The Migrating Desktop – the General Entry Point to the Grid

Mirosław Kupczyk, Rafal Lichwała, Norbert Meyer, Bartek Palak, Marcin Płóciennik, Maciej Stroiński, Paweł Wolniewicz

Poznan Supercomputing and Networking Center,
ul. Z. Noskowskiego 12/14, Poznan, Poland
{miron, syriusz, meyer, palak, marcinp, stroins, pawelw}@man.poznan.pl

Introduction

In this work, we present a current status of development and integration of Migrating Desktop and corresponding grid tools. Migrating Desktop (MD) is an advanced GUI, which allows working with grid resources, it hides the complexity of the grid middleware and frees the user from knowing where the resources physically reside. This facility is not only GUI for well-known Globus commands, it offers much more: flexible personalised working environment, scalability and portability, set of tools, single sign-on mechanism, Roaming Access Server (RAS) interfaces, support for multiple grid infrastructure [1, 2]. The more detailed outline is described here.

The Migrating Desktop overview

A number of Grid middleware projects are currently working on user interfaces for interaction with grid applications, however due to the dynamic and complex nature of the Grid, efficient job submission and job management is still difficult to ordinary scientists. Through its user- and application-centric approach, the Migrating Desktop brings the new quality to High Performance Computing (HPC) user interfaces. Our solution is a complete, production-deployed software environment with special focus on interactive grid applications. These applications are simultaneously compute- as well as data-intensive and are characterized by the interaction with a person in a processing loop. This research was done under the EU CrossGrid project (IST-2001-32243). The homepage of the Migrating Desktop encloses the tool, and user guide [3].

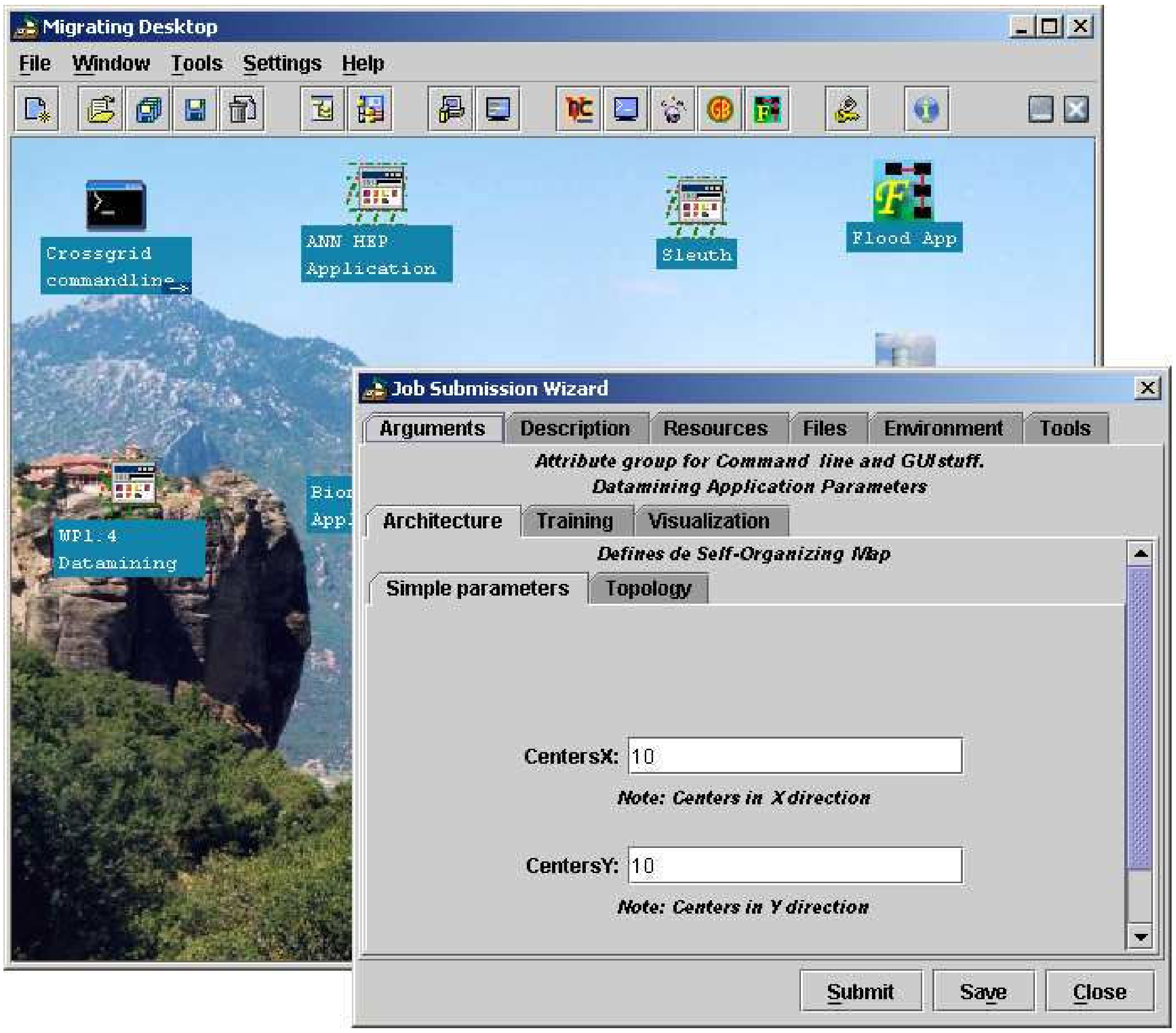
Migrating Desktop is just a front end to Remote Access Server, which intermediates between different grid middleware and applications. The RAS offers a well-defined set of web-services that can be used as an interface for accessing HPC systems and services (based on various technologies) in a common, standardized way. All communication bases on web services technology. This architecture provides an important future direction with respect to the general acceptance of services and protocols.

