifikovať sledované parametre (ich množstvo, rozsah, a pod.),
(iii) spĺňajú určité kvalitatívne štandardy,
(iv) časové, t.j. informácie s časovým priebehom (informácie vzťahované k vybraným časovým obdobiam, v čase extrapolované informácie),
(v) informácie o alternatívach, t.j. informácie, ktoré reflektujú na možný alternatívny vývoj sledovaných parametrov vplyvom rôznych vstupných podmienok.

Predkladaný príspevok je zameraný na priblíženie všeobecných aspektov architektúry vytvorenej aplikácie pre modelovanie environmentálnych indikátorov, na prezentáciu postavenia tejto aplikácie v rámci širšie navrhnutého systému socio-ekonomického hodnotenia využívania krajiny a diskusiu o možnostiach jej ďalšieho vývoja smerom k využívaniu distribuovaných výpočtových zdrojov.

Všeobecné poznámky k systému modelovania environmentálnych indikátorov

Pojem systém modelovania v tomto príspevku chápeme ako systém, ktorý je vytvorený z troch základných prvkov, ktorými sú (i) požiadavky na nové informácie (resp. údaje a nové znalosti), (ii) údaje tvoriace vstupy pre modelovanie a (iii) znalosti (Fig. 1.).

Požiadavky na nové informácie v tomto systéme predstavujú prvotný impulz, na základe ktorého je definovaný rámec v ktorom bude vytvorená architektúra celého

systému. Požiadavky môžu byť definované na základe potrieb výskumu v danej oblasti alebo môžu vyplývať z potrieb organizácie a riadenia na rôznej úrovni.

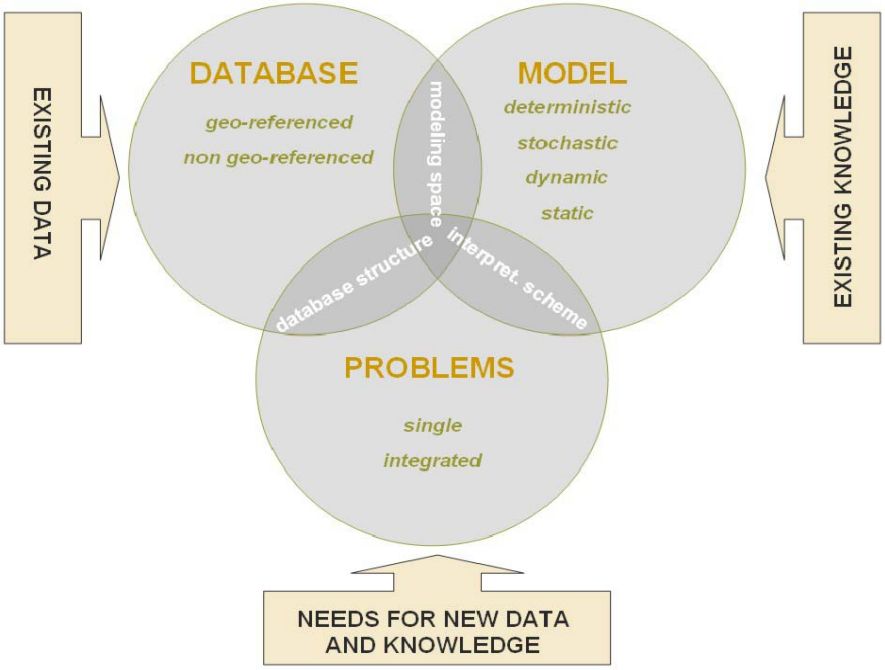


Fig. 1. Všeobecná schéma systému modelovania environmentálnych indikátorov (General scheme of system for modeling environmental indicators.)

Údaje sú prvkom systému modelovania, ktorý v zásadnej miere ovplyvňuje kvalitu výsledkov modelovania. Do procesu modelovnia zameraného na krajinný priestor vstupujú okrem klasicky ponímaných údajov (údaje bez vzťahu k priestoru) aj údaje priestorové. V špeciálnom prípade priestorových údajov vzťahujúcim sa k povrchu zeme hovoríme o georeferencovaných údajoch. Georeferencované údaje na rozdiel od klasicky ponímaných údajov obsahujú v údajovej báze okrem kvalitatívnych atribútov aj údaje o geometrii prvkov, geografickej lokalizácii prvkov a údaje o vzájomných topologických vzťahoch medzi týmito prvkami.

Znalosti sú v našom ponímaní systému modelovania definované ako účelovo reprezentované poznatky o správaní sa systémov, ktoré sú objektom záujmu. Poznatky sú reprezentované vo forme matematických modelov. Z hľadiska prezentovanej aplikácie sú tieto modely (i) deterministické a (ii) dynamické, t.j. sledujú časový vývoj stavových veličín. Modely sú pre systém modelovania environmentálnych indikátorov dostupné vo forme účelových aplikácií fungujúcich pod bežnými operačnými systémami.

Vzťahy medzi jednotlivými prvkami systému modelovania sú definované prostredníctvom štruktúry údajovej bázy (vzťah medzi požiadavkami a dostupnými údajmi), interpretačného rozhrania (vzťah medzi existujúcimi znalosťami a

požiadavkami) a prostredníctvom modelovacieho priestoru (vzťah medzi dostupnými údajmi a existujúcimi znalosťami)

Modelovací priestor definujeme ako abstraktný priestor vytvorený dostupnými údajmi. V tomto priestore je aplikovaný zvolený model. Modelovací priestor plní v systéme modelovania funkciu (i) atribútového priestoru, t.j. priestoru, ktorý zabezpečuje atribútové údaje zodpovedajúce požiadavkám aplikovaného modelu na vstupy, pričom údaje sú uložené v zodpovedajúcej forme/štruktúre a (ii) geografického priestoru, t.j. priestoru pre aplikáciu modelu, ktorý zabezpečuje väzbu modelovaných premenných k danému geograficky explicitne vyjadrenému priestorovému objektu na úrovni základnej ďalej nedeliteľnej priestorovej jednotky (elementárna modelovacia jednotka).

Architektúra aplikácie

Na obrázku (Fig. 2.) je schematicky znázornená architektúra systému pre modelovanie environmentálnych indikátorov. Prvky vytvorenej architektúry sú reprezentované (i) jednotlivými vstupmi (vstupné údaje), (ii) nástrojmi transformácie a tvorby nových údajov (model), (iii) nástrojmi komunikácie medzi údajovou bázou a modelom a (iv) výstupmi.

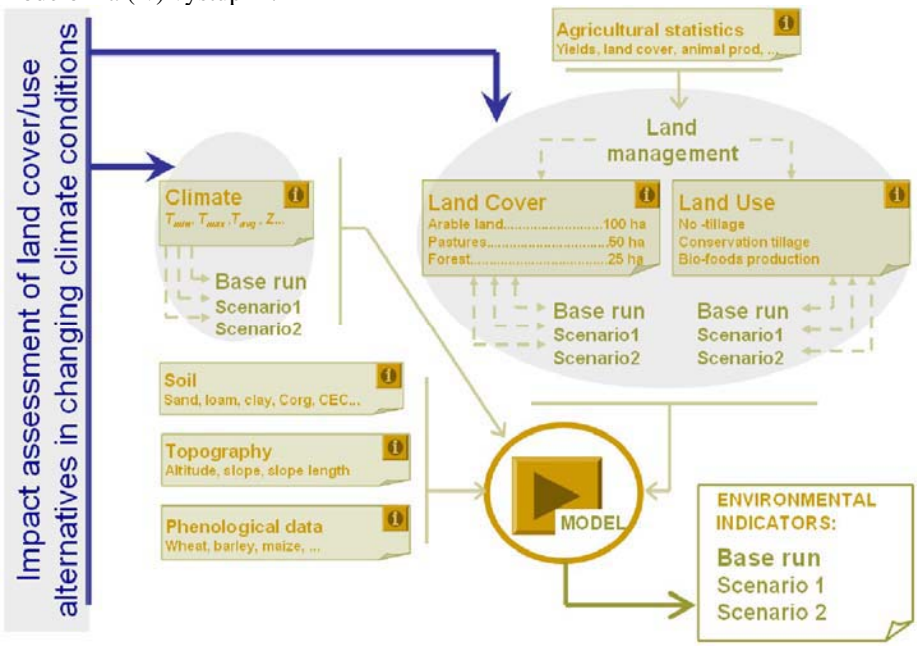


Fig. 2. Architektúra aplikácie pre modelovanie environmentálnych indikátorov (Environmental indicators modeling - application scheme.)

Georeferencované údaje sú v tejto architektúre reprezentované údajmi o (i) topografii územia, (ii) pôde/pôdnej pokrývke, (iii) klíme a (iv) krajinnej pokrývke a administratívnej jednotke. Georeferencované údaje vytvárajú geografický priestor v ktorom je aplikovaný zvolený model. Atribútový priestor pre aplikáciu modelu je zabezpečený kvalitatívnymi atribútmi priestorových prvkov (údaje o hodnotách parametrov klímy, parametrov pôdy, a pod) a údajmi bez priamej priestorovej referencie (štatistické údaje o poľnohospodárskej a lesnej produkcii, údaje o spotrebe hnojív a pesticídov, údaje o spôsobe hospodárenia a pod.).

Vzťah prvkov s nepriamou priestorovou väzbou k elementom geografického priestoru je definovaný prostredníctvom známej väzby na niektorý z priestorových prvkov (napr. štatistiky o poľnohospodárskej produkcii platné pre administratívne alebo manažérske jednotky, údaje o využívaní krajiny platné pre vybraný typ krajinnej pokrývky a zároveň daný administratívny región a pod.).

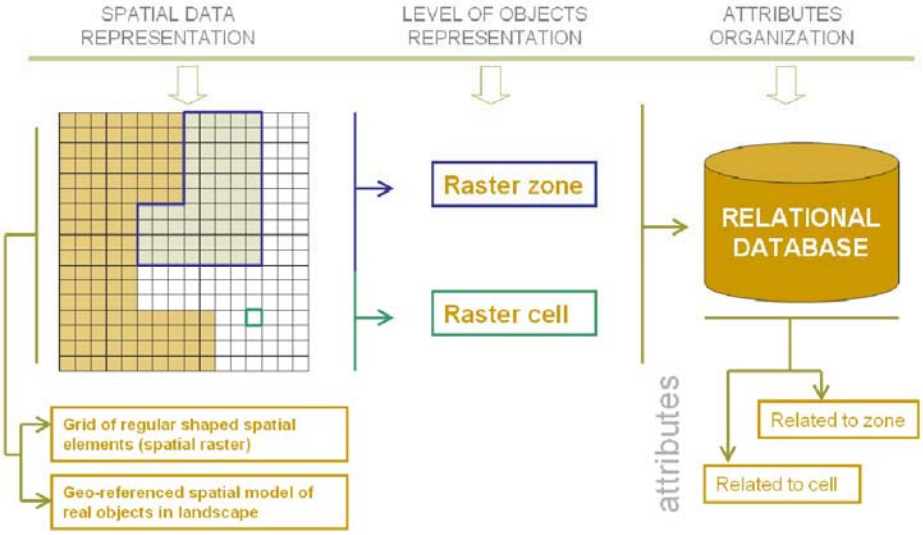


Fig. 3. Organizácia priestorových a atribútových údajov. (Spatial and attribute data organisation.)

Na obrázku (Fig. 3.) je schematicky znázornená organizácia priestorových údajov a vzájomná väzba priestorových a atribútových údajov. Priestorový element prostredníctvom ktorého sú reprezentované georeferencované údaje je pravidelný priestorový element štvorcového tvaru (pixel). Množina všetkých priestorových elementov vytvára priestorový rámec pre reprezentáciu priestorových údajov na najnižšej úrovni. Vzhľadom na praktické dôvody (úroveň detailu vstupných údajov, existujúce väzby z atribútovými údajmi, výpočtové možnosti) sú však jednotlivé priestorové elementy na základe predchádzajúcej geografickej analýzy organizované

do tzv. zón, ktoré reprezentujú účelové priestorové prvky vyššieho rádu (element geografického priestoru, elementárny modelovací priestorový element). Ku každému elementu geografického priestoru je vytvorená väzba na atribútové údaje. Atribútové údaje môžu byť organizované v samostatnej, štruktúrovanej relačnej databáze.

Modelovací balík EPIC bol vyvinutý pre potreby simulácie vplyvu rôznych systémov hospodárenia na pôde, pestovanie rôznych plodín na rôzne zložky krajiny (voda, živiny, erózia pôdy, organická hmota pôdy, produkcia biomasy a pod.). Na obrázku (Fig. 4) je schematicky znázornený modelovaný systém. Modelovaná je časová bilancia vstupov a výstupov v systéme atmosféra – rastlina (biomasa) – pôda – pôdna/podzemná voda. Bilancia je rátaná s denným krokom pre ľubovoľný väčší časový interval. Územne je bilancia rátaná pre homogénny priestorový element, ktorý je homogénny z hľadiska hodnôt parametrov vstupov alebo javu v krajine.

Vzájomný vzťah medzi údajmi a modelom je zabezpečený vstupnými údajovými súbormi. Vstupné údajové súbory zabezpečujú špecifickú organizáciu atribútových údajov vhodnú pre zadávanie vstupných parametrov pre model. Rovnako výstupy z modelu sú špecificky organizované vo výstupných súboroch. Údajová štruktúra vstupov a výstupov však nie je zhodná s údajovou štruktúrou databázy, v ktorej sú organizované vstupné údaje a preto je nevyhnutné zabezpečiť nástroje na bezproblémovú výmenu údajov medzi údajovou bázou a modelom (tzv. technologické rozhranie medzi údajmi a modelom, ktoré zabezpečuje transformáciu vstupných a výstupných formátov, pričom nijakým spôsobom neovplyvňuje samotný obsah transformovaných údajov.).

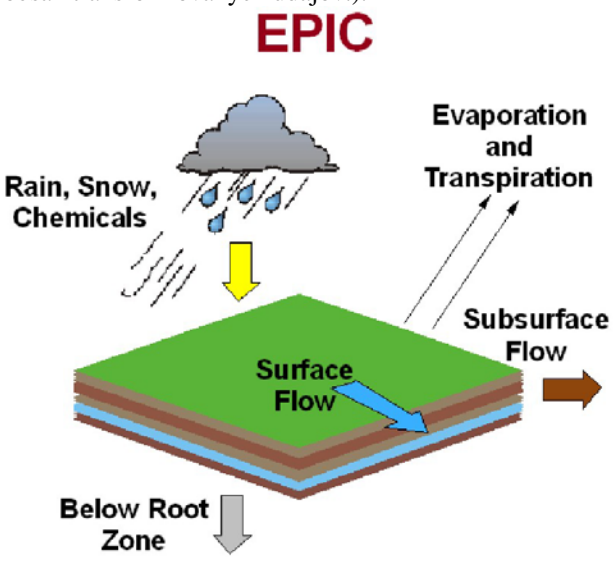


Fig. 4. Model EPIC, všeobecný princíp modelovania krajinného systému (EPIC model, general principle of landscape system modelling).

Výstupy modelu sú organizované vo forme časových radov pre jednotlivé modelované parametre. Časový krok v rámci výstupnej časovej rady môže byť

zhodný s minimálnym časovým krokom samotného modelu (24 h v prípade prezentovanej aplikácie), alebo časovým krokom rôzne agregovaným (týždeň, mesiac, rok, desaťročie). Výstupy modelovania môžu byť priestorovo reprezentované prostredníctvom väzby na priestorové elementy geografického priestoru.

Architektúra aplikácie je navrhnutá a vytvorená tak, aby umožňovala bezproblémovo modelovať rôzne alternatívy správania sa systému vedúce k rôznym výstupným hodnotám sledovaných environmentálnych indikátorov. Simulácia alternatívneho chovania krajinného systému je v rámci vytvorenej architektúry riešená prostredníctvom aplikácie rôznych scenárov definovaných pre záujmové územie – napr. klimatický scenár, protierózny scenár hospodárenia na pôde, bezorbové scenáre a pod. Navrhnuté scenáre sú realizované vo forme účelovej zmeny (priestorovej či kvalitatívnej) vybraných vstupných údajov.

Postavenie aplikácie v rámci širšie koncipovanej úlohy

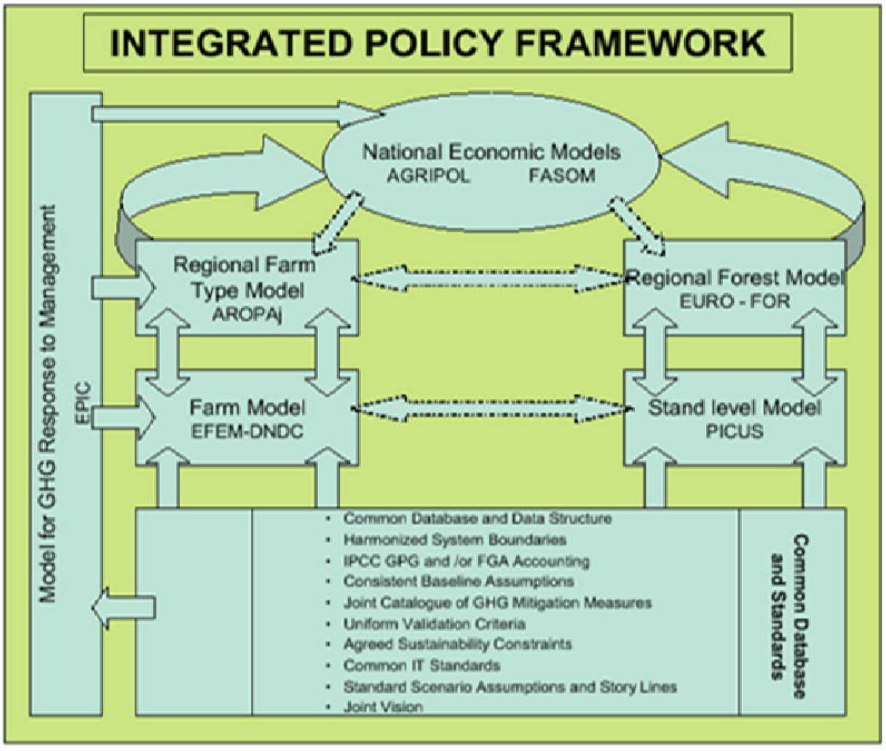


Fig. 5. Postavenie prezentovanej aplikácie v rámci širšie koncipovanej úlohy – projekt INSEA (Position of presented application within wider defined problems – INSEA project).

V súčasnom období je prezentovaný systém modelovania environmentálnych indikátorov aplikovaný a testovaný ako súčasť širšie koncipovaných úloh riešených v

rámci medzinárodného projektu INSEA (Integrated Carbon Sink Enhancement Assessment, www.iiasa.ac.at/Research/FOR/INSEA). Základným cieľom systému modelovania environmentálnych indikátorov je zabezpečiť vhodné vstupy pre socio-ekonomické modely, ktoré zabezpečujú účelové hodnotenie navrhnutých opatrení pre znižovanie emisií GHG a pre zvyšovanie ich sekvestrácie v rezervoároch krajiny. Socio-ekonomické hodnotenie je realizované na úrovni administratívnych regiónov (NUTS2), národnej úrovni (NUTS0) a európskej úrovni (EU25)

Na obrázku (Fig. 5.) je znázornená výmena údajov v rámci projektu INSEA. Vzájomná väzba jednotlivých modelov pre environmentálne indikátory (EPIC, PICUS, EURO-FOR) a modelov pre socio-ekonomickú evaluáciu (AROPA, EFEM-DNDC, AGRIPOL, FASOM) je v rámci projektu INSEA zabezpečená priamou výmenou údajov a znalostí medzi modelmi a nepriamo aj prostredníctvom spoločnej údajovej bázy. Postavenie prezentovanej aplikácie je znázornené obdĺžnikom v ľavej časti obrázku (Model for GHG response to management). Modelovanie environmentálnych indikátorov predstavuje v rámci projektu INSEA hierarchicky najnižšiu úroveň tvorby výstupov a zároveň má prierezový charakter.

Záverečné poznámky

Prezentovaný systém modelovania environmentálnych indikátorov (samostatne alebo ako súčasť širšie koncipovaných úloh) predstavuje perspektívnu aplikáciu pre implementáciu do gridového prostredia a to aj napriek tomu, že súčasné nároky tejto aplikácie na výpočtové zdroje nie sú kritické.

Je však zrejmé, že rozvoj metód získavania geografických, kvantitatívnych a dynamických informácií, stále sa zvyšujúce požiadavky na úroveň detailu v modeloch krajinného systému, ako aj požiadavky na komplexnosť sledovaných vzťahov v krajine budú do budúcnosti predpokladať výrazne vyššie nároky na výpočtovú kapacitu. Rovnako treba zdôrazniť požiadavky na vizulizáciu časovo-priestorového priebehu environmentálnych indikátorov.

Literatúry

1. EC, 2002. Communication of 16 April 2002 from the Commission to the Council, The European Parliament, the Economic and Social Committee and the Committee of the Regions – Toward a Thematic Strategy for Soil Protection.
2. Intergovernmental Panel on Climate Change (IPCC), 2000. Watson R., I., Noble, B. Bolin, N., Rahindranath, D., Verardo, D., Dokken (Eds). Land Use, Land-Use Change, and Forestry. Cambridge University Press, Cambridge, UK, 377 pp.
3. Knisel, W.G., 1980. A field scale model for chemicals, runoff, and erosion from agricultural management systems. U.S. Dept. Agric. Conserv. Res. Rept. No.26.
4. Leonard, R.A., Knisel, W.G., Still, D.A., 1987. GLEAMS: Ground water loading effects on agricultural management systems. Trans. ASAE, 30, 5, 1403-1428.

5. Parton, W.J., D.S. Ojima, Cole, C.V., Schimel, D.S., 1994. "A general model for soil organic matter dynamics: sensitivity to litter chemistry, texture and management". In Quantitative modeling of soil forming processes, SSSA Spec. Public., Madison, WI: SSSA, No. 39, p. 147-167.
6. Sharpley, A.N., Williams, J.R. (eds.), 1990a. EPIC – Erosion/productivity impact calculator: 1. Model documentation. USDA Technical Bulletin No. 1768, Washington DC, 235 pp.
7. Sharpley, A.N., Williams, J.R. (eds.), 1990b. EPIC – Erosion/productivity impact calculator: 2. User Manual. USDA Technical Bulletin No. 1768, Washington DC, 127 pp.
8. UNFCCC, 1998. Report of the Conference of the Parties on its Third Session, Held at Kyoto from 1 to 11 December 1997. Addendum. Document FCCC/CP/1997/7/Add1, http://www.unfccc.de
9. Williams, J.R. et al., 1996. Using soil erosion models for global change studies. Journal Soil and Water Cons., 51, 5, p. 381-385.
10. Williams, J.R., Nicks, A.D., Arnold, J.G., 1985. SWRRB, a simulator for water resources in rural basins. Journal ASCE Hydr., 111, 6, p. 970-986.
11. Williams, J.R., Renard, K.G., 1985. Assessment of soil erosion and crop productivity with process models (EPIC). In: Follett, R.F. and Stewart B.A. (eds.). Soil erosion and Crop Productivity, ASA/CSSA/SSSA, Madison, WI, p. 67-103.

Grids and User Applications

Viera Šipková

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
viera.sipkova@savba.sk

Abstract. Collaborative science requires access to large, heterogeneous data collections, large-scale computing resources, high-performance networking and visualization. These environments combine aspects of traditional distributed and parallel computing systems - Grid systems. In this document we give a concise answer on the question: What is the Grid and Grid computing?. First, the Grid concepts and technologies will be introduced. Most of the definitions are undertaken from the publications of the fathers of the Grid vision: Ian Foster and Carl Kesselman. In the second Section we describe briefly the Grid architecture. The third Section is focused on the Globus Toolkit 4 (GT4) which supports the development of service oriented distributed applications and infrastructures. The Section 4 makes analysis of computational problems, and outlines some categories of them from gridifying point of view. The last Section presents some examples of how a computational task can be submitted to run on a Grid using the GT4 infrastructure.
This work was made out from several documents originated within the Globus Alliance community [2], and EGEE [16] project.

Introduction

The Grid: *A New Infrastructure for 21st Century Science* [6]. The term Grid was coined by Ian Foster and Carl Kesselman (Grid bible [1]). Whereas the World Wide Web provides the seamless access to information stored in millions of different geographical locations, the Grid is an emerging infrastructure that provides seamless access to computing power and data storage capacity distributed over the globe. By providing scalable, secure, high-performance mechanisms for discovering and negotiating access to remote resources, the Grid promises to make it possible for scientific collaborations to share resources on an unprecedented scale, and for geographically distributed groups to work together in ways that were previously impossible [4, 5]. A special effort for the implementation of Grids comes from the fact that resources exist within different administrative domains, they run different software, and they are subject to different security and access control policies. A set of disparate organizations and/or individuals which are enabled to share resources in a controlled fashion form so called Virtual Organization (VO).

Much of the worldwide activity on Grids these days is directed to the issue of common, open standards, so that applications made to run on one Grid will run on all

others. Both the Internet, and the Web have key standards such as TCP/IP and HTTP, which have been critical for the progress in these communities. These standards have been set by standards bodies, the IETF [8] for the Internet, and W3C [13] is one for the Web. Grid-specific standards are currently being developed by the Global Grid Forum [9], known as Open Grid Services Architecture (OGSA) [17], which defines the fundamental components of the Grid, describes their purpose and function, and indicates how these components should interact with one another. The goal of OGSA is to standardize practically all the services one commonly finds in a grid application by specifying a set of standard behaviors and interfaces for what could be termed a Grid service: a Web service that can be created dynamically and that supports security, lifetime management, resource manageability, and other functions required in Grid scenarios.

Grid Architecture

The architecture of the Grid is often described in terms of the *hourglass* model, a series of *layers* of different widths, each providing a specific function.

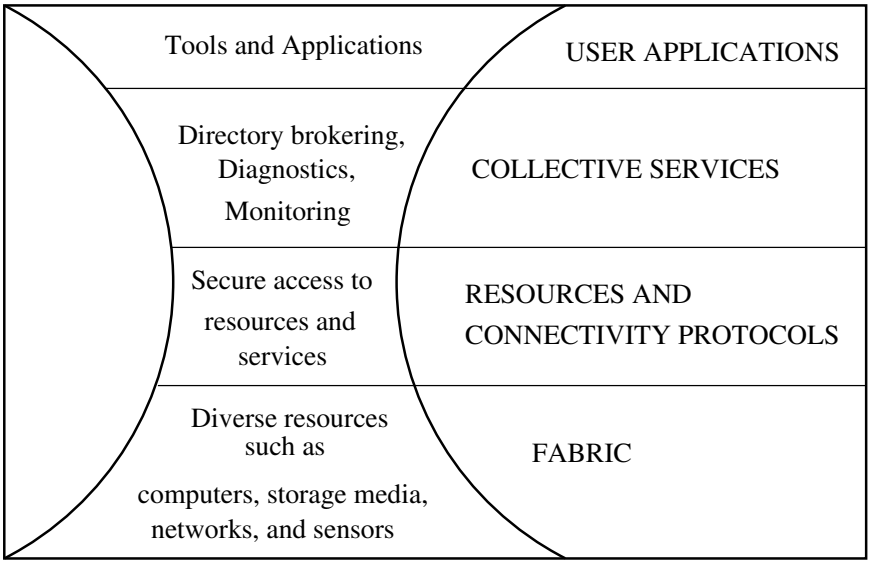


Fig. 1. Grid Architecture

As Fig. 1 shows schematically, the lowest level, the *fabric*, constitutes the physical infrastructure - devices or resources that Grid users want to share and access, including computers, storage systems, catalogs, networks, and various forms of sensors.

Above the fabric, within the middleware layer, there are the *connectivity* and *resource* tiers. Protocols in this layer must be implemented everywhere, and therefore,

they must be relatively small in number. The connectivity tier contains the core communication and authentication protocols required for Grid-specific network transactions. Communication protocols enable the exchange of data between resources, whereas authentication protocols provide cryptographically secure mechanisms for verifying the identity of both users and resources. The resource tier contains protocols that exploit communication and authentication protocols to enable the secure initiation, monitoring, and control of resource-sharing operations.

The *collective services* are based on protocols, services, and APIs that implement interactions across collections of resources. They combine and exploit components from the resource and connectivity layer and implement a wide variety of tasks. Examples of collective services include directory and brokering services for resource discovery and allocation, monitoring and diagnostic services, data replication services, and others.

The top layer of any Grid system comprises the *user applications* and tools operating within a VO. They are constructed in terms of, and call on, the components in any other layer. Generally, an application that needs to execute several independent tasks, might proceed by:

- *Obtaining* necessary authentication credentials.
- *Querying* an information system and replica catalog to determine availability of computers, storage systems, and networks, the location of required input files, or services.
- *Submitting* requests to appropriate computers, storage systems, and networks to initiate computations or move data.
- *Monitoring* the progress of the various computations and data transfers, notifying the user when all are completed, and detecting and responding to failure conditions.

In order to do all of the above, it is clear that an application that a user may have written to run on a stand-alone PC will have to be adapted in order to invoke all the right services and use all the right protocols. So, the running an application on the Grid will require users to invest some effort into the *gridifying* it. However, once gridified, thousands of people will be able to use the same application and run it trouble-free on the Grid.

