Software Virtualization of the Computing Infrastructure to Interface DAMEGRID web application on the S.Co.P.E. Grid

Academic Tutor: Prof. Adriano Peron
Company Tutor: Dr. Massimo Brescia
Candidate: Giovanni Vebber
Matricola: 566/346

Academic Year 2010/2011
DAME (Data Mining & Exploration)

It is a program for the development of Web 2.0 services, dedicated to data mining on massive data sets, based on distributed computing infrastructure and machine learning methods.
HW Hybrid DAME Infrastructure

User & Data Archives (300 TB dedicated)

DM Models Job Execution (300 multi-core processors)

Cloud facilities
16 TB
15 processors
Driver Component Grid Extension

The DAME architecture

Client-server AJAX (Asynchronous Java-Xml) based; interactive web app based on Javascript (GWT-EXT);

Restful, Stateless Web Service experiment data, working flow trigger and supervision Servlets based on XML protocol

FRAMEWORK WEB-SERVICE Suite CTRL

DRIVER FILESYSTEM & HARDWARE I/F Library

HW env virtualization; Storage + Execution LIB Data format conversion

REGISTRY & DATABASE USER & EXPERIMENT INFORMATION

DATA MINING MODELS Model-Functionality LIBRARY RUN

clustering regression

MLP

DMPlugin
DM Functionalities SVM, MLP, PPS...
DM Models SVM, MLP, PPS...
DM Library wrappers JN, SWIG...
DM Libraries libsvm, liblinear...
Low Level Libraries bias, lapack, gpl,...

user

FRONT END WEB-APPL. GUI

CALL

CALL

XML

XML

Stand Alone
GRID
CLOUD
USER INFO
USER SESSIONS
USER EXPERIMENTS

Driver Management System

Framework

Execution
Storage
Data Processing

GRID Infrastructure
Stand Alone Machine
Bachelor Activities

► SCOPE Grid User Interface (UI) installation and configuration on DAME Cloud machine

► DriverGRID Module design and development
  ▪ Grid job execution functionalities (Java environment)
    ► Glite commands handled via system call

► DRMS integration with DriverGRID
  ▪ Hierarchical structure of DRMS in two layers:
    ► First layer: platform selection/scheduling GRID / Stand Alone (SA)
    ► Second layer: platform job handling (DriverGRID / DriverSA)
The new DRMS

**DRMS**
- Execution
- Grid Job cancel
- storing and file conversion

**Driver Manager**
- Platform selection/scheduling
- driverGrid activation
- driverSA activation

**DriverGrid**
- Proxy / MyProxy creation
- job execution/cancel
- job output/status

**DriverStandAlone**
- job execution
- job output/status

**DBMS**
- Registration
- Authentication
- Working session

**Framework**
- Experiment launch
- experiment cancel

**DMPlugins**

**DAMEWARE**
Other components

**Database**

**FileStore**

**Grid-SCOPE System**

**System Call**

**Logical Flux**

**Data flux**

---

**DA ME**

[Image of DA ME logo]
UI Installation

► UI installation through gLite 3.2 INFNGRID
  ▪ gLite 3.2 x86_64
  ▪ Scientific Linux 5.5

► The UI is installed with the packet handler YUM

Command:
  `yum groupinstall ig_UI_noafs`

File Repository
- `glite-ui.repo`: UI node info retrieval
- `lcg-ca.repo`: get Certification Authority list
- `dag.repo`: get added packets of SL5.5
- `ig.repo`: get sw packets of gLite
UI configuration

► UI Node configuration through YAIM_INFNGRID
► YAIM is a simple way to configure a Grid node
► It is distributed through rpm packets

Command:

```
/opt/glite/yaim/bin/ig_yaim -c -s <your-site-info.def> -n <nodetype>
```
Access to the Grid is done by a digital certificate X.509
- Next, people authentication is required
- Proxy certificate generation
  - Avoids iterated periodical authentication (limited to 12/24 hours)
- Myproxy certificate generation
  - Extends account connection (tipically 7 days)
- Subscribe to a virtual organization (unina.it)

Proxy Creation:

```
[gvebber@notredame ~]$ voms-proxy-init --voms unina.it
Enter GRID pass phrase:
Your identity: /C=IT/O=INFN/OU=Personal Certificate/L=Federico II/CN=Giovanni Vebber
Creating temporary proxy .................................................. Done
Contacting voms01.scope.unina.it:15003 [/C=IT/O=INFN/OU=Host/L=Federico II/CN=voms01.scope.unina.it] "unina.it" Done
Creating proxy .............................................................. Done
Your proxy is valid until Thu Jan 13 00:23:57 2011
```
GRID Activities

- Main Grid job activities...
  - launch (file jdl)
  - status
  - Output
  - Cancel

...respective gLite commands:
- `glite-wms-job-submit -a jdl`
- `glite-wms-job-status Id_job`
- `glite-wms-job-output Id_job`
- `glite-wms-job-cancel -a Id_job`
Job status analysis

- One thread for each launched job on the Grid
- Periodical polling on job status (1min)
- The monitoring thread stops when:
  - Done (Success)
  - Done (Failed)
  - Aborted
  - Cancelled
  - Cleared
gLite Java System Call vs API

► Why system calls?
  ▪ Can use all gLite commands available
  ▪ Fast modification of any command
  ▪ More portability between gLite versions

► Why not APIs?
  ▪ Limited handling of gLite commands number
  ▪ gLite 3.2 integration not easy
COMING NEXT...

► Complete integration into DAMEWARE webapp
► Intelligent (self-adaptive) job scheduling system improvement
► Deeper testing sessions

http://dame.dsf.unina.it/
Technical and management info
Documents
Science cases
Newsletters

http://dame.dsf.unina.it/beta_info.html
DAMEWARE Web application Beta Version