School of Engineering & Design

Electronic & Computer Engineering

MSc in Data Communications Systems

Grid Monitoring

Theofylaktos Papapanagiotou

Dr.Paul Kyberd

DRAFT

March 2011

A Dissertation submitted in partial fulfillment of the

requirements for the degree of Master of Science

School of Engineering & Design

Electronic & Computer Engineering

MSc in Data Communications Systems

Grid Monitoring

Student's name: Theofylaktos Papapanagiotou

Signature of student:

Declaration: I have read and I understand the MSc dissertation guidelines on plagiarism and cheating, and I certify that this submission fully complies with these guidelines.

DRAFT

Abstract

Develop an interface which allows the customisable aggregation and display and information on the performance of a computational grid.

In EGI era of grid computing in Europe, MyEGEE and Nagios are taking the place of SAM in performance monitoring of the grid, in NGI oriented infrastructures. SAM Framework had a significant role in reporting the service availability. That monitoring model was needed to be replaced by a new Multi Level Monitoring architecture. The need of establishing a central point in regional level, where metrics gathered from each information system of the grid, lead to the adoption of MyOSG. MyEGEE is the tool that extends MyOSG to european NGI's and provides an interface to customize the display of aggregated metrics of the performance of NGI sites. Nagios is the main technological choice to provide that regional monitoring system.

DRAFT

Contents

DRAFT

List of Tables

Listings

DRAFT

List of Figures

DRAFT

DRAFT

Chapter 1

Introduction

1.1 Context

Performance monitoring of a grid is a key part in grid computing. Based on the reports of grid performance, decisions on capacity planning are being made. Visualization of performance status in different levels helps scientists and managers focus on the exact point of the infrastructure where a bottleneck on service exists. Current interfaces delivers performance graphs without following the standard topology schema that is presented by the grid information system.

WSRF aggregation framework and GLUE schema are examined to understand the gathering process of metrics. Ganglia's hierarchical delegation to create manageable monitoring domains is an important aspect. MyEGI which is based in Django python framework, access easily the unified metrics database. Performance in the aspect of how many jobs are served by each site is not examined in this project. Whether it is possible to integrate Ganglia performance graphs in MyEGI and Nagios interfaces, using the standard Information System of ATP.

Build a lab to gather performance data and start working on the development of the integration parts. It is assumed that the environment is a grid site, that already have the components needed to work together. Ganglia daemons on each node, presented by the GLUE schema on site BDII, Nagios/MyEGI monitoring frameworks. A web interface is available to present the work of the integration of Ganglia into Nagios/MyEGI.

DRAFT

1.2 Aims & Objectives

Different role users are going to use a portal to get information about the performance status of the grid, to export the appropriate report for their job. This project aims to develop these particular pieces of code to support the aggregation of the metrics from Nagios, to allow the web based customization of the visualization of the reports. These metrics are needed to report the availability and reliability of NGIs and particular sites of the grid.

The procedures that are going to be used in order to achieve the above aims should include at the beginning some opening and exploration of the environment where the interface is going to be placed. The usage of grid computing in the world should be well known, so a visibility of the importance and the possible uses of the software will be recognized. The appropriate access to the infrastructure should be gained, on different platforms and levels. Brunel University site and GridPP/NGS VO at the beginning, as long as the UKI ROC operations may be a good point of collaboration with researchers to reach the bests possible requirements and data to analyze. The middleware used in both these VOs should be examined so with the knowledge of running projects and global usage of them may target to export better specifications. Existing operations on the grid should also be discovered. The European initiative milestones on the operations of the regional level should be considered as a route, and registration to news about the upcoming research projects that are going to use the grid should also be take place.

After that wide-opening to get the whole picture, a targeted and focused view should follow. Existing monitoring tools must be used to check the problems and search for requirements. The experience of SAM, Gridview, Gridmap, Gstat, GridICE, etc should be taken in order to merge their functionality as possible as it is. Information systems that already reside over the infrastructure, must also be learned. Standards and specifications should be examined, on how the message bus works and delivers the data in an hierarchical manner. A contact with the CERN team working on MyEGEE and Indiana University's MyOSG team should be established, to collaborate on the core of MyOSG source. Changes submission to subversion system as long as ticket closure of the development project tool will help to get to know the core of MyEGEE and Nagios. It is possible to create and upstream a Nagios customized web interface, to create different views of Nagios resources scheme to grid topology oriented architecture. Nagios, NRPE and Ganglia installations should be deployed across the CE&SE nodes of Brunel's sites

DRAFT

to have a working production environment to work on. Attention should be taken on the potential performance impact of these sensors deployment. UKI MyEGEE validation/testing portal will be used as a pre-production environment to check changes. PNP should be fixed in GridPP Nagios to be evaluated. Statistical access log analysis of existing tools may have results on trends of users/admins preferred views.

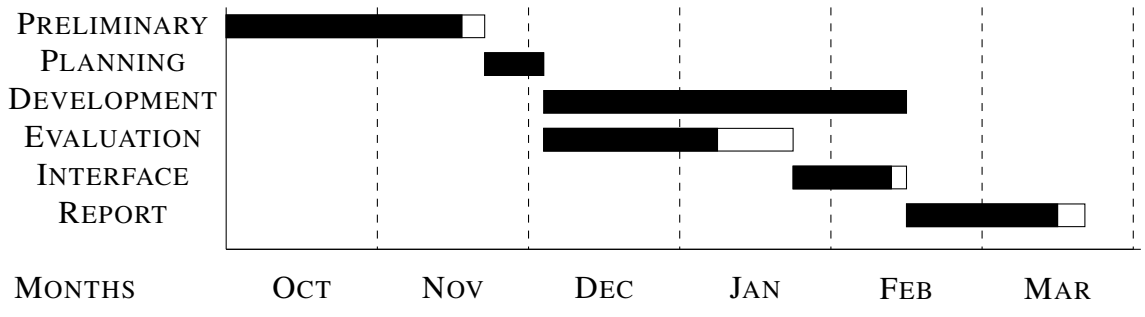
Various tools are going to be used to track changes and collaborate. Monitoring articles in GridPP wiki & CERN twiki should be made. Snippets upstream & status changes must be a regular operation in SVN/JIRA/Trac in CERN interfaces. Ongoing task through the dissertation project is the reading of papers and methodical updates of Mendeley citation management tool to have the bibliography organized. Possible changes suggestions to MSc on DCS course notes about grid monitoring may by made, as long as the EGI roadmap updates. Finally with the appropriate supervision and follow-up of meetings and presentations, a paper publishing might take place.

1.3 Organization

1.3.1 Tools

This project was developed in LaTeXusing Vi editor. Its releases may be found in Google Code, where Mercurial was used for source control. Citation management through Mendeley software. Papers obtained by becoming member of IEEE, ACM and USENIX. Operating Systems Laboratory of Technological Education Institute of Piraeus was used to build a testbed of grid site and tools to study existing monitoring tools.