Globus Toolkit

Practically all major Grid projects are being built on protocols and services provided by the Globus Toolkit [3], a software work-in-progress which is being developed by the Globus Alliance [2]. The Globus Alliance is a community of organizations and individuals developing fundamental technologies behind the Grid. The last version, Globus Toolkit 4 (GT4) [7] is a set of software components for building *distributed systems*: systems in which diverse and discrete software agents interact via message exchanges over a network to perform some tasks. More specifically, GT4 software components make extensive use of *Web services* mechanisms to define interfaces and structure of its components. Web services provide a standard means of interoperating between different software applications running on a variety of platforms and/or

frameworks. They implement flexible, extensible, and widely adopted XML-based mechanisms for describing, discovering, and invoking network services. These document-oriented protocols are well suited to the loosely coupled interactions - *service-oriented architectures*, in which service interfaces are described, operations invoked, access secured, etc., all in uniform way. While end-user applications are typically concerned with domain-specific operations, distributed computing ultimately requires the manipulation and management of infrastructure. GT4 provides a set of *Grid infrastructure services* that implement interfaces for managing computational, storage, and other resources.

The GT4 software itself consists of a variety of components and capabilities which are aggregated into five major modules: security, data management, execution management, information services, and common runtime components.

1. *Security Tools*: Security tools are concerned with protecting communications, establishing the identity of users or services (*authentication*), and determining who is allowed to perform what actions (*authorization*), as well as with supporting functions such as managing user credentials and maintaining group membership information.
2. *Data Management:* Data management tools are concerned with the location, transfer, and management of distributed data. GT4's primary data management tools are: *GridFTP* for high-performance and reliable data transport, *Reliable File Transfer* service for managing multiple transfers, and *Replica Location Service* for maintaining location information for replicated files.
3. *Execution Management*: Execution management tools are concerned with the initiation, monitoring, management, scheduling, and/or coordination of remote computations. GT4 supports the *Grid Resource Allocation and Management* (GRAM) interface as a basic mechanism for these purposes. The GT4 GRAM server is typically deployed in conjunction with the Delegation and RFT service to address data staging, delegation of proxy credentials, and computation monitoring and management in an integrated manner.
4. *Information Services:* The *Monitoring and Discovery System* (MDS) is a suite of web services to monitor and discover resources and services on Grids. This system allows users to discover what resources is part of a VO and to monitor those resources. MDS services provide query and subscription interfaces to arbitrarily detailed resource data and a trigger interface that can be configured to take action when pre-configured trouble conditions are met.
5. *Common Runtime Components:* The common runtime components provide GT4 web services with a set of libraries and tools that allows these services to be platform independent, to build on various abstraction layers, and to leverage functionality lower in the web services stack. These components are architecturally diverse and it is thus hard to identify a common structure. One of few sub-themes is *Java WS Core* which implements specifications: *WS-Addressing* (WSA) [14], *WS Resource Framework* (WSRF) [11], *WS-Notification* (WSN) [12] and others. The WSA specification defines transport-neutral mechanisms to address Web services and messages. The WSRF specifications define a generic and open framework for modeling and accessing stateful resources using Web services. The WSN specifications define a pattern-based approach allowing Web services to

disseminate information to one another. This framework comprises mechanisms for notifications.

To describe all of the components in more detail is beyond the scope of this paper. In the next Sections we focus primarily on the user applications, especially, on the *gridifying* computational problems, and on the job execution management using the GT4 GRAM.

Computational problems

Let's say you have a computational problem which is simply too complex to be executed on just one computer. Difficult computational problems can be categorized in different ways. From a computer scientist's point of view, you can divide them by looking at their general computational nature, or you categorize problems by whether they are computation-centric, data centric or community-centric.

One way of categorizing a computational problem in computer science is by its *degree of parallelism*. If your problem can be split into many smaller sub-problems that can be worked on by different processors in parallel, then you can speed up your computation a lot by using many computer resources from across a business, a company or an academic institution. The network of computers is then used as a single, unified resource. This solution is called *distributed computing*, and this term refers to just about any system where many computers solve a problem together. *The Grid computing*, in a sense, just one species of distributed computing, there are many others (peer-to-peer computing, internet computing, etc).

Another category is the *granularity* of the problem. If each sub-problem is highly dependent on the results of other sub-problems, you are dealing with *fine-grained* parallel calculations. In practice, fine-grained parallel calculations require very clever programming to expose the most of their parallelism so that the right information is available to processors at the right time. The fine-grained calculations are the best suited to a big, monolithic supercomputers, cluster of computers, or at least very tightly coupled clusters of computers, which have lots of identical processors with an extremely fast, reliable network between the processors to ensure there are no bottlenecks in communications. This type of computing is often referred to as *high-performance computing* or *supercomputing*. For the development of applications for high-performance computing systems mostly a well-known low-level parallel programming model - *Message Passing Interface* (MPI), has been applied. The MPI standard specification defines a library of routines that implement the message-passing model. The MPICH [20] is a highly portable implementation of the MPI-1 standard [18] with extensions defined in MPI-2 standard [19] for a wide variety of parallel and distributed computing environments.

At the opposite end of the granularity scale are *coarse-grained* or *embarrassingly* parallel calculations, where each sub-problem is relatively independent of all others. Such calculations are ideal for a more loosely-coupled network of computers, since

delays in getting results from one processor will not affect the work of the others. These types of calculations are often referred to as *high-throughput computing*. For such computing specialized, often high-level distributed, programming models are applied, including various forms of object systems, Web technologies, CORBA, workflow systems, or compiler based systems. The MPICH-G2 - Grid-enabled MPI [21], is a complete implementation of the MPI-1 standard that uses Globus Toolkit services to support efficient and transparent execution in heterogeneous Grid environments.

At first sight, it seems that the Grid is only good for high-throughput computing. But in fact, many of the challenging problems in science require a combination of fine- and coarse-grained approaches. And this is where the Grid can be particularly powerful. In this case many independent calculations which are itself fine-grained, could be distributed over many different clusters on the Grid, thus adding coarse-grained parallelism and saving a lot of time.

Computer-centric problems are the domain of high performance computing. Many computer-centric applications can benefit from the Grid to combine large computational resources in order to tackle problems that cannot be solved on a single system, or at least to do so much more quickly.

Data-centric, also called *data-intensive* problems, is the primary driving force behind the Grid at present, and will likely continue to be for some time in the future. Over the next decade, huge amounts of scientific data will come from everywhere. The Grid will be used to collect, store and analyze data maintained in geographically distributed repositories, digital libraries, and databases.

Community-centric problems, also referred to as *collaborative applications*, are concerned primarily with enabling and enhancing human-to-human interactions, attempting to bring people or communities together for collaborations of various types. They are often structured in terms of a VO which enables the shared use of computational resources such as data archives and complex simulations.

GT4 Execution Management

GT4 comprises a collection of software components, each providing services, associates APIs and *client* libraries that support the development of grid applications. In particular, for execution management, GT4 contains the *Grid Resource Allocation and Management* (GRAM) interface as a basic mechanism for the initiation, monitoring, management, scheduling, and/or coordination of remote computations. The WS GRAM implementation is built on Web services technologies, where the WSRF core is used to implement distributed communications and service state.

Let's suppose you want, for example, to:

- run an executable on a remote computer,
- run a parallel program across multiple distributed computers,
- run a set of loosely coupled tasks,
- make a program available as a network service.

These tasks all fall within the scope of job management. In GT4 jobs represent computational tasks (executables, scripts), which may perform input/output operations while running that affect the state of the computational resource and its associated file systems. In practice, such jobs require coordinated staging of data into the resource prior to job execution, and out of the resource following execution. Users benefit also from accessing output data files as the job is running. Monitoring consists of querying and subscribing for status information such as job state changes. Some jobs may be parallel, meaning that they consist of more than one simultaneous task. GRAM provides a mechanism for task rendezvous (processes coordination) which may be used directly by application code or by intervening middleware libraries.

The new job description XML schema *Resource Specification Language* (RSL), representing the job to be submitted, allows to specify either a *single* job, or a *multi-job*, i.e. a job that is itself composed of several executable jobs. In order to execute and monitor the resources of the compute element an optional local job scheduler is required. Currently, GRAM services provide secure job submission to PBS [24], LSF [22], and Condor [23] schedulers. Scheduler adapters translate the WS GRAM job description document into commands understood by the local scheduler. For execution of simple time-sharing jobs a standard Unix Fork method is used.

GRAM is based on component architecture at both the protocol and software implementation levels. The job management with GRAM makes use of multiple types of the following services:

- *Job management services:* represent, monitor, and control the overall job life cycle. WS GRAM allows for remote execution and management of programs through the creation of a managed job. Each submitted job is exposed as a distinct resource qualifying the generic *ManagedJob* service which provides an interface to monitor the status of the job or to terminate the job. Each compute element, as accessed through a local scheduler, is exposed as a distinct resource qualifying the generic *ManagedJobFactory* service which provides an interface to create ManagedJob-resources of the appropriate type. A WS GRAM client must create a job that will then go through a life cycle where it eventually completes execution and the resource is then eventually destroyed.
- *Credential management services:* are used to control the delegation of rights among distributed elements of the GRAM architecture based on users' application requirements. WS GRAM utilizes the WSRF functionality to provide for authentication of job management requests as well as to protect job requests from malicious interference. The user may delegate some of its rights to WS GRAM and related services in order to facilitate file staging, and also for use by the job process itself.
- *File transfer services:* provides for reliable, high-performance transfers of files between the compute resources and external data storage elements before and after the job execution. GridFTP servers are required to access remote storage elements as well as file systems accessible to the local compute elements that may host the job. The *ReliableFileTransfer* service is invoked by WS GRAM services to effect file staging before and after job computations.

Job Submission Usage Scenarios

GT4 comprises both a predefined service implementations on the server side with associated APIs and also client libraries.

The official job submission client for WS GRAM is the *globusrun-ws*. It is a program for submitting and managing jobs to a local or remote job host. It provides secure job submission for users who have the right to access a job hosting resource in a Grid environment. The client uses *secure transport* for all https-endpoints and *secure message* for http.

The *globusrun-ws* client offers many additional features which may be specified by several options. For example, on-line and batch submission modes are supported; host/self/identity authorization rules may be used by the client to verify that the service is using an appropriate credential; streaming can be used to monitor the standard output and standard error files of the job; delegated credentials can be passed for use by the WS GRAM services; and also several options for job monitoring are available. In the following, some simple examples of job submitting are introduced.

Submitting a simple job

Fig. 2 shows a simple job submitting without writing a job description document. The '*-c*' option specifies that a job description will be automatically generated assuming the first argument is the executable and the remaining are job's arguments. For every job the client creates a Submission ID and outputs status information.

```
>globusrun-ws -submit -c
/test/maxcut/test/adjmatrix.dat
Submitting job...Done.
Job ID: uuid:b9a19984-55d7-11da-8b0f-000cf1bd4f9a
Termination time: 11/16/2005 12:59 GMT
Current job state: Active
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
```

Fig. 2. Job1

Submitting a simple job to a remote host

In Fig. 3 the same simple job is submitted, however, to a remote host. The remote host is specified by the '-F' option, so called contact string. If it is not supplied the default local host is taken: *https://localhost:8443/wsrf/services/ManagedJobFactoryService*

```
>globusrun-ws -submit -F
https://host:8443/wsrf/services/ManagedJobFactoryServic
e
```

```
-c /test/maxcut /test/adjmatrix.dat
Submitting job...Done.
Job ID: uuid:28e90568-55e0-11da-94df-000cf1bd4f9a
Termination time: 11/16/2005 14:00 GMT
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
```

Fig. 3. Job 2

Submitting a simple job with the job description

Fig. 5 shows submitting a job defined by the job description XML file. Let us suppose the following simple job description as given in Fig. 4. The <directory> element specifies the current directory for the execution of the command on the execution machine to be */test*, job argument and the standard output paths are specified relatively to this.

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
    <executable>/test/maxcut</executable>
    <directory>/test</directory>
    <argument>adjmatrix.dat</argument>
    <stdout>stdout.maxcut</stdout>
    <stderr>stderr.maxcut</stderr>
</job>
```

Fig. 4. Job description