General concept of MD was to provide the containers - frameworks for plug-ins written by application developers. Such approach allows increasing functionality in an easy way without need of architecture changes. It is possible to add various tools, applications and support visualisation of different formats using that mechanism. As an example: MD on demand loads from network appropriate plug-in for visualisation of non-standard graphics format file. That makes our product independent of specialized tools designed only for specific application. MD supports different kinds of accessing remote/HPC applications.

The Migrating Desktop as a support for developers of the grid applications

We have focused mainly on supporting frameworks for grid applications, unifying different interfaces into one common solution. MD calls appropriate web services on the RAS. The server job submission interface gives uniform access to different resource brokers. Clients can submit all jobs with this interface and then appropriate plug-ins are called to convert job description to the specific language. Currently we are using plug-in for CrossGrid/DataGrid resource broker to convert job description to JDL and Progress plug-in to convert job description to XRSL. Grid applications available for users are grouped in a user-friendly way in a Job Wizard in the MD. This Wizard simplifies the process of specifying parameters and limits, suggesting user defaults or recently used parameters. The Wizard is responsible for proper preparation of the user's job and consists of several panels. One panel is application specific plug-in and the rest can be used to set job information, resource requirements, files and environment variables. Application plug-in asks user for application parameters and these parameters are passed to the command line later on.

For application developers that do not like to prepare plug-in we created generic XML plug-in that interprets argument description given in XML to a graphical form. Example of plug-ins is presented in Fig. 1. MD also allows running remote Java tools. In contrast to grid applications that are submitted to the resource broker, tools are Java applets downloaded from different network localisations on demand. Some applications require a graphical visualisation of job results. In our approach visualisators are tools and they also base on the Tool Plug-in.



Attribute group for Command line and GUI stuff.
Datamining Application Parameters

Defines de Self-Organizing Map

CentersX: 10
Note: Centers in X direction

CentersY: 10
Note: Centers in Y direction

Fig. 1 Migrating Desktop Window with Job Submission Wizard.

Architecture overview

The architectures of the entire CrossGrid project is presented in appropriate documents [8]. For the purpose of this paper we show only basic components and interfaces between corresponding modules that are used for supporting job processing (Fig. 2).