1.3.2 Time-plan (Gantt Chart)



DRAFT

Task	Start date	End date	Duration in days
Preliminary	09/29/10	10/24/10	20
- Identify Concepts	09/29/10	10/08/10	8
- Gain Access	10/08/10	10/24/10	12
Planning	11/12/10	12/04/10	17
- Explore existing technologies	11/12/10	11/28/10	12
- Write Interim Report	11/28/10	12/04/10	5
Experimental-Development	12/04/10	02/14/11	51
- Evaluate performance monitoring tools	12/04/10	12/25/10	15
- Information/topology databases	12/17/10	12/29/10	8
- Develop Customized Interface	12/29/10	02/14/11	34
— Coding of information aggregation	12/29/10	01/21/11	16
— Development of the frontend	01/21/11	02/10/11	14
— Complete the interface (auth, scale, etc)	02/10/11	02/14/11	4
Report	02/16/11	03/29/11	32
- Begin Writing	02/17/11	03/01/11	11
- Submit Draft & Make Changes	03/01/11	03/14/11	9
- Prepare Final	03/14/11	03/29/11	11

Table 1.1: Key activities necessary to complete the project

DRAFT

Chapter 2

Literature Review

2.1 Grid Computing

Grid computing [1] is the most recent decade's technology innovation in high performance computing. A large number of scientists working on the operations of this huge co-operative project of EU. Monitoring & information architecture [2] has been standardized in the initial state of that project, to succeed in today scale of 150.000 cores in production. Use of grid computing nowadays takes place in academic and research environments. Also, applications in industry-based needs such as promising Power Grid control [3] are emerging.

Grid computing may be the infrastructure over which Cloud Computing may reside. Cloud computing promise that it will change how services are developed, deployed and managed. The elastic demands of education and research community is a good place where cloud computing may be developed. Many datacenters all over Europe which are currently serving grid computing infrastructure for LHC, could later share the resources to help some other big academic projects scale up as needed.

2.2 Resource Brokers

Resource Brokers [4] where developed to manage the workload on Computer elements and Resource elements. Globus is a non-service based RB, and gLite RB which is service based. A Workload Management System (WMS) exists in gLite to do the distribution and management

DRAFT

of the Computing and Storage oriented tasks.

Based on the middleware that resource brokers rely on, they use the equivalent information system. From resource broker's point of view, the relevant information is the data store and query. There are two main categories of information systems in middlewares. The Directory-based and the Service-based. They are used for resource mapping by the brokers when they access the resource data.

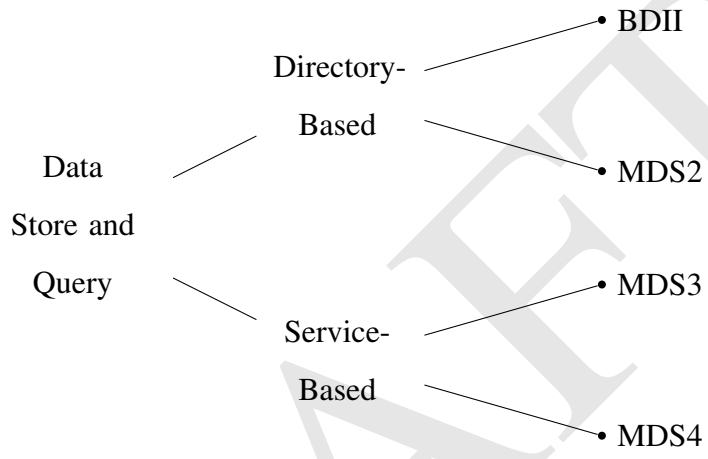


Figure 2.1: Grid Resource Brokers grouped by Information Systems[4]

2.2.1 Globus

Globus Toolkit is an open source toolkit used to build grids. It provides standards such as OGSA, OGSI, WSRF and GSI, and the implementations of OGF protocols such as MDS and GRAM.

Monitoring and Discovery Service (MDS) is part of Globus Toolkit, and provides the information for the availability and status of grid resources. As a suite of Web Services, it offers a set of components that help to the discovery and monitoring of the resources that are available to a Virtual Organization.

DRAFT

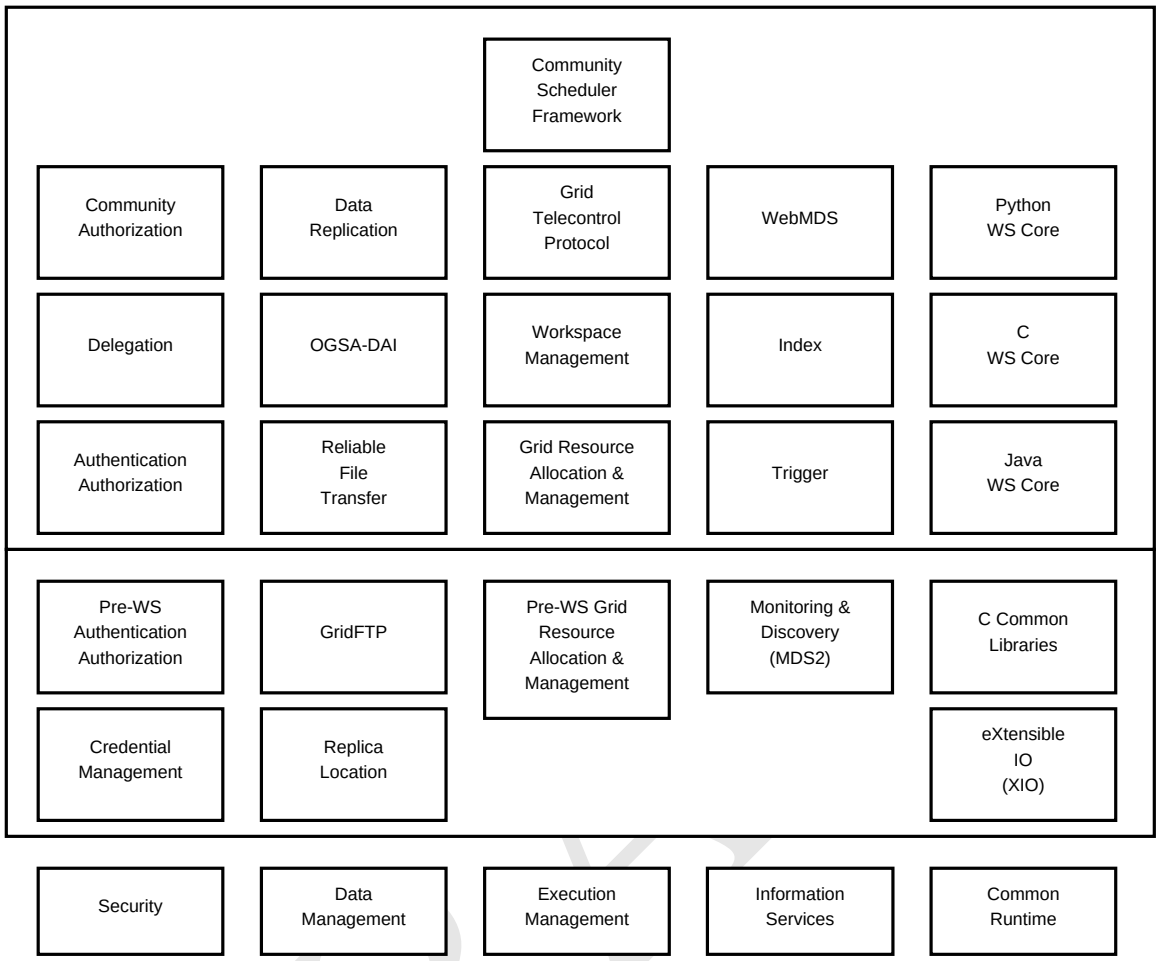


Figure 2.2: Globus Toolkit version 4 (GT4)

2.2.2 gLite

gLite is a middleware which was created to be used in the operation of the experiment LHC in CERN. The user community is grouped in Virtual Organizations, and the security model is GSI. A grid using gLite consists of User Interface, Computer Element, Storage Element, Workload Management System and Information Service.