```
>globusrun-ws -submit -f /test/maxcut\_desc.xml
Submitting job...Done.
Job ID: uuid:463c0212-55ec-11da-9a45-000cf1bd4f9a
Termination time: 11/16/2005 15:27 GMT
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
```

Fig. 5. Job3

Acknowledgements

This work is supported by projects EGEE EU 6FP INFSO-RI-508833, VEGA No. 2/3132/23 and VEGA No. 2/6103/6

Reference

1. The Grid 2: Blueprint for a New Computing Infrastructure. Edited by Ian Foster and Carl Kesselman. Morgan Kaufman (Elsevier), 2003.
2. Globus Alliance. http://www.globus.org/
3. Globus Toolkit. http://www.globus.org/toolkit http://www.globus.org/toolkit/docs/4.0/key/GT4 Primer 0.6.pdf
4. I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001. http://www.globus.org/alliance/publications/papers.php
5. I. Foster, C. Kesselman, J. Nick, S. Tuecke: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002. http://www.globus.org/alliance/publications/papers.php
6. Ian Foster: The Grid: A New Infrastructure for 21st Century Science. Physics Today, 55(2):42-47, 2002. http://www.aip.org/pt/vol-55/iss-2/p42.html
7. Ian Foster: Globus Toolkit Version 4: Software for Service-Oriented Systems. Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, 2005. http://www.globus.org/alliance/publications/papers.php
8. The Internet Engineering Task Force. http://www.ietf.org
9. Global Grid Forum. http://www.ggf.org
10. OASIS Consortium. http://www.oasis-open.org/
11. OASIS Web Services Resource Framework TC. http://www.oasis-open.org/committees/ home.php?wg abbrev=wsrf
12. OASIS Web Services Notification TC. http://www.oasis-open.org/committees/tc home.php?wg abbrev=wsn
13. World Wide Web Consortium. http://www.w3.org
14. W3C Web Services Addressing Working Group. http://www.w3.org/2002/ws/addr/
15. Grid Cafe. http://gridcafe.web.cern.ch/gridcafe/
16. Enabling Grids for E-sciencE. http://www.eu-egee.org
17. The Open Grid Services Architecture, Version 1.0. EGEE Deliverable DJRA1.1. https://edms.cern.ch/document/476451/1.0/architecture.pdf
18. Message Passing Interface Forum. MPI: Message Passing Interface standard. Int. Journal of Supercomputing Applications, 8(3/4), pp. 165-414, 1994.
19. Message Passing Interface Forum. MPI2: Message Passing Interface standard. Int. Journal of High Performance Computing Applications, 12(1-2), pp. 1-299, 1998.
20. W. Gropp, E. Lusk, N. Doss, and A. Skjellum: A high-performance, portable implementation of the MPI message passing interface standard. Parallel Computing 22(6), pp. 789-828, 1996.
21. N. Karonis, B. Toonen, and I. Foster: MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface. Journal of Parallel and Distributed Computing(JPDC), Vol. 63, No. 5, pp. 551-563, May 2003.
22. Platform LSF. http://www.platform.com/Products/Platform.LSF.Family/Platform.LSF/
23. Condor - High Throughput Computing. http://www.cs.wisc.edu/condor/
24. Portable Batch System. http://www.openpbs.org

Collaboration and Knowledge Sharing in Grid Applications

Michal Laclavik, Emil Gatial, Zoltan Balogh and Ladislav Hluchy

Institute of Informatics, Slovak Academy of Sciences,
Dubravska cesta 9, 845 07 Bratislava, Slovakia
laclavik.ui@savba.sk

Abstract. In this paper we describe a solution for collaboration and knowledge sharing based on text notes entered by a user. Theory, implementation and use of such system – EMBET is described. The key idea is that a user enters notes in a particular situation/context, which can be detected by the computer. Such notes are later displayed to other or the same users in a similar situation/context. The context of user is detected from computerized tasks performed by user. In the K-Wf Grid, grid services are semi-automatically composed to workflows, which should solve a user problem. It was identified that even when services and input and output data are well semantically described, there is often no possibility to compose an appropriate workflow e.g. because of missing specific input data or fulfillment of a user and application specific requirements. To help user in workflow construction it is appropriate to display notes and suggestions entered by the same or different users. Thus experts can collaborate and fill up application specific knowledge base with useful knowledge, which is shown to users in the right time. The solution was used and evaluated in the Pellucid IST project and it is further developed in the K-Wf Grid IST project.

Introduction

The experience management solutions are focused on knowledge sharing and collaboration among users in organizations. A lot of companies have recognized knowledge and experience as the most valuable assets in their organization. Experience is something that is owned by people only, not obtainable by computer systems. Any-how, according to the state of the art in the area we can a create experience management system, which will manage (not create) experience and will be able to capture, share and evaluate experience among users. We can understand experience through text notes entered by a user. Such form of experience is the most understandable for humans, but it can be grasped by a computer system, though only partially. A computer system needs to return experience in a relevant context. Thus we need to model the context of the environment and capture and store the context of each entered note. In this paper we describe such solution for the experience management based on text notes entered by users.

The key idea is that a user enters notes in a particular situation/context, which can be detected by the computer. Such notes are later displayed to other or the same users

in a similar situation/context. The context of a user can be detected from many sources - received emails, a step in a business process or a workflow management system, used files or detection of other events performed in the computer. Not entire detected context is relevant to the entered note and the system with user's assistance needs to detect a relevant context based on the text of the note. The detection of the context is based on techniques such as indexing, semantic annotation or similarity of cases. In addition, the solution uses ranking and voting mechanisms for updating relevance of the text notes. Not the entire EMBET solution is described in this article in detail. The focus is on the context detection from a new entered text note. The solution was used and evaluated in the Pellucid IST project [1] and it is further developed in the K-Wf Grid IST project [2].

The main objective of the solution is to provide a simple and powerful experience management infrastructure, which can be applicable in many areas with users sharing and collaborating through experience. The idea is to return experience to users when they need it. Therefore it is crucially important to model and capture a user context and the described solution can be used only in applications where actions/tasks performed by a user are computerized and can be captured and reported to the system in the form of events.

The EMBET system can be applied in a different application. In scope of this article we focus on its use to support grid applications particularly contraction of grid workflows. When constructing a workflow from grid services, a user needs to have knowledge about problems, services, models or input and output data. Such knowledge can be formalized only partially and workflows solving a user problem can be composed only semi-automatically with user help. Experience/notes entered by experts can help users to create the best workflow for their needs.

The article first discusses a theoretical approach of the EMBET solution and its architecture, followed by examples given for the Flood Forecasting grid application.

Theory of the Approach

According to Bergman [3], the experience management is simply the capability to collect lections from the past associated to cases. We have a person who has a problem p, which is described in a certain space, called the Problem Space (P). In the experience Management system we have Case-Lesson pairs (c, l), where we have a Case Space (C) and a lesson space (L). We could have a single multidimensional vector in which we distinguish a case part and a lesson part. To be able to collect a lesson we must first devise a problem transformation function that maps problem space to case space.

$$c = f(p) \tag{1}$$

This function should be mono valued. For the simplest cases this function is the identity function, because the developer of the experience management system (EMS) has chosen to characterize the problem with the same attributes of the case. But, even in this situation, the important fact is that we are faced with a semantic bridge. We do

not store problems (which are infinite and cannot be predicted) but cases, which are a formal representation of problems solved in the past, or situations which have happened in the past.

To be able to re-use a particular lesson we should:
1. Characterize a problem (which of course is a separate process, that requires effort)
2. Transform the problem from the space P to the space C.
3. Choose from the cases the most "useful" lesson from the case-lesson pairs stored in the database
4. Apply that lesson.

The point "Choose" introduces the function of utility that, for each problem (transformed) outputs a number, which states the utility of the use of that lesson for the problem. Unfortunately this function is not known in advance, but could be verified only AFTER the lesson is applied to the problem at hand. To overcome this, Bergman introduces the similarity function and postulates that the most useful lesson will come from the most similar case. The problem, of course, is to develop this similarity function, which is quite a complex problem.

In EMBET the described steps are recognized as:
1. User context detection from the environment which describes problem P
2. Our Model is described by ontology and Notes are stored with an associated context, which describes space C
3. Notes represent learned lesson L which is associated with space C (note context). The note context is matched with a user problem described by the detected user context. The user context is wider than the note context and as a result all applicable notes are matched and returned.
4. Applying the lesson is left to the user be reading appropriate notes.

For context modeling we are using CommonKADS [4] methodology. We are able to model and detect context when the application domain is well modeled.

The role of EMBET is to assist the user in relevant knowledge/suggestions, which are applicable to his/her current situation. This is often called “the experience management”. Thus EMBET does not create but only manage experience and is able to capture, share and evaluate experience among users of the application.

In EMBET experience is understand through notes entered by the user. Such form of experience is the most understandable for humans, but it can be partially understood by a computer system. A computer system needs to return experience in a context where experience is relevant. Thus we need to model a context of environment and capture and store a context of each entered note. From the text of the note we need to detect what kind of current context of a user is important for the note and what kind is not. We can do this with user’s assistance, where the user will get pre-checked list of the detected context and will change it or submit the detected

context unchanged. When the context is properly assigned, the note can be displayed to the same or other users in the same or similar context.

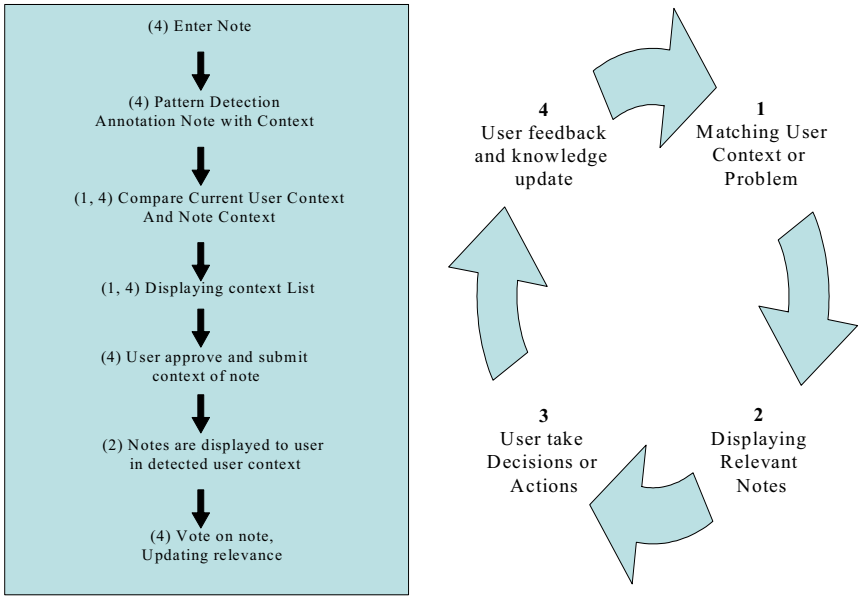


Fig. 1. Experience Management Cycle of EMBET

For the proper experience or knowledge management we need to have a closed knowledge cycle (see Figure 1, on right side). The most crucial point in experience and knowledge management systems is step 4 – "User feedback on knowledge and knowledge update". In this article we describe mainly this step as solved in EMBET system, which can be seen also in Fig. 1 on the left side. Updating of knowledge and experience in EMBET consists of the following steps:

- A user submits a Note, if s/he thinks s/he has relevant knowledge for the current or past situation
- The text of the note is processed and patterns of a semantic annotation are matched and notes are annotated with knowledge concepts. Context concepts are detected.
- Compare the current user context with the Note detected context.
- Displaying a Context List to a user with pre-selected context items detected in the Note.
- The user approves the context and submits the context of the note.
- The notes can be later displayed to users in a similar detected user context
- Users can Vote on Note relevance. The relevance of the note is then updated. If a user does not get relevant knowledge in any situation, and s/he gathers experience, s/he may enter a new note with his/her learned lesson.

The closed knowledge cycle means that experience is managed in the system the way that system together with users controls all phrases: capture capitalization and

reuse of information and knowledge and that all the phrases are closed in the system and manages by a regular user of the system. The knowledge management systems often allow a user to see or search for knowledge but introduction of knowledge and the user feedback on knowledge is complicated and can be managed by knowledge experts only.

Architecture and Infrastructure

Architecture of EMBET consists of 3 main elements:

- Core of the system
- Graphical User Interface (GUI)
- System Memory

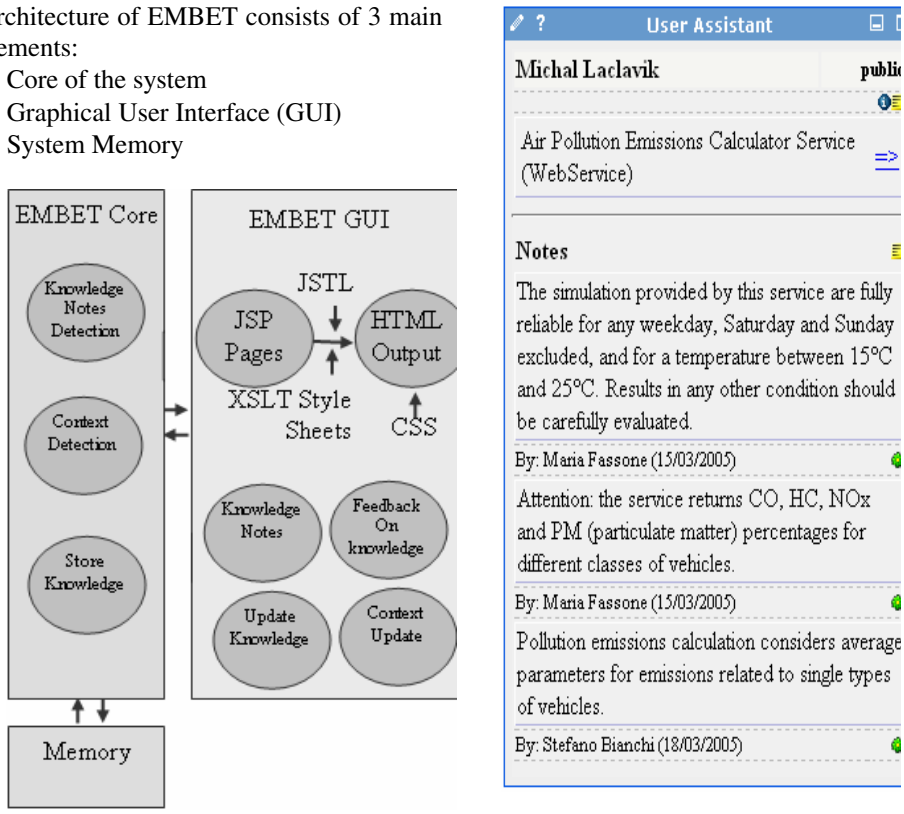


Fig. 2. EMBET Architecture and Graphical User Interface

EMBET Core provides the main functionality of EMBET. It determines a User context and searches for the best knowledge (in a form of text notes) in its Memory. The knowledge is subsequently sent through XML-RPC [5] or SOAP to EMBET GUI. When a user enters a note, the EMBET Core processes the note and determines the context of it from the current user's context and the context detected in the note. When the user confirms the context, the EMBET Core stores the note and the user's feedback. The core also handles a user the state (context).

EMBET GUI visualizes the knowledge and the user's context information to the user. Furthermore it informs the EMBET core about user context changes. The context can be reported also directly to the EMBET core from external systems (e.g. from workflow systems, received emails, or file system monitors). EMBET GUI visualizes knowledge based on XML [6] transformation to HTML through XSL [7] Templates processing. Moreover EMBET GUI has an infrastructure for a note submission and context visualization. It further provides a user with feedback (voting) on knowledge relevance. In addition it contains a user interface for knowledge management by experts where an expert can change a note and its context.

EMBET Core - EMBET GUI interface is used for an XML data exchange between EMBET Core and EMBET GUI. The Interface will be based on the SOAP protocol where both components act as web services; currently we use the XML-RPC protocol for an XML message exchange.

Interface to Memory is used for information and knowledge extraction and storage. It is based on RDF/OWL data manipulation using Jena API, which EMBET Core uses to extract and store knowledge.

Experience is represented by text notes, an unstructured text, entered by a user. For the context and environment modeling we use ontology, thus we use a Protégé ontology editor for ontology based modeling. Ontology is stored and managed in the Web Ontology Language (OWL) [8]. The Jena Semantic Web Library [9] is used for knowledge manipulation and knowledge storing. The Java technology is used for developing the system and user Interface is based on the JSP technology. The XSL templates are used to transform XML generated from OWL to displayed HTML. Since the Java technology is chosen as background for the EMBET, a choice of the web server – Jakarta Tomcat [10] and implementation middleware is reasonable. XML is a widely used language for web development; it is regarded as a universal format for structured documents and data on the Web. Therefore in EMBET XML is used in various forms/degrees from description of ontology to the communication content language. The XML/XSLT technology permits visualization of XML documents and display of information presented to the user by EMBET. The JSTL[11] is a native Java library for XML processing.

Example of Use

To better illustrate the use of EMBET in the process of user assistance, we present the following example from the K-Wf Grid project's flood forecasting application, which extends the flood prediction application of the CROSSGRID (IST-2001-32243) [12] project. The application's main core consists of simulation models series for meteorological, hydrological and hydraulic predictions. The models are organized in a cascade, with each step of the cascade being able to employ more than one model. For example, the first step - the meteorological weather prediction - can be computed using the ALADIN model, or the MM5 model.

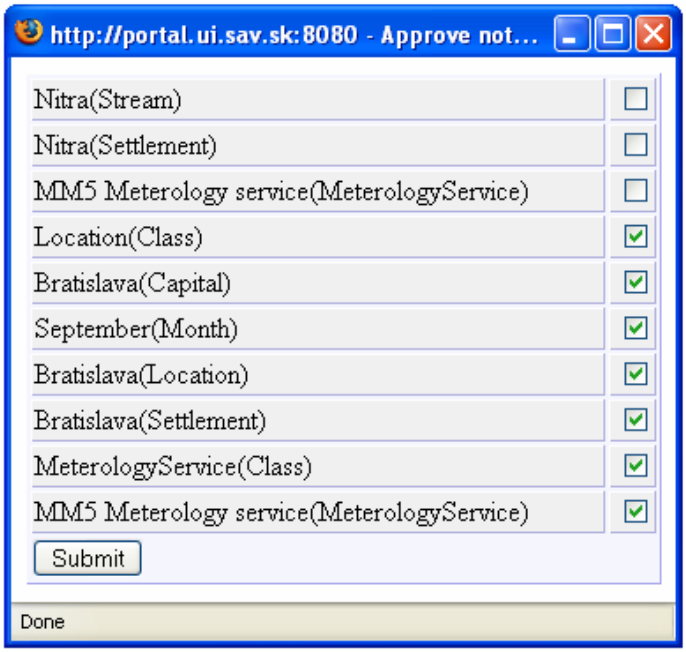


Fig. 3. Note Context Detection: Unchecked items are current user context; checked items are elements detected from text of the note. User selects only those items which are relevant.

Consider that the user has used the MM5 meteorology model and he/she wants to describe its properties (gathered knowledge), which may be relevant for other users. The proposed model of interaction is as follows:

- A user enters a note through EMBET, stating that "the MM5 model is not appropriate for weather forecast in September for Bratislava because it gives results which differ approx. 50% from reality".
- From the workflow in which the user states this note, we know directly the current user context (unchecked items on Figure 3)
- Some of current context can be relevant to note and some does not have to be. The note is processed and its text related to the context, as well as the relevant context items are selected (figure 3). In this case, by finding the text MM5 we can assume that "Service type MM5" is the relevant part of the context. There is other context relevant information which can be detected like "September", the time in which this note is valid.
- After the context detection, the user is presented with a checklist (Figure 3) where the user may select only the relevant parts of the context, which will trigger this note.
- A user gets pre-selected parts of the context, which were detected by the sys-tem as really relevant. He/she can subsequently modify the contents of the list and finally submit the note.

- Each time anyone works with the MM5 service for Bratislava area in September, the note is displayed.
- Each note can be voted by a user as being "good" or "bad" and the current results are always displayed along with the vote.
- This model gives a good basis for experience management and knowledge sharing in a virtual organization as well as for application-related collaboration among users.

Conclusion and Future Work

Our solution was evaluated on a selected administration application in the Pellucid IST project [1], where the context or the problem of a user was detected in the Workflow Management Application. Currently we are applying this solution in the K-Wf Grid IST project [2], focused on building grid-based workflows, where users need to collaborate and share knowledge about different applications, computational models, grid resources or data. In case of K-Wf Grid we detect a user problem from the grid environment. Such solution may be applied in many further areas where the user problem can be detected. Usually this is in any business process where actions are performed via a computer, e.g. workflow processes, document management, supply chain management or dynamic business processes where email communication is in use.

People like to enter notes or memos. This is a natural way of notifications for themselves or others to remind them of problems, activities or solutions. Therefore we think that such solution can be successfully applied and commercialized with good results.

In the K-Wf Grid [2] project grid services are semi-automatically composed to work-flows which should solve a user problem. It was identified that even when services and input and output data are well semantically described, there is often no possibility to compose an appropriate workflow e.g. because of missing specific input data or fulfillment of a user and application specific requirements. To help user in workflow construction it is appropriate to display notes and suggestions entered by the same or different users. Thus experts can collaborate and fill up application specific knowledge base with useful knowledge which is shown to users in the right time.

Acknowledgments

This work is supported by projects KWf-Grid EU RTD IST FP6-511385, NAZOU SPVV 1025/2004, RAPORT APVT-51-024604, VEGA No. 2/3132/23 and VEGA No. 2/6103/6.

References

1. Pellucid Consortium: Pellucid IST Project Website, 2004, http://www.sadiel.es/Europa/pellucid/
2. K –Wf Grid Consortium: K –Wf Grid IST Project Website, 2005, http://www.kwfgrid.net/
3. Ralph Bergmann: Experience Management: Foundations, Development Methodology, and Internet-Based Applications (Lecture Notes in Artificial Intelligence), 2002
4. August Schreiber et al.: Knowledge Engineering and Management: the Common-KADS methodology, ISBN: 0262193000, 2000
5. XMLRPC.com : XML-RPC Home Page, 2005, http://www.xmlrpc.com/
6. W3C : Extensible Markup Language (XML), http://www.w3.org/XML/
7. W3C : XSL Transformations (XSLT), 2005, http://www.w3.org/TR/xslt
8. W3C: Web Ontology Language OWL, 2005, http://www.w3.org/TR/owlfeatures/
9. HP Labs and Open Source Community, Jena Semantic Web Library, 2004, http://www.sf.net/
10. Apache Open source Community: The Jakarta Site – Apache Tomcat, http://jakarta.apache.org/tomcat/
11. Sun: JavaServer Pages Standard Tag Library, 2005 http://java.sun.com/products/jsp/jstl/
12. Hluchy L., Tran V.D., Habala O., Simo B., Gatial E., Astalos J., Dobrucky M.: Flood Forecasting in CrossGrid project. In: Grid Computing, 2nd European Across Grids Confer-ence, Nicosia, Cyprus, January 28-30, 2004, LNCS 3165, Springer-Verlag, 2004, pp. 51-60, ISSN 0302-9743, ISBN 3-540-22888-8É

K-Wf Grid Demonstration

Ondrej Habala

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
habala.ui@savba.sk

Abstract. The K-Wf Grid Project (The Knowledge-based Workflow System for Grid Applications, IST-2004-511385 K-Wf Grid) is aimed at providing workflow and knowledge management tools, as well as a complete framework for grid-based and SOA-based workflow construction and execution, with several pilot applications.

The K-Wf Grid Project

Fig. 1 presents architecture of the system components of the workflow orchestration and execution environment in the project K-Wf Grid.
The main user interface for developing semantic-based Grid applications is the User Assistant Agent (UAA), which contacts the Grid Workflow Execution Service (GWES) that manages the process of composing and executing the services. The automated semantic service composition is partly delegated to the Automatic Application Builder (AAB), the Workflow Composition Tool (WCT), and the user (by means of the UAA). The AAB and the WCT are knowledge-based semi-automatic modeling services, which in cooperation with the User Assistant Agent (UAA) can propose known solutions to problems solved in the past. When parts of the workflow are ready to be executed on the Computing Grid, the Grid Workflow Execution Service asks the Scheduler for the optimal resource, due to some user-defined metrics. Then, the corresponding Web Service operation is invoked remotely on the Grid middleware using WSRF protocols. The events triggered by the workflow orchestration and execution will be published by means of an event system. The Knowledge Assimilation Agent (KAA) consumes these events and generates knowledge that is stored in the Grid Organizational Memory (GOM). This knowledge can be later reused by the components of the workflow orchestration and execution environment.

WCT provides the functionality of composing abstract workflows of Grid applications from simple user requirements. It employs semantic reasoning techniques over OWL¬S descriptions (i.e. subsumption, classification) and it tries to propose a solution to the user's problem by using provided descriptions of available resources. Such a solution is delivered in form of an abstract workflow instance composed of service operations. This workflow of operations is based on the Petri nets model,

which has several advantages over the directed acyclic graphs, for a detailed description see [GWES]. The main input to the WCT is a description of data (results) which is to be produced by the target application and, optionally, a set of user-provided data (input) to be used by the composed application. It is also possible to upload an incomplete workflow as an input in order to complete it automatically. The main output of the composition process is a refined description of the abstract workflow. During the composition and refinement of the workflow the User Assistant Agent is used to guide the user according to the experience it gained in the past compositions.

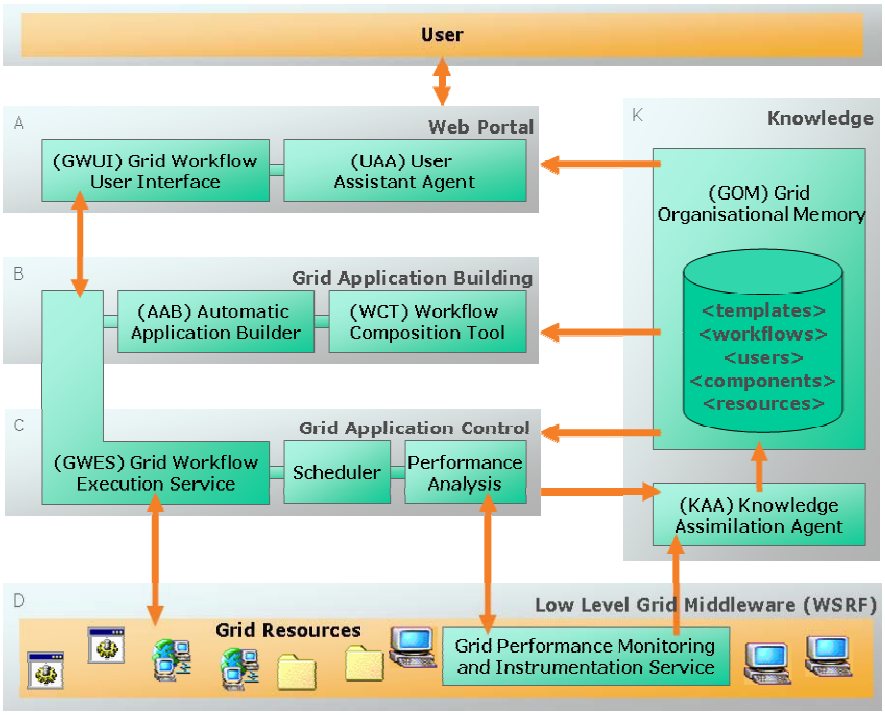


Fig. 1. K-Wf Grid architecture

Application Scenario