System components:

Migrating Desktop framework allows the user to access the Grid resources, run interactive applications, monitoring and visualization and manage data files. The MD provides a front-end framework for embedding some of the application mechanisms and interfaces, and allows the user virtual access to Grid resources from other computational nodes.

The Roaming Access Server is a set of web services that mainly provide grid functionality. It is the integration level for interfaces of different middleware. It provides services like job submission service, job monitoring service, interactive session manager and interactive job channels forwarding service.

Scheduling Agents and Resource Broker (Workload Manager Service, WMS), components are directly dependent on the DataGrid project. WMS services extend the basic functionalities of the DataGrid software on the field of parallel and interactive jobs.

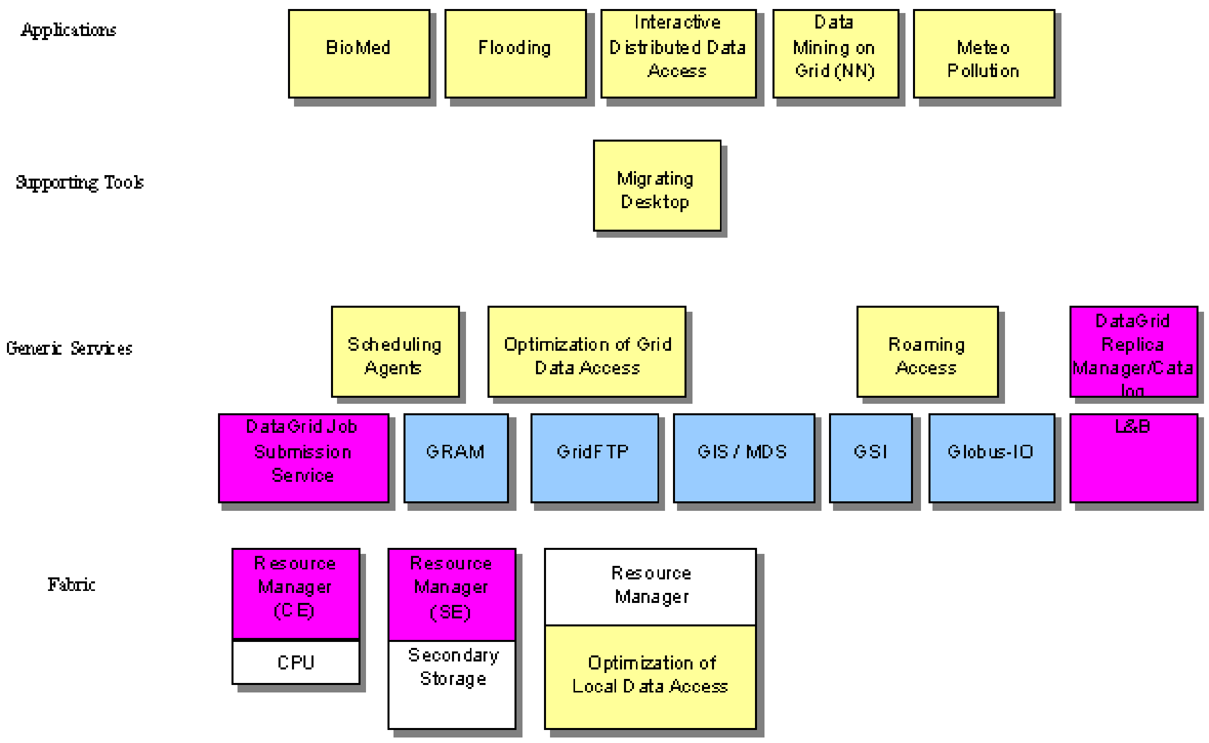


Fig. 2 CrossGrid layered architecture. The yellow blocks indicate CrossGrid components, the blue blocks are Globus components, and the purple components from the DataGrid project.