The information service in version 3.1 of gLite is similar to MDS of Globus middleware, except that the GRIS and GIIS are provided by BDII (see Section BDII) which is an LDAP based service.

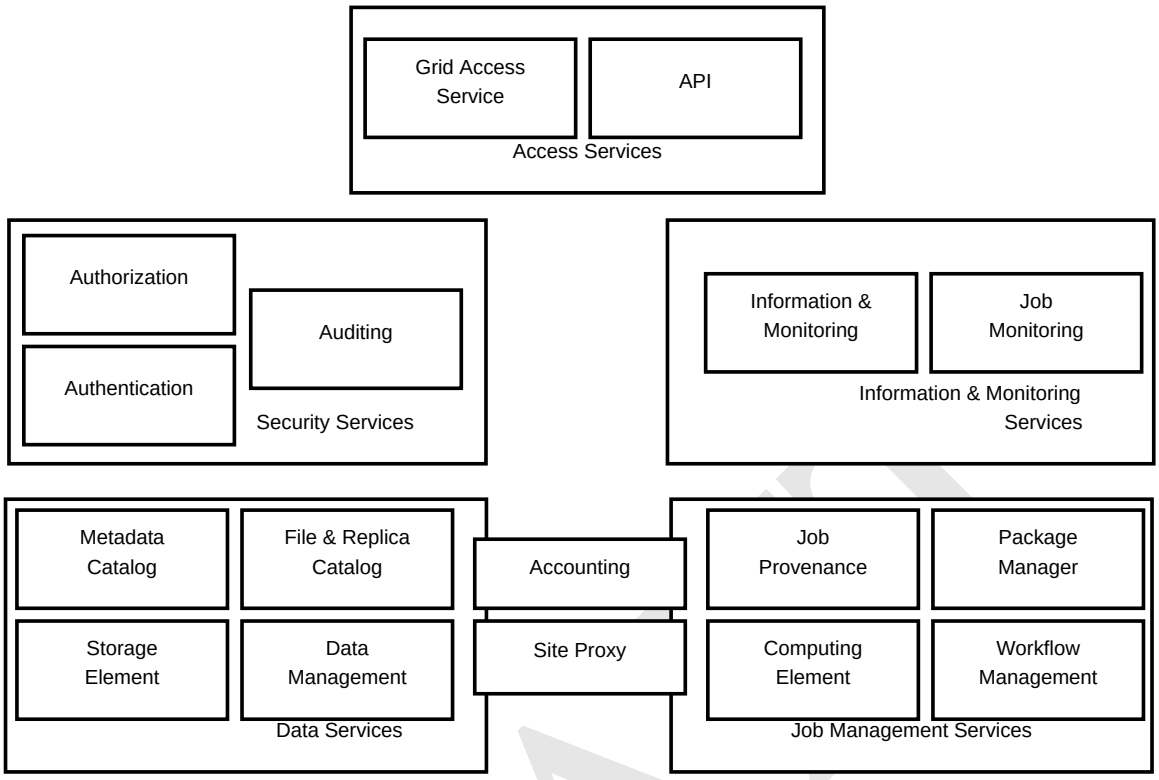


Figure 2.3: gLite architecture

2.3 Information Services

A Grid Monitoring Architecture [5] was proposed in early 2000's. Information systems were developed to create repositories of information needed to be stored for monitoring and statistical reporting reasons. Such an organized system later was specified by the Aggregated Topology Provider (ATP) definition. The largest world grids adopt that model, forming OIM in OSG (USA) and GOCDB as that information base in EGEE (Europe). Message Bus was also defined as a mean to transfer the underlying data, and well known tools came up such as Gstat, GOCDB and BDII with Glue specification. Grid performance monitoring and keeping of such an information system has also impact in the performance of the system it shelf [6], so various methods were developed to give the solution to the scaling and performance problem, such as MDS2 (GIIS & GRIS), GMA and R-GMA [7], which offers relational environment [8], has experience on production systems [9] and scales to reach huge needs such as CMS project [10, 11].

2.3.1 MDS

Monitoring and Discovery Services is about collecting, distributing, indexing and archiving information of the status of resources, services and configurations. The collected information is used to detect new services and resources, or to monitor the state of a system.

Globus Toolkit was using LDAP-based implementation for its information system since its early versions, back in 1998 [12]. MDS2 in Globus Toolkit fully implemented referral with a combined GRIS and GIIS, using mds-vo-name=local to refer to the GRIS and all other strings to refer to a GIIS. It was widely accepted as a standard implementation of a grid information system [13], with good scalability and performance [14].

MDS 4 consists of the Web Services Resource Framework and a web service data browser, WebMDS. The WSRF Aggregator Framework includes:

1. MDS-Index, which provides a collection of services monitoring information and an interface to query such information.
2. MDS-Trigger, which provides a mechanism to take action on collected information.
3. MDS-Archive, is planned for future release of MDS, to provide access to archived data of monitoring information.

External software components that are used to collect information (such as Ganglia)[15] are called Information Providers.

2.3.2 Glue

As long as Information Services are used to connect different infrastructures, the schema of its structure had to be standardized. To inter-operate EU and USA grids, DataTAG developed the GLUE schema implementation. GLUE specification quickly adopted by the communities and currently its recommended LDAP DIT is specified in GLUE specification v.2.0 from GLUE Working Group of OSG.

Many objectclasses of the Glue schema define a Computer Element, a Storage Element, etc. As seen in Figure 3.2 in later chapter, performance monitoring attributes such as processor load are defined in objectclasses that extends Computer Element objectclass.

DRAFT

2.3.3 BDII

BDII is used by gLite as the Information Index Service of the LHC experiment. It is LDAP based and may be at top-level or site-level. The GIIS has been replaced by site BDII, which is fundamental for a site in order to be visible in the grid.

Top-level BDII contains aggregated information about the sites and the services they provide. Site BDII collects the information from its Computer Elements, Storage Elements, etc as long as every configured service that is installed on the site.

Information about the status of a service and its parameters is pushed on BDII using external processes. An information provider is also used (such as in WSRF) to describe the service attributes using the GLUE schema.