The user may use several tools in order to create and execute his/her application workflow. One of these tools is an integrated logging panel, where he/she can see all messages from K-Wf Grid modules; including information, warning and error messages (see Fig. 2). Another important tool is the ontology browser portlet, where

the user may view (and later also modify) the knowledge stored in the system (see Fig. 3).

The core of the system is the UAA panel with the Grid Workflow User Interface (GWUI), a Java applet (see Fig. 4). In the UAA panel the user may view notes of the application components (where any notes are available), add new notes for future users, and most importantly compose workflows, which are visualized and executed using the GWUI applet.

User initiates the process of workflow construction and execution by stating a problem using the small yellow icon in the UAA panel, as a free text sentence. This problem sentence is then analyzed for keywords, and based on these keywords its context is selected. After the user confirms the proper context of the problem, UAA initiates new workflow, which is immediately displayed in the GWUI applet. Then the user controls workflow instantiation and execution from the GWUI applet, by using its toolbar located in the bottom of the applet's window.

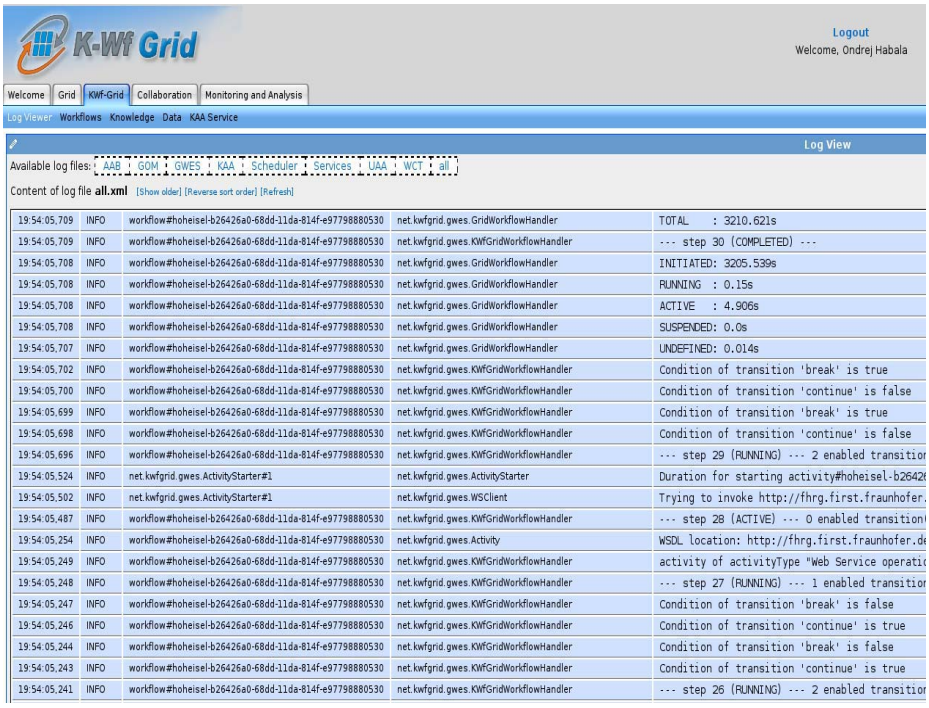


Fig. 2. Logging panel

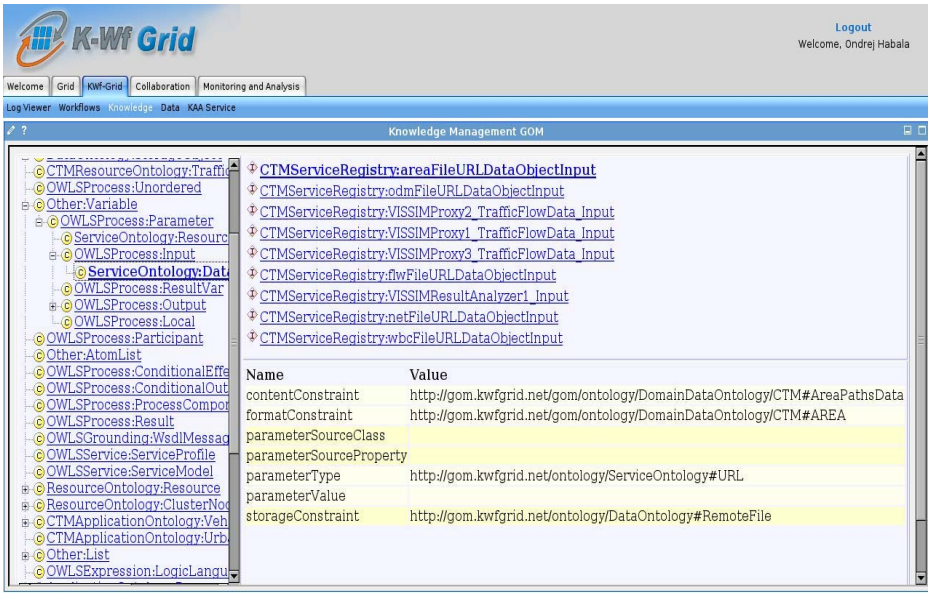


Fig. 3. Knowledge Management GOM portlet

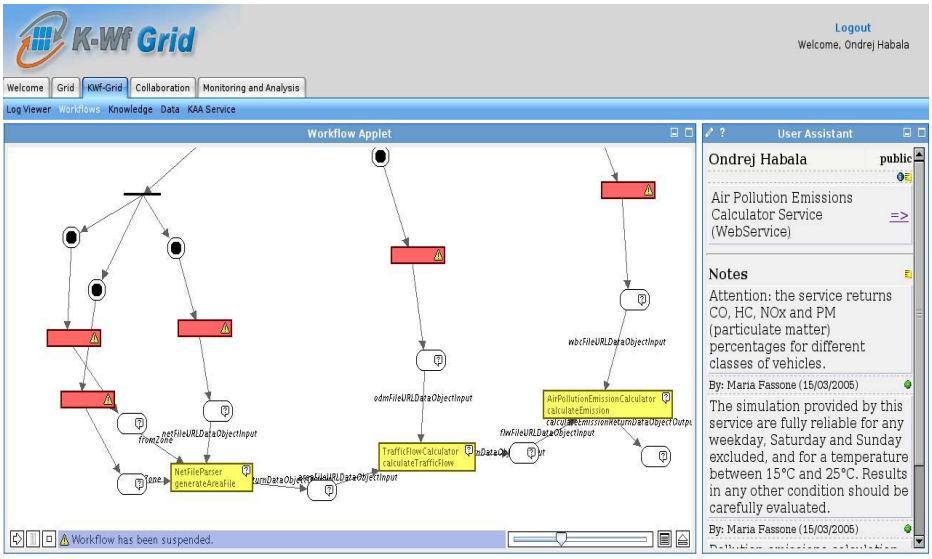


Fig. 4. User Assistant (right) and Grid Workflow User Interface (left) panels

Acknowledgements

This work is supported by projects KWf-Grid EU RTD IST FP6-511385, NAZOU SPVV 1025/2004, RAPORT APVT-51-024604, VEGA No. 2/3132/23 and VEGA No. 2/6103/6.

Reference

[GWES] Hoheisel, A.: User Tools and Languages for Graph-based Grid Workflows. In: Special Issue of Concurrency and Computation: Practice and Experience, Wiley, 2005.

FloodGrid demostration in the CrossGrid project

Miroslav Dobrucký

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
miroslav.dobrucky@savba.sk

Abstract. CrossGrid project has developed, implemented and exploited new grid components for interactive computation- and data intensive applications like simulation and visualisation for surgical procedures, flooding crisis team decision support system, distributed data analysis in high-energy physics, and air pollution combined with weather forecasting. This presentation will show our FloodGrid application developed in the CrossGrid project, that consist of cascade of three simulations (meteorological, hydrological and hydraulic). Application configuration (preparation of parameters) and interactive steering of the FloodGrid application is done via our web portal, through which also visualization of simulation resultscan be shown.

Introduction - Flood application

The food forecasting application developed within the scope of the CrossGrid project [1] consists of several simulation models (meteorological, hydrological and hydraulic) and appropriate post-processing tools connected together, thus constituting a workflow [3{5].

The meteorological model is used to forecast precipitation, to be used by the hydrological model for computation of discharge of the river. That is used in the final step for actual computation of possible flood by the hydraulics model. All the models generate binary output data, which are then used by post-processing tools to generate pictures visualizing the situation. These pictures are then used by respective experts for situation evaluation.

The flood forecasting application has two user interfaces to enable users to interact with the application in a more user-friendly way. One interface is implemented as a web portal accessible by standard web browser. It consists of a set of portlets - reusable web components - that are placed in the portlet portal framework. Another user interface is implemented as plug-in for Migrating Desktop (MD) - a desktop user environment for working with grids developed in [1]. While the portal interface focuses mainly on the flood application, MD is a general tool that enables a user to work with grid in a flexible way.

The flood forecasting application framework with appropriate simulation models enable users to easily run the desired sequence of simulations and respective post-processing tools, browse the results of simulations, register results into the replica management service and applicable metadata into the metadata catalog for later search and retrieval.

FloodGrid portal

In following, we will briefly show how to use our CrossGrid portal. The first step is that the user log into the portal with his/her login and password, which was assigned to him/her by portal administrator. Then the user can switch to one of his workspaces, i.e. "FloodGrid portal". Then s/he should load his/her grid certificate proxy from some myproxy server, which it was previously stored in, e.g. "*rosnicka.ui.savba.sk*", by clicking to "*Grid Proxy*" tab, see Fig. 1. S/he can specify the validity duration of new generated proxy, i.e. number of hours and wheather it will be destroyed when s/he logout. The proxy can be inspected (view) for time left and can be destroyed at any time.

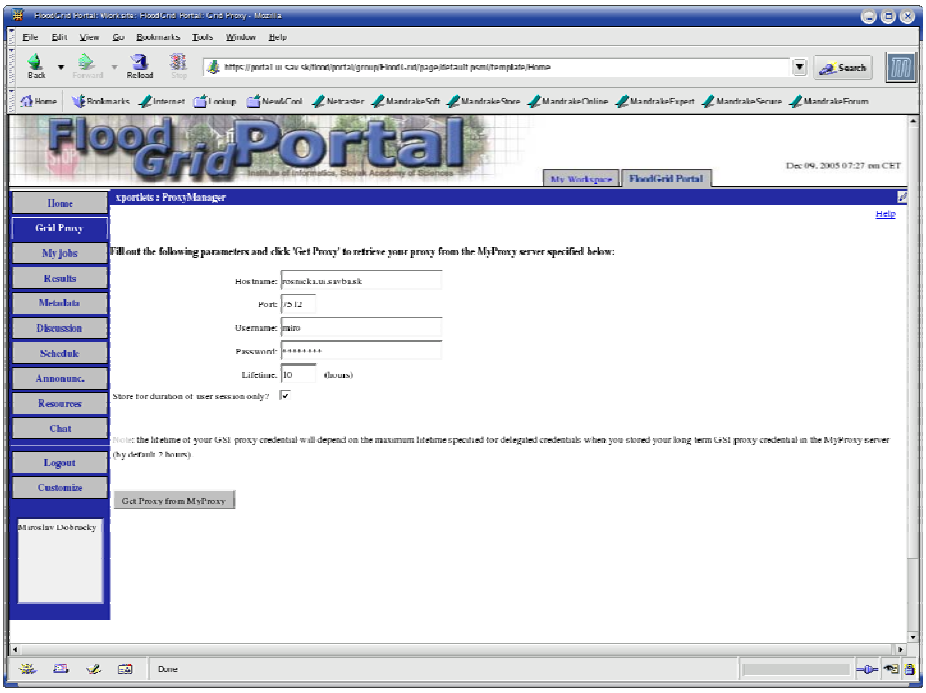


Fig. 1. User loads his/her grid proxy certificate

Then "*My jobs*" tab can be clicked to get list of job templates and jobs that were/are/will be run. The user can choose from predefined workflow templates, e.g. a

single simulation, a single visualization job, or one of the workflows – more jobs connected into a cascades, see Fig. 2.

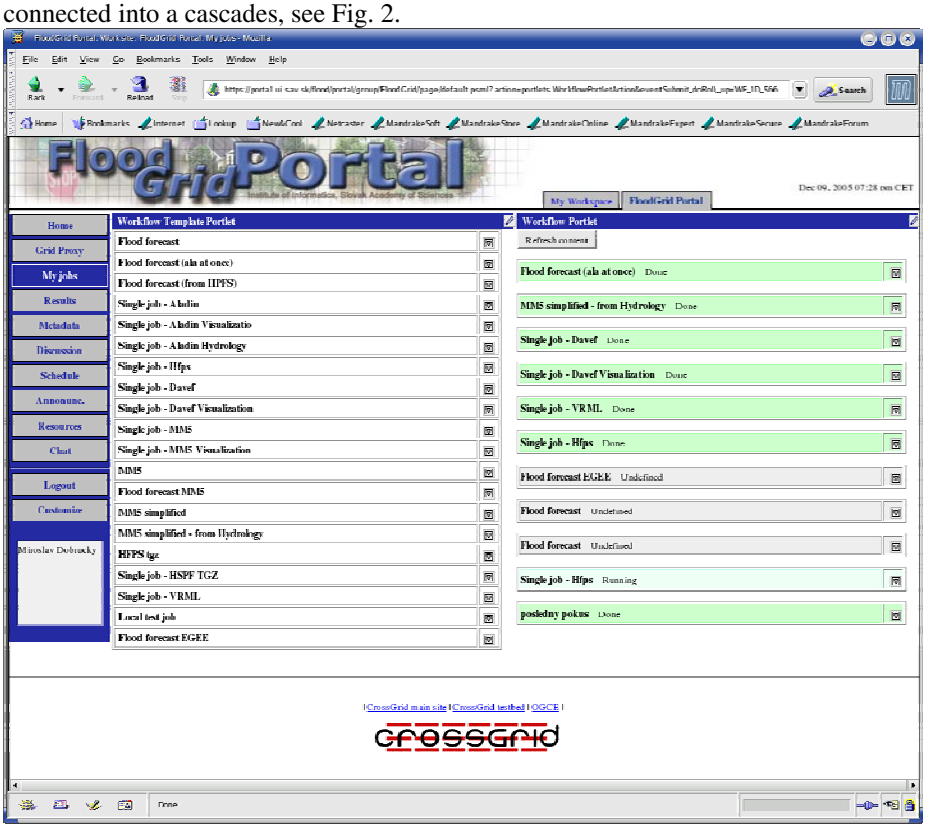


Fig. 2. List of workflow templates and instances

Then s/he can set for each individual job in the "*Undefined*" workflow its parameters, most of which have been assigned some default values. Mostly s/he will change the date/time. Then s/he click "*Save*" followed by clicking "*Submit*" and can fold up the workflow details., see Fig.3.

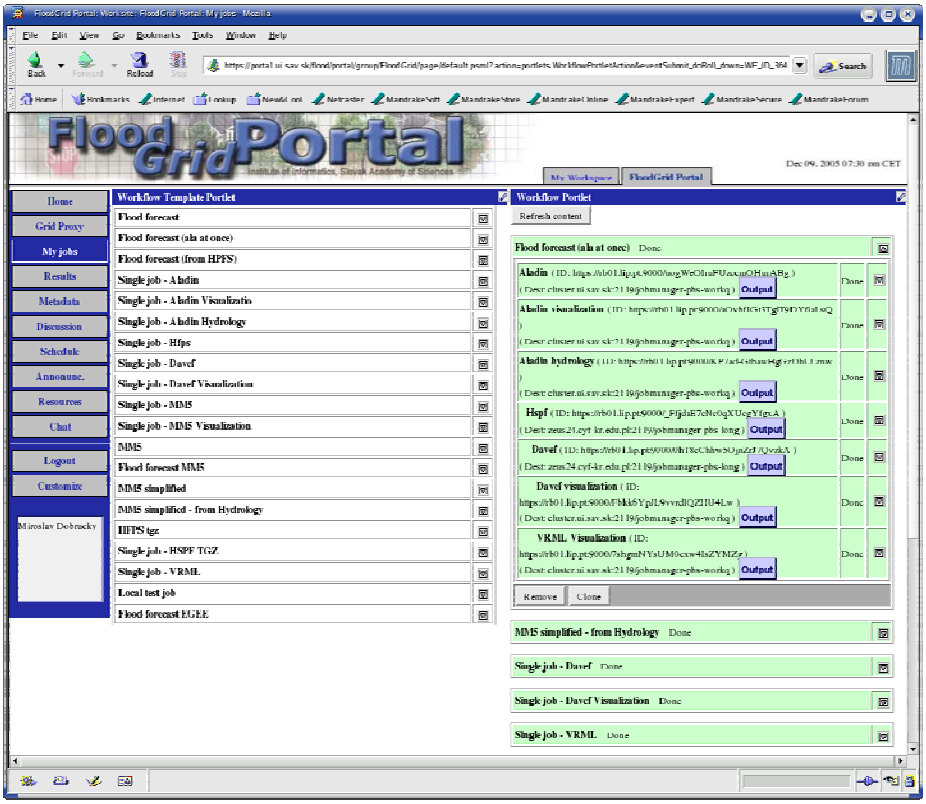


Fig. 3. A composite job (workflow) - the cascade of simulation and visualization jobs

Then the user should wait and can click the "*Refresh content*" button to obtain the most recent status of his/her jobs. When some job changes his status to "*Done*", its outputs can be shown, by clicking to "*Output*" button (Fig.3) in the relevant job instance, or browsing in the whole output directory by clicking "*Results*" in the left menu bar, but in this later case the user should know the job's ID number. For outputs evolving in the simulation time, which are generated as a bunch of files, the animation tool can be used,(see Fig.4)

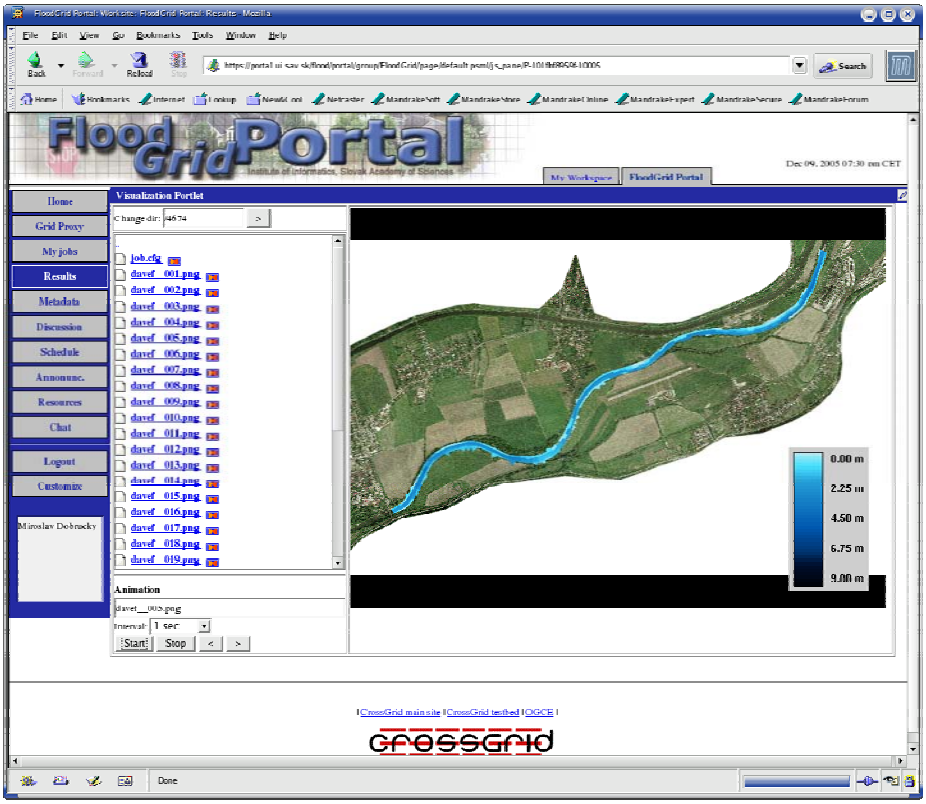


Fig. 4. An example of 2D+time output - flood extent evolving in a time placed into the orto-photo map – animation

Before running some simulation job, the user may wish to find the relevant input or output files. S/he can browse the metadata catalog, be entering keywords and conditions (Fig.5) and then examining the relevant metadata files.

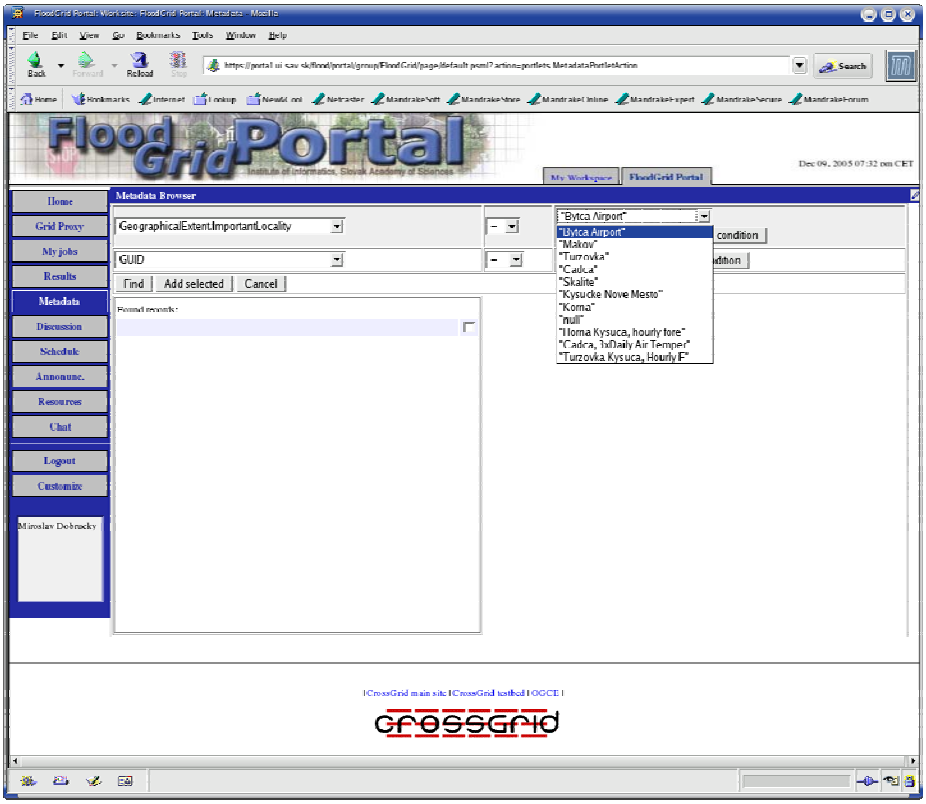


Fig. 5. The metadata browser - entering the query.

Conclusion

Our FloodGrid portal has also usual collaboration tools like chat, schedule calendar, discussion, announcements, shared working place for shared resources (files, URLs etc.), that means more people can work together in the flood prediction problem together in the cooperative manner.

New version of our FloodGrid portal are customized for the usage in other our project EGEE ([2], in which scope we are preparing it to be used for Earth Science Research Virtual Organization. New technologies are introduced to our portal for its usage in other our project MEDIGRID ([6]), to be able to cooperate in mixed operation systems platforms (MS Windows + Linux), by full employment of GT4 ([7]).

Acknowledgements

This work is supported by projects EU 5FP CROSSGRID IST-2001-32243, EGEE EU 6FP INFSO-RI-508833, VEGA No.2/3132/23 and VEGA No. 2/6103/6.

Reference

1. CrossGrid project: http://www.eu-crossgrid.org
2. EGEE project: http://public.eu-egee.org/
3. L. Hluch□y, O. Habala, V.D. Tran, B. □Simo, J Astalo□s, M. Dobruck□y, Infrastructure for Grid-based Virtual Organizations., Proc.of International Conference on Computational Science, Part III, LNCS 3038, Springer-Verlag, 2004, pp. 124{131.
4. L. Hluchy, V.D. Tran, O. Habala, B. Simo, E. Gatial, J Astalos, M. Dobrucky, Flood Forecasting in CrossGrid project., Grid Computing, 2nd European Across Grids Conference, LNCS 3165, Springer-Verlag, 2004, pp. 51{60.
5. L. Hluchy, J Astalos, M. Dobrucky. O. Habala,, B. Simo, V.D. Tran, Food Forecasting in a Grid Environment., Proc. of 5-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2003, R.Wyrzykowski et.al. eds., 2004, LNCS 3019, Springer-Verlag, pp. 831{839.
6. MEDIGRID project: http://www.eu-medin.org/projectPage.php?acronym=MEDIGRID
7. Globus Toolkit ver.4: http://www.globus.org/toolkit/

Grid Infrastructures in Central Europe

Ján Astaloš

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 84507, Bratislava, Slovakia
astalos.ui@savba.sk

Abstract. The paper presents current state of Grid infrastructures in Central European countries. It aims to provide information for discussion about future of Grids in Slovakia, by presenting how the national Grid initiatives are organized in surrounding countries. After brief introduction to different types of Grid infrastructures, national Grid infrastructures are presented in more detail with focus on type of coordination at national level. While some countries have Grids initiatives established as projects funded at national level, others have Grid centers coordinating national Grid related activities. At the end, the current state of Grid infrastructure in Slovakia is presented.

Introduction

Grids are a crucial enabling Information and Communication Technology for meeting the goal of the 'Lisbon Strategy' [1]. The term "the Grid" was coined in the mid1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. Since then, Grid concept evolved to support the sharing and coordinated use of diverse resources in dynamic multi-institutional virtual organizations. Research projects in Europe and in the world are developing necessary technologies to build world-wide Grid infrastructure which will enable researchers and businesses to better share knowledge and resources.

Grid infrastructures

Many Grid infrastructures (or shortly Grids) were built with different purpose, organization, amount of resources and security levels. Early Grid infrastructures were often called *''Testbeds'* as the grid technologies were under development and Grids were used mainly for testing. As the number of users using grids for their work and number of sites involved had increased, testbeds had to transform to production Grids. Questions about smooth operation, security, user training and support become crucial. *'Production grids'* in Europe (EGEE, NorduGrid, SEE-Grid ...) span through multiple application domains and connect thousands of CPUs. Third kind of Grids is infrastructures dedicated to *'demonstration and training'* (e.g. Gilda). Grid infrastructures differ also in their scope. Projects developing new technologies usually

set-up dedicated grid testbeds among partners (e.g. DataGrid or CrossGrid projects). Application domains that need Grids for sharing resources are building Grid infrastructures to support collaboration inside their communities (virtual organizations). Examples are NEESgrid (research of earthquakes), AstroGrid or BioGrid. Other kinds of Grids are regional or national Grid infrastructures (e.g. AustrianGrid, DutchGrid or Swegrid). Most of the countries in Europe already have (or are building) their national Grid infrastructures. Largest Grids span through multiple countries often on different continents (EGEE, TeraGrid, Open Science Grid, Nordugrid).

National Grid infrastructures in central Europe

EGEE Grid infrastructure ('*flagship*' in EU with funding of over 30 million Euro from European Commission) aims to join national and regional Grid infrastructures and create robust and secure Grid available to scientists 24 hours-a-day. 180 participating sites from 41 countries with more than 18,000 CPUs are organized to 12 federations. Slovakia with Poland, Czech Republic, Austria, Slovakia, Hungary, Slovenia and Croatia form the Central European federation. Most of the countries in our federation have national grid initiatives either as projects funded on national level or as centers coordinating national Grid activities.

Poland

Polish Ministry for Science and Information Society Technologies (MNiI) issued in 2000 a national programme called PIONIER (Optical Internet - Advanced Applications, Services and Technologies for the Information Society) with main goal to build a national infrastructure to support high demanding national applications, national and international scientific co-operation. High bandwidth optical network built in the scope of the programme connects all city networks, including universities and HPC centers. Multiple projects were supported by the programme:

- *Clusterix* (National Cluster of Linux Systems) [2] (2004-2006) is supported also by Intel company and it aims to build distributed PC-cluster of Itanium2 CPUs connecting 12 partners. Peak setup had more than 800 CPUs.
- *SGIGrid* (High Performance Computing and Visualization with the SGI Grid for Virtual Laboratory Applications) [3] (2003-2005) is focusing on connecting SGI supercomputers from Polish centers for visualization, load balancing, resource management and remote access of scientific instrumentation. It is supported by Silicon Graphics.
- *Progress* (Polish Research On Grid Environment for Sun Servers) [4] co-funded by SUN Microsystems.
- *Virtual Laboratory* [5] (2003-2005) is a research project for providing a remote access and control to various kind of scientific laboratory equipments embedded in Grid environment.

Czech Republic

In Czech Republic national Grid programme is covered by project METACentre [6] which coordinates national grid activities. It started in 1996 with the main aim to interconnect largest supercomputing centers in Czech Republic and provide common interface to users. In the project participate:

- CESNET (Czech Education and Scientific Network) operator
- Supercomputing centre Brno (Masaryk university Brno)
- West Bohemian supercomputing centre (West Bohemian university Pilsen)
- Supercomputing centre UK (Charles university Prague)
- CESNET participates also on national and EU grid related projects (EU DataGrid, GridLab, CoreGrid and EGEE)

Austria

In Austria, national Grid initiative is covered by project AustrianGrid [7] supported by Federal Ministry for Education, Science and Culture. It has 37 national and 7 international partners from research institutes in advanced computing technologies along with partners from multiple application areas. Currently the Austrian Grid infrastructure includes 9 sites with 229 CPUs (32- and 64-bit) in heterogeneous Grid environment. The project is divided into work packages and covers infrastructure setup and support, development of new grid technologies (security and authentication, visualization and interfaces, multimedia and compression, databases, data mining and data retrieval, self-adapting computational grid software, programming paradigms and methods for the Grid, mobility support in the Grid) and application areas (medical sciences, high energy physics, applied numerical simulations astrophysical simulations and solar observations, meteorology and geophysics and environmental applications).

Hungary

In Hungary, national grid activities are coordinated by Hungarian Grid Competence Centre (MGKK) [8] established in 2003 by five most active Grid centers in Hungary.

Its main goal is to intensively promote and coordinate high-quality research and development activities in the field of Grid computing at a national level. Participants of the MGKK are:

- Computer and Automation Research Institute of the Hungarian Academy of Sciences (SZTAKI)
- Eotvos Lorand University of Science, Budapest (ELUB)
- Office for National Information and Infrastructure Development (NIIFI)
- KFKI Research Institute for Particle and Nuclear Physics of the Hungarian Academy of Sciences (KFKI-RMKI)
- Budapest University of Technology and Economics (BUTE)

National long-term programme called the Hungarian Information Society Strategy issued by Ministry of Informatics and Communication (IHM) mentions Grid technology explicitly as a major pillar of the programme. Multiple Grid related projects were also funded in the framework of different IT-oriented R&D programmes such as the IKTA program organized by the Ministry of Education between 2000 and 2003 and the GVOP program organized by the recently established National Office for Research and Technology (NKTH) and were strongly coordinated by MGKK. As a result of Grid projects four different kinds of national Grid system have been successfully developed and tested in Hungary: Hungarian Cluster Grid, Hun Grid, JGrid, SZTAKI Desktop Grid.

Croatia

In Croatia, national grid initiative is covered by the CRO-GRID poly-project [9] is a cooperative research and development effort of several Croatian institutions in the field of distributed processing, Grid systems and middleware. The CRO-GRID consists of three separate projects:

- CRO-GRID *Infrastructure*, whose leader is the SRCE - University computing centre, University of Zagreb and has main goal to build clusters in scientific and academic centres, connect them using CARNet and integrate them to international infrastructures.
- CRO-GRID *Middleware* systems, whose leader is the Faculty of Electric Engineering and Computing, University of Zagreb with main goal to research and develop new grid technologies.
- CRO-GRID *Applications*, whose leader is the Rudjer Boskovic Institute with goals to educate the scientific community for the self standing development of distributed applications, to develop models and methods of distributed application development and to develop at least 3 new applications and test them on the CRO-GRID system.

Slovenia

SiGNET (Slovenian Grid NETwork) is the pilot project of the emerging Slovenian production Grid. Reference site at Jožef Stefan Institute has 42 Opteron CPUs (60 Opteron processors are planed to be added in the near future). The cluster is owned by the local High Energy Physics group and financed through an equipment grant by the Ministry of Higher Education, Science and Technology. It is integrated in NorduGrid and EGEE infrastructures. Currently the resources contributed to SiGNET are HEP exclusive but in the future addition of astrophysics is foreseen. There are also clusters owned by computational chemistry and meteorology research groups that will be targeted for addition to the production Grid.

Situation in Slovakia

Unlike other countries in central Europe, Slovakia has no national funding program for supporting the development of Grid technologies and building national Grid infrastructure. Currently there are five grid sites included in EU Grid infrastructures (IISAS, IEPSAS, TU-Kosice and FMPhI-UNIBA in EGEE, and UPJS in NorduGrid) and two sites in other organisations are in preparation.

- Institute of Informatics SAS provides 37 CPUs in multiple Grid infrastructures (24 in production EGEE infrastructure, 8 in testbeds for projects CrossGrid, MediGrid and KWfGrid. 5 CPUs are in demonstration and training infrastructure Gilda). IISAS also provides site and user support including operation of SlovakGrid Certification Authority issuing digital grid certificates for users and hosts from Slovakia.
- Technical University of Kosice participates with 21 CPUs in EGEE
- Institute of Experimental Physics SAS participates with 14 CPUs in EGEE
- Department of Mathematics, Physics and Informatics of Comenius University in Bratislava has 7 CPUs in EGEE (soon to be extended to 35 CPUs).
- University of Pavel Jozef Safarik announced to put 9 CPUs to NorduGrid infrastructure.

There are other institutes that already have or are building their computing clusters (e.g. from computational chemistry application domain). Selected universities submitted join project to Ministry of Education and got funding for central grid services. They will build experimental Grid connecting unused capacity of workstations in PC labs using Condor local resource management system and EGEE middleware. Despite the relatively high number of sites participating in Grid infrastructures, the number of users from Slovakia comparing to other countries in Central Europe is still quite low.

This work is supported by projects EGEE EU 6FP INFSO-RI-508833, VEGA No. 2/3132/23 and VEGA No. 2/6103/6.

References

1. http://www.cordis.lu/ist/grids/
2. http://clusterix.pcz.pl
3. http://www.wcss.wroc.pl/pb/sgigrid/
4. http://progress.psnc.pl
5. http://vlab.psnc.pl
6. http://meta.cesnet.cz
7. http://www.austriangrid.at
8. http://www.mgkk.hu
9. http://www.cro-grid.hr

Mediterranean Grid of Multi-Risk Data and Models

B. Simo, M. Ciglan, P. Slizik, M. Maliska, M. Dobrucky

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 84507 Bratislava, Slovakia
branislav.simo@savba.sk

Abstract. The paper presents the MEDIGRID project that aims to create a distributed framework for multi-risk assessment of natural disasters and it will integrate models of forest fire behavior and effects, flash floods and landslides. These models will be upgraded to web applications in order to be able to run remotely as web services over the internet. A distributed repository with earth observation data combined with field measurements will be created. In the paper the MEDIGRID system architecture and simulation models are described, with conclusions towards the project's future.

Introduction

In order to make prevention, forecasting and mitigation of natural disasters possible a lot of effort has been invested into research of phenomena like forest fires, floods, landslides, soil erosion and others. Currently, there are models available that are capable of computing various aspects of these hazards. However, the data, the models or the results of simulations are not easily available to the audience that could make proper use of them; resources are not available to computationally intensive models and so on.

We are working to overcome some of these problems in the context of the MEDIGRID [1] project by employing the grid technology. The grid technology allows us to make the models and data accessible via internet in a secure manner for all partners and, possibly, other parties in the future. It also allows us to exploit computing resources connected to the grid for execution of demanding simulation models. The aim of the project is to create a distributed framework for multi-risk assessment of natural disasters that will integrate models, which have been developed in previous projects funded by European Commission. These include models for simulation of forest fire behavior and effects, flood modeling and forecasting, landslides and soil erosion simulations. Also, a distributed repository with earth observation data, combined with field measurements is being created, which will provide data to all models using data format conversions when necessary.

The multi-risk assessment of natural disasters in MEDIGRID project means a collection of various simulation models that can be run on the data stored in the data warehouse in order to simulate, estimate, and predict possible natural disasters as forest fires, floods, landslides and soil erosion.

Architecture

We had to choose which grid middleware to use on our computing resources in order to create our virtual organization (VO) for multi-risk assessment of natural disasters. Because each of the partners will provide some of the services on his own resources that run both Linux and Windows, we have chosen the implementation of the WSRF specification by the Globus alliance, which runs on both platforms. It is a Java implementation of core web (grid) services with security, notifications and other features. Each of the system components – simulation models, data providers, information services or other supporting services – will be exposed as a web service. We will use WSRF as a basic technology that will both serve as an implementation framework for individual services and glue the individual components together.

So far, we have implemented data transfer service and job submission service and integrated the data transfer service with a metadata catalog and replica location service. Metadata catalog and replica location are central services that will run on one central machine. The data transfer and job submission services will potentially run on every machine connected to the testbed.

The whole system will be accessible via a web portal. We have chosen GridSphere portal framework [7] for its support of portlet specification. Application specific portlets will allow users to invoke all the simulation services plugged into the system in application specific manner; for example using maps for selection of a target area or an ignition points for forest fire simulations. There will be portlets for browsing results, metadata describing those results, testbed monitoring and others.

Data management