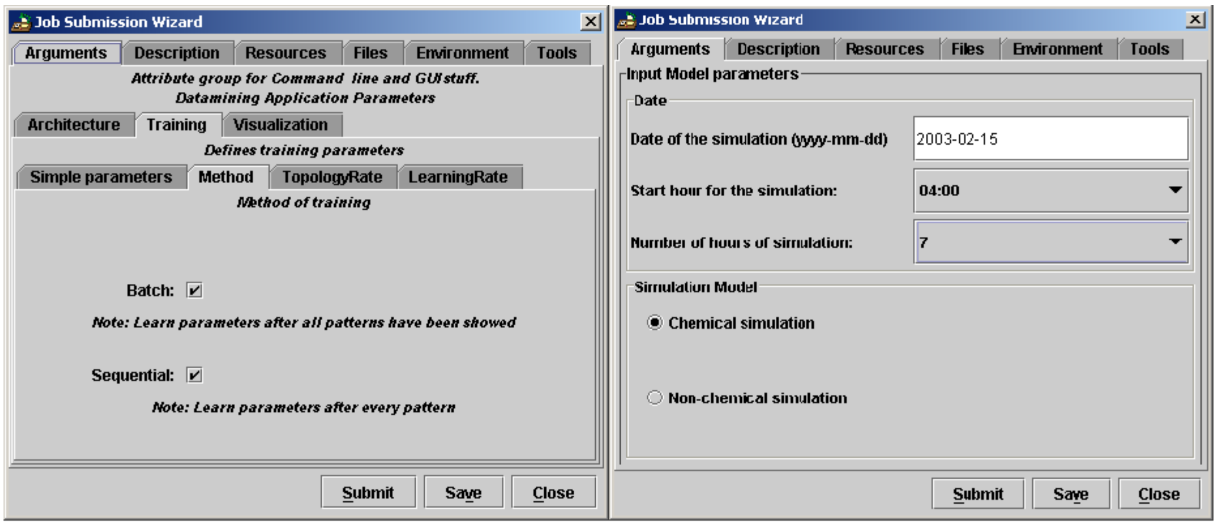
Computing Element (CE) is the combination of a Gatekeeper with its Worker Nodes (WN). Gatekeeper the system that provides the gateway through which jobs are submitted to local farm nodes. The gatekeeper is the interface through which grid enabled systems can use the local resources. Worker Node is a local farm computing node where jobs are actually executed. Jobs received by the Gatekeeper are sent to the WNs through the local scheduling system. These components are taken from Datagrid Project

Logging and Bookkeeping (L&B) is responsible to store and manage logging and bookkeeping information generated by the various components of Workload Management System (WMS): bookkeeping information: refers currently active jobs, i.e. jobs within the WMS, logging information: concerns the WMS itself. That components is taken from Datagrid Project

Interfaces to grid applications
General concept of the Migrating Desktop was to provide the containers – frameworks for plug-ins written by application developers. Such approach allows increasing functionality in an easy way without need of architecture changes. It is possible to add various tools and applications and support visualisation of different formats using that mechanism. As an example: Migrating Desktop on demand loads from network appropriate plug-in for visualisation of non-standard graphics format file. That makes our product independent of specialised tools designed only for specific application.

The Migrating Desktop supports different kinds of accessing remote/HPC applications. We have focused mainly on supporting frameworks for grid applications, unifying different interfaces into one common solution. The Migrating Desktop calls appropriate web services on the RAS. The server job submission interface gives uniform access to different resource brokers. Clients can submit all jobs with this interface and then appropriate plug-ins are called to convert job description to the specific language. Currently we are using plug-in for CrossGrid/DataGrid resource broker to convert job description to JDL and Progress plug-in to convert job description to XRSL. In the future Job submission interface will support XML job description, which is under development by the Job Submission Language group from the Global Grid Forum [5].