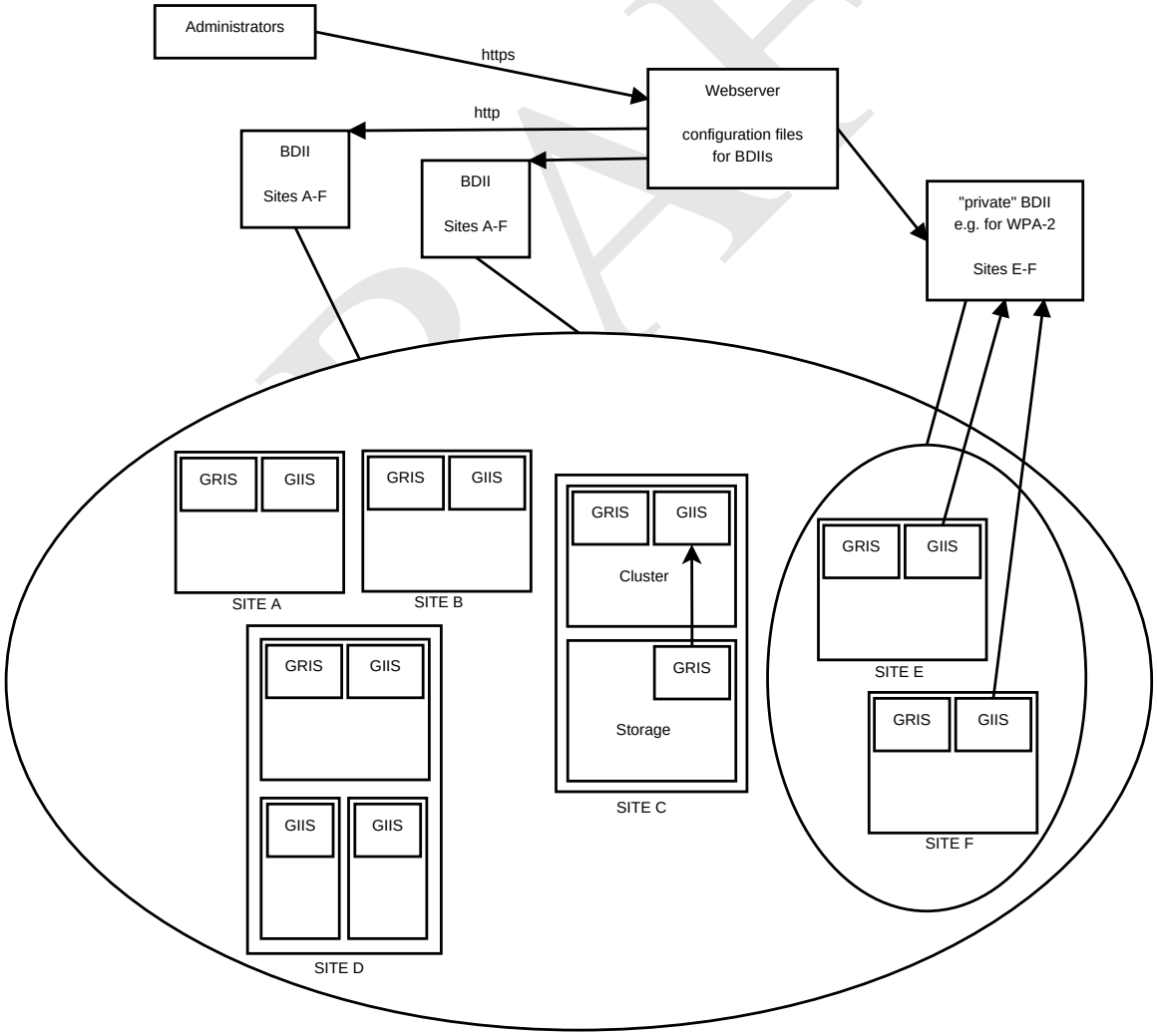


Figure 2.4: Berkeley Database Information Index

2.4 Performance Monitoring

Standards are being published about the operational models that the grid computing initiative will use. Last decade the EGEE I, II & III was adopted by european universities to fund and establish a collaborative community of researchers under a central point, the CERN oriented research project in Particle Physics. After EGEE, the European Grid Initiative were formed to lead to the explode of that community into regional initiatives. Performance and availability monitoring tools and views also follow that format, phasing out commonly used SAM [16] and having the adoption of Nagios as the monitoring of regional performance tool.

A taxonomy effort has been made [17] to present the differences of performance monitoring systems of the grid, and later a more general [18] taxonomy paper was published to give a more general visibility of these tools. GridICE was generally used to aggregate the performance metrics of ROCs in high level reports [19]. Later GridICE was left as long as the SAM left, to meet the milestone of EGI to have a regional monitoring tool (Nagios) to report the reliability of the joined sites and report the values for SLA reasons.

Grid performance can be also measured using benchmark tools in different levels of the grid architecture, using the micro-benchmarks at the Worker Node level, the Site (CE) level and the Grid VO level. Various benchmarks exist in these levels, using different libraries and algorithms, such as This project focuses on mathematically compute of the performance of a grid based on the metrics that are taken at the Worker Node level.

Different metrics and benchmarks exist, such as the measurement of the performance of CPUs in **MIPS using EPWhetstone** and the evaluation of the performance of a CPU in **FLOP/s and MB/s using BlasBench**. GridBench [20] provides a framework to collect those metrics using its own description language, **GBDL**.

GcpSensor [21] introduce a new performance metric called WMFLOPS. It uses PAPI [22] (Performance API) to access the hardware performance counters. For data distribution it uses MDS information system which provides dynamic metrics for CPU load average, one for 1, for 5 and for 15 minutes load.

DRAFT

2.4.1 Ganglia

Ganglia is a monitoring tool which provides a complete real time monitoring environment. It is used by both academia and industry to monitor large installations of clusters, grids. Any number of host metrics may be monitored in real time using the monitoring core, a multithreaded daemon called Gmond. It runs on every host that is in scope of monitoring. Its four main responsibilities are:

1. Monitor the changes that happen in the host state
2. Multicast over the network, the changes that has been made
3. Listen to network for changes that other ganglia nodes are multicasting and
4. Answer the status of the whole cluster to specific requests, using XML.

All the data that are gathered of the multicast channel are written to a hash table in memory. The metric data of each node that runs gmond and sends information over the multicast channel are been processed and saved. Data sent over the multicast channel is happening using external data representation (XDR). When there is a request over a TCP connection, the response is in XML.

2.5 European Grid Infrastructure

Latest EGI directive to form regional operation tools pushed the use of Nagios [23] as the main tools of availability & performance (an so reliability) monitoring of the grid. Each NGI/ROC (regional level) has its own interface, and hierarchically there is a Super Nagios interface to report the top level view of general system availability. Nagios offers extensions such as NRPE to remotely invoke check commands in inaccessible/private installations. Another important add-on to Nagios is the NdoUtils, which offers an SQL store of history data to the monitoring interface. Nagios Configuration Generator was introduced to help the automatically generation of the configuration based on the information system of nodes and services.

Finally, there has been proposed an integration of SAM views to a Nagios customized interface, to offer the last good known SAM interface to the old users. Nagios also integrates

DRAFT

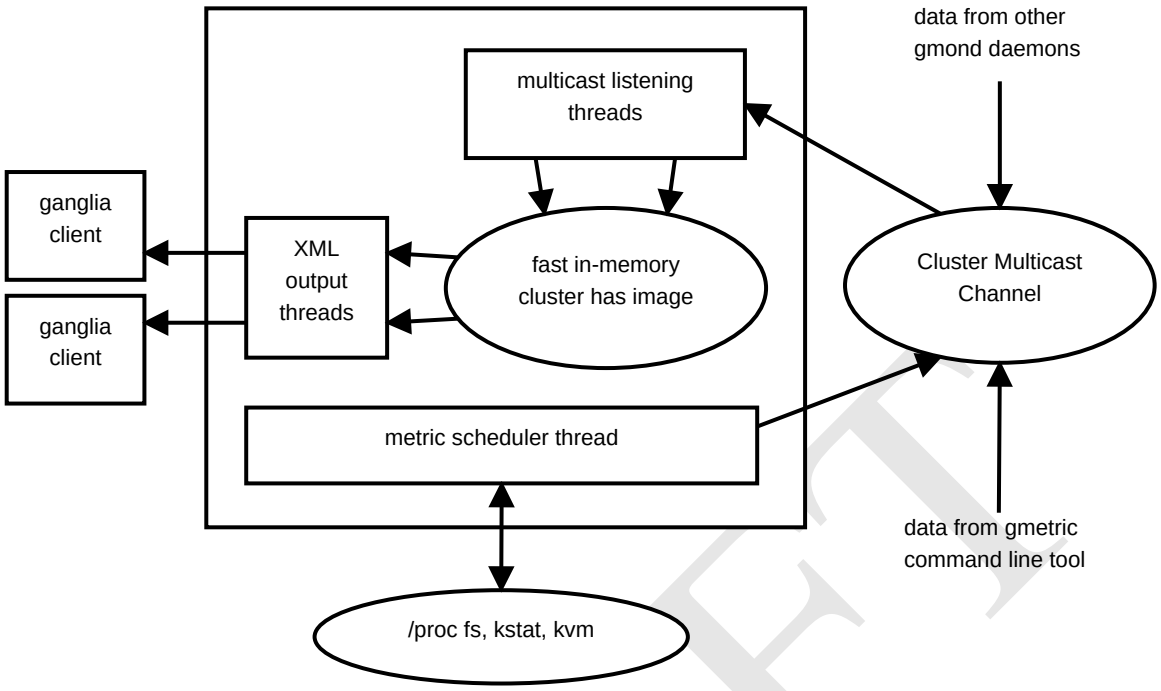


Figure 2.5: Ganglia Data Flow

with GGUS, a ticketing system that european grid initiative uses. Monitoring infrastructure in EGI is fully distributed using regional Nagios servers and the corresponding regional MyEGI portals.