An important aspect of simulation applications supported in the MEDIGRID project is that some of them are closely bound to the Windows operating system while others can be executed only on Linux platform. This means that underlying data grid infrastructure must be operable on heterogeneous resources. This is not an easy task as several standard grid data management components are operable on Linux systems only, e.g. current implementation of the GridFTP [4] - the basic grid transportation mechanism - requires Linux platform.

There are data that can be used only by members of certain organization due to the license agreement issues. Owners of the data must then have absolute control over data access policies.

Many input data sources for the applications and many simulation results must be stored, maintained and accessed in the MEDIGRID system. To facilitate usage of the data managed in the MEDIGRID, rich metadata must be kept and made available to the users - to discover the data of their interest.

In order to fulfill those requirements we have developed new data transfer service and integrated it with the metadata catalog and replica location service.

Data transfer service

Three important features of the data transfer service are:
- integration with the RLS and MCS
- fine grained data access policies
- data transfer mechanism for heterogeneous resources

In grid environment, files are specified by several levels of naming. Logical File Name (LFN) is a unique name, identifier of the file in the distributed environment, Physical File Name specify exact location of the file in the system (e.g. Protocol, site/service URL, path to file in local file system). We use those naming concepts in the MEDIGRID system. The integration with RLS and MCS means, that after creating a new file replica by the data transfer capability, replica is automatically registered in the central catalogs (RLS, MCS). User can also specify only the LFN for the data transfer (e.g. Request: 'deliver file with LFN A to the site S1') and the transfer service will automatically determinate exact physical location of required data sets.

To allow definition of fine-grained data access policies, we exploit the concept of authorized data resources. In the scope of MEDIGRID we use the term data resource with the following meaning: data resource is a directory in the file system of MEDIGRID site, which is registered in the site's data transfer service and is associated with the resource access list. Each site can contain multiple data resources.

Resource access list is a file with the list of distinguishes names of the grid users who are authorized to access the resource. Access privileges are also defined (e.g. Read only access, read/write access). The users specified in resource access list are authorized to access all files stored in the data resource. This simplifies the administration of data access rights, compared to the model with access control list for each distinct file. To make the system more flexible, we allow the creation of resource access list for each subdirectory of the data resource, which overrides the default one.

Metadata catalog and presentation layer

Metadata Catalog Service (MSC) [6] is used in the MEDIGRID project for metadata management. MCS is a grid based metadata catalog that addresses the need of grid environment to facilitate data publishing, discovery and access for large-scale data sets. MSC supports concepts of Logical collections and Logical views. Logical collections are user-defined aggregations that can consist of zero or more logical files and/or other logical collections. Logical views are another type of aggregation that can consist of zero or more logical files, collections and/or other logical views.

We have proposed and implemented Grid Virtual Directory System (VDS) - an extension of Replica Location Service and Metadata Catalog Service. VDS allows creating structures of virtual directories for simplification of the logical organization of the data files distributed across the grid. VDS hides from user most of the data management related operations that must be performed in grid environment (e.g. data replication, manipulations with the catalog services – RLS, metadata catalogs). This

concept permits the user to view and manipulate the files in the grid in much the same way he works with the local file system on his workstation.

Job management

The main constraint of the data transfer service – a required ability to support applications bound to both Windows and Linux also applies to job management, thus limiting the choice of available job management grid tools.

We have chosen the Java WSRF implementation provided by Globus toolkit [3] as a base for implementing our platform independent job submission service. When designing our job services we have put an important restriction on the functionality of the service – it is not possible to specify the binary file that will be executed as part of the job submission. The executable is always fixed for specific instance of the service. This constraint allows us to view the instance of job submission service as an entity providing fixed application function, limits the security threats by not allowing users to run arbitrary code, makes the application environment better maintainable and frees us from having to consider platform differences upon job submission as platform is completely hidden behind the service.

Job submission service provides the ability to run the executable associated with it with parameters provided along with the job submission request. Currently, jobs are started locally using the "fork"-like mechanism on both Linux and Windows with parameters passed as environment variables. Job requests are queued by the service and are run in the "first come first served" manner in order not to overload the host (see Figure 3). The jobs are internally implemented as WSRF [5] resources and their status data are exposed as resource properties, thus allowing queries in standard manner and also support change notification. Job cancellation is implemented using platform specific wrappers as Java does not provide a way to cancel a job tree.

We are aware of advantages of classical unrestricted computational grids but we feel fixed-binary job services are better suited here. However, in case of computationally demanding jobs (sequential and parallel) we plan to use classical computational grid as a back-end for execution of such jobs.

Visualization service

The purpose of the visualization service is to present the results of simulations in the form of pictures. This service differs from data and job services; however it provides a fundamental function for multi-risk assessment framework.

The visualization algorithms may differ according to the visualized computation type, their input and output data are usually very similar in the case of natural disasters. The input data fall in one of the following categories:

- 3D terrain model, usually referred to as DEM (Digital Elevation Model) or DTM (Digital Terrain Map) – raster map;
- orthophotomap, a photograph of the terrain surface taken from a plane (raster map);
- the actual results of simulations, possibly in various formats, raster or vector; and
- other files, in raster or vector formats.

The last two cases expect the data to be in various raster and vector format. However, in MEDIGRID, mostly raster data will be used. To simplify data exchange, a few common formats for data have been agreed upon. For raster data, ArcInfo ASCII Grid format and GRASS ASCII Grid format have been chosen.

The visualization subsystem will be built using the client-server architecture. It will consist of two loosely-coupled parts – visualization service and visualization server. The task of visualization service is to prepare data in appropriate format for the visualization server. The service is also responsible for copying the data to a suitable place and creating (updating) the necessary configuration files. The visualization server will transform the input data into actual pictures.

Standard web-mapping operations, like panning and zooming, will be supported. Communication between server and client will adhere to the OpenGIS Consortium WMS standard [2], providing larger extent of independence between the two communicating parts.

Simulation models

There are several applications in MEDIGRID: floods, landslides, forest fires and soil erosion. All applications except flood use proprietary models developed by respective project partners.

The flood application consists of several third-party simulation models (meteorological, hydrological and hydraulics) and appropriate post-processing tools. MM5 meteorological model forecasts precipitation, which is an input to the HSPF hydrological model, which computes the river discharge. The DaveF hydraulics model then computes the possible flood in the given area. All the models generate binary output data, which are then used by post-processing tools to generate pictures visualizing the situation.

Future work

With the core services implemented, our work will focus on enhancing their functionality – job submission to computational grids, implementation of various metadata categories, etc. Also, higher level services will be built: distributed data warehouse using data transfer services together with the metadata catalog and searchable set of simulation model services.

Moreover, we are implementing web based user interface using the Gridsphere [7] portal framework integrating all abovementioned services.

Acknowledgement

This work is supported by projects EGEE EU 6FP INFSO-RI-508833, MediGrid EU 6FP RTD GOCE-CT-2003-004044, NATO-project EST.EAP.CLG 981032, VEGA 2/3132/23 and VEGA No. 2/6103/6.

References

1. The Medigrid project home page, http://www.eu-medin.org/projectPage.php?acronym=MEDIGRID (visited December, 2005)
2. OpenGIS Consortium Web Map Server (WMS) standard, http://www.opengeospatial.org/specs/?page=specs (visited November, 2005)
3. The Globus toolkit home page, http://www.globus.org/toolkit/ (visited December, 2005)
4. W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, S. Tuecke: GridFTP: Protocol Extensions to FTP for the Grid, http://www-fp.mcs.anl.gov/dsl/GridFTP-Protocol-RFC-Draft.pdf (visited December, 2005)
5. Karl Czajkowski, Donald F Ferguson, Jeffrey Frey, Steve Graham, Igor Sedukhin, David Snelling, Steve Tuecke, William Vambenepe: The WS-Resource Framework, http://www.globus.org/wsrf/specs/ws-wsrf.pdf (visited December, 2005)
6. Gurmeet Singh, Shishir Bharathi, Ann Chervenak, Ewa Deelman, Carl Kesselman, Mary Manohar, Sonal Patil, Laura Pearlman: A Metadata Catalog Service for Data Intensive Applications. Proceedings of Supercomputing 2003 (SC2003), http://www.globus.org/alliance/publications/papers/mcs_sc2003.pdf (visited December, 2005)
7. Gridsphere portal framework project, http://www.gridpshere.org (visited December, 2005)

Knowledge Assimilation for Performance Prediction of Grid Services for optimal Workflow Execution

Zoltán Balogh and Ladislav Hluchý

Institute of Informatics, Slovak Academy of Sciences,
Dubravska cesta 9, 84507 Bratislava
{balogh.ui, hluchy.ui}savba.sk

Abstract. As Grid technology becomes more service oriented and the distributed applications are constructed as complicated Grid service workflows, many new challenges arise during the composition and execution of such workflows. One of such problems is the selection of most suitable web service (WS) deployments for a concrete workflow task execution during the process of work-flow scheduling. A service is needed which is able to estimate the behavior of each web service deployment in the Grid. Herein we present the design of such service capable of estimating WS behavior and performance measures including run time, availability, accessibility, stability and others. This work exploits many scientific concepts and methodologies such as instance based learning and case based reasoning. Presented prediction service also implements a novel approach to WS performance prediction through the refinement of case retrieval process through semantic description of discrete features and service input data.

Introduction

This article deals with run time prediction of stateful web services (WS) for the purposes of optimal WS workflow construction and scheduling. Work presented in this paper is a part of effort to design a Knowledge Assimilation Agent (KAA) responsible for WS run time prediction in scope of the 6th IST FP called K-Wf Grid (Knowledge-based Workflow System for Grid Applications) [1].

At the present it is difficult to predict the behavior of a WS which carries out a job in a shared distributed computing environment such as Grid. The more complex the job is the more difficult is to predict the time needed by a web service to complete the execution. The run time of a WS is influenced by several aspects: internal WS performance, run times of other services utilized by WS during its operation and load of Grid resources which are used by WS during its execution. Additional complexity is introduced when constructing composite WS – workflows constructed of several interconnected WS. For the purpose of this article we work with stateful web services, i.e. web services which are stateless in nature, but which manage resources with state. Such stateful WS are responsible for creation, maintenance and destruction of WS

resources with state. There have been already attempts to predict application performance in Grids [2] however existing approaches do not deal with WS prediction in context of WS workflows. The main difference of our approach is that we investigate the service performance in dependence of invocation parameters and related resource properties extended by a novel approach to WS performance prediction through the refinement of case retrieval process through semantic description of discrete features and service input data.

Construction of Grid Service Workflows

This chapter describes the context in which WS behavior is being predicted. The presented approach is being implemented in the K-Wf Grid project and is an effort of several members from the project consortium.

The overall objective of the workflow orchestration and execution environment is to provide services that allow users to assemble and invoke simple as well as complex workflows on dynamic Grid resources concerning interactive alterations and refinement of the workflow during runtime. The aim is to develop a very generic workflow orchestration and execution environment that is suitable for any application domain dealing with loosely coupled Grid services.

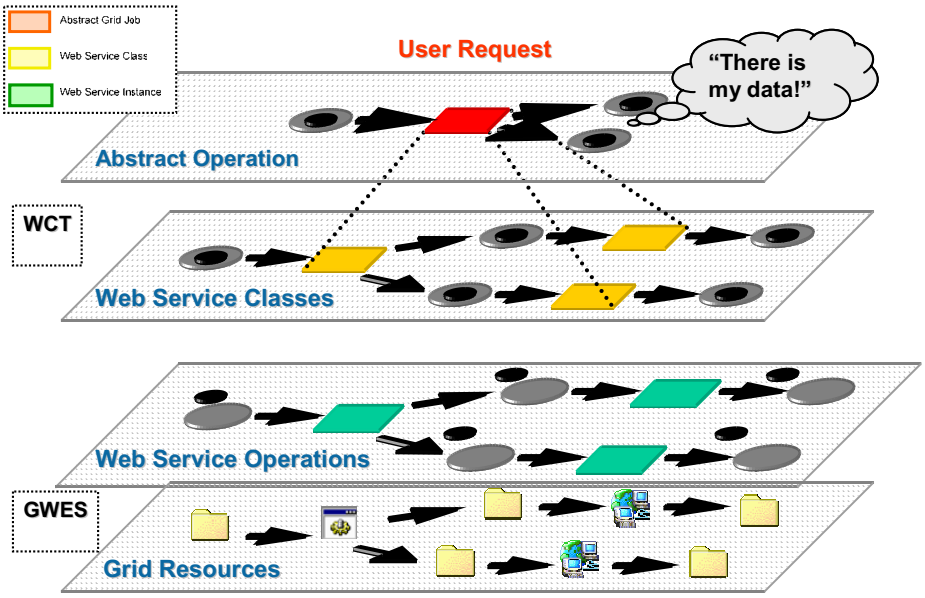


Fig. 1. Layered visualization of Grid service workflow composition and execution

The construction of workflows is being enacted in several layers (Figure 1), where each layer falls into competence of a certain component. The Grid Workflow Execution Service (GWES) is the Grid workflow enactment engine of the K-Wf Grid

system, which coordinates the creation and execution process of Grid workflows. It implements the highly dynamic workflow concept. It provides interfaces to the Web Portal for user interaction and to the Low-Level Grid Middleware for the invocation of application operations. The mapping of abstract to concrete workflow is mainly dele-gated to the system components WCT, AAB, and Scheduler.

The Automatic Application Builder (AAB) is a tool used in the semi-automatic process of workflow construction. The Scheduler is a component of the Grid Application Control layer. It determines which instances of a list of alternative Web Service Operation (WSO) instances are selected in order to be executed in the current workflow execution. The goal is to consider non-functional properties of the Web Services to choose between functionally equivalent instances of the same WSOs, and make the workflow ultimately concrete.

The Workflow Composition Tool (WCT) provides the functionality of composing abstract workflows of Grid applications from simple user requirements. It employs semantic reasoning techniques and it tries to propose a solution to the user's problem by using provided descriptions of available resources. Such a solution is provided in form of an abstract workflow instance composed of operations on Web Service classes. The main input to the WCT is a description of data (results) which is to be produced by the target application and, optionally, a set of user-provided data (input) to be used by the composed application.

Requirements for prediction and the Knowledge Assimilation Agent

WS workflows are used in the K-Wf Grid project to interconnect separate WS into logically coherent Grid applications. The construction of workflow is supported by knowledge in order to optimize grid resource usage and comply with requirements defined by a user.

The Scheduler as introduced in the previous chapter is a component of the Grid Application Control layer. It determines which instances of a set of alternative Web Service Operation instances will be selected in order to be executed in the current workflow execution. As input, the Scheduler receives the workflow description and searches for nodes that are mapped to a WS (a list of WS instances). Then, all possible alternative instances of each WS are compared considering different possible com-parison criteria (metrics) to choose the best one in the given situation.

The most important criterion used to choose the instance is the execution time. Scheduler requires a service which is able to predict the behavior of certain WS deployments. It is in general very hard to predict the behavior (for example the execution time) of a Web Service Call. The Scheduler is often in situation to decide which in-stance of WS to use for a certain computation. In distributed environments it is rare to have full control over all the computing resources and therefore it is desirable to have an infrastructure which would be able to measure performance, keep records and make predictions about WS classes and their individual deployments (instances). Such pre-dictions are in K-Wf Grid produced by the Knowledge

Assimilation Agent (KAA). For example runtime of individual WS operations is generated by KAA. Some other criteria (e.g., fault tolerance) may also be predicted. As a result one of the WS in-stances referenced by its URI is returned to the Scheduler. Comparison criteria are inserted and taken from the Grid Organizational Memory (GOM). The criteria are usually experience-based and are derived from earlier executions of the WSs. The information collected from the previous runs contains execution time, reliability, availability rate, and other possible metrics.

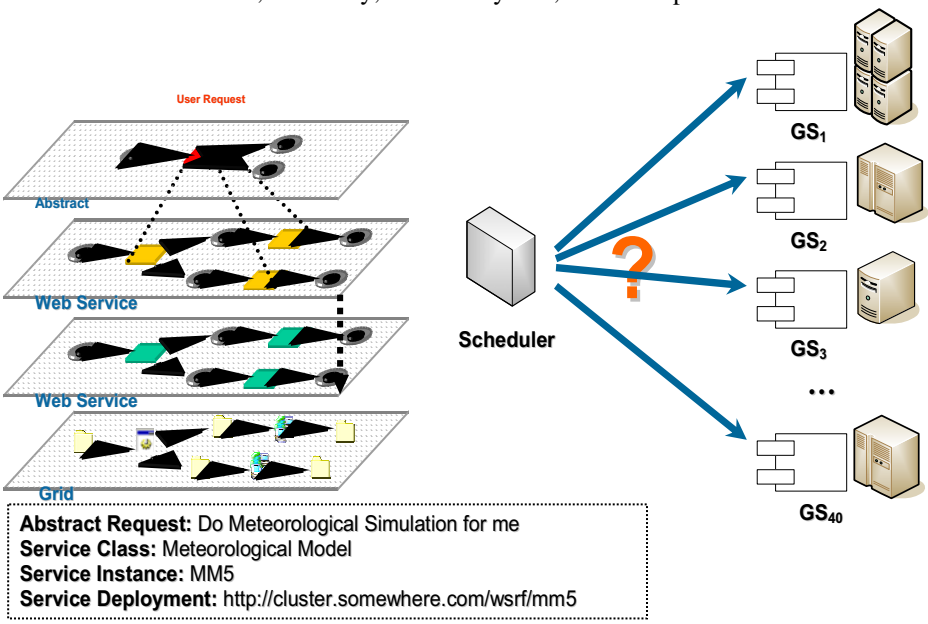


Fig. 2. KAA assists Scheduler during the transition of the workflow from WS Classes to WS Resources through WS Operations. KAA helps Scheduler to resolve a problem which WS deployment to use in a current context.

KAA is the component which extracts experience from historic data which are used to improve scheduling decisions made by the Scheduler (Figure 2). The selection of the most suitable WS Instances among several alternatives is based on historical analysis of individual WS Instance invocations.

Generic Ontology for WS Operation Run-time Prediction

In this article we consider WS operation (WSO) run-time prediction as the best representative WS performance measure. Similar approach can be chosen to predict other WS performance measures, such as:

– WS reliability prediction – a probability of WS operation failure before it delivers the desired result,

– WS availability prediction – a probability that WS is present or ready for immediate use and
– WS accessibility prediction – a probability denoting the chance of a successful service instantiation at a point in time.

The exact relationship between the WSO input and the corresponding run-time is complex– making it necessary to learn the relationship [4] calls the relationship concept). The main concerns which must be addressed during resource-usage prediction, but which must be taken into account also for runtime prediction are:

1. necessity to learn the relative importance of the features (input), because the extent to which individual features affect the relationship is unknown,
2. scaling factors employed for numerical stability have to be determined on the fly – because the range and distribution of the values of the features are not known in advance, and
3. the relationship (concept) has often non deterministic component and unobservable features.

Case representation is crucial for capturing information about WS operation invocations. We model the case structure using ontology. The case which will represent a single WS operation invocation is depicted on the following figure:

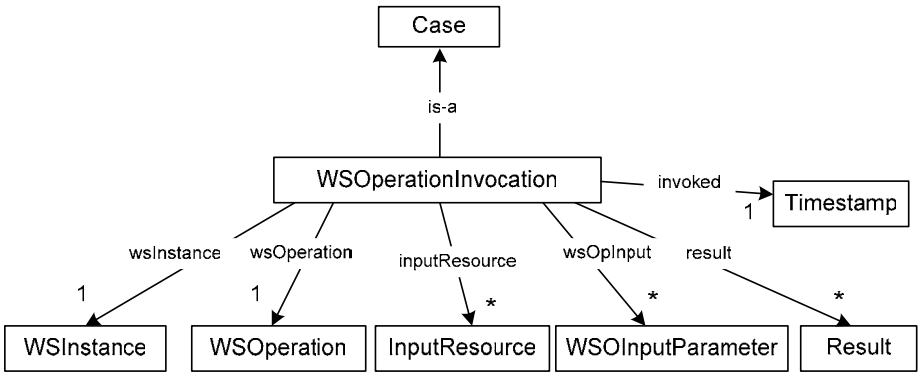


Fig. 3. Case representation for WS operation performance prediction.

The central part of the ontology is the class `WSOperationInvocation` which is inherited from a `Case` class, obviously implying that "`WSOperationInvocation` is a `Case`". Class `WSOperationInvocation` has the following properties:

– `wsInstance` (an instance of `WSInstance` class) – which is a concrete WS deployment on a concrete location and on a concrete computer system,
– `wsOperation` (an instance of `WSOperation` class) – which is the name of operation invoked in scope of the current case,
– `wsOpInput` (an instance of `WSOInputParameter` class) – which is any number of input parameters we can gather about the current case,
– `invoked` – time of initial invocation of the current case and

– `result` (an instance of `Result` class) – which will describe different resulting aspects generated by the current case (also described bellow).

A case can have several types of results. One result can be the length of execution; another is whether the WS operation failed to finish successfully or how much time elapsed between the request and real start of invocation of an operation. The class called Result is used to represent all such resulting effects. The Result class can be therefore further extended by subclasses as follows:

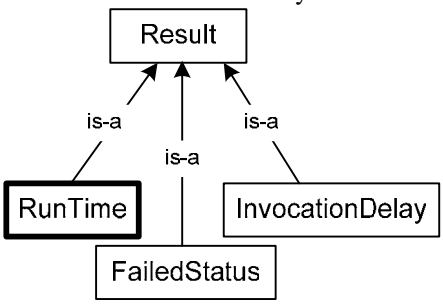


Fig. 4. The Result class possible extension

Capturing Information about WS Performance and assimilation of various data

Run-times of WS operations need to be monitored in order to predict the length of WSO executions in the future. There are several types of events, each containing different type of information about WS invocations. These various events must be aggregated (assimilated) into a single semantic representation - in our case we use the ontology with the super-class Case described in the previous chapter. In addition to information extracted from monitored events, we need to assimilate information from and will require access to:

– Input Resource Ontology,
– Web Services Ontology and
– WS Property Ontology.

Unification of cases gives us uniform representation of data which we will use for further reasoning. Aggregation of events into one ontology format will need to be performed continuously either in predefined time periods or after occurrence of given number of events. The result of aggregation will be therefore a history of WS operation invocations in an aggregated format, prepared for further processing. A concrete example of a case with data assimilated from several sources is on the following figure:

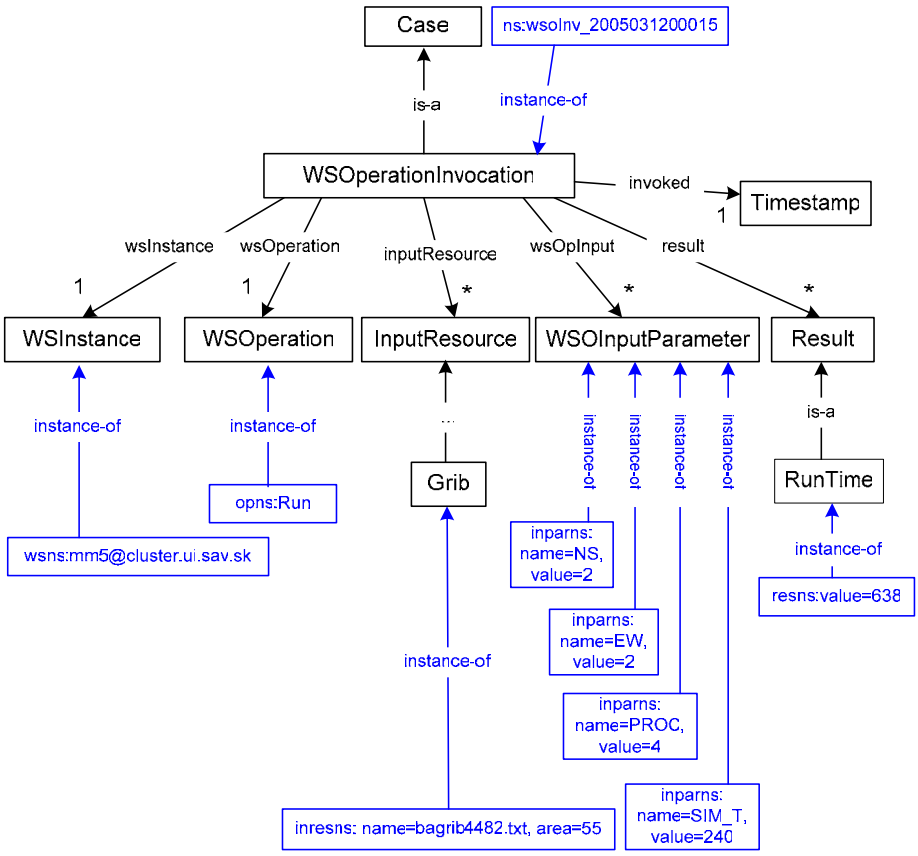


Fig. 5. Sample Case with class instances generated from a web service operation

We make reasoning about each WS instance and its operations individually. It is be-cause we are always interested in the individual WS instance run-times which are then compared and evaluated.

Refinement of case retrieval through semantic description of input data

Retrieval of similar cases in case-based reasoning (CBR) is in general solved by comparing feature vector of current case with the feature vectors of all cases from the case base. For discrete features exact match is done for feature values of one case with another case. Such exact matching has its limitations (returns a very limited set of cases), therefore we propose to model discrete features using ontology. The ontology

can be used to find similar concepts thus broaden the set of most similar cases using simple inference.

For example ontology from Figure 6 can be used to refine the retrieval of cases from the case base. We start with retrieval of all the OWL cases. If we detect that retrieved cases are not sufficient to deliver a satisfactory prediction, the retrieval is repeated with super class concept of the current concept (OWL), which is RDF.

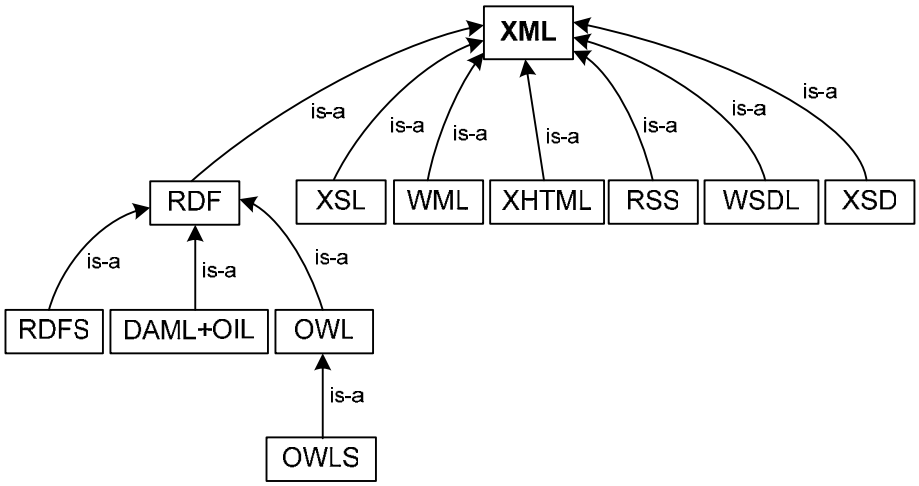


Fig. 6. Input data description Ontology example

Application of the Approach

The K-Wf Grid [1] project deals with knowledge supported Web Service workflow construction. As mentioned earlier in this article performance prediction of individual Web Service Instances is crucial for decision making process of both the Scheduler and Application Builder Agent (AAB).

