
ZeliGrid: an architecture to deploy applications with dynamic non-functional aspects on computational grids

Rodrigo Souza Granja e Alexandre Sztajnberg

DICC/IME e PEL/FEN - UERJ
Rio de Janeiro, RJ, Brazil
{rodrigoss, alexszt}@ime.uerj.br

1

Motivation

- Applications with non-functional requirements in Grid environments
 - Local resources (ex.: CPU, memory)
 - Communication resources (ex.: BW, latency)
- To provide the required dynamic support we need
 - to describe non-functional requirements and policies
 - adaptation and reconfiguration mechanisms
 - monitoring mechanisms
 - management of the policies for resource use

2

Goals

- Develop the support for applications with non-functional requirements in a grid environment
 - Based on the QoS contracts from the CR-RIO architecture
- Integrate largely used tools
 - Globus: security, remote job execution and management
 - NWS: resource monitoring
 - LDAP: directory service
- Develop the additional modules
 - Contract interpretation
 - Contract management
 - Reconfiguration mechanisms

Globus Toolkit

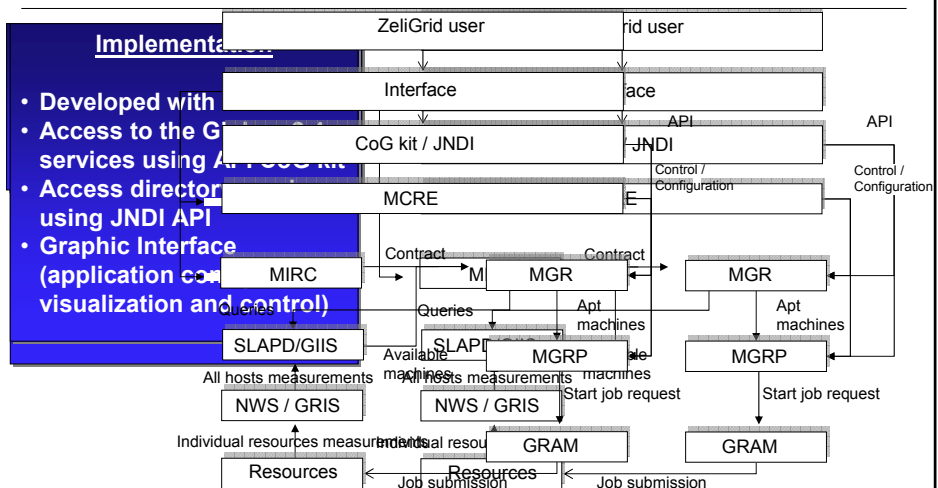
- Middleware – Grid basic services (v 2.4)
 - MDS – Directory service
 - GRIS and GIIS
 - GRAM – Remote job execution and management
 - GSI – Security and authentication
 - Java CoG Kit – Java API
 - JNDI – Java Directory access API

Network Weather Service

- Resource monitoring and forecasting distributed system
- Active and static measurement
 - Operating System information (*uptime* and *vmstat*)
 - Probes
- CPU, latency, bandwidth
- Information available through LDAP
 - SLDAP server

5

Proposed architecture



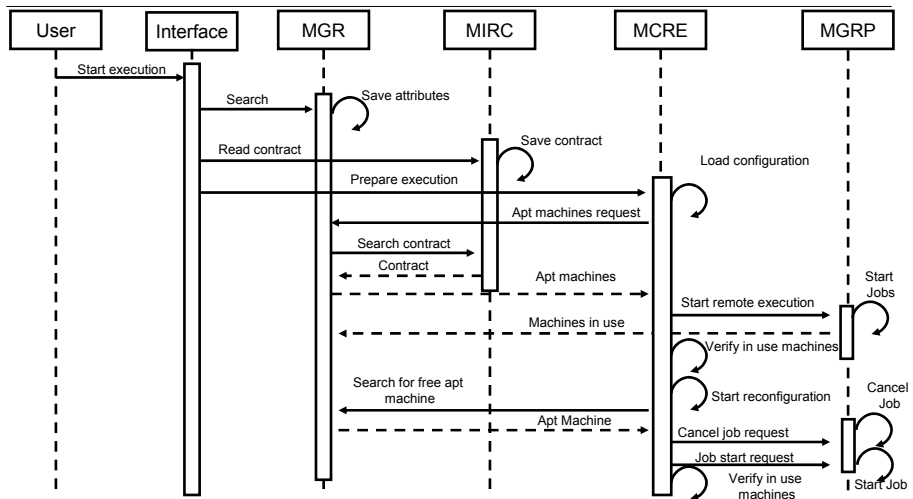
6

Contract Example

```
01 [profile1]
02   osName=Linux
03   bandwidth>78
04   cpuAvail>0.6
05   cpuSpeed>2000
06   freeMemory>200
07 [profile2]
08   osName=Linux
09   bandwidth>50
10   cpuAvail>0.5
11   cpuSpeed>1800
12   freeMemory>100
```