2.5.1 NGS and GridPP

Brunel University takes part in regional and european initiatives. 5 different Computer Elements exist, and 3 Storage Elements, consisting the UKI-LT2-Brunel site. LT2 stands for London Grid, a co-operation with other London Universities. GridPP and NGS are two collaboration groups that Brunel University is member of, and papers on the web interface [24] and real time visualization of the grid status were presented [25] by GridPP

In GridPP, regional monitoring tools exist to provide distributed monitoring services in UK. Regional Nagios and MyEGI/MyEGEE instances co-exist in Oxford University that offer service availability monitoring for all UK sites. Ganglia installations exist in site level deployments, and a Ganglia frontend which aggregates Tier-1 sites is offered through RAL.

DRAFT

Chapter 3

Design/Methods

3.1 Approach Adopted

Grid performance monitoring in this project is examed using **GMA**, an architecture that established to provide the standards for a distributed monitoring system. The technologies that will be discussed here are about the Information Infrastructure that provides the metrics to the users/applications.

The metrics are generated using **Linux kernel**'s load average functions. **Ganglia** is used to take that metrics and synchronize all cluster nodes with the relevant information, over the **multicast channel**.

Nagios is configured using a **custom script** that takes the information for the cluster nodes, and periodically queries the **Gmond** to get the metrics for the discovered nodes. The results are stored in its repository and using RRDTool and pnp4nagios, graph reports are generated on demand.

To pass the information, two different information systems are examined, **BDII and WSRF**. Both are used in modern grid implementations and are described in **MDS specification**. BDII queries event source (Gmond) using Perl/Python LDAP libraries. The results taken, fill the directory schema which has been extended using **Glue schema** specification for Proccessor Load in Computing Element structure.

MDS4 introduces the use of **WSRF** in grid information system. A **Ganglia Information Provider** using **XSLT** takes the XML output from Gmond and aggregates the metrics using

DRAFT

WSRF **Aggregation Framework**. In front of it, a Tomcat instance serves the **WebMDS** frontend to allow **XPath** queries to the results that has been aggregated.

Finally, two sample small applications has been developed to provide a homogeneous interface that display the same information using the two different information systems.

3.2 Design Methods

3.2.1 Grid Monitoring Architecture

By definition [3] Grid Monitoring Architecture consists of three components, as shown in Figure 3.1:

1. **Directory Service** which supports the publish and discovery of the information
2. **Producer component**: which is responsible for the availability of the performance data that takes from the event source and
3. **Consumer component**: the one that requests the performance data and receives the metrics from the producer.

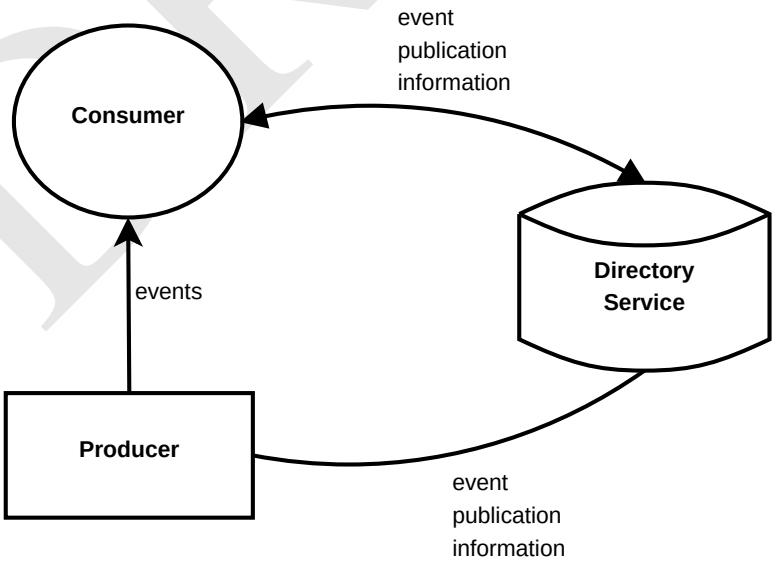


Figure 3.1: Grid Monitoring Architecture

DRAFT

In GMA, all metrics that are transmitted by the producer are handled as events with a timestamp, so performance data should be accurate. These events are transmitted to the consumer directly, and not through the directory service (whose role is just to advertise producers to consumers and vice versa). The GMA recommends that the structure of the data should be following a schema definition.

Grid Monitoring Architecture supports two models to handle the communication between producers and consumers:

- **Streaming publish/subscribe model**
- **Query/Response model**

The directory service is used by both producers to discover consumers and consumers to discover producers. The information of the availability of each producer/consumer is published to the directory service, and each component may initiate a connection to another type of component that has discovered in the directory service. Even though the role of the directory service is so centric in the discovery of components between each other, the performance data messages are transfered between the producer/consumer directly and not via the Directory Service.

3.2.2 GLUE Schema

gained wide acceptance given its adoption by Globus MDS3

GLUE schema came to provide the interoperability needed between US and European Physics Grid Projects. As a standard, a common schema was introduced to describe and monitor the grid resources. Major components include:

- Computing Element (CE)
- Storage Element (SE)
- Network Element (NE)

The implementation of Glue schema may be using LDAP, XML or SQL. The MDS implementation of the Glue schema in this project includes the core Information Provider and the Ganglia Interface for the cluster information.