A sample scenario is the use of the platform for the flood prediction platform in the K-Wf Grid project. The system is responsible for suggesting best workflow composed of several concrete WS implementations. KAA in the flood forecasting application can be used to discover for instance dependencies between execution time of a WS and the area for which we make the forecast. The area is stored in a special file – resource – which must be semantically described. In the simplest case the se-mantic description might contain information about the area and the density of the area grid. When flood prediction will be launched, the KAA will be notified about invoca-tion parameters used by a WS – which can be the URL of a file containing a descrip-tion of the geographical area. The KAA determines the description of the resource used as input for WS and stores all information into Grid Organizational Memory (GOM).

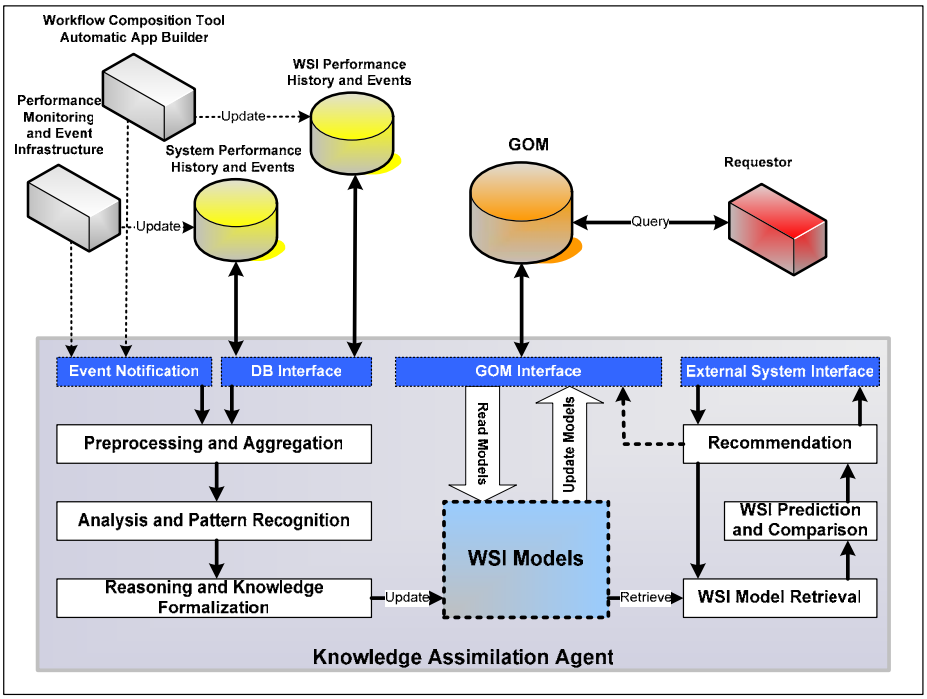


Fig. 7. Architecture of the Knowledge Assimilation Agent for the KWf-Grid

The KAA will be also notified about forecast completion and will store information about several such forecasting computations, thus having a base for successful fore-casting. If the KAA discovers any dependencies between an input resource parameter and WS performance it stores the model into GOM. Having relevant models, the KAA is able to predict how a concrete WS will perform, based on the provided input re-source description.

Conclusion

In this article we have presented our approach for predicting runtime of web services. First we have explained the construction of Grid Service workflows – which is the context in which performance prediction is being enacted. Than we described the problem which must be solved by the Scheduler and solution to which is provided by the Knowledge Assimilation Agent. Generic Ontology design for WS Operation Run-time Prediction was presented followed by an example and short description of WS monitoring (capturing) possibilities. A brief insight into the refinement of case re-trieval through semantic description of input data was also presented. Lastly we have

provided a short example of applicability of the KAA component in the context of the K-Wf Grid project. Other challenges for the future comprise inclusion of input data (metadata) description properties, prediction of other qualitative performance measures (such as availability or reliability of WS) and adjustment of weights according to comparison between predicted and really measured results.

Acknowledgments

This work is supported by the EU IST K-Wf Grid Project IST-2002-511385, VEGA No. 2/3132/23, VEGA No. 2/6103/6, APVT-51-024604 and SPVV 1025/2004.

References

1. EU IST K-Wf Grid Project IST-2002-511385 http://www.kwfgrid.net/
2. Faerman M., Su A., Wolski R., Berman F., Adaptive Performance Prediction for Distributed Data-Intensive Applications, August 9, 1999, Proceedings of the ACM/IEEE SC99 Conference on High Performance Networking and Computing, Portland
3. Menasce, D., A., Virgilio, A., F.: Capacity Planning for Web Services – Metrics, Models, and Methods. Prentice Hall, 2002. ISBN 0-13-065903-7
4. Nirav H. Kapadia, José A. B. Fortes, Carla E. Brodley, Predictive Application-Performance Modeling in a Computational Grid Environment, Eight IEEE International Symposium on High Performance Distributed Computing, pp. 47-54, August 1999.

Selected Part of Solving Sparse System over Z_2 via Block Lanczos Algorithm

Mgr. Marek Sýs and Ing. Vladislav Novák *

KAIVT FEI STU Bratislava

Abstrakt Nowadays Lanczos algorithm belongs to mostly used methods solving large sparse linear systems. In our article we focus on solving systems over field Z_2. It has some advantages opposite to real case, because elements of this field can be represented as bits and therefore we can work with whole block of bits. But this needs effective implementation of matrix multiplication discussed below.

1 Úvod

Jednou z najrýchlejších faktorizačných metód, ktorá sa používa na faktorizáciu RSA modulov je metóda zovšeobecneného sita číselných polí (GNFS). Táto vyžaduje nájsť niekoľko nenulových vektorov $x \in Z_2^n$ takých, že

$$Bx = 0, \tag{1}$$

kde matica B je rozmerov $n_1 \times n$. V celom texte budeme "0" označovať nulový vektor, nulový prvok, prípadne nulovú maticu.
Matica B je veľmi riedka, vo všeobecnosti obdĺžniková, takmer štvorcová matica nad poľom Z_2, získaná pomocou GNFS v preosievacej fáze. Každý z jej rozmerov je pre veľké RSA moduly (viac ako 500 bitov) rádovo $10^6 - 10^7$. Na hľadanie týchto vektorov sa môže použiť Gaussova eliminácia (GE), prípadne jej varianty. Tieto všetky však majú časovú zložitosť $\mathcal{O}(n^3)$ a teda výpočet pre takto veľké matice trvá príliš dlho (týždne až mesiace).

V súčasnosti medzi najpoužívanejšie metódy na hľadanie riešení veľkého riedkeho systému $Ax = u$ a teda aj $Ax = 0$ nad ľubovoľným poľom, patrí Lanczosov algoritmus (LA) [1]. Ten na rozdiel od GE dokáže využiť skutočnosť, že matica systému je riedka. Vyžaduje však jej symetrickosť. Ak položíme za $A = B^T B$, dostaneme systém

$$Ax = B^T Bx = b, \tag{2}$$

potom už matica A je symetrická a Lanczosovým algoritmom získané riešenie tohto systému, je s pravdepodobnosťou väčšou ako (1/2) aj riešením (1).

Existujú rôzne varianty LA, z nich asi najznámejšie a tiež najpoužívanejšie sú jeho blokové verzie [2], ktoré umožňujú nájsť naraz viacero riešení sústavy nad poľom Z_2 a súčasne urýchľujú samotný výpočet. Je to z toho dôvodu, že využívajú celú šírku počítačového slova N.

* Tento článok bol vytvorený s podporou grantu VEGA 1/0161/03.

Od tohto N závisí potom aj rýchlosť výpočtu. Pre maticu s priemerným počtom d nenulových prvkov na riadok je časová zložitosť $\mathcal{O}(dn^2/N)+\mathcal{O}(n^2)$, je teda omnoho rýchlejší ako GE. Cenou za rýchlosť pri LA je možnosť zlyhania, pretože LA je na rozdiel od GE pravdepodobnostný. Pravdepodobnosť úspechu je však vysoká a tak môžeme toto riziko podstúpiť. Ďalšia nevýhoda LA je, že nájde len niekoľko riešení (maximálne 2N), kdežto GE nájde všetky.

2 Lanczosov algoritmus

Opíšme si teraz klasický Lanczosov algoritmus, ktorý bol navrhnutý pre riešenie sústav lineárnych rovníc tvaru

$$Ax = b, \quad b \neq 0 \tag{3}$$

nad poľom $\mathbb{R}$, kde A predstavujú reálnu štvorcovú plnohodnostnú symetrickú maticu rozmerov $n \times n$, $b \in \mathbb{R}^n$ je vektor konštánt a $x \in \mathbb{R}^n$ je vektor neznámych.

V prvom kroku sa zvolí

$$w_0 = b.$$

Každý ďalší vektor sa vypočíta podľa vzťahu

$$w_i = Aw_{i-1} - \sum_{j=0}^{i-1} c_{ij} w_j \qquad (i > 0), \tag{4}$$

kde pre konštantu c_{ij} platí

$$c_{ij} = \frac{w_j^T A^2 w_{i-1}}{w_j^T A w_j}.$$

Toto sa opakuje až pokiaľ pre nejaké m bude $w_m = 0$. Konštanty c_{ij} sú volené tak, aby každý vektor w_i bol prostredníctvom matice A kolmý na všetky predchádzajúce vektory w_j vypočítané skôr, tj. $w_i^T A w_j = 0$ pre $0 \leq i < j$. Keďže je matica A symetrická platí tiež $w_j^T A w_i = 0$. V konečnom dôsledku potom môžeme písať

$$w_j^T A w_i = 0 \qquad i \neq j.$$

Hľadaný vektor x má vyjadrenie

$$x = \sum_{i=0}^{m-1} \frac{w_i b}{w_i A w_i} w_i. \tag{5}$$

Lanczosov trik je v tom, že $c_{ij} = 0$ pre $|j - i| > 2$ [1], a teda

$$w_i = Aw_{i-1} - c_{i,i-1} w_{i-1} - c_{i,i-2} w_{i-2} \qquad (i \geq 2). \tag{6}$$

Vráťme sa teraz k systému (2) nad poľom Z_2. Keďže LA počíta len riešenia $Ax = b$, $b \neq 0$ môžeme b nahradiť vektorom Ay pre náhodne zvolené y.

Nami hľadané riešenie potom bude $x - y$. Na tento nový systém môžeme aplikovať Montgomeryho variantu blokového LA. Tento vyzerá nasledovne: zvolíme si náhodnú maticu V_0 rozmeru $n \times N$ a počítame iteračne

$$W_i = V_i S_i$$

$$V_{i+1} = AW_i S_i^T + V_i + \sum_{j=0}^{i} W_j C_{i+1,j} \quad (i > 0) \tag{7}$$

kým pre nejaké m bude $V_m A V_m = 0$. Matice S_i rozmerov $N \times N_i$ vyberajú vhodné stĺpcové vektory V_i aby k $W_i^T A W_j$ ešte existovala inverzná matica. Matice C_{ij} sa vypočítajú ako

$$C_{i+1,j} = (W_j^T A W_j)^{-1} W^T A (A W_i S_i^T + V_i) \tag{8}$$

a sú volené tak, aby $W_j^T A V_{i+1} = 0$ pre $j \leq i$.
Riešenie x potom nájdeme ako

$$x = \sum_{i=0}^{m-1} W_i (W_i^T A W_i)^{-1} W_i V_0. \tag{9}$$

Ako sme už povedali my potrebujeme získať viacero riešení (1) a teda potrebujeme vlastne spustiť BLA viac krát pre rôzne náhodné vektory y. Teraz môžeme využiť skutočnosť, že pracujeme v poli Z_2, ktorého prvky môžeme reprezentovať ako bity v počítači. Ak chceme realizovať výpočet na počítači s N-bitovou architektúrou, môžeme naraz spustiť vlastne až N takýchto BLA naraz a to riešením systému $AX = AY$, kde matica Y rozmerov $n \times N$ pozostáva z našich náhodných vektorov y. Teda na počítači s 32 bitovou architektúrou bude šírka matice v BLA $N = 32$. Po nagenerovaní matice Y inicializujeme $V_0 = AY$ a aplikujeme blokový LA až kým $V_m^T A V_m = 0$ pre nejaké m. Počítame

$$X = \sum_{i=0}^{m-1} W_i (W_i^T A W_i)^{-1} W_i V_0. \tag{10}$$

Ak V_m je nulová matica potom máme $AX = AY$. Často však skončí algoritmus s $V_m A V_m = 0$ a $V_m \neq 0$. Zrejme potom ani $AX \neq AY$. V tomto prípade sa dá nájsť nekoľko riešení (2) použitím GE na maticu získanú prilepením matice $A(X - Y)$ k AV_m. Podobne ako pri klasickom LA sa aj tu dá iteračný vzťah zjednodušiť a to na :

$$V_{i+1} = AV_i S_i S_i^T + V_i D_{i+1} + V_{i-1} E_{i+1} + V_{i+2} F_{i+1} \tag{11}$$

pre $i \geq 0$, kde

$$D_{i+1} = I_N - W_i^{inv} (V_i^T A^2 V_i S_i S_i^T + V_i A V_i)$$

$$E_{i+1} = -W_{i-1}^{inv} V_i^T A V_i S_i S_i^T \tag{12}$$

$$F_{i+1} = -W_{i-2}^{inv}(I_N - V_{i-1}^T A V_{i-1} W_{i-1}^{inv})(V_{i-1}^T A^2 V_{i-1} S_{i-1} S_{i-1}^T + V_{i-1} A V_{i-1}) S_i S_i^T$$

a

$$X = \sum_{i=0}^{m-1} V_i W_i^{inv} V_i^T V_0$$

pričom

$$W_i^{inv} = S_i (S_i^T V_i^T A V_i S_i)^{-1} S_i^T .$$

Pre $i < 0$ definujeme $W_i^{inv} = 0$ a $V_i = 0$ a $S_i = I_N$. Prakticky nám stačí v každej iterácii počítať AV_i, $V_i(AV_i)$, $(AV_i)^T(AV_i)$, pretože členy s menšími indexami už poznáme z predchádzajúcich výpočtov. Člen AV_i sa počíta ako $B^T BV_i$ pretože násobenie maticou A by bolo časovo omnoho náročnejšie. Postup výpočtu matíc W_i^{inv} a S_i je uvedený v [2] na strane 116.

3 Násobenie matíc

Prevažnú väčšinu výpočtového času zaberajú násobenia matíc, preto si pri implementácii násobenie vyžaduje najväčšiu pozornosť.

Vo všeobecnosti je vzorec pre výpočet prvkov matice $P = QR$ nasledovný:

$$P_{i,j} = \sum_k (Q_{i,k} R_{k,j}). \tag{13}$$

V našich výpočtoch patria prvky matíc P, Q a R do Z_2 (preto operáciu sčítania realizujeme pomocou logickej operácie xor - exlizívny súčet). Ďalej sú riadky matíc P a R bitovo reprezentované slovom počítača.

Označme i-ty riadok matice P symbolom $\boldsymbol{p_i}$ a k-ty riadok matice R symbolom $\boldsymbol{r_k}$. Potom môžeme napísať vzťah:

$$\boldsymbol{p_i} = \sum_k (Q_{i,k} \boldsymbol{r_k}). \tag{14}$$

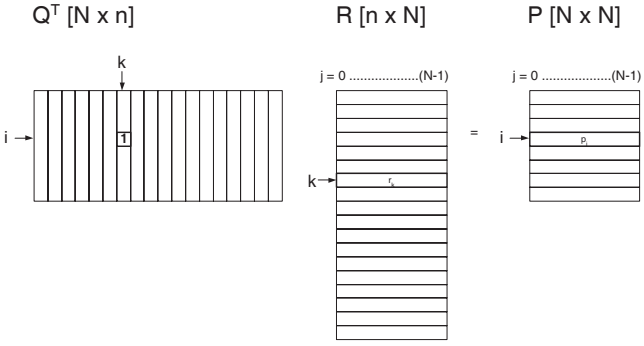
Ten sa dá využiť pre zrýchlenie násobenia matíc, pretože člen xor-sumy: $Q_{i,k}\boldsymbol{r_k}$ môžeme vypočítať naraz (za predpokladu, že šírka riadku $\boldsymbol{r_k}$ je rovná alebo menšia ako počet bitov registra počítača). Prakticky:

$$Q_{i,k}\boldsymbol{r_k} = \begin{cases} \mathbf{0} & \text{pre} \quad Q_{i,k} = 0 \\ \boldsymbol{r_k} & \text{pre} \quad Q_{i,k} = 1 \end{cases}$$

preto na výpočet $\boldsymbol{p_i}$ stačí sčítať tie $\boldsymbol{r_k}$, pre ktoré $Q_{i,k} = 1$.

Ak za maticu Q dosadíme maticu B alebo B^T zo systému (1) reprezentovanú zoznamom jednotiek, potom je výhodné pri výpočte použiť blokový algoritmus násobenia matíc, ktorý znižuje počet prístupov do hlavnej pamäte počítača (RAM) [3].

V nasledujúcom texte si opíšeme príklad implementácie násobenia bitovo reprezentovaných matíc zobrazeného na obrázku 1. Rozmery matíc sú: $Q^T[N \times n]$, $R[n \times N]$, $P[N \times N]$. Budeme predpokladať $N = 32$. V matici Q^T sú slovami počítača (napr. údajovým typom `unsigned int` jazyka C) reprezentované stĺpce, v maticiach R a P sú slovami počítača reprezentované riadky.

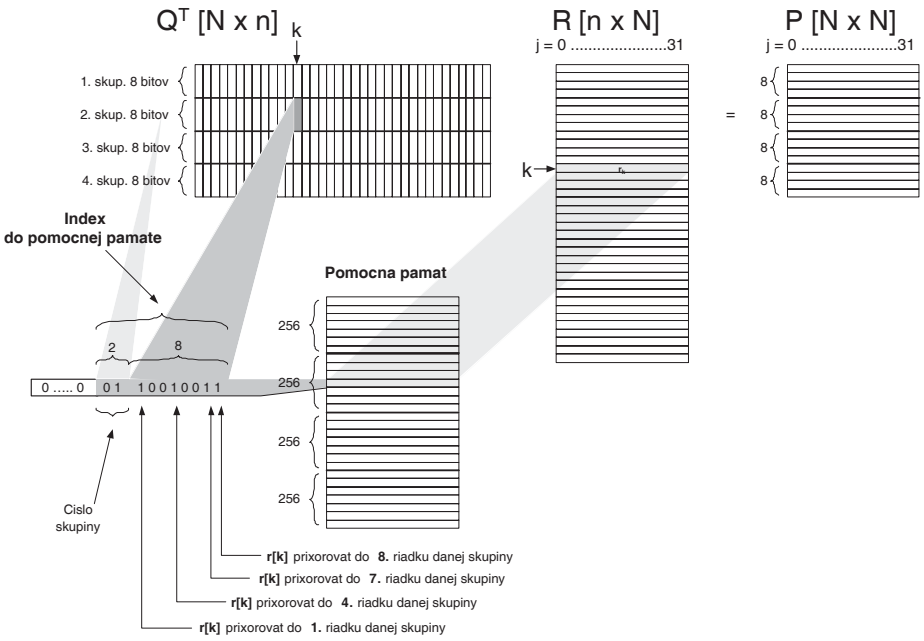


Obrázok 1. Násobenie bitovo reprezentovaných matíc

V implentácii rýchleho násobenia matíc sú stĺpce matice Q^T rozdelené do 4 skupín po 8 bitov (obr. 2). Ďalej sa používa pomocná pamäť rozdelená na 4 časti, každá jej časť má $2^8 = 256$ riadkov. Prvá časť pomocnej pamäte zodpovedá prvým 8 riadkom matice Q^T a matice P, druhá jej časť zodpovedá druhým 8 riadkom matice Q^T a matice P, atď. Algoritmus má 2 časti.

V prvej časti sa najprv vynuluje pomocná pamäť a potom sa postupne do nej pripočítavajú riadky matice R. Algoritmus postupne prechádza cez všetky stĺpce matice Q^T a v každom stĺpci po 4 skupinách bitov. Označme písmenom k index práve čítaného stĺpca matice Q^T. Vtedy sa na určité miesta do pomocnej pamäti pripočítava k-ty riadok matice R. Index miesta v pomocnej pamäti, do ktorej sa pripočíta k-ty riadok matice R, sa určí podľa hodnôt bitov v danej skupine a podľa poradia skupiny nasledovným spôsobom. Nech je napr. práve spracovávaná 2. skupina bitov v k-tom stĺpci matice Q^T, ktorá má hodnotu 10010011 (obr. 2). Posledných 8 bitov indexu je určených hodnotou skupiny bitov t.j. 10010011. 9.a 10. bit indexu sprava, je určený číslom práve spracovávanej skupiny bitov, takže pre 2. skupinu bude hodnota týchto bitov 01 (1. skupina bitov má číslo 00, 2. skup. 01, 3. skup. 10, 4. skup. 11). Predchádzajúce bity indexu sú nulové.

V druhej časti algoritmu sa podľa obsahu pomocnej pamäte vypočíta hodnota výslednej matice P. Ukážeme si ako sa zostaví 9. až 16. riadok matice R, zodpovedajúci 2. skupinám bitov v matici Q^T. Hodnoty ostatných riadkov sa vypočítajú analogicky.



Obrázok 2. Efektívnejšia násobenie bitovo reprezentovaných matíc

Najprv sa vynuluje obsah matice R. 1. bit 2. skupiny bitov v každom stĺpci matice Q^T určuje, či sa zodpovedajúci riadok matice R má pripočítať do 9.(=8+1.) riadku matice P. Preto sa do 9. riadku matice P pripočítajú všetky hodnoty s pomocnej pamäte, ktoré zodpovedajú 2. skupine bitov a ich index na 8. mieste sprava obsahuje hodnotu 1. t.j. do 9. riadku matice R sa sčítajú tie hodnoty z pomocnej pamäte, ktorých index vyhovuje maske: 0....0 01 1xxxxxxx (x=ľubovolná hodnota) . Do 10. riadku matice R sa sčítajú všetky hodnoty z pomocnej pamäte, ktorých index výhovuje maske: 0....0 01 x1xxxxxx. Do 11. riadku sa sčítajú hodnoty z pomocnej pamäte, ktorých index vyhovuje maske: 0....0 01 xx1xxxxx, atď. Do 16. riadku sa sčítajú hodnoty z pomocnej pamäte, ktorých index vyhovuje maske: 0....0 01 xxxxxxx1. Týmto postupom sa (pri spracovávaní 2. skupín bitov) k-ty riadok matice R sa pripočíta do 9., 12. 15. a 16. riadku matice R. Pravdaže k-ty riadok matice R sa pravdepodobne pripočíta aj do iných riadkov, podľa toho aké hodnoty sú v 1., 3. a 4. skupine bitov k-teho stĺpca matice Q^T.

Z implementačných dôvodov sú bity zoskupené po 8 alebo 16 bitov. Voľba počtu bitov v skupine závisí od veľkosti matíc a od veľkosti rýchlej vyrovnávacej pamäte (cache).

4 Záver

Metóda GNFS sa skladá z dvoch hlavných častí: „preosievacia“ fáza a fáza lineárnej algebry riešiaca veľký riedky systém nad poľom Z_2.

Paralelizácia prvej fázy faktorizácie metódou GNFS je dnes podrobne rozpracovaná tak, že umožňuje dosiahnuť vysokú (takmer maximálnu) efektivitu paralelného spracovania. Preto zvyšovaním počtu použitých počítačov rastie takmer priamo úmerne aj dosiahnutý výkon.

Úzkym hrdlom pri faktorizácii sa preto v súčasnosti stáva práve riešenie systému (1). Najlepší algoritmus na jeho riešenie je dnes blokový LA, ktorý už v klasickej podobe nestačí na skutočne veľké sústavy (rozmerov 10^7) a vzniká tak potreba jeho paralelizácie . Problémom však je, že jej efektivitu do značnej miery znižuje skutočnosť, že si jednotlivé výpočtové uzly potrebujú počas výpočtu matíc $V_i^T(AV_i)$, $V_i^T A^2 V_i = (AV_i)^T(AV_i)$ vymienať medzivýsledky, predstavujúce veľký objem údajov.

Otázka návrhu minimalizujúceho časové straty pri vzájomenej komunikácii výpočtových uzlov tak zatiaľ zostáva stále otvorená.

Referencie

[1] C.Lanczos *Solution of sytems of linear equations by minimized iteration* Journal of the National Bureau of Standards,Vol. 49,(1952),pp 409-436

[2] Peter L. Montgomery *A Block Lanczos algorithm for Finding Dependencies over GF(2)* Advances in Cryptology -EUROCRYPT '95,Lecture Notes in Computer Science ,vol 921,Springer-Verlag, pp. 106-120

[3] Aberdeen D., Baxter J.: Emmerald: a fast matrix-matrix multiply using Intel's SSE instructions. Concurrency and Computation: Practice and Experience. Pract. Exper. 2001; 13:103-119

[4] Coppersmith D.: Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. Mathematics of Computation 62 (1994), no. 205, 333-350.

[5] Dongarra J., Duff I., Sorensen D., Van der Vorst H.: Numerical Linear Algebra for High-performance Computers. SIAM Press, 1998.

Latest advances in fully coupled thermal structural calculations using finite element method (FEM) with new energy conservation equation

Ladislav Écsi[1]

[1] Department of Strength of Materials, Faculty of Mechanical Engineering
Slovak University of Technology in Bratislava, Námestie slobody 17
812 31 Bratislava, Slovak Republic
ladislav.ecsi@stuba.sk

Abstract. In this paper some latest advances in fully coupled thermal structural calculations are presented. The analysis is based on a new energy conservation equation, which the author believes represents the most complete formulation of the first principle of thermodynamics. The equation was implemented into a finite element code using large strain/ large deflection formulation utilizing the updated Lagrange method and the extended NoIHKH material model for cyclic plasticity of metals. If the new energy conservation equation proves to be experimentally correct, it will open new perspectives in the study of all coupled thermal-structural problems, mainly in the area of fast/ultra fast thermoelasticity or thermoplasticity.

1 Introduction

In contemporary numerical simulations of coupled-thermal structural problems, either a sequentially coupled thermal structural analysis [3] or a fully coupled thermal structural analysis have been used, where the latter is usually based on a heat equation [2] derived from an internal energy material derivative formulation. Such a heat equation performs well as an energy conservation equation in stationary or static calculations, but in high-speed deformation regions its results may not necessarily be satisfactory. In this paper a fully coupled thermal-structural analysis is presented, using a new energy conservation equation [5] which is complete with respect to the mechanical and thermal energy contributions, and thus presupposes a better solution, mainly in the area of fast/ultra fast thermo-elasticity or thermo-plasticity.

2 Theory Background

The governing equations to describe deformable body behavior during fully coupled thermal structural calculation using the FEM and large strain/large deformation for-

mulation with an updated Lagrange method [1] are the force equilibrium equation [9] and the energy conservation equation [5] known in the following variational form:

$$\int_{\Omega}\rho\dot{\mathbf{v}}\bullet\delta\mathbf{v}dv+\int_{\Omega}\boldsymbol{\sigma}:\delta\mathbf{d}dv=\int_{\Omega}\mathbf{b}\bullet\delta\mathbf{v}dv+\int_{\partial\Omega}\mathbf{t}\bullet\delta\mathbf{v}ds+\sum_{i=1}^{NNode}\mathbf{f}_i\bullet\delta\mathbf{v}_i\,, \tag{1}$$

$$\begin{aligned}&\int_{\Omega}\delta T\rho\dot{\mathbf{v}}\bullet\mathbf{v}dv+\int_{\Omega}\delta T(\boldsymbol{\sigma}:\mathbf{d})dv+\int_{\Omega}(\nabla\delta T)\bullet(\boldsymbol{\sigma}\bullet\mathbf{v})dv+\int_{\Omega}\delta T\rho c\dot{T}dv-\int_{\Omega}(\nabla\delta T)\bullet\mathbf{q}dv=\\&=\int_{\Omega}\delta T\mathbf{b}\bullet\mathbf{v}dv+\int_{\partial\Omega}\delta T\mathbf{t}\bullet\mathbf{v}ds+\int_{\partial\Omega}\delta Tq_n ds+\int_{\Omega}\delta Trdv+\sum_{i=1}^{NNode}\delta T_i\mathbf{f}_i\bullet\mathbf{v}_i+\sum_{i=1}^{NNode}\delta T_iQ_i\end{aligned}\,, \tag{2}$$

with

$$\mathbf{q}=-\mathbf{K}\bullet(\nabla T)\,.$$

Where $\mathbf{v},T$ are the velocity vector and the temperature, $\boldsymbol{\sigma},\mathbf{d},\mathbf{K}$ stand for the Cauchy stress tensor, the strain rate tensor and the heat conductivity tensor, $\mathbf{b},\mathbf{t},\mathbf{f}_i,\mathbf{q},q_n,r,Q_i$ denote the volume force vector, surface traction vector, nodal force vector, heat flux vector, normal heat flux, specific heat source and nodal heat flux and ρ,c represent the material density and specific heat respectively. Considering that the strain rate tensor of an isotropic material has the additive decomposition $\mathbf{d}=\mathbf{d}^e+\mathbf{d}^p+\alpha\dot{T}\mathbf{I}$, where α is the coefficient of thermal expansion and $\mathbf{I}$ is an identity tensor, eqs. (1) and (2) are supplemented with the following constitutive and evolution equations:

$$f=\sigma_{eq}-\sigma_y-R\le 0\,,\ \sigma_{eq}=\sqrt{\frac{3}{2}\left(\hat{\boldsymbol{\Sigma}}-\hat{\mathbf{X}}\right):\left(\hat{\boldsymbol{\Sigma}}-\hat{\mathbf{X}}\right)}\,, \tag{3}$$

$$R=Q\left(1-e^{\left(-b\varepsilon^p\right)}\right), \tag{4}$$

$$\overset{\circ}{\hat{\mathbf{X}}}=\mathbb{C}_{cycl}:\hat{\mathbf{d}}^p-\gamma\left(\varepsilon^p\right)\hat{\mathbf{X}}\dot{\varepsilon}^p\,,\ tr\left(\hat{\mathbf{X}}\right)=0\,,\ \gamma\left(\varepsilon^p\right)=\gamma_\infty-\left(\gamma_\infty-\gamma_0\right)e^{\left(-\omega\varepsilon^p\right)}\,. \tag{5}$$

Equations (3)-(5) represent the extended NoIHKH material model for cyclic plasticity of metals [4], modified for large strain/ large deformation, using combined isotropic and kinematic hardening with associative plasticity [7],[8], where $\mathbb{C}_{cycl}$ is a fourth order cyclic material tensor, $\hat{\boldsymbol{\Sigma}},\hat{\mathbf{X}},\hat{\mathbf{d}}^p$ are the rotated deviatoric stress tensor, the rotated back stress tensor and the rotated plastic strain rate tensor, $\varepsilon^p,\dot{\varepsilon}^p$ denote the accumulated plastic strain and the accumulated plastic strain rate and, $Q,b,\gamma_0,\gamma_\infty,\omega$ stand for the material properties. The model is based on the original NoIHKH material model [6], initially proposed for small strain cyclic plasticity of metals. In the Cauchy stress update calculation as well as in the rotated back stress evolution equa-

tion (5) derivation was used the Jaumann objective rate in the form of the Green-Naghdi objective rate utilizing the following rotation tensors at the midpoint and endpoint configuration of the body:

$$\mathbf{R}^{n+\frac{1}{2}} = \exp\left[\frac{\Delta t}{2}\mathbf{W}^{n+\frac{1}{2}}\right]\bullet\mathbf{R}^{n}, \tag{6}$$

$$\mathbf{R}^{n+1} = \exp\left[\Delta t\mathbf{W}^{n+\frac{1}{2}}\right]\bullet\mathbf{R}^{n}, \tag{7}$$

where $\mathbf{W}^{n+\frac{1}{2}}$ is the spin tensor at midpoint configuration, $\mathbf{R}^{n}$ stands for the rotation tensor at the previous configuration and Δt denotes the time step value. In the tensor exponential function mathematical expression the Rodriguez formula [9] was used.

3 Numerical Example

As a numerical example a solid bar loaded in cyclic bending was studied. In the numerical analysis an 8 node three-dimensional solid element with linear shape functions was used. One end of the bar was fixed and the second end underwent a predefined cyclical vertical deformation of one third of the bar length, while it was fixed in the remaining two directions. Only one loading cycle was realized. The bar was initially at rest with zero initial temperature. Heat convection through all surfaces of the bar was considered, applying zero bulk/environmental temperature. In the calculation no heat source was considered. The case was calculated as static, using a time step of 1 second. The prescribed cyclic vertical deformation at the moving end of the bar was determined with a sine function applying 15-degree increment in each time step. It is however necessary to emphasize that the word “static” applies in the sense that no inertia force was considered, but the deformation velocity was nonzero. As a result, this type of “static” calculation is dependent on the time step value.

3.1 Material Properties

In the calculation properties of a low carbon steel material should have been used, but due to the 32bit computer structure limitation on which the analysis was run, some of these properties had to be changed, to prevent the global stiffness matrix from being nearly singular. As a simplification, in the calculation all material properties were considered to be constant. Table 1 outlines the used material properties in the numerical simulation.

Table 1: Material properties

$c = 15.0\mathrm{J}/\left(\mathrm{kg}\cdot\mathrm{K}\right)$	$\mu = 0.3$
$k_{xx}, k_{yy}, k_{zz} = 50.0W/\left(m\cdot K\right)$	$\mu_{cycl} = 0.3$
$\alpha_x, \alpha_y, \alpha_z = 0.000012K^{-1}$	$\sigma_y = 250000.0Pa$
$h = 0.5W/\left(m^2\cdot K\right)$	$Q = 100000.0Pa$
$T_{bulk} = 0.0K$	$b = 3.0$
$\rho = 7800.0kg/m^3$	$\gamma_\infty = 20.0$
$E = 2100000.0Pa$	$\gamma_0 = 10.0$
$E_{cycl} = 550000.0Pa$	$\omega = 10.0$

3.2 Numerical Results

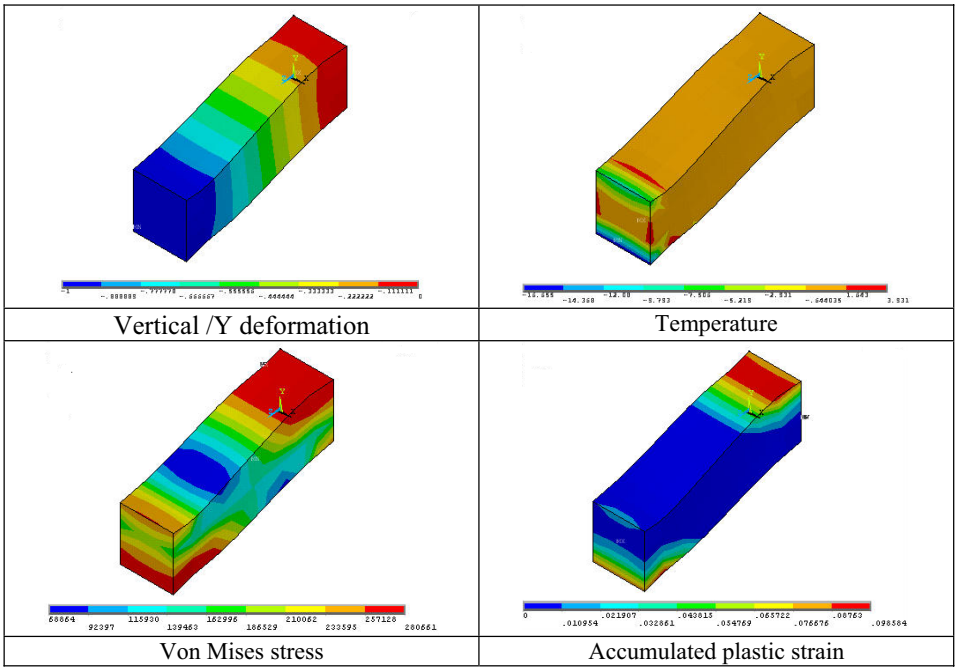


Fig.1. Some results corresponding to the maximum downward deformation of the bar

Figure 1 and figure 2 depict some results corresponding to the limiting deformation of the bar. These are the vertical or Y deformation, the temperature distribution, Von Mises stress and accumulated plastic strain distribution at the maximum prescribed displacement of the bar in downward/upward direction during the cyclic loading of the body.

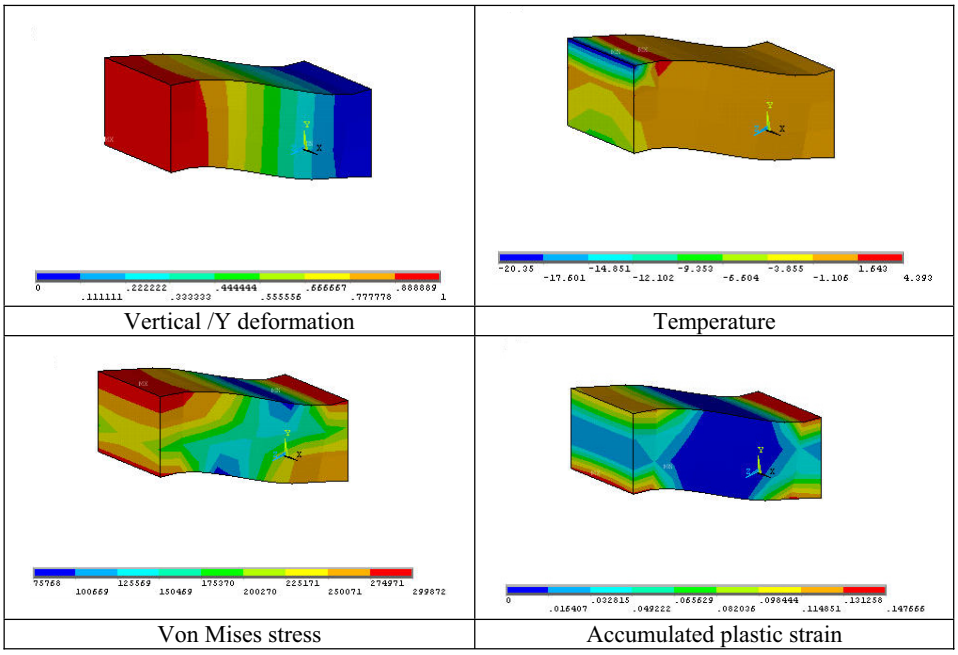


Fig.2. Some results corresponding to the maximum upward deformation of the bar