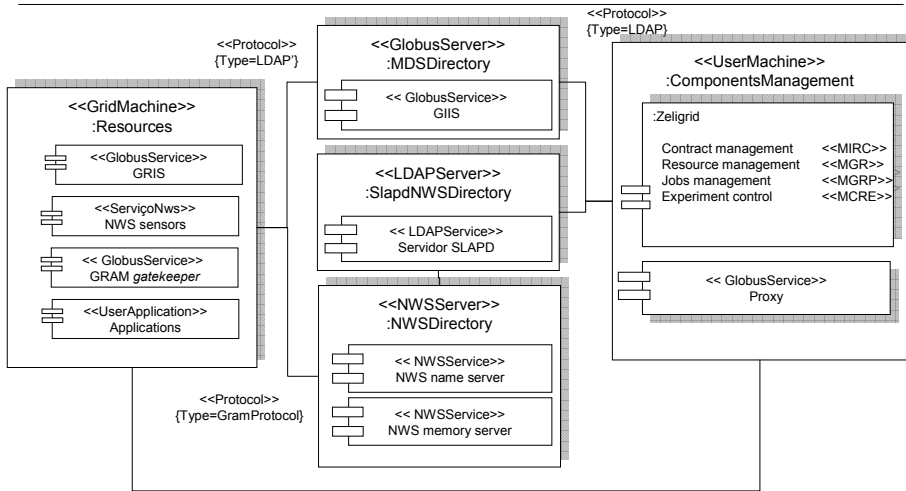
7

Sequence diagram



8

Deployment



9

Resources

- Nodes that execute the application jobs
- Runs:
 - GRIS instance
 - Gather static information (OS variables)
 - NWS Sensors
 - Active probe measurements
 - CPU availability
 - Bandwidth between each slave and the master
 - Gram Gatekeeper

10

Information Servers

- May be deployed in distinct machines
- MDS directory (GIIS)
 - Concentrate and deliver GRIS measurements
- NWS SLAPD directory
 - Deliver NWS measurements trough LDAP
- NWS servers
 - Memory server
 - Name server

11

User machine

- Controls the architecture
- Where ZeliGrid is deployed
 - MIRC
 - MGR
 - MCRE
 - MGRP
- Must have an authenticated proxy
 - Access Globus secure services

12

Main window

The screenshot shows the main window of the software. It features a menu bar at the top, a toolbar with icons, and a status bar at the bottom. The main area contains several panels: a Messages panel on the left, a central Information trees panel, and a Log messages panel at the bottom. Red arrows point to these components with labels: Menu Bar, Tool Bar, Remote Jobs, Information trees, Log messages, and Status Bar. A box on the left lists the Messages panel contents: Status Bar, Log messages, Date and time, Profile, Stdout, and StdIn.

Messages

- Status Bar
- Log messages
- Date and time
- Profile
- Stdout
- StdIn

Labels in the screenshot:

- Menu Bar
- Tool Bar
- Remote Jobs
- Information trees
- Log messages
- Status Bar

13

Configuration

The Configuration dialog box is shown with the following fields circled in red:

- NWS Address: `ldap://colossus.ime.uerj.br:2112/o=grid`
- Mds Address: `ldap://colossus.ime.uerj.br:2135/o=grid`
- Mds-Vo-name: `gls`
- Master Host Address: `colossus.ime.uerj.br`
- Master executable: `/usr/lib/java/bin/java`
- Master arguments: `-classpath /jobs MasterJob`
- Nws max. delay (sec.): `300`
- Restart completed: `true`
- Reconfiguration delay(sec.): `30`
- Status check Delay(sec.): `2`

Trabalho em grupo

14

Experiment

- Linux Slackware 10 / Ethernet 10/100 network
- Master-slave software architecture
- Qualitative tests
 - Verify if the mechanisms behaved as expected
 - Measurements made by the log messages analysis
 - External applications created to consume resources (CPU and memory) forcing reconfigurations

15

Results

- Jobs are deployed on available machines that best fits the QoS contract requirements
- During operation the monitoring mechanisms could consistently detect contract violations
- Reconfiguration mechanisms were able to
 - relocate remote jobs to other machines
 - switch to the 2nd profile, according to the specified profiles

16

Future works

- New reconfigurations mechanisms
 - State persistent
- More complex contracts
 - Closer to CR-RIO's semantics
- Use ZeliGrid to deploy other applications

17

Contacts

- Rodrigo Souza Granja
 - rodrigoss@ime.uerj.br
 - <http://www.ime.uerj.br/~rodrigoss>
- Alexandre Sztajnberg
 - alexszt@ime.uerj.br
 - <http://www.ime.uerj.br/~alexszt>
- Apoio
 - PIBIC CNPq e UERJ
 - Projeto CARAVELA (Proj. GIGA / RNP-CPqD)

18

Integration steps

- Implantação do Globus
 - Geração, autenticação e aplicação de certificados
 - Ativação dos serviços e configuração
- Implantação do NWS
 - Integração com o SLAPD
 - Acesso aos dados através do servidor LDAP
- Acesso do Globus ao repositório LDAP
 - JNDI

Implementation

- Developed in Java
 - Architecture configuration interface
 - Remote jobs visualization
 - Resources state monitoring
 - Remote jobs reconfiguration
 - Remote jobs reallocation