DRAFT

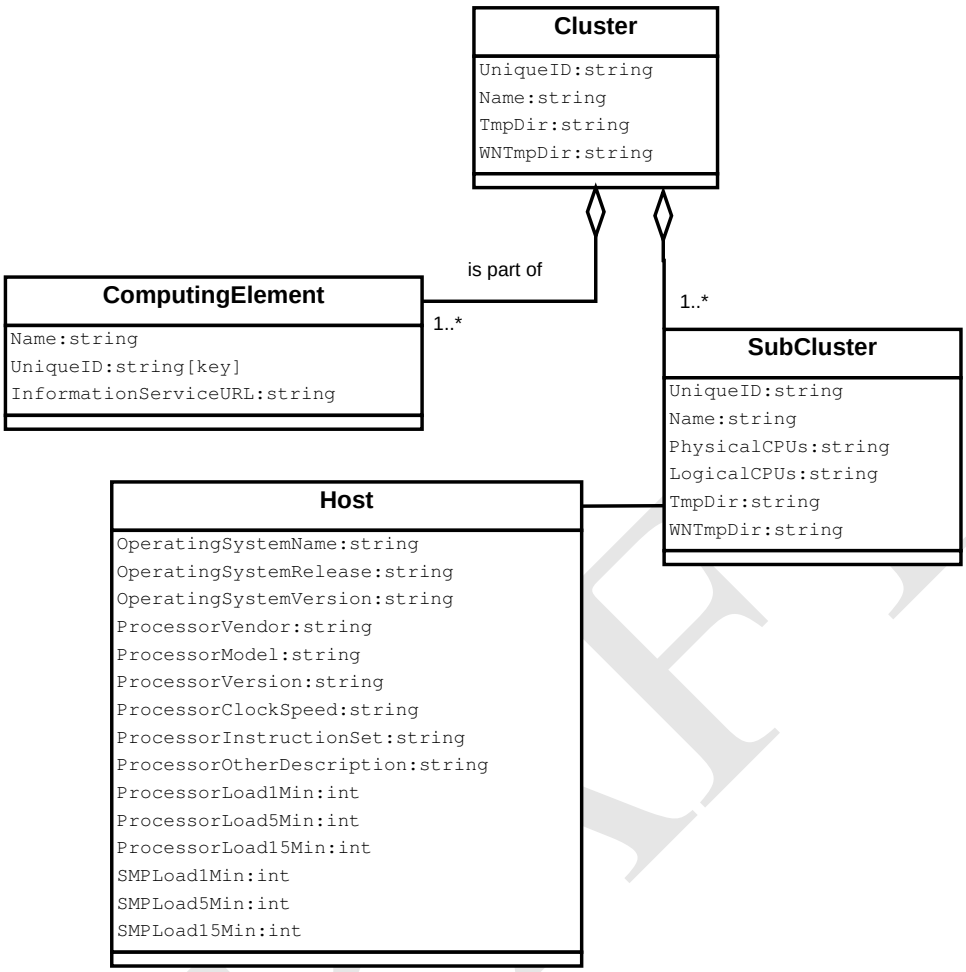


Figure 3.2: GLUE schema 2.0 extention for Host and SMP Load

3.2.3 Information Infrastructure

To design the Information Infrastructure for distributed computing applications some requirements have been considered such as performance, scalability, cost and uniformity.

Because grid computing applications usually operate in large scale installations, there are performance requirements for the information infrastructure. It should allow rapid access to configuration information that is frequently used, using **caching to query periodically each host or index server for the metrics**.

The number of components in a grid infrastructure scales up to hundreds of thousands of nodes, and these components should be available for queries by many different tools. That information should be discoverable using information indexes.

Deployments, maintenance and operations in a large installation of many systems has op-

erational costs for human resources. The information system should automaticaly discover and serve the availability paths for applications and grid resources/servives.

Because of the large number of different heterogeneous networks of nodes and clusters, there is a need of uniformity. Uniformity helps to simplify the developers to build applications that give better configuration decisions. APIs for common operations and data models for the representation of that information. Resources are devided in groups of computing, storage, network elements, etc.

The solution proposed by GLUE standard and X.500 (Directory Service) is the key feature to scale, and get uniformity. It may be used to provide extensible distributed directory services. It is optimised for reads, its binary-tree like hierarchy and usually backend data structure provides a framework that well organize the information that need to be delivered by an Information Infrastructure.[26]

3.3 Data-acquisition Systems

3.3.1 Metrics

CPU load is taken using the pseudo /proc/loadavg file which in turn is filled by Linux kernel's CALC_LOAD macro. This function takes 3 parameters. The load-average bucket, a y constant that is calculated using formula

$$y = \frac{2^{11}}{2^{((5log_2(e))/60x)}}$$

for values $x = 1$, $x = 5$ and $x = 15$ (where x represent the minutes and y the exponent constant), and the number of how many processes are in the queue, in running or uninterruptible state.

3.3.2 Ganglia

The metrics about load in one, five and fifteen minutes are taken from Gmond daemon through the proc filesystem as seen in Figure 3.3. These values are multicasted using a UDP message on the network, only if the value has been changed from the previous one taken. There is also a time thresohold that after that time the value is been sent again, even if it haven't changed, so new hosts on the network may gather the data needed for their Gmond. Each host of a cluster

DRAFT

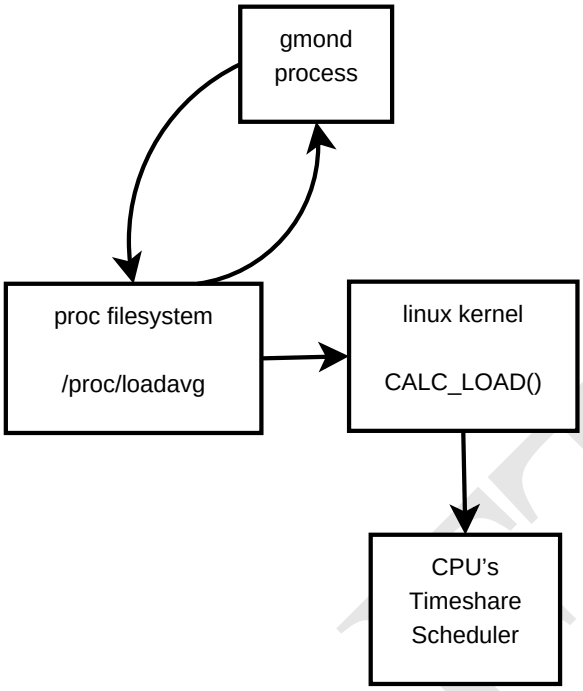


Figure 3.3: Load Average calculation

have the information about the metrics of itself and each other node, so it stores the whole cluster state. Using loopback interface, every Gmond sends its metrics to itself.

If a TCP connection on the Gmond listening port 8649 is made, Gmond writes a full cluster state of metrics in XML including its DTD. There is a typical access list in the configuration called trusted hosts, and of course every node that is in the cluster that a specific node is configured to be part of, is allowed to connect to get the XML.

3.3.3 Nagios

Nagios is the core monitoring tool that is used for grid computing monitoring as Multi Level Monitoring architecture proposes, to meet the needs of EGEE/EGI. Following SAM and Gridview, Nagios instances have been deployed in many levels of grid infrastructure, enhansing the functionality of scheduling and execution of site tests. The message bus that uses is MSG, which offers an integration between Nagios and the other monitoring tools of grid.

CERN provides MSG-Nagios-bridge, a mechanism to transfer test results between different levels of Nagios deployment (regional, project, site). MSG-Nagios-bridge both submit tests to other Nagios installations and consume results from them.