Grid applications available for users are grouped in a user-friendly way in a Job Wizard in the MD. This Wizard simplifies the process of specifying parameters and limits, suggesting user defaults or recently used parameters. The Wizard is responsible for proper preparation of the user's job and consists of several panels. One panel is application specific plug-in and the rest can be used to set job information, resource requirements, files and environment variables. Application plug-in asks user for application parameters and that parameters are than passed to the command line. For application developers that do not like to prepare plug-in we created generic XML plug-in that interprets argument description given in XML to a graphical form. Example plug-ins are presented in Fig. 3.


Job Submission Wizard

Arguments | Description | Resources | Files | Environment | Tools

Attribute group for Command line and GUI stuff.
Datamining Application Parameters

Architecture | Training | Visualization

Defines training parameters

Simple parameters | Method | TopologyRate | LearningRate

Method of training

Batch: ✓

Note: Learn parameters after all patterns have been showed

Sequential: ✓

Note: Learn parameters after every pattern

Submit | Save | Close

Job Submission Wizard

Arguments | Description | Resources | Files | Environment | Tools

Input Model parameters

Date

Date of the simulation (yyyy-mm-dd): 2003-02-15

Start hour for the simulation: 04:00

Number of hours of simulation: 7

Simulation Model

● Chemical simulation

○ Non-chemical simulation

Submit | Save | Close

Fig. 3 Example Application plug-ins. Left – generated from XML, right – Java class.

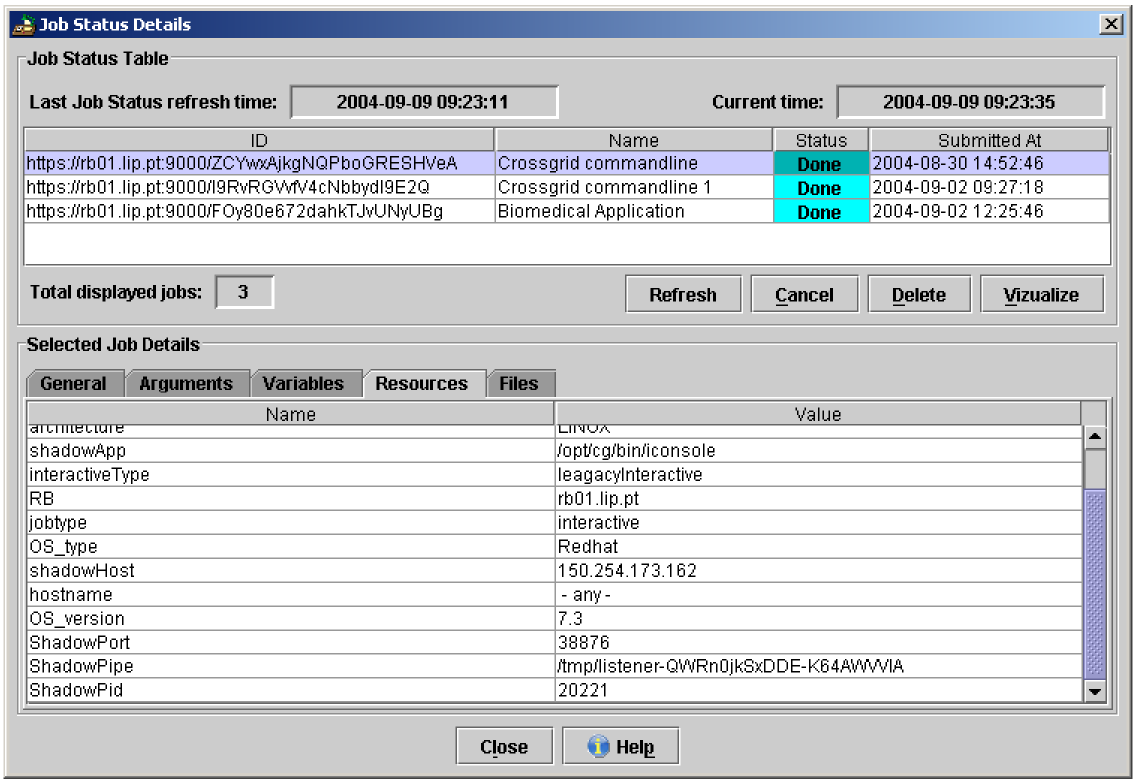
The Migrating Desktop also allows running remote Java tools. In contrast to grid applications that are submitted to the resource broker, tools are Java applets downloaded from different network localisations on demand.

