



Enabling Grids for E-scienceE

GridICE: a monitoring service for Grid Systems

*Giuseppe Misurelli
INFN-CNAF (Italy)
giuseppe.misurelli <at> cnaf.infn.it*

*First Latin American Workshop for
Grid Administrators
Merida 2005, 23 November*

www.eu-egee.org



Information Society



- **GridICE Server Installation**
 - Brief Introduction
 - System Requirements
 - Core Packages & Dependencies
 - APT Installation
 - Apache Configuration
 - PostgreSQL Optimization
 - The GridICE Configuration Script
 - After The Discovery Process

- **Why Monitoring**
 - A Use Case Perspective
 - VO manager viewpoint
 - Grid Operations viewpoint
 - Site Administrator viewpoint

- **What is Grid Monitoring**
 - Our Definition
 - Concepts & Terminology
 - Requirements
 - The Four Main Phases of Monitoring

- **The GridICE Approach**
 - Generating Events
 - Distributing Events
 - Presenting Events

- **Monitoring a Grid**
 - Challenges for Data Collection
 - Challenges for Data Presentation

- **GridICE@Work**
 - VO manager utilization
 - Grid Operations manager utilization
 - Site Administrator utilization

GridICE Server Installation

GridICE:

- is a distributed monitoring tool for grid systems
- integrates with local monitoring systems
- offers a web interface for publishing monitoring data at the Grid level
- fully integrated in the LCG-2 Middleware
 - gridice-clients data collector installation and configuration for each site realized by the Yaim scripts.

- **Suggested Operating system is Scientific Linux with a minimal installation**
- **The GridICE server should be installed on a performant machine**
 - PostgreSQL service - RAM intensive demand
 - Apache web server - RAM-CPU intensive demand

The GridICE server software is composed by three core packages:

1. gridice-core
(setup and maintenance scripts / discovery components)
2. gridice-www
(web interface scripts and components)
3. gridice-plugins
(monitoring scripts)

Plus several dependencies:

- Apache http web server
- PostgreSQL database server
- Nagios monitoring tool
- ...

- Fully automated process thanks to APT package manager

Add in the `/etc/apt/source.list.d` the GridCE repository:

```
### GridCE APT Repository ###
```

```
rpm http://inf Forge.cna f.infn.it gridce/packages/sl /3.0.3/i386 \
gridce
```

Update your new repository list with the command:

```
[root@localhost]# apt-get update
```

Upgrade your system with the command:

```
[root@localhost]# apt-get upgrade
```

(it takes a while)

Install the GridCE meta-package with the command:

```
[root@localhost]# apt-get install gridce-server
```

1. HTTPD to disable dir indexes and manage the .htaccess related files:

In `/etc/httpd/conf/httpd.conf`

Modify `Options Indexes FollowSymLinks`

with `Option -Indexes FollowSymLinks`

After few lines modify `AllowOverride None`

with `AllowOverride All`

Save and exit

Then create the two symbolic links for the jpgraph and ADODB libraries

In `-s /var/www/jpgraph-<version> /var/www/html/gridice/external/jpgraph`

In `-s /var/www/adodb /var/www/html/gridice/external/adodb`

1. All PostgreSQL databases and configurations files are located in **`/var/lib/pgsql/data`**
If this directory does not exist (or it is empty) then launch:

```
[root@localhost]#su - postgres
[root@localhost]#initdb -D /var/lib/pgsql/data
```

2. For a database performances optimization we suggest to set the following attributes/values in the **`/var/lib/pgsql/data/postgresql.conf`** file as follows:

```
listen_addresses = '*'
max_connections = 256
work_mem = 2048
maintenance_work_mem = 32768
fsync = false
enable_hashjoin = true
enable_indexscan = true
enable_nestloop = true
enable_seqscan = true
enable_tidscan = true
effective_cache_size = 10000
random_page_cost = 2
```

- Choosing your Grid being monitored...

Create a GridICE server configuration file

```
[root@localhost]#cd /opt/gridice/setup/
[root@localhost]#cp gridice-server.cfg.template gridice-server.cfg
```

giisgroup

BDII list to use for the Grid being monitored

Note that

- Every *giisgroup* indicates a *BDII* so that you can have more than one monitored Grid
- For each group you can insert more than one *BDII* for backup pupose in terms of *ldap* queries to the related *BDII*.

Now you can launch the GridICE configuration scripts:

```
[root@localhost]#/opt/gridice/setup/start-conf.pl --cfg \
/opt/gridice/setup/gridice-server.cfg
```

- **GridICE Database creation (plus patches for the new geo view)**

```
[root@local host]#su - postgres
[root@local host]#psql -U gridiceadmin GridICEdb < \
/opt/gridice/setup/pgsql /mondb.sql
```

- **GridICE cron jobs to perform maintenance routines and periodic discovery**

```
[root@local host]#cp /opt/gridice/uti ls/gridice-cronj obs /etc/cron.d
```

- **GridICE discovery script to explore and collect all the monitoring data about your Grid**

(It queries the Information Service of your Grid and inserts into the RDMS all the data retrieved)

```
[root@local host]#/usr/li b/nagi os/dscv/start-dscv.pl
```