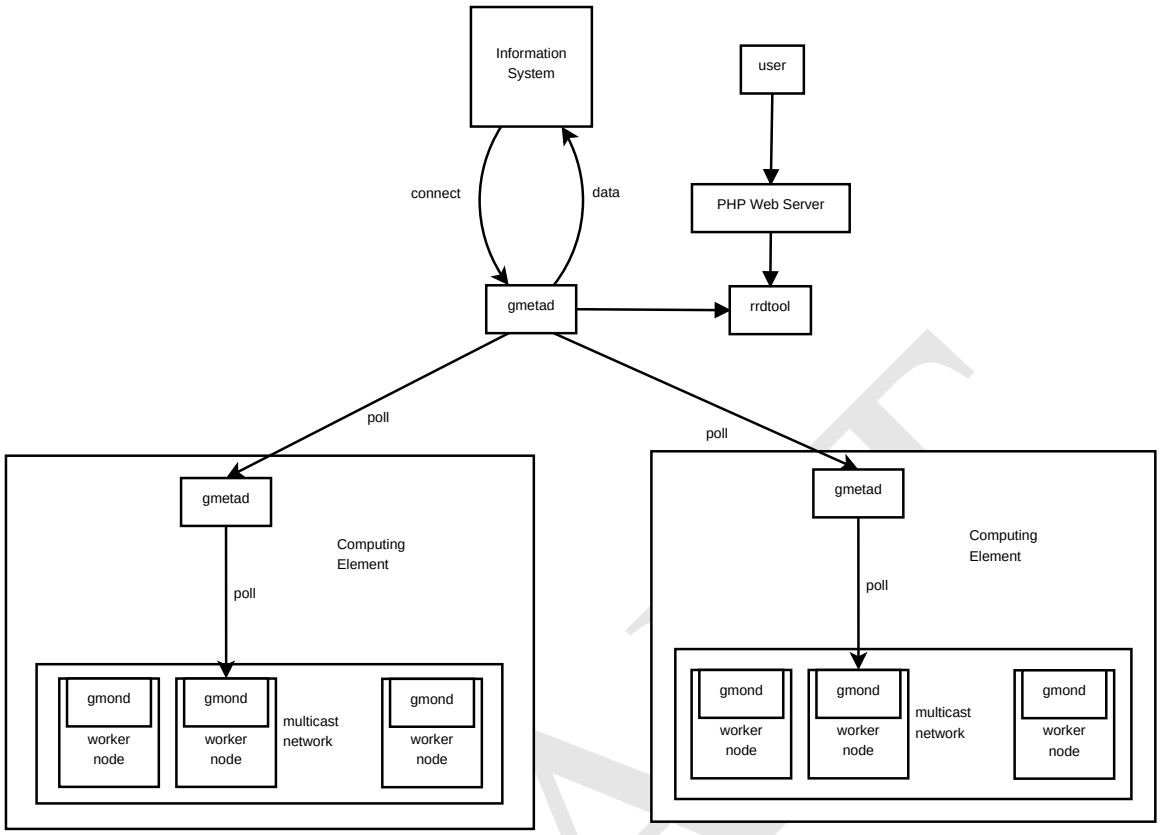


Figure 3.4: Ganglia Network Communications

A Regional Metric Store is also used by nagios. It is a database that provides a back-end to Nagios current and historical metrics, and connected with the frontend and the message bridge. The adaptor that provides such functionality called NDOUtils, and may have a MySQL/PorstgreSQL or Oracle backend.

In the front-end, users are allowed to discover the nodes and services provided in the monitoring levels by regions, projects and sites, using cgi scripts that are part of the Nagioc core distribution. Access control between levels of Nagios instances and between users and Nagios installations, is performed using the standard methods of grid, which is GOCDB as described in ATP. User authentication is done by user certificates.

Listing 3.1: Ganglia to Nagios script

```
#!/bin/bash
if [ ! $1 ]
```

```
then
        echo "Please HOST argument"
        echo "ex. ganglia_to_nagios 10.0.0.1"
        exit
fi
/usr/src/redhat/SOURCES/ganglia-python-3.3.0/ganglia.py --host $1 ↩
   --live | while read host
do
        echo ";$host.oslab.teipir.gr
define host{
        use     gmond-host
        host_name       $host.oslab.teipir.gr
        alias           $host
        address         $host.oslab.teipir.gr
        hostgroups      worker-nodes
}
"
done > /etc/nagios/teipir/hosts.cfg
```

Bulk Mode with NPCD NPCD: spool directory to process bulk data create graphs using RRDTOOL

3.4 Range of cases examined

Publish to Information System [27]

3.4.1 LDAP based - MDS/BDII

To integrate Ganglia with MDS in early versions of Globus, there schema of OpenLDAP should be extended using the Glue-CE definitions from the DataTAG web site (MDS version 2.4). A Ganglia Information Provider was the native ganglia client on python, given by the ganglia development team itshelf.

DRAFT

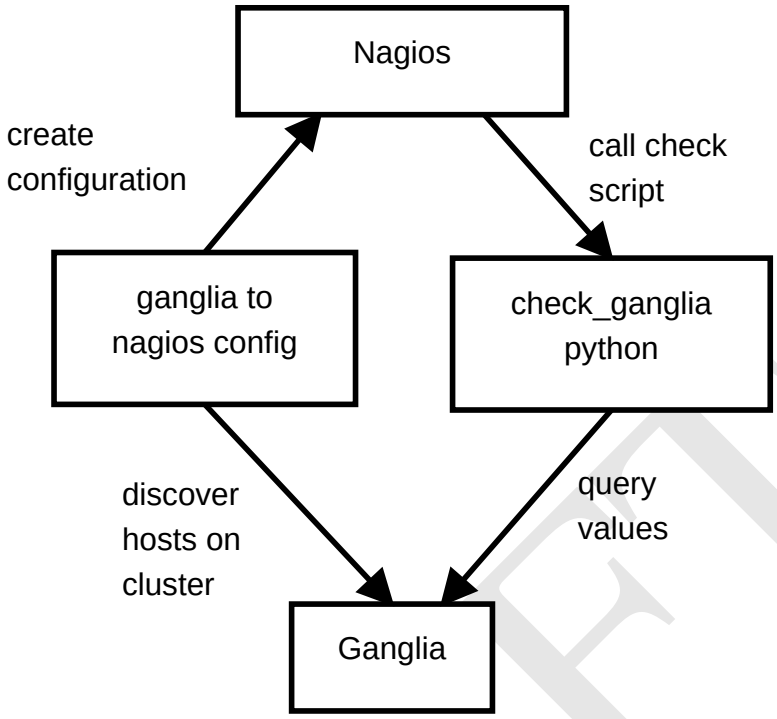


Figure 3.5: Nagios configuration and check ganglia values

1. Python ganglia client script: `http://globus.org/toolkit/docs/2.4/mds/gangliaprovider.html`
2. Perl gaglia-IP tool: `http://www.star.bnl.gov/public/comp/Grid/Monitoring/SimpleGangliaIP.h` and `http://www.star.bnl.gov/public/comp/Grid/Monitoring/ganglia_ip`

```
# which glue services does the information base provides?
[root@osweb ~]# ldapsearch -h dgc-grid-44.brunel.ac.uk -p 2170 -x -b "mds-vo-name=uki-lt2

# installation of bdii
-install glite-ui which installs bdii
-ldapsearch bdii
-test perl & python scripts to export mds from gmond
-configure connection to

[root@osweb ~]# /opt/glite/yaim/bin/yaim -c -s site-info.def -n BDII_site

# configure BDII
```

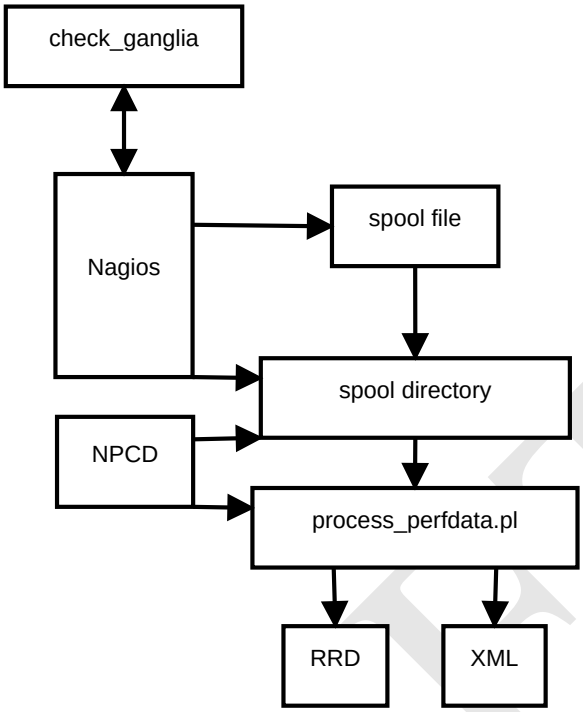


Figure 3.6: PNP 4 Nagios data flow

```
site-info.def:
# BDII
CE_HOST="osweb.teipir.gr"
SITE_BDII_HOST="osweb.teipir.gr"
SITE_EMAIL="theofpa@teipir.gr"
SITE_LAT=37.979166
SITE_LONG=23.674719
SITE_DESC="TEI of Piraeus"
SITE_LOC="Athens, Greece"
SITE_WEB="http://oslab.teipir.gr"
SITE_SECURITY_EMAIL=$SITE_EMAIL
SITE_SUPPORT_EMAIL=$SITE_EMAIL
SITE_OTHER_GRID="EGEE"
BDII_REGIONS="oslab.teipir.gr"
```