Some applications require a graphical visualisation of job results. In our approach visualisators are just tools and can also base on the Tool Plug-in. Tool Plug-ins can visualise a single file or a single job. All available file and jobs visualisators are registered in the Migrating Desktop database, and the appropriate Tool Plug-in is chosen depending on the file extension or type of the job.

Application state monitoring

Monitoring can be performed on different moments/levels of the application life. We assumed to present all monitoring types supported by application providers. At the moment we focus mainly on the grid jobs state monitoring framework.

The user can check the state of the launched application/submitted job, suspend or delete the selected job using the Job Monitoring tool in the MD. The Job Monitoring dialog presented on Fig. 4 is a useful tool for tracing the status of the previously submitted jobs. This dialog contains all the information about the submitted job including an extended job status and job log. Some information like the extended status can have different formats, because they are returned by different grid information systems. The RAS plug-ins read information from a specific grid information system and presents them to users in a text or XML format.


Job Status Details

Job Status Table

Last Job Status refresh time: 2004-09-09 09:23:11 Current time: 2004-09-09 09:23:35

ID	Name	Status	Submitted At
https://rb01.lip.pt:9000/ZCYwxAjkgNQPboGRESHVeA	Crossgrid commandline	Done	2004-08-30 14:52:46
https://rb01.lip.pt:9000/I9RvRGWfV4cNbbydI9E2Q	Crossgrid commandline 1	Done	2004-09-02 09:27:18
https://rb01.lip.pt:9000/FOy80e672dahkTJvUNyUBg	Biomedical Application	Done	2004-09-02 12:25:46

Total displayed jobs: 3

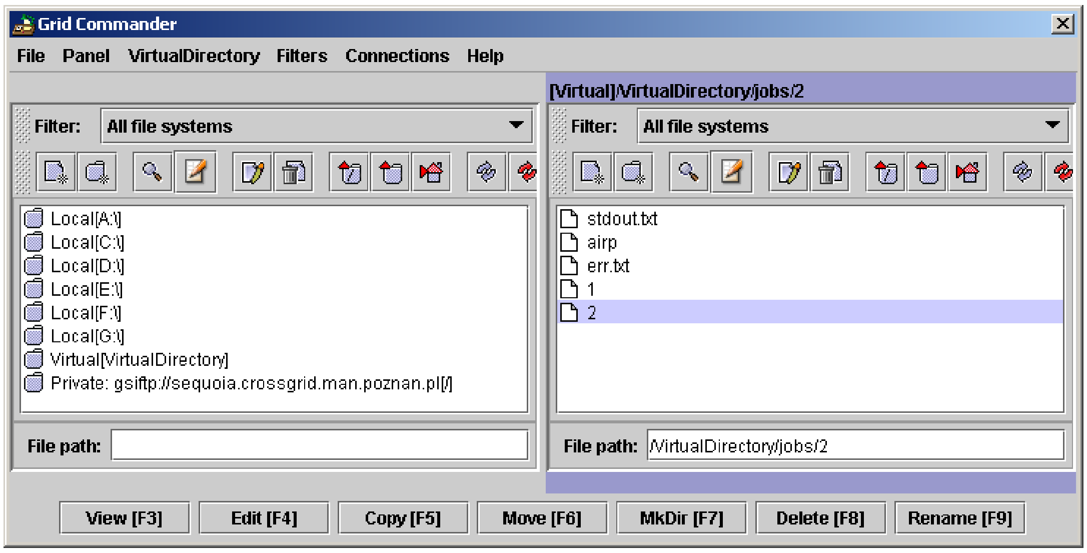
Selected Job Details

Name	Value
architecture	LINUX
shadowApp	/opt/cg/bin/iconsole
interactiveType	leagacyInteractive
RB	rb01.lip.pt
jobtype	interactive
OS_type	Redhat
shadowHost	150.254.173.162
hostname	- any -
OS_version	7.3
ShadowPort	38876
ShadowPipe	/tmp/listener-QWRn0jkSxDDE-K64AWVIA
ShadowPid	20221

Fig. 4 Job Monitoring Dialog.