- ✓ **Be sure that the following services are running**
 1. nagios
 2. postgresql
 3. httpd (check also if the http port is open)

- ✓ **To see your Grid monitored data, point the web browser to the URL:**
http://<fqdn_of_your_gridice_server>/gridice

Why Monitoring

- **Grid resources availability is subject to failures.**
- **Resources observability is necessary for the Grid utilization.**

Need for analyzing the usage, behavior and performance of a Grid depending on different users:

- 1. *VO manager***
- 2. *Grid operations manager***
- 3. *Site administrator***

- **Visualization of the actual set of resources accessible to its members.**
- **Evaluation of members' demand satisfaction on the Grid mapping functionalities.**
- **Evaluation of the Service Level Agreement (SLA) for the global Grid service offers.**

- **Detection and prediction of fault situations related to wide area distributed resources.**
- **Coordination of the deployment and upgrade of the Grid middleware installed at several sites.**
- **Investigation on Grid resources for statistical purpose.**

- **Detection of fault situations related to the own resources.**
- **Control how the own resources are used and appear to the Grid.**

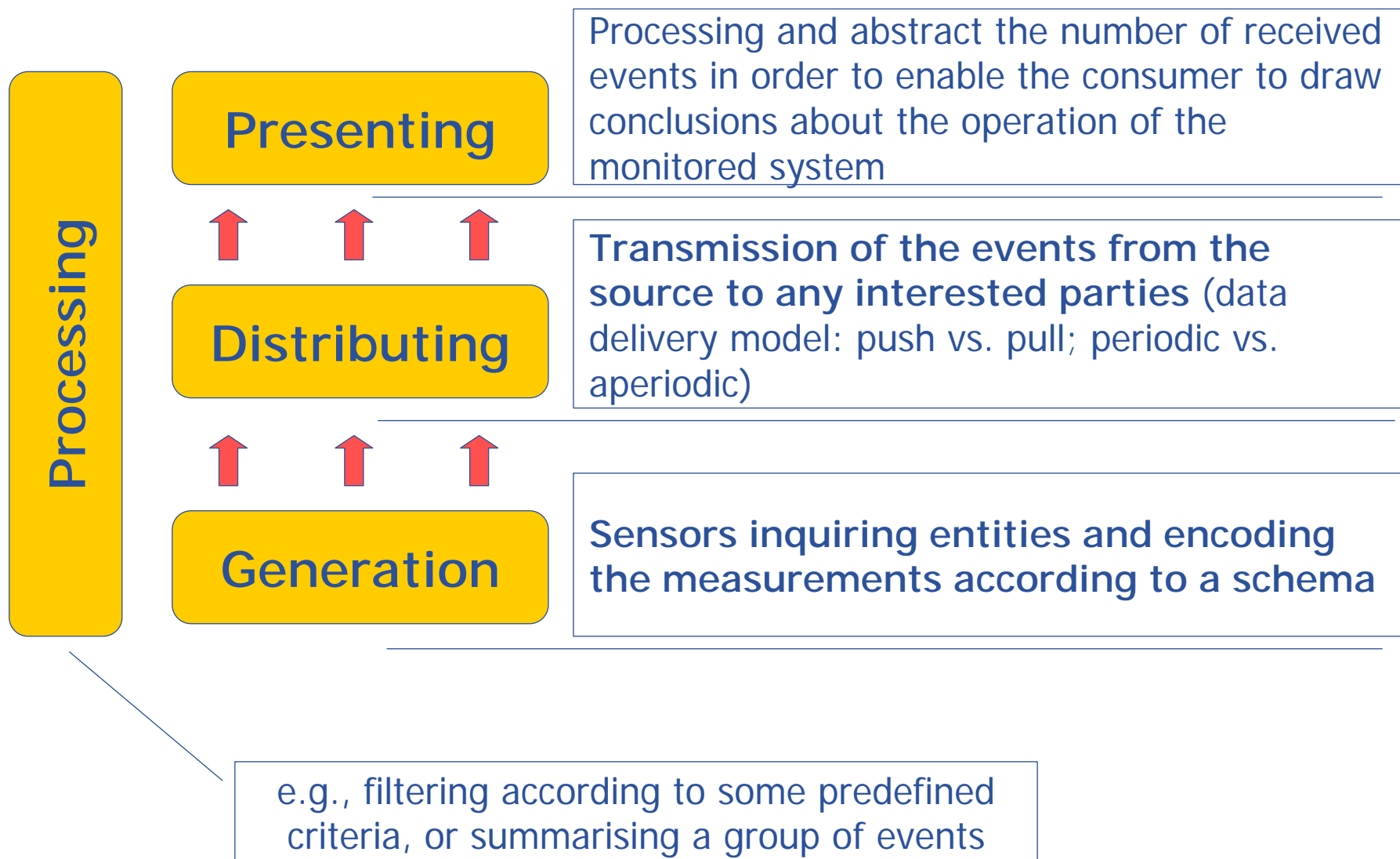
What is Grid Monitoring

- **Grid Monitoring**
 - the activity of **measuring** significant **Grid resources related parameters**
 - in order to
 - **analyze usage, behavior and performance of the grid**
 - **detect and notify fault situations**

- **Entity:** any networked and useful resources having a considerable lifetime (e.g. processors, memories, disk capacity, etc.).
- **Events:** collection of timestamped data, associated with the attribute of an entity.
- **Event Schema (or Schema):** the typed structure and semantics of the all events, so that given an event type, one can find the structure and interpret the semantics of the corresponding event.
- **Sensor:** process monitoring an entity and generating events.