DRAFT

Common Name	Attribute	Objectclass
Hostname	GlueHostName	GlueHost
Unique ID assigned to the host	GlueHostUniqueID	GlueHost
Processor Load, 1 Min Average	GlueHostProcessorLoadLast1Min	GlueHostProcessorLoad
Processor Load, 5 Min Average	GlueHostProcessorLoadLast5Min	GlueHostProcessorLoad
Processor Load, 15 Min Average	GlueHostProcessorLoadLast15Min	GlueHostProcessorLoad
SMP Load, 1 Min Average	GlueHostSMPLoadLast1Min	GlueHostSMPLoad
SMP Load, 5 Min Average	GlueHostSMPLoadLast5Min	GlueHostSMPLoad
SMP Load, 15 Min Average	GlueHostSMPLoadLast15Min	GlueHostSMPLoad
Number of CPUs	GlueHostArchitectureSMPSize	GlueHostArchitecture
Processor Clock Speed (MHz)	GlueHostProcessorClockSpeed	GlueHostProcessor
Network Interface name	GlueHostNetworkAdapterName	GlueHostNetworkAdapter
Network Adapter IP address	GlueHostNetworkAdapterIPAddress	GlueHostNetworkAdapter
The amount of RAM	GlueHostMainMemoryRAMSize	GlueHostMainMemory
Free RAM (in KBytes)	GlueHostMainMemoryRAMAvailable	GlueHostMainMemory

Table 3.1: GLUE schema for Host Processor Information Provider

Perl:

```
$[root@mon ~]# ./ganglia_ip -h mon -p 8649 -o mds
```

Python:

```
$[root@mon ~]# /opt/ganglia/bin/ganglia --format=MDS
```

```
[root@osweb ~]# cat /opt/glite/etc/gip/provider/glite-info-provider↩
   -service-ganglia-wrapper
#!/bin/bash
/opt/bin/ganglia_ip -h 195.251.70.54 -p 8649 -o mds
```

DRAFT

3.4.2 Web Service based - WSRF

container

using WSRF (GT4, information services, information providers) Ganglia Resource Provider MDS Index Service GLUE CE

OASIS standard

container

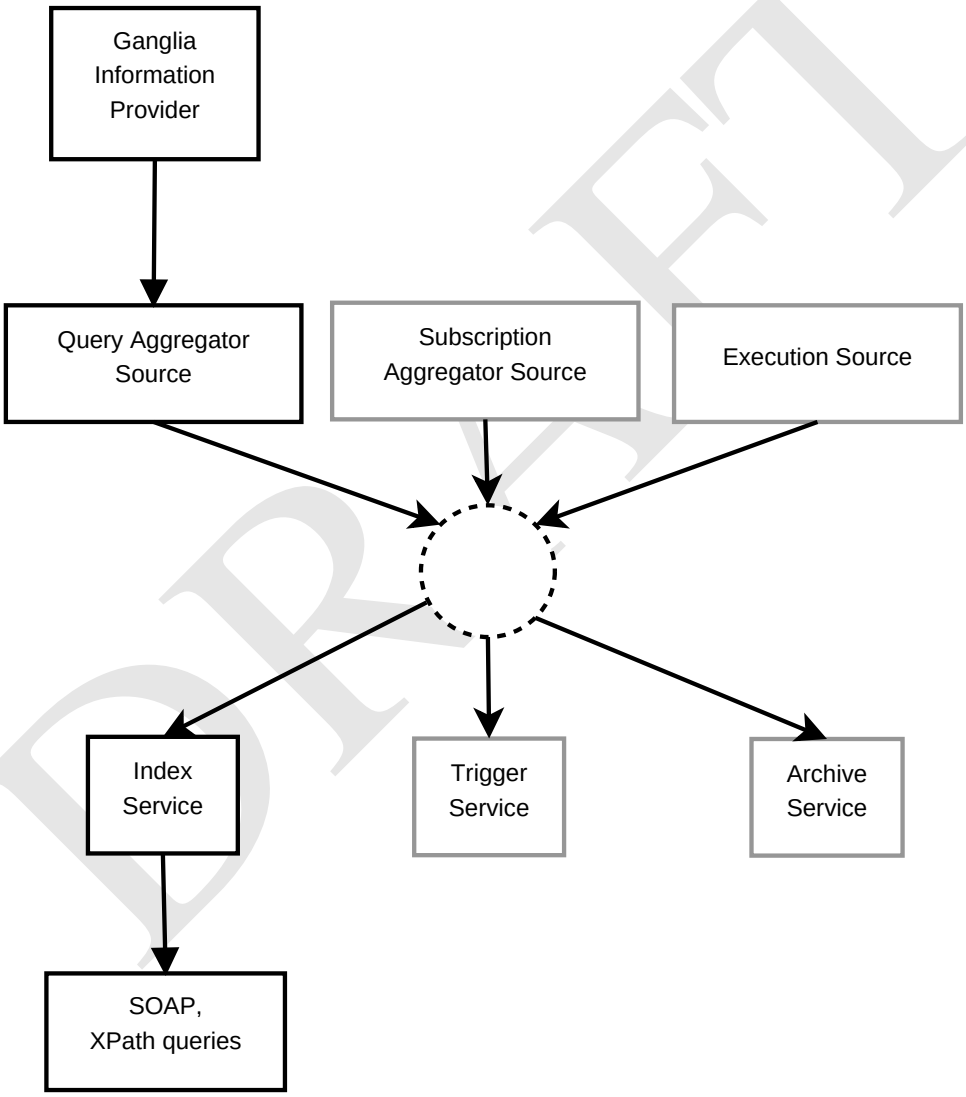


Figure 3.7: Web Service Resource Framework

DRAFT

3.4.3 information provider

wssd

and

rp xml

and

hierarchy.xml maybe in an ”aggregation” section

and

deserialization of MDS query in gmond to WSRF

```
# installation of globus, wsrf, ganglia:
-download binary 4.0.7 and extract to /opt/globus
wget http://www-unix.globus.org/ftppub/gt4/4.0/4.0.7/installers/bin/gt4.0.7-x86_rhas_4-ir
export $JAVA_HOME=/usr/java/default
[root@osweb ~]# export JAVA_HOME=/etc/alternatives/java_sdk
export PATH=$JAVA_HOME/bin:$PATH
./configure
make
make install
-install postgresql, create user & db
-configure globus to connect to postgresql
-create init script for wsrf container
-create ganglia resource provider for wsrf and connect to gmond
-call wsrf to get ganglia metrics

# query WSRF for ganglia data:
[root@osweb ~]# grid-proxy-init -verify -debug
[root@osweb ~