It is necessary to emphasize here that the results in this paper correspond to a non-realistic material and the presented numerical values are only informative. The author believes that, in order to use real material properties, such calculations will have to be run on a 64bit structure computer. As of today at the Department of Strength of Materials we haven't the possibility to rerun the simulation on a 64bit computer, although we have tried all we could do to proceed in that direction. The presented numerical example however shows that the calculation is capable of running on a 32bit computer and this fact presupposes that it will run on a 64bit computer too. As soon as numerical results for a real material are available, it will be possible to realize an experimental verification of the calculation. These experiments however will not be simple, as the verification will have to be carried out at high-speed deformations of the solid body.

4 Conclusion

In this paper a fully coupled thermal structural analysis with large strain/large deformation formulation was presented. The calculation is based on a new energy conservation equation, which is complete with respect to the thermal and mechanical and energy contributions, and thus presupposes a better solution of the coupled thermal structural problems, mainly in the area of fast/ ultra fast thermoelasticity or thermoplasticity. In the numerical solution the extended NoIHKH material model modified for large strain/ large deflection cyclic plasticity of metals was used. The presented paper represents one of the first outcomes of a three-year project devoted to the fully coupled thermal structural simulations, which the author considers to be positive. If the new energy conservation equation proves to be experimentally correct it will open new perspectives in the study of all coupled thermal structural processes which take place at high-speed deformations. Depending on future experimental verification, the new energy conservation equation can still be improved, if necessary, by introducing a heat source in it, to take into account the amount of energy dissipated into heat during plastic deformation. However, to propose a mathematical formula for the heat source is more an experimental problem than a mathematical one.

References

1. Bathe, K.J.: Finite Element Procedures in Engineering Analysis, Prentice-Hall, Englewood Cliffs, N.Y., (1982)
2. Batra, R.C., Love, B.M.: Mesoscale analysis of shear bands in high strain rate deformations of tungsten/ nickel-iron composites, Journal of Thermal Stresses, Vol 28, No. 6-7, (2005) 747-782
3. ESI Group: Sysweld 2000, Theory Manual, (2000)
4. Écsi, L.: Extended NoIHKH model usage for cyclic plasticity of metals, In proceedings of the 7th International Conference on Applied Mechanics, Hrotovice, Czech Republic, 29.3-1.4. (2005)
5. Écsi, L.: Numerical behavior of a solid body under various mechanical loads using finite element method with new energy balance equation for fully coupled thermal-structural analysis, In proceedings of the 6th International Congress on Thermal Stresses, TS2005, Vienna, Austria, 26-29. May (2005), 543-546
6. Lemaitre, J.: Handbook of Materials Behavior Models 1, Academic Press, N.Y., (2000), 213-222
7. Maugin, A.G.: The Thermomechanics of Plasticity and Fracture, Cambridge University Press, Cambridge, (1992)
8. Simo, J.C., Hughes, T.J.R.: Computational Inelasticity, Springer, N.Y., (2000)
9. Souza, E.A., Peric, D.J., Owen, D.R.J.: Computational Plasticity, Small and Large Strain Finite Element Analysis of Elastic and Inelastic Solids, Swansea, to be published

Experimenty ATLAS, ALICE v CERN a ich výpočtové nároky na GRID

P. Chochula[1], S. Kapusta[1], M. Babik[2], T. Darányi[5], A. Jusko[2], Ivan Králik[2], B. Pastirčák[2], P. Stríženec[2], P. Šťavina[3], M. Zagiba[3], and J. Urbán[4]

[1] CERN, Ženeva Švajčiarsko
[2] 'UEF SAV, Košice Slovenská republika
[3] FMFI UK, Bratislava Slovenská republika
[4] PF UPJŠ, Košice Slovenská republika
[5] UI SAV, Bratislava Slovenská republika

Abstract. The Large Hadron Collider (LHC) currently being built at CERN will produce enormous amount of data (nearly 15 PetaBytes of data annually), which the community of high energy physics around the world will access and analyse. This paper describes the computing requirments and specifications of ALICE and ATLAS LHC experiments and discuss the contributions of Slovak institutes involved in these experiments from this point of view.

1 Úvod

ATLAS a ALICE sú dva zo štyroch experimentov, ktoré sa uskutočnia na urýchľovači LHC (Large Hadron Collider) v CERN (European Organization for Nuclear Research) v Ženeve. V roku 2007, keď začne LHC operovať, bude poskytovať najenergetickejšie zrážky elementárnych častíc s najintenzívnejšími doteraz dosiahnutými zväzkami v laboratórnych podmienkach. Experimentálne zariadenia snímajúce tieto zrážky budú najväčšími a najefektívnejšími detektorovými systémami, ktoré boli v oblasti jadrovej a časticovej fyziky skonštruované. LHC bude urýchľovať zväzky protónov a iónov (najťažšími budú ióny olova Pb). To umožní štúdium protón-protónových (p-p) zrážok pri energii 14 TeV a Pb-Pb zrážok pri energii 5.5 TeV/nukleón. Najvyššia luminosita protónového zväzku bude bude $10^{34}\ cm^{-2}s^{-1}$.

Obe zariadenia, ATLAS aj ALICE, sú budované v širokej medzinárodnej spolupráci. Kolaborácia ATLAS zahŕňa 150 inštitúcií a participuje v nej asi 1300 vedeckých pracovníkov z 34 krajín. Kolaborácia ALICE je trocha menšia, experimentu sa zúčastňuje 83 inštitúcií resp. 1000 pracovníkov z 28 krajín.

Zo Slovenska sa na týchto dvoch experimentoch v CERN zúčastňujú 3 pracoviská: **FMFI UK Bratislava, PF UPJŠ Košice, ÚEF SAV Košice**. Účastníkov experimentu ATLAS je 20, experimentu ALICE 25.

1.1 Experiment ATLAS

Experimentálne zariadenie ATLAS je najväčším z experimentov na LHC v CERN. ATLAS bude študovať základné otázky fyziky vysokých energií ako je pôvod

hmotnosti, procesy, ktoré viedli k prevahe hmoty nad antihmotou vo vesmíre, príznaky supersymetrie, či možno aj produkciu čiernych mikrodier.

Detektor ATLAS pozostáva z množstva subdetektorov rôznych technológií (polovodičové, plynové, scintilačné detektory, detektory pracujúce s kvapalným argónom slúžiacim ako signálne prostredie, ...) s obrovským množstvom čítacích kanálov. Usporiadanie experimentálneho zariadenia ATLAS s jeho systémom subdetektorov je na obr. 1.

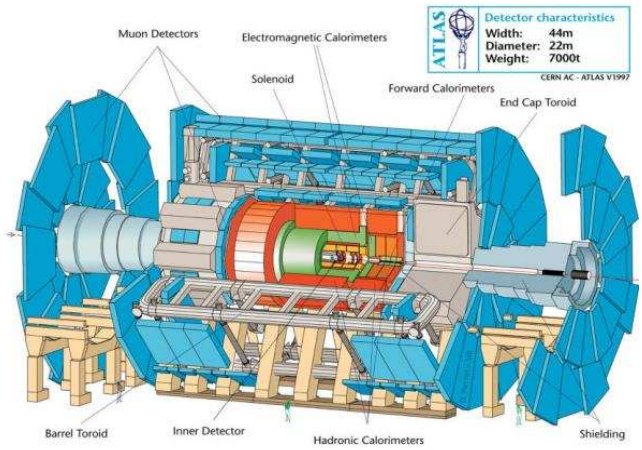


Fig. 1. Experimentálne zariadenie ATLAS

1.2 Experiment ALICE

Experiment ALICE je ako jediný z experimentov na LHC dedikovaný na štúdium silnointeragujúcej hmoty pri extrémnych teplotách a/alebo hustotách energie, kedy vzniká nový stav hmoty - kvark-gluónová plazma (QGP), čo sa dosahuje zrážkami ťažkých iónov. Podrobný výskum tohto nového stavu hmoty sľubuje objasniť kľúčové otázky teórie silnej interakcie - kvantovej chromodynamiky (QCD), ako je narušenie chirálnej symetrie, štruktúra vákua QCD, hadronizácia. Štúdium kvark-hadrónového fázového prechodu je zároveň v úzkom vzťahu s charakterizáciou podmienok, ktoré prevládali v rannom vesmíre niekoľko mikrosekúnd po tzv. *veľkom tresku*.

Komplexný systém subdetektorov ALICE umožní simultánne štúdium viacerých signatúr QGP, ktoré sa prejavia sa vo vlastnostiach sekundárnych hadrónov, miónov, elektrónov a fotónov produkovaných v zrážkach ťažkých iónov. Celý detektor obsahuje viac ako 15 miliónov detekčných elementov [3].

Okrem zrážok ťažkých iónov (Pb-Pb), bude experiment študovať aj zrážky ľahších jadier (Ar-Ar), ale aj protón-protónové a protón-jadrové interakcie. Schéma experimentálneho usporiadania s hlavnými subdetektormi ALICE je na obr. 2.

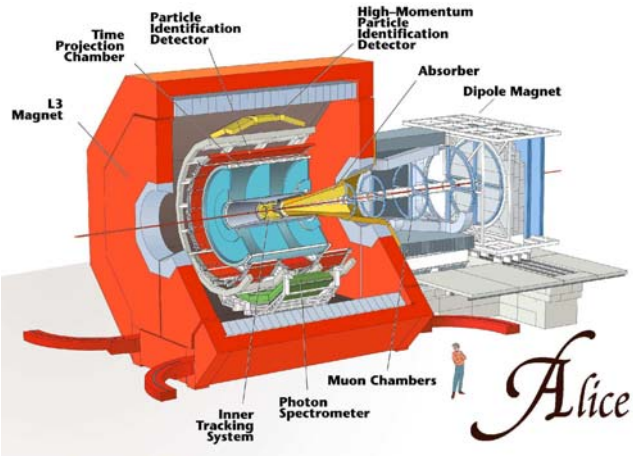


Fig. 2. Experimentálne zariadenie ALICE

V oboch experimentoch prechádzajú signály z detektorov viacerými úrovňami triggrov až sú nakoniec zberané dáta s frekvenciou rádovo 100 Hz. Namerané ale aj simulované dáta prechádzajú ďalej v spracovaní cyklom rekonštrukcia-kalibrácia-selekcia-analýza.

2 Distribuované počítanie v experimentoch ATLAS a ALICE

V dôsledku vyššie popísanej situácie budú v porovnaní s predošlými skúsenosťami fyzici konfrontovaní s bezprecedentným objemom dát, rádovo 10 PB/rok (cca 20 miliónov CD nosičov), ktoré bude nutné každoročne spracovať a ktoré budú musieť byť spracované desiatkami pracovísk. Samotná analýza dát vyžaduje pritom výpočtovú kapacitu na úrovni dnešných 70000 najvýkonnejších procesorov. Nie je možné klasicky splniť uvedené nároky. Riešením je zužitkovať výpočtové kapacity distribuované po celom svete. CERN je preto vedúcou silou pri vývoji technológie počítačového GRIDu. Úlohou je zjednotiť výpočtové centrá do jednej virtuálnej organizácie (VO), ktorá poskytne dostatočnú kapacitu na analýzu dát produkovaných v rámci LHC. V rámci tejto VO budú dáta a výpočtová kapacita distribuované medzi príslušnými pracoviskami, ktoré budú prepojené vysokorýchlostnou počítačovou sieťou. Prípravou hierarchickej počítačovej štruktúry, správou zdrojov a vývojom potrebného programového pre experimenty na LHC v CERN sa zaoberá projekt LHC Computing Grid (**LCG**) [5]. Výpočtové kapacity **LCG** sú operované cez **EGEE** *middleware*, čo je projekt, ktorý bude slúžiť clému európskemu výskumnému priestoru, medziiným aj komunite i fyziky vysokých energií. **LCG** je iniciatívou pre vytvorenie gridového prostredia pre analýzu, rekonštrukciu a spracovanie dát, ako aj pre simulácie primárnych protón-protónových a ťažko-iónových interakcií. Existuje

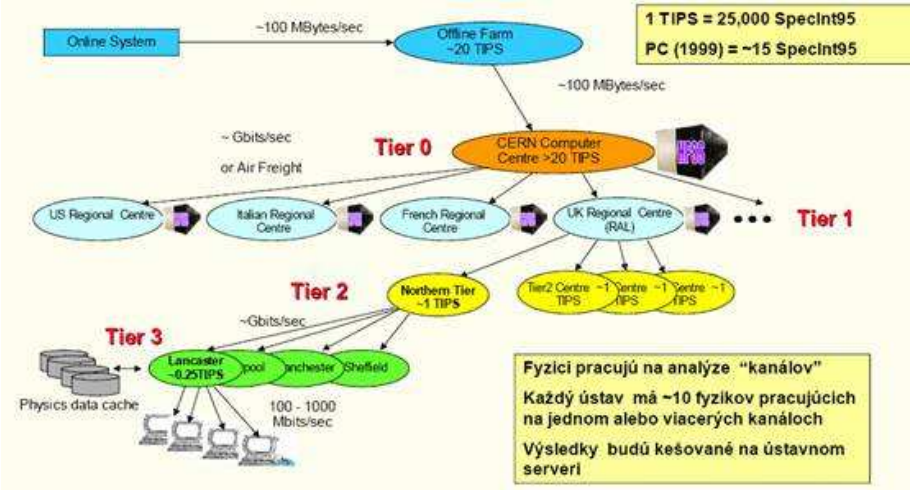


Fig. 3. LCG model

viacero možných implementácií pre vytváranie gridovej infraštruktúry vo fyzike vysokých energií (LCG, AliEn, NorduGrid, ...) ich princípy sú však viac-menej rovnaké, ale treba pripomenúť, že v tomto bode nie sú jednotlivé CERN experimenty jednotné. Experiment ATLAS prioritne využíva **LCG2** [2], experiment ALICE [4] je orientovaný na **AliEn**. Oba balíky dokážu vzájomne komunikovať a môžu byť inštalované paralelne na tej istej počítačovej infraštruktúre.

2.1 Architektúra LCG

Architektúra LHC Gridu je striktne hierarchická, vyššie úrovne (TIER) poskytnú dáta nižším. TIER0 je najvyššia vrstva hierarchie a bude fyzicky inštalovaná v CERN, Jej primárnou úlohou je uchovať dáta produkované experimentami. Spolu s úrovňou TIER1 má za úlohu zabezpečiť kalibráciu a primárnu rekonštrukciu dát. Vrstvy TIER2 a nižšie sú určené predovšetkým na fyzikálnu analýzu takto predpripravených údajov a na produkciu Monte-Carlo simulácií. Vrstva TIER3 je charakteristická pre menšie univerzitné resp.ústavné centrá. Pre prístup k TIER3 sa využívajú PC koncových užívateľov (TIER4), viď obr. 3.

2.2 AliEn

AliEn (Alice Environment) je gridová štruktúra vyvinutá kolaboráciou ALICE a už niekoľko rokov je úspešne používaná. Systém je vybudovaný z *Open Source components*, používa *WEB Services model* a štandardné sieťové protokoly. Web Services hrajú centrálnu úlohu pre použitie AliEnu ako prostredia na distribuované počítanie. Užívatelia komunikujú výmenou správ v protokole *SOAP (Simple Object Acces Protocol)*. **AliEn** pozostáva z nasledujúcich komponent a

služieb: autentifikácia, autorizácia, workload a data management systémy, file a metadátové katalógy, informačný servis, grid a job monitoring servis, archivačné a výpočtové elementy [6].

2.3 Požiadavky experimentov ALICE a ATLAS na GRID

Požiadavky experimentov vychádzajú z odhadov založených na veľkosti a frekvencii jednotlivých prípadov interakcie. Predpokladané hodnoty za prvé dva roky prevádzky LHC sú v nasledujúcej tabuľke:

	ATLAS	ALICE
Frekvencia prípadov	140 Hz	30/100 Hz
Veľkosť prípadov	1,6 MB	86,5/2,5 MB
Množstvo uložených dát/prípad	$0,8 * 10^9$/rok	$1,0 * 10^9$/rok
Počet simulovaných prípadov	$1,0 * 10^{10}$/rok	$0,5 * 10^{10}$/rok
Čas rekonštrukcie	0,64 kSI95 s/prípad	300 kSI2000 s/prípad
Čas simulácie	3,00 kSI95 s/prípad	20 MSI2000 s/prípad
Úhrnné množstvo dát	10 PB/rok	5 PB/rok

Pre ALICE prvý údaj zodpovedá zrážke iónov Pb, druhý p-p zrážke. Požiadavka na výkon na úrovni TIER2 farmy je 350 resp. 450 kSI200 pre ATLAS resp. ALICE a požadovaná disková kapacita by mala byť 158 resp. 187 TB opäť pre ATLAS resp. ALICE.

3 Slovenská účasť v experimentoch ATLAS a ALICE

V rámci účasti slovenských pracovísk v LHC experimentoch je nutné pristúpiť ku koordinovanej príprave pre účasť na ich fyzikálnom programe. Vyžaduje to vybudovanie počítačovej infraštruktúry umožňujúcej prístup k experimentálnym dátam aj ich samotnú analýzu. V súčasnosti pracujú na vyššie spomínaných ústavoch štyri skupiny, ktoré riešia nasledujúce softwarovo zamerané úlohy:

- **ALICE Bratislava** (prof. RNDr. Branislav Sitár,DrSc.)
  - štúdium charakteristík a simulácie pixel.detektorov
  - vývoj programového vybavenia pre kontrolný systém experimentu
- **ALICE Košice** (RNDr.Ladislav Šándor,CSc.)
  - programové zabezpečenie centrálneho triggera ALICE
  - radiačné simulácie pre experiment ALICE
  - fyzikálne simulácie a vývoj softwaru pre analýzu údajov ALICE
- **ATLAS Bratislava** (Doc.RNDr. Stanislav Tokár, RNDr.)
  - hadrónová kalibrácia kalorimetrie (spolu s ATLAS KE)
  - participácia naa analýze produkcie a štúdium vlastnosti top kvarku
- **ATLAS Košice** (Doc.RNDr.Dušan Bruncko, CSc.)
  - *validization* GEANT4 pre hadrónové procesy
  - on-line kalibrácia hadrónových end-cap kalorimetrov

Efektívna analýza dát z LHC experimentov a tým aj náš podiel na fyzikálnom programe CERN si vyžaduje veľké množstvo výpočtových zdrojov nedosiahnuteľné bežnými prostriedkami. Prístup k experimentálnym dátam bude možný iba v rámci hierarchie GRID. Povinný HW vklad našich ústavov do experimentov ráta s poskytnutím výpočtovej kapacity. Vďaka technológii GRID možno vklad realizovať sprístupnením časti kapacity PC fariem. N základe týchto potrieb a s ohľadom na odlišnosť úloh jednotlivých skupín bol navrhnutý projekt SLCG na vybudovanie infaštruktúry kompatibilnej s LCG. V rámci tohto projektu sú budované dve počítačové farmy - v Bratislave a Košiciach. Obe farmy budú poskytovať rovnaký výkon a ich financovanie bude symetrické a bude sledovať profil výkonu ako by sa mal zvyšovať u jednej farmy v nasledujúcich rokoch podľa tejto tabuľky:

Rok	Počet CPU	Disková kapacita [TB]
2005	15	2
2006	25	4,4
2007	50	9,4
2008	80	17,4
2009	90	25,4
2010	100	35,4

3.1 Súčasný stav

V súčasnosti je košická PC farma obsadená 14 CPU a poskytuje 1,5 TB diskového priestoru. Bratislavská PC farma po zakúpení nových nódov bude mať 35 CPU a 3,0 TB disku. Obe farmy boli úspešne pripojené do LCG2. Mnoho z výpočtových aktivít už beží, predovšetkým simulácie, či vývoj softwaru pre spracovanie dát. Postupne prebieha v oboch experimentoch séria výpočtových kampaní, tzv. Data Challenge, ktoré v narastajúcej zložitosti testujú vyvíjané komponenty softwaru. Našou snahou hlavne v experimente ALICE je zapojiť sa čo najskôr do účasti na Data Challenge, čo je typická aplikácia pre GRID.

Na KE PC farme boli už v rámci LCG v posledných dvoch mesiacoch počítané špeciálne simulácie odozvy kalorimetrov pre ATLAS. Budú slúžit na kalibráciu kalorimetra, tj. na rekonštruovanie celej energie skutočne pohltenej kalorimetrom zo signálov z aktívnych častí kalorimetra. Simulačné joby sú spúšťané pomocou LCG2 *middleware*, s pomocou *resource brokers* v CERNe a v INFN Taliansko.

References

1. Atlas Technical Proposal - CERN-LHCC-94-43,
2. Computing Technical Design Report - CERN-LHCC-2005-022,
3. Technical Proposal ALICE - CERN-LHCC-95-71,
4. ALICE TDR Computing - CERN-LHCC-2005-018,
5. http://lcg.web.cern.ch/LCG
6. P.Saiz et al., Nucl.Instrum.Methods **A502** (2003) 437-440

e-Science: Experiences with utilization of the grid computational model based on using of the ARC middleware

Jozef Černák, Alexander Dirner, and Martin Savko

University of P. J. Šafárik in Košice, Faculty of Sciences, Institute of Physics, Jesenná 5, Košice, Slovak Republic

Abstract. The aim of this contribution is to present the long term personal experiences with the grid computing which were obtained during utilization of the ARC middleware. The ARC middleware is provided by the open academic grid project: NorduGrid. In May 2002 NorduGrid built up a Grid infrastructure which has been suitable for production-level research tasks. Around this university project an international multidisciplinary community of users grew up. We successfully applied the ARC middleware which is robust, reliable and mainly user friendly, to solve the research tasks regarding to complex avalanche dynamics and an image pattern recognition of microscopic video sequences. In the near future the computational tasks from another research areas such as biophysics, biochemistry, computational chemistry and high energy physics (CERN LHC experiments) are expected.

1 ARC middleware (NorduGrid)

ARC middleware [1] provides an implementation of the fundamental Grid services [2], such as information services, resource discovery and monitoring, job submission and management, brokering and data management and resource management. The middleware is built upon standard Open Source solutions like the OpenLDAP, OpenSSL, SASL and Globus Toolkit (GT) libraries. It is based on innovative solutions which are essential for a production quality middleware: Grid Manager, gridftpd, the information model and providers. A "personal" broker shown in the Fig. 1 is integrated into User Interface. This intelligent brokering client mainly contributes to the robustness of the whole Grid system i.e. no critical central elements of the Grid are present. The jobs which are considered to run in the Grid must be described by the extended Resource Specification Language (xRSL) [3] which is derived from the standard RSL [4]. The current status of submitted jobs and available resources in the NorduGrid are reachable using the clear monitoring system [5].

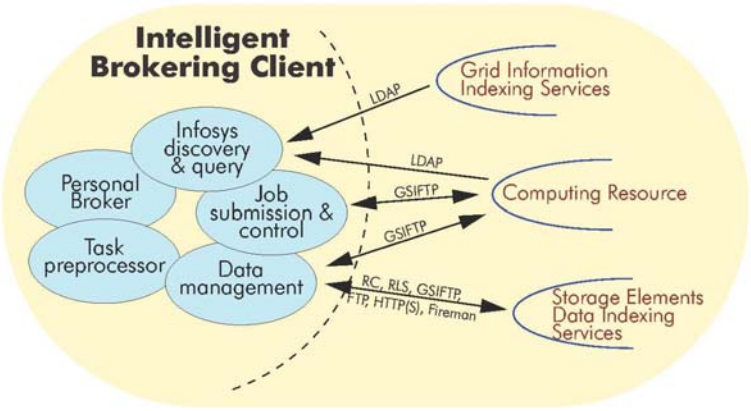


Fig. 1. Intelligent brokering client i.e. any user has own broker witch is a part of his client.

2 Computer simulations of cellular automaton and pattern recognition in digital images

The current realization of the ARC middleware enable us to very easy porting of any non-interactive computer programs written in C, C++ (and another script languages bash, perl, python or Java) which are running in the OS Linux to the Grid. In the project NorduGrid there are no special limitations regarding the research field, i.e. no specific research field is preferred. This openness is very fruitful because a strong community of grid users from different research ares such as: particle physics (VOs: ATLAS and ALICE), statistical physics, biophysics, chemistry and informatics evolved naturally [6].

At University of P.J. Safarik in Kosice two qualitatively different problems were ported in the Grid. To do this we need to create simple scripts written in xRSL [4,3] since more advanced solutions are based on a design of user specific clients or on utilization of grid portals [7].

2.1 Specification of the tasks solved by the grid computational model

Avalanche dynamics investigated in the frame of self-organized-criticality The goal of this study is to obtain basic information about avalanches propagation in the simplest computer models, which belong to the cellular automaton. The computer simulations take place on two-dimensional lattice of the linear sizes $L = 256 - 2048$. At any site we define threshold $E_c$where the hight E at any site means the energy, stress, or density. The energy $\delta E = 1$ is randomly added to the sites of the lattice until a moment when $E \geq E_c$ i.e. the site begins to be unstable and a some amount of the energy is redistributed to the neighbours to by again stable $E < E_c$. The avalanche is propagated until all sites are

stable $E < E_c$ [8]. Avalanche size distributions follow power law $P(a) \sim a^{-\tau a}$, to find the correct characteristic exponents we need obtain a good statistic where more than 10^7avalanches at different lattice sizes $L = 256, 512, 1024$ and 2048 are necessary to eliminate the finite size effects. This computational task is considered for a parametric task [9] where the parameters are initial deployment of disturbing sites or their density.

Pattern recognition The research of aggregation phenomena, for example colloids, is based on microscopic observations [10]. The current CCD image technology enables grabbing of the video information directly into the PC. On the other hand, the video sequences contain not only investigated information but many disturbing information, for example an unwanted noise, shadows and etc. (see Fig. 2 (left)).

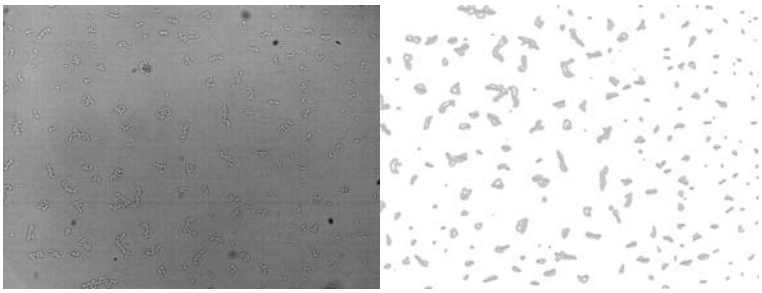


Fig. 2. One of original microscope images (left) and its final version after image processing (right). The black lines represent the maximal lengths of the recognized objects.

To obtain a relevant information we must use a workflow which contains partial activities (programs) to eliminate unwanted information, for example noise reduction, contrast enhancements, thresholding and etc. This computational task is a typical parametric task [9] where the input files are pictures, the output files are files with desired information in our case a list which contains a size of any cluster (see Fig. 2 right).

2.2 Characterization of computational tasks

Computer simulations of avalanche like dynamics were carried out for a few parameters and lattice sizes $L = 256, 512, 1024$ and 2048. The total number of long runs ($t_{CPU} = 30 - 4700$ mins., $CPU@2.4GHz$ Xeon) was about 440 jobs. The number of significant events is about 10^7 for any parameter since the total amount of data is approximately $8\,GB$. During computer simulations in NorduGrid about $30 - 100\,CPUs$ were available.

Single image analysis do not need so much CPU time as in the first application, typically one image is analyzed in time $t_{CPU} < 10$ mins. ($CPU@2.4GHz$

Xeon). However, the total amount of input transfered data (from user to the Grid) is approximately the same, because one image sequence contains about 1200 picture frames since any picture is 5 MB big i.e $6\,GB$ of input data.

2.3 Job submission in NorduGrid

To describe a task an extended version of xRSL derived from the RSL is used. The syntax of this language is relatively very easy. For the simplest tasks with a few input or output files the script are easy and short. They can be manually designed. On the other hand, for larger number of tasks manual preparation is not effective. Therefore we designed a simple sell scripts which are able to generate a submission script and a script which will collect the results from the grid automatically. Then, the user activities are described by a simple workflow as is shown in the Fig. 3.

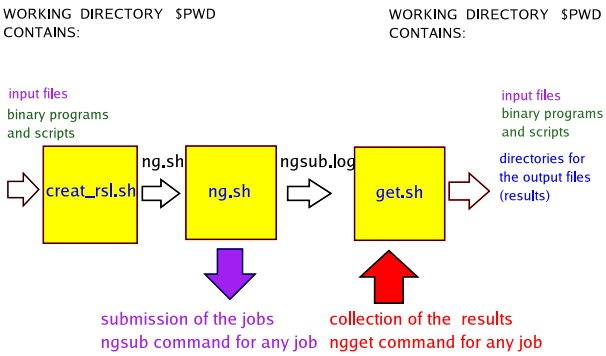


Fig. 3. The user activities

The input data, programs and control scripts are located in the working directory $PWD. The shell *create_rsl.sh* is started by user to create automatically a submission script *ng.sh*. It takes into account input data, programs and their mutual relationships also it considers the user constrains to generate bash script *ng.sh*. The script *ng.sh* is written in xRSL language and it contains a main command ***ngsub*** (ARC middleware) which submits a single job into the Grid. In the next step *ng.sh* script will be started. All input data together with programs and scripts are submitted into the NorduGrid. The details of submission process, for example job identification number, are recorded into the file **ngsub.log**. The user script *get.sh* reads the submission record **ngsub.log** and collects the results from the NorduGrid. The most significant command of the shell *get.sh* is **ngget** which copies the output data into directory of the job. These directories are placed in the working directory $PWD.

The same sell scripts *create_rsl.sh* and *get.sh*, with mirror changes, were used for both computational tasks.

3 Results

During the year 2005 more than 2500 jobs were submitted from our servers (vls.science.upjs.sk, alice.grid.upjs.sk) into NorduGrid, since the 96% of them was correctly finished. This factor is comparable with computing on the local clusters. The all tasks generated at last $20 - 40$ GB of data. It was observed that the most of the jobs finished in the time shorter than 10 hours see the Fig. 4. At present time three servers (vls.science.upjs.sk, alice.grid.upjs.sk, amos.grid.upjs.sk) at University contain 11CPUs and they can serve the students and researchers.

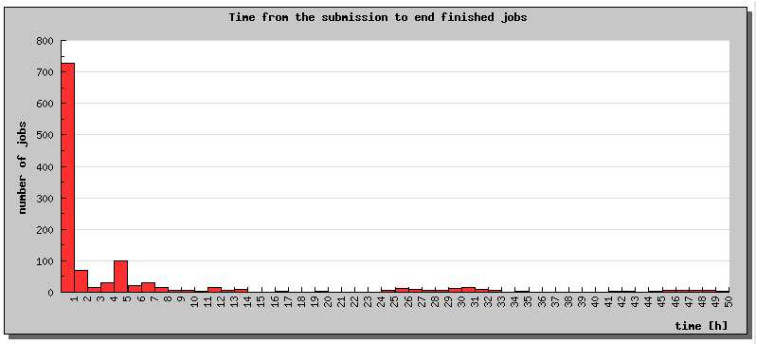


Fig. 4. Task time distribution, for a specific user, obtained during 10 months in 2005.

4 Perspective

In the future we will effort to contribute in the grid computing deployment. Our activities will be focused mainly on the grid users from physics (high energy physics, biophysics, statistical and theoretical physics) and users from the life science. We will continue to compute the tasks which were successfully ported into the NorduGrid. In cooperation with Institute of Informatics at University we are planning to solve selected tasks regarding the grid development such are: a grid security, an interoperability, performance analyses and useful tools for users. Our ambition is to play a significant role in the efforts to build a national grid project which is still missing.

References

1. http://www.nordugrid.org
2. http://www.globus.org
3. O. Smirnova, *"XRSL (Extended Resource Specification Language)"* in http://www.nordugrid.org/papers.html

4. http://www.globus.org/toolkit/docs/3.0/gram/
5. O. Smirnova, *"The Grid Monitor: Usage Manual"*
in http://www.nordugrid.org/papers.html
6. http://www.nordugrid.org/NorduGridV)/
7. GridBlocks Portal, http://gridblocks.sourceforge.net/portal.html
8. J. Cernak, Self-organized criticality: Robustness of scaling exponents, Phys. Rev. E **65**, 046141 (2002).