Data management

Data management is a very important and complicated part in every system. After many analyses a common set of interfaces that allow operation on data and metadata was created. The designed framework allows extending MD/RAS infrastructure easily so that many management systems could be attached as a plug-in. The main file Management tool in the Migrating Desktop is called GridCommander (Fig. 5). The Grid Commander is a two-panel application similar to the Commander family tools. A single panel can represent a local directory, gridftp or ftp directory or other protocol to the native storage. Each protocol is defined as a plug-in that makes file management in MD easy to extend to other protocols. From the technical points of view graphical elements operate on generic file systems that invoke appropriate functions within the supported plug-in.


Grid Commander

[Virtual]/VirtualDirectory/jobs/2

Filter: All file systems

Filter: All file systems

Local[A:\]
Local[C:\]
Local[D:\]
Local[E:\]
Local[F:\]
Local[G:\]
Virtual[VirtualDirectory]
Private: gsiftp://sequoia.crossgrid.man.poznan.pl[/]

stdout.txt
airp
err.txt
1
2

File path:

File path: /VirtualDirectory/jobs/2

Fig. 5 Grid Commander window.

In order to support Grid data-intensive applications and to give easy and intuitive access to grid data we have created the User Virtual Directory that is an abstract filesystem that contains information about all user files independently of their physical location. Each branch in the Virtual Directory tree can be physically placed on different location or even can be placed on storage with different way of accessing (e.g. ftp, gridftp or any other project native protocol). Virtual Directory was designed to standardise data access, and to create a user-friendly, uniform view of Grid and local files.

Used technology

Migrating Desktop base on Java applet technology. It can be launched using Java Webstart Technology or using web browser with appropriate Java Plugin included in JRE. There are three layers of MD: graphical layer, grid and web services communication. We are basing on Swing libraries for designing graphical user interface, Java CoG Kit [4] version 1.1a with some extensions is being used as interface to Globus functionality and Axis ver. 1.1 web services client for communication with Roaming Access Server. We use Java CoG Kit for operation on proxy and GridFTP/FTP.[6][7]

RAS server is based on newest version of AXIS webservices version 1.1. Axis is third generation of APACHE SOAP, open-source software that implements a framework for constructing SOAP processors like client, server, gateways. This SOAP engine with some extensions is also being used in Globus Toolkit Project. We decided to use Axis to have possibility for easy migration to OGSA when it becomes stable standard. As an server for webservices we use open-source servlet container, Apache Tomcat version 4.1.

References

1. M. Kupczyk, R. Lichwala, N. Meyer, B. Palak, M. Plociennik, P. Wolniewicz, „Mobile Work Environment for Grid Users", Across Grids 2003, LNCS 2970, pp. 132-138, 2004
2. http://ras.man.poznan.pl/crossgrid/md1.avi
3. http://wp3.crossgrid.org
4. G. von Laszewskia, I. Foster, J. Gawor, and P. Lane, .A java commodity grid kit,. Concurrency and computation: Practice and Experience, vol. 13, no. 8-9, pp. 643.662, 2001, http://www.cogkits.org/.
5. Sujoy Basu, Vanish Talwar, Bikash Agarwalla1, Raj Kumar Interactive Grid Architecture for Application Service Providers – Proceedings of International Conference on Web Services (ICWS '03) 23-26 June 2003, Las Vegas, Nevada
6. Apache Axis Project http://ws.apache.org/axis/
7. Apache Jakarta Project: Apache Tomcat http://jakarta.apache.org/tomcat/index.html
8. Bubak, M., Malawski, M., Zajac, K., "Current Status of the CrossGrid Architecture", Proceedings of the Cracow '02 Grid Workshop, December 11-14, 2002, Cracow, Poland