- **Scalability:** monitoring systems have to cope efficiently with a growing number of resources, events and users.
- **Extensibility:** monitoring systems must be extensible with respect to the supported resources.
- **Data delivery models:** monitoring systems must integrate different measurement policies (e.g. periodic, on-demand).
- **Portability:** any encapsulated measurement must be platform independent.
- **Security:** monitoring systems must deal with security concerns such as privacy, data integration and confidentiality.

The Four Main Phases of Monitoring



The GridICE Approach

- **Generation of events:**
 - Sensors: typically perl scripts or c programs.
 - Schema:
 - GLUE Schema v.1.1 + GridICE extension.
 - *System related (e.g., CPU load, CPU Type, Memory size).*
 - *Grid service related (e.g., CE ID, queued jobs).*
 - *Network related (e.g., Packet loss).*
 - *Job usage (e.g., CPU Time, Wall Time).*
 - All sensors are executed in a periodic fashion.

- **Distribution of events:**
 - Hierarchical model.
 - **Intra-site:** by means of the local monitoring service
 - *default choice, LEMON (<http://www.cern.ch/lemon>).*
 - **Inter-site:** by offering data through the Grid Information Service.
 - **Final Consumer:** depending on the client application.
 - Mixed data delivery model.
 - **Intra-site:** depending on the local monitoring service (push for lemon).
 - **Inter-site:** depending on the GIS (current choice, MDS 2.x, pull).
 - **Final consumer:** pull (browser/application), push (publish/subscribe notification service coming on the next release).

- **Data stored in a RDBMS used to build aggregated statistics.**
- **Data retrieved from the RDBMS are encoded in XML files.**
- **XSL to XHTML transformations to publish aggregated data in a Web context.**

Monitoring a Grid

- **The distribution of monitoring data is strongly characterised by significant requirements (e.g., Scalability, Heterogeneity, Security, System Health)**
- **None of the existing tools satisfy all of these requirements**
- **Grid data collection should be customized depending on what are the needs of your Grid users selected**

- **Different Grid users are interested in different subset of Grid data and different aggregation levels**
- **Usability principles should be taken into account to help users finding relevant Grid monitoring information**
- **A sintetic data aggregation is crucial to permit a drill-down navigation (from the general to te detailed) of the Grid data**

GridICE@Work

Mostly interested in:

- **Resources available to the VO**
 - Computing elements where VO users can submit jobs.
 - Storage elements where VO users can store/retrieve data.

- **Job monitoring**
 - How many jobs are running or queued?
 - *For the whole VO? In each site? Submitted by a certain RB?*
 - How many jobs have been executed?
 - *For the whole VO? In each site? Submitted by a certain RB?*

Mostly interested in:

- General status of the managed Grid
 - How many sites compose the managed Grid and where they are located.
 - How many resources (cpu#, WN, etc.) are available.

- Highlighted problems
 - Is there any Grid service (e.g., CE, SE, BDII) which related processes have problems?
 - Is the Grid Information Service working properly?

Mostly interested in:

- Status of their resources
 - What is the cpu load at the moment?
 - What is the percentage of the busy storage space?
 - Are there any jobs running or queued in my site and in which Worker Node?

- Highlighted problems
 - Is there any Grid service (e.g., CE, SE, BDII) which related processes have problems?
 - Is the Grid Information Service working properly?

- **Dissemination Web site:** <http://grid.infn.it/gridice>
- S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone, M. C. Vistoli **GridICE: a monitoring service for Grid systems**, Future Generation Computer System 21 (2005) 559–571
- B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, R. Wolski, **A Grid Monitoring Architecture**, GFD-I.7.
- S. Zaniolas, R. Sakellariou, **A taxonomy of grid monitoring systems**, Future Generation Computer Systems 21 (2005) 163–188.
- M. Franklin, S. Zdonik, **“Data In Your Face”**: Push Technology in Perspective, ACM SIGMOD '98, Seattle, WA, USA.
- S. Andreozzi, A. Ciuffoletti, A. Ghiselli, C. Vistoli. **Monitoring the connectivity of a Grid**. Proceedings of the 2nd International Workshop on Middleware for Grid Computing (MGC 2004) in conjunction with the 5th ACM/IFIP/USENIX International Middleware Conference, Toronto, Canada, October 2004.
- S. Andreozzi, N. De Bortoli, S. Fantinel, G.L. Rubini, G. Tortone. **Design and Implementation of a Notification Model for Grid Monitoring Events**. CHEP04, Interlaken (CH), Sep 2004.