9. J. Cernak, Zaciname programovat v NorduGrid-e (in Slovak),
http://bioflab.upjs.sk/~cernak/sys_prog/nordugrid-g.pdf
10. J. Cernak, G. Helgesen, and A.T. Skjeltorp, Aggregation dynamics of nonmagnetic particles in ferrofluid, Phys. Rev. E **70**, 031504 (2004).

E-Veda v EGEE – virtuálne organizácie

Tomáš Darányi, Adrián Tóth,
Ladislav Hluchý, Miroslav Dobrucký, Ján Astaloš

Ústav informatiky SAV
Dúbravská cesta 9
845 07 Bratislava,
Tomas.Daranyi@tuke.sk, Adrian.Toth@tuke.sk,
hluchy.ui@savba.sk, dobrucky.ui@savba.sk, astalos.ui@savba.sk,
http://www.ui.savba.sk

Abstrakt Tento dokument podáva prehľad oblastí vedeckého výskumu využívajúce gridovú infraštruktúru, ktorá je budovaná v rámci projektu EGEE. Vysvetlíme pojem virtuálnej organizácie a jej vzťah k používateľom. Je popísaný postup vytvárania nových virtuálnych organizácií výborom EGAAP. Popíšeme špeciálnu virtuálnu organizáciu dteam, následne sú uvedené jednotlivé existujúce virtuálne organizácie zastrešené projektom EGEE spolu s popisom jednotlivých výskumných domén. Na záver je uvedený prehľad niektorých virtuálnych organizácií navrhnutých pre schválenie výborom EGAAP.
Dokument je určený výskumníkom, ktorí majú záujem využívať prostriedky gridu projektu EGEE v rámci vlastného výskumu, buď prostredníctvom zapojenia do už existujúcej virtuálnej organizácie, alebo vytvorením novej.

This document gives a brief overview of the fields of research that uses the EGEE grid infrastructure. Description of the concept of virtual organisation and the relationship between VO and its users is given. The method of creating new VOs by the EGAAP advisory board is explained. The special dteam VO is described. In the following a brief overview of existing virtual organisations covered by EGEE is given, with a description of the respective domains of research. Several proposed projects which are submitted to the EGAAP board are also described.
Document is targeted towards researchers who are considering to use EGEE grid resources in their research by joining one of the existing VOs or by proposing a new one.

1 Virtuálne organizácie a používatelia gridu

Virtuálna organizácia je abstraktná entita, ktorá začleňuje používateľov, ústavy a zdroje do jednotnej administratívnej domény. Môže zastupovať reálne organizácie, projekty (napr. ATLAS, Alice) alebo skupinu výskumníkov z rovnakej oblasti výskumu (napr. Biomed). Existuje tiež virtuálna organizácia založená na regionálnom princípe – jedná sa o VOCE, ktorá podporuje výskumníkov zo strednej Európy. Špeciálnou virtuálnou organizáciou je dteam (deployment team), ktorá začleňuje administrátorov jednotlivých gridových uzlov.

Od každého používateľa gridu sa vyžaduje, aby bol súčasťou aspoň jednej virtuálnej organizácie – bez členstva nie je možné využívať gridové služby. Členstvo používateľovi prideľuje určité práva, napr. prístup k dátam a zdrojom danej virtuálnej organizácie. Počas registračného procesu do LCG si používateľ musí vybrať jemu zodpovedajúcu VO, následne preposlať registračné údaje manažérovi danej VO s účelom validácie. Aby sa stal členom, používateľ musí splniť stanovené požiadavky a dodržiavať podmienky členstva danej virtuálnej organizácie.

Členovia VO v rámci infraštruktúry EGEE musia prijať pravidlá používania prostriedkov EGEE (*EGEE resource usage policy*) a zúčastniť sa na školeniach EGEE (*EGEE training courses*). Je žiadúce, aby aplikácie pripojené do EGEE infraštruktúry pravidelne vykazovali svoje aktivity a uviedli využívanie prostriedkov. Výstupy takýchto výkazov sú vyhodnocované externými recenzentmi.

1.1 EGAAP

EGAAP (EGEE Generic Application Advisory Panel) je výbor, ktorý formou odporučení podáva návrhy manažmentu EGEE na začlenenie nových aplikačných oblastí do infraštruktúry EGEE. Tým EGAAP zastupuje potencionálnych nových používateľov z vedeckej komunity smerom k EGEE.

Organizácie, pracovné skupiny, ktoré majú záujem využívať gridovú infraštruktúru EGEE, podávajú žiadosť o začlenenie výboru EGAAP, ktorý žiadosi vyhodnocuje a výsledok posiela manažmentu EGEE.

2 Oblasti výskumu v rámci EGEE

V tejto časti budú v krátkosti uvedené jednotlivé oblasti výskumu, ktoré využívajú infraštruktúru EGEE.

2.1 Fyzika vysokých energií

Výskumy v oblasti fyziky vysokých energií, konkrétne plánované experimenty na urýchľovači LHC (Large Hadron Collider) boli hybnou silou a motiváciou pre budovanie gridu LCG – *LHC Computing Grid*-u. Urýchľovač LHC obsahuje štyri detektory pre štyri experimenty – ATLAS, Alice, CMS a LHCb. Z toho sú odvodené aj názvy jednotlivých virtuálnych organizácií zastrešujúce výpočty v jednotlivých experimentoch.

V rámci projektov z oblasti fyziky vysokých energií sa grid bude využívať na rekonštrukciu tzv. „eventov" – zrážok častíc v detektore urýchľovača LHC, Monte-Carlo simulácie a rôzne iné spracovanie experimentálnych dát, ktorých množstvo vysoko presahuje možnosti výpočtovej kapacity CERN-u.

ATLAS bude skúmať fundamentálne vlastnosti hmoty a základné sily tvoriace univerzum. Výskum bude prebiehať pomocou zrážok protónov s veľmi vysokou energiou. ATLAS je v súčasnosti najväčšia kolaborácia v oblasti fyziky – na projekte spolupracuje 1800 fyzikov z 34 krajín sveta.

Bližsie informácie nájdete v [1].

CMS je projekt vytvorený na nájdenie hypotetickej častice – tzv. bozónu Higgs, ako aj na otestovanie niektorých zatiaľ experimentálne nepotvrdených teórií o štruktúre hmoty.

Bližsie informácie nájdete v [2].

Alice je zameraná na skúmanie zrážok ťažkých iónov, čo v podmienkach urýchľovača LHC dáva predpoklad k vzniku nového skupenstva hmoty – kvark-gluónovej plazmy.

Ďalšie informácie a kontakt na VO manažéra nájdete v [3].

LHCb je experiment študujúci narušenie tzv. CP-symetrie počas zrážok častíc v urýchľovači.

Ďalšie informácie a kontakt na VO manažéra nájdete v [4].

2.2 Biomed

Skupina zaoberajúca sa aplikáciami z oblasti biológie a medicíny. Možné aplikácie zahŕňajú spracovanie obrazov pre medicínu, bioinformatiku, resp. akékoľvek spracovanie biologických/medicínskych dát, kde vznikne potreba použitia gridovskej infraštruktúry. Okrem iného, táto skupina sa zaoberá aj integráciou existujúcich externých (t.j. EGEE partnermi nepodporovaných) aplikácií do projektu EGEE.

Aplikácie zahŕňajú analýzu snímkov tomografickej emisie, magnetickej rezonancie, analýzu makromolekulárnych štruktúr, modelovania evolúcie genómu.

Domovská stránka a kontakt na VO manažéra nájdete v [5].

2.3 CompChem

Cieľom virtuálnej komunity CompChem (akronym od *Computational Chemistry*) je vývoj aplikácie na výpočet parametrov chemických parametrov na úrovni molekúl, využívajúc výpočtové prístupy, ktoré sú založené na základných fyzických zákonoch bez použitia empirických parametrov.

Viac informácií o VO CompChem (prístup na webstránku je podmienenýplatným EGEE certifikátom) v [6].

2.4 EGEODE

Skratka virtuálnej organizácie vznikla ako akronym od *Expanding Geosciences On Demand*. Jej cieľom je nasadenie gridovskej infraštruktúry pre geo-vedy (*geosciences*) a čiastočne aj pre zdieľanie a spracovanie dát v geofyzikálnom priemysle. Hlavným iniciátorom vzniku tejto VO je firma CGG (*Compagnie Générale de Géophysique*), ktorá má vedúce postavenie na trhu v oblasti služieb týkajúcich sa hlavne ropného priemyslu.

Táto VO sa pýši prvou priemyselnou aplikáciou – *Geocluster, an industry Seismic Processing Solution* – bežiacou v rámci EGEE.

Viac informácií o VO EGEODE (prístup na webstránku je podmienený platným EGEE certifikátom) v [7].

2.5 ESR

Súčasťou výskumu virtuálnej skupiny ESR – skratka od *Earth Science Research* – je sledovanie a štúdium pevniny, oceánov a atmosférických procesov ako aj sledovanie ich vzájomnej interakcie.

Hlavné zamerania výskumu v rámci VO:

1. pozorovanie Zeme,
2. klimatológia,
3. hydrológia,
4. fyzika pevniny – hlavne seizmológia.

Aplikácie zahŕňajú analýzu dát (seizmológia, meteorológia), modelovanie (klimatológia) a simulácie (meteorológia).

Viac informácií o VO ESR (prístup na webstránku je podmienený platným EGEE certifikátom) v [8].

2.6 Magic

Hlavné zameranie skupiny je astronómia a astrofyzika. Teleskop MAGIC vznikol v roku 2003, jeho cieľom je pozorovanie gama žiarenia v pásme od 30 GeV do 1 TeV. Hlavnými iniciátormi projektu boli krajiny Nemecko, Taliansko a Španielsko. Samotný teleskop je umiestnený na Kanárskom súostroví, na ostrove La Palma. Cieľom vzniknutia VO s rovnakým názvom bolo vytvorenie distribuovaného výpočtového systému pre všetky zapojené strany. V súčasnosti pracuje skupina vývojárov na gridifikácii aplikácie *MMCS – Magic Monte Carlo Simulation*.

Viac informácií o VO Magic (prístup na webstránku je podmienený platným EGEE certifikátom) v [9].

2.7 Planck

Misia agentúry ESA s rovnomenným názvom bude spustená v roku 2007. Jej cieľom bude zmapovanie oblohy zatiaľ bezprikladnou kombináciou presnosti, frekvenčného spektra, nezávisloti od systematickej chyby, stability a citlivosti. Medzitým už bolo vytvorené značné množstvo simulácií sledujúc dva hlavné ciele:

- testovanie algoritmov, potrebných na riešenie kritických problémov,
- vyhodnotenie dopadu systematikých vplyvov na celkový vedecký výsledok misie predtým, než budú k dispozícii reálne dáta.

Každý simulačný modul je preverený samostatne, následne sa skúma vzájomná spolupráca modulov po ich spojení. Cieľom VO je portovanie aplikácie na overenie funkčnosti takýchto simulačných blokov.

Viac informácií o VO Planck (prístup je podmienený platným EGEE certifikátom) v [10].

3 Plánované projekty

V tejto sekcii je uvedený zoznam niektorých projektov, ktoré sú v súčastnosti v procese schvalovania v rámci výboru EGAAP.

3.1 ArcheoGRID

Táto VO zameraná na archeologické aplikácie umožňuje využívať výpočtové a pamäťové zdroje gridu EGEE pre generovanie, spravovanie a analýzu dát, ktoré môžu pochádzať z prieskumov terénu, vykopávok alebo laboratórií, tiež pre vytváranie modelov a simulácií z dát s veľkým počtom premenných.

3.2 Fusion

VO zastrešuje výskumníkov z oblasti termonukleárnej fúzie. Hlavné nároky sú na masívne distribuovanú výpočtovú kapacitu gridových zdrojov, na distribuované spracovanie a uskladňovanie dát. Budú sa vykonávať Monte-Carlo simulácie, výpočty sledovania lúčov, alebo modelovania dynamiky častíc.

3.3 Glucosyltransférase a BIOINFOGRID

Tieto projekty sú z oblasti bioinformatiky. V rámci Glucosyltransférase sa plánuje analýza génov a proteínov pomocou gridu. V spolupráci s EGEE bude možné vybudovať prostriedky pre rýchlu analýzu genetických informácií veľkých rozmerov.

BIOINFOGRID navrhuje prepojiť existujúce služby a aplikácie pre používateľov z oblasti molekulárnej biológie s gridovou infraštruktúrou EGEE. Plánujú sa aplikácie z oblasti genomiky, transkriptomiky, proteinomiky a molekulárnej dynamiky. Projekt sa uchádza o členstvo vo VO Biomed.

4 Záver

Tento dokument bol zameraný na gridovú infraštruktúru EGEE z hľadiska aplikačných domén. Boli popísané jednotlivé virtuálne organizácie, ktoré projekt EGEE podporuje a vysvetlené postupy registrácie používateľov, resp. navrhovanie nových aplikačných oblastí.

Výskumní pracovníci, ktorí majú záujem využívať distribuované výpočtové a pamäťové prostriedky gridu, by mali získať prehľad o prebiehajúcich a taktiež o plánovaných aktivitách projektu EGEE. To by malo pomôcť pri začleňovaní sa do niektorej VO a pri používaní alebo vývoji gridových aplikacií.

Poďakovanie

Článok vznikol s podporou projektov EGEE – Enabling Grids for E-sciencE – (2004-2006) INFSO-RI-508833, VEGA No. 2/6103/6, VEGA No. 2/3132/23.

Literatúra a odkazy

[1] *http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/* ,
VO manažér: *project-lcg-vo-atlas-admin@cern.ch*

[2] *http://cms-project-ccs.web.cern.ch/cms-project-ccs/* ,
VO manažér: *project-lcg-vo-cms-admin@cern.ch*

[3] *http://aliweb.cern.ch/offline/* ,
VO manažér: *project-lcg-vo-alice-admin@cern.ch*

[4] *http://lhcb-comp.web.cern.ch/lhcb-comp/* ,
VO manažér: *project-lcg-vo-lhcb-admin@cern.ch*

[5] *http://egee-na4.ct.infn.it/biomed/* ,
VO manažér: *legre@clermont.in2p3.fr*

[6] *https://cic.in2p3.fr/index.php?id=vo* ,
VO manažér: Osvaldo Gervasi, *osvaldo@unipg.it*

[7] *https://cic.in2p3.fr/index.php?id=vo* ,
VO manažér: Thomas Dominique, *dthomas@cgg.com*

[8] *https://cic.in2p3.fr/index.php?id=vo* ,
VO manažér: Som de Cerff Wim, *sdecerff@knmi.nl*

[9] *https://cic.in2p3.fr/index.php?id=vo* ,
VO manažér: Kornmayer Harald, *harald.kornmayer@iwr.fzk.de*

[10] *https://cic.in2p3.fr/index.php?id=vo* ,
VO manažér: Taffoni Giuliano, *taffoni@ts.astro.it*

IEPSAS-Kosice: experiences in running LCG site

Marian Babik[1], Dusan Bruncko[2], Tomas Daranyi[1], Ladislav Hluchy[1] and Pavol Strizenec[2] *

[1] Department of Parallel and Distributed Computing, Institute of Informatics, Slovak Academy of Sciences
[2] Department of High Energy Physics, Institute of Experimental Physics, Slovak Academy of Sciences
Marian.Babik@saske.sk, Dusan.Bruncko@saske.sk, Tomas.Daranyi@saske.sk, Ladislav.Hluchy@savba.sk, Pavol.Strizenec@saske.sk

Abstract. The article presents lessons learned in running and maintenance of the cluster IEPSAS-Kosice. We will highlight the history of the high performance computing at the Institute of Experimental Physics (IEP) and provide a brief overview of the high energy physics experiments, where IEP participates. Further, we will present architecture, configuration and network topology of the cluster and provide details about our day-to-day operations. The process of the integration into the EGEE/LCG2 will be presented, as well as statistics about the past and present usage. We will also briefly describe the supported virtual organizations and the role they have played in the forming of the requirements on the cluster. We will conclude with our plans for the future.

1 Introduction

Institute of Experimental Physics (IEP) [2] interests in Grid development are derived primarily from the computing needs in the Large Hadron Collider (LHC) era, where IEP concentrates on the two main LHC experiments Atlas, Alice [4, 8, 9]. Undoubtedly, many petabytes will be generated by the experiments and it will require a significant amount of computational and data storage capacity to analyze the data and generate the Monte-Carlo samples needed for interpretation. These needs will have to be fulfilled and the solution will probably require a close collaboration among the existing regional centers. Use of the Grid based technologies will greatly facilitate implementation of such distributed model and will enable more efficient use of the resources by the various experiments.

Experimental activities of the IEP are connected with the participation in the H1 experiment at HERA [7], DESY [6], in heavy ion programme at CERN and in the CERN Large Hadron Collider (LHC) programme - experiments Atlas and Alice [5, 8, 9]. IEPSAS-Kosice computing facility contains over 30M SI2k [18]

* Acknowledgments: The research reported in this paper has been partially financed by the EU within the project IST-2004-511385 Enabling Grids for E-Science in Europe (EGEE) and Slovak national project, VEGA No. 2/3132/23: "Effective tools and mechanism for scalable virtual organizations"

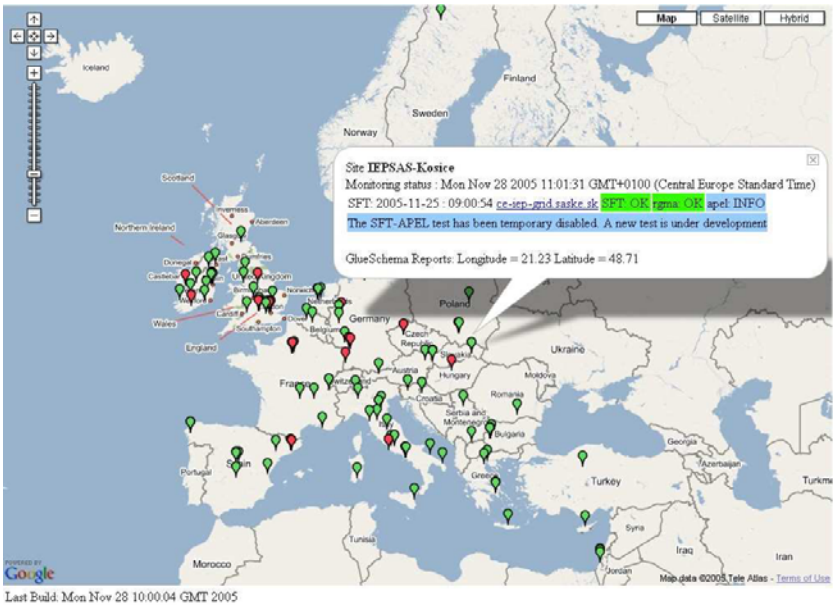


Fig. 1. LCG-2 Infrastructure

of raw computational power and about 2 TBytes of the disk space. It is operated by the computational group at the IEP and is used currently by the experiments Atlas and H1. Integration into the Alice grid computing (Alien [9]) is in its final stages. The group closely collaborates with other Slovak computing facilities i.e., FMPhI-Bratislava [12], TU-Kosice [11] and IISAS-Bratislava [13]. In the following we will describe the LHC Computing Grid in which IEPSAS-Kosice participates, the topology and configuration of the cluster and the supported virtual organizations. We will also present the current cluster resources as well as the statistics about the past and present usage.

2 LHC Computing Grid

The Large Hadron Collider (LHC), currently being built at CERN near Geneva, is the largest scientific instrument on the planet [4]. When it begins operations in 2007, it will produce roughly 15 Petabytes (15 million Gigabytes) of data annually, which thousands of scientists around the world will access and analyze. The mission of the LHC Computing Project (LCG) is to build and maintain a data storage and analysis infrastructure for the entire high energy physics community that will use the LHC [3].

The data from the LHC experiments will be distributed around the globe, according to a four-tiered model. A primary backup will be recorded on tape at CERN, the "Tier-0" center of LCG. After initial processing, this data will

be distributed to a series of Tier-1 centers, large computer centers with sufficient storage capacity for a large fraction of the data, and with round-the-clock support for the Grid.

The Tier-1 centers will make data available to Tier-2 centers, each consisting of one or several collaborating computing facilities, which can store sufficient data and provide adequate computing power for specific analysis tasks. Individual scientists will access these facilities through Tier-3 computing resources, which can consist of local clusters in a University Department or even individual PCs, and which may be allocated to LCG on a regular basis. IEPSAS-Kosice is currently registered as a Tier-3 center and can be seen in the context of the European LCG google map in Fig. 1 [17].

LCG is the primary production environment for the EGEE project. The Enabling Grids for E-sciencE (EGEE) project [15] is funded by the European Commission and aims to build on recent advances in grid technology and develop a service grid infrastructure which is available to scientists 24 hours-a-day.The project aims to provide researchers in academia and industry with access to major computing resources, independent of their geographic location. The EGEE project will also focus on attracting a wide range of new users to the Grid.

The project will primarily concentrate on three core areas:

- The first area is to build a consistent, robust and secure Grid network that will attract additional computing resources.
- The second area is to continuously improve and maintain the middleware in order to deliver a reliable service to users.
- The third area is to attract new users from industry as well as science and ensure they receive the high standard of training and support they need.

3 IEPSAS-Kosice

The history of the IEPSAS-Kosice is dated back to the year 2000, when initial experiments with PC clusters were made at the IEP. They were mostly based on the cooperation with the CDF collaboration at Fermilab [16]. Since then, the cluster has gained in both computational power and available storage.

The topology of the cluster can be seen in Fig. 2. The four main components of the topology are computing element (CE), storage element (SE), user interface (UI) and worker nodes (WN). Computing element (CE) hosts the basic monitoring and job scheduling software based on the well known Globus toolkit. It is supported by the MAUI batch scheduler, which schedules jobs to be run on the Worker Nodes (WN). The storage element's main responsibility is to provide scalable and effective way of storing files. Currently it is based on the classical LCG storage element, but plans are made to upgrade it to the dcache repository. The user interface serves as a gateway for the users to enter the LCG grid as well as the local resource at the IEPSAS-Kosice. Furthermore, there are additional elements in the topology including the main gateway, vobox and installation server. The main gateway serves also as the transparent firewall and host proxy, filtering the farms traffic. The vobox element is a newly added computer, which

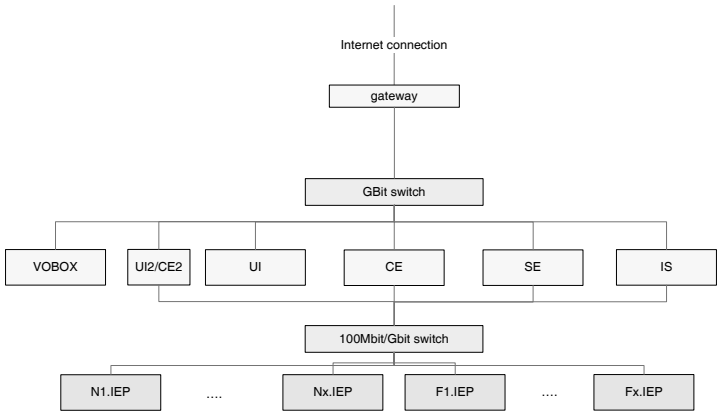


Fig. 2. Topology of the IEPSAS-Kosice cluster

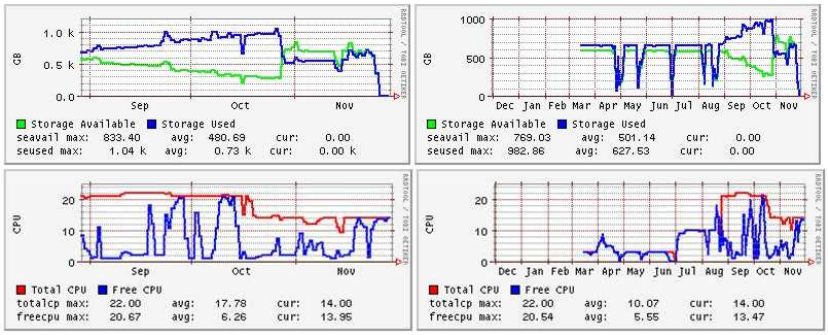


Fig. 3. Computational and storage resources of the IEPSAS-Kosice cluster.

goal is to support virtual organization's specific software requirements, such as Alice's Alien system. The installation server (IS) is running the Quattor system, which provided necessary support for the installation and maintenance of the worker nodes. The overall computational and storage resources and their stability over the last year can be seen in Fig. 3.

4 Virtual Organizations

4.1 H1

H1 is an international collaboration performing fundamental research in the field of High Energy Physics also known as Elementary Particle Physics [10].The H1 collaboration has built and operates the H1 detector, one of the big experiments

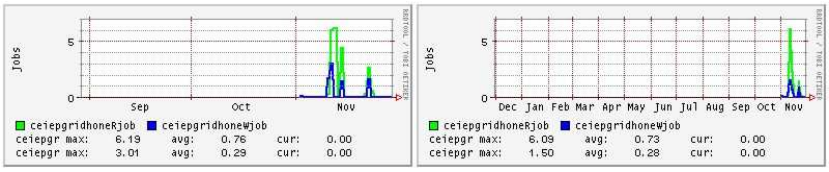


Fig. 4. Number of jobs submitted to the IEPSAS-Kosice from H1 experiment.

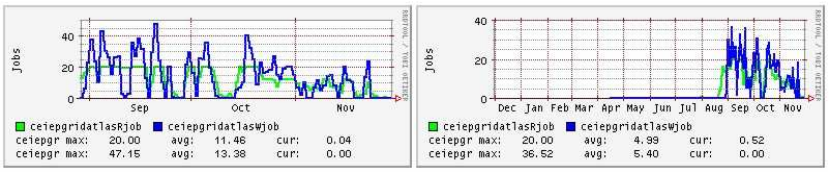


Fig. 5. Number of jobs submitted to the IEPSAS-Kosice from Atlas experiment.

taking data with colliding positron-proton beams at HERA. HERA is one of the facilities at the German national laboratory for elementary particle physics DESY (Deutsches Elektronen-Synchrotron) in Hamburg [7, 6]. The H1 Detector is a very complex apparatus designed to detect particles which are created when high energy electrons and protons collide. Technologies used to build such a detector range from high precision mechanical engineering to modern electronics and up to date computing. The main interest of research of the H1 collaboration is to measure the structure of the proton, to study the fundamental interactions between particles, and to search for physics beyond the Standard Model of the elementary particles. IEPSAS-Kosice is one of the several sites participating in the Monte Carlo simulations for the H1 detector. The overall statistics of submitted jobs and production performance can be seen in Fig. 4.

4.2 Atlas

The ATLAS calorimeters are designed to measure the energy of the products of the proton collisions [8]. The collision creates many elementary particles, and most of them reach the calorimeters, where its energy should be measured. The ATLAS calorimeter is non-compensating, it means, that its response to the different kind of particles with the same energy is different (two basic set of particles are distinguished - those interacting electromagnetically, and those interacting strongly). Therefore some software compensation algorithms should be applied, to compute the proper particles energy out of signals coming from calorimeters. This procedure is called calorimeter calibration.

To find such algorithms and their parameters a very detailed simulations of the calorimeters response should be fulfilled. One of those simulations tasks was running in IEP grid farm, the simulation of so called di-jet events. Di-jet events

are such proton-proton collisions, which produce the two jets of particles, with the opposite momenta. Such events are very often in the LHC collider, so it's knowledge is a basis for the calorimeter calibration.

The description of the original hard proton-proton scattering was obtained from Monte-Carlo generator elsewhere, and the files with produced particles description was transferred to the IEP. Here a simulation of the calorimeters response to these particles was fulfilled within the ATLAS software framework and using the simulation engine GEANT4, widely used in the HEP for the simulation of particle interactions with the matter.

The simulation is performed in three steps (simulation, digitization, reconstruction), each one covered by a separate job. The simulated events was divided to different sets, according jets transversal momenta, each sets contains 20 000 collisions. So far 3 such sets were fully computed, the fourth is running now. The simulation jobs are running 50 events each, so for each set the 400 jobs were done with the CPU time from 16.5h to 23.5h per job (depending on momenta). The digitization and reconstruction jobs are running 250 events each, so together 160 jobs per set was runned, with CPU time app. 1h each. The output data files amounts app. 150GB per set. The job statistics for the Atlas computations at the IEPSAS-Kosice can be seen in Fig. 5.

References

1. Globus Toolkit, http://www-unix.globus.org/toolkit/
2. Institute of Experimental Physics, http://www.saske.sk/Uef/index.php?lang=en
3. LHC Computing Grid, http://lcg.web.cern.ch/LCG/
4. Large Hadron Collider, http://lhc.web.cern.ch/lhc/
5. European Organization for Nuclear Research, www.cern.ch
6. Deutches Elektronen Synchrotron in der Heimholtz-Gemeinshaft, www.desy.de
7. Hadron Electron Ring Accelerator, http://giroz.desy.de/bluebook/ch26/
8. The Atlas Experiment, http://atlas.web.cern.ch/Atlas/index.html
9. The Alice Experiment, http://aliceinfo.cern.ch/index.html
10. H1 Collaboration at DESY, http://www-h1.desy.de/
11. Cluster of the Technical Univeristy of Kosice, http://goc.grid.sinica.edu.tw/gstat/TU-Kosice/
12. Cluster of the Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, http://goc.grid.sinica.edu.tw/gstat/FMPhI-UNIBA/
13. Cluster of the Institute of Informatics, Slovak Academy of Sciences, Bratislava, http://goc.grid.sinica.edu.tw/gstat/IISAS-Bratislava/
14. Cluster of the Institute of Experimental Physics, Slovak Academy of Sciences, Kosice, http://goc.grid.sinica.edu.tw/gstat/IEPSAS-Kosice/
15. Enabling Grids in E-Science in Europe, http://public.eu-egee.org/
16. The Collider Detector at Fermilab, http://www-cdf.fnal.gov/
17. Google map of the LCG infrastructure, http://goc03.grid-support.ac.uk/googlemaps/lcg2.html
18. The Standard Performance Evaluation Corporation (SPEC), http://www.spec.org/

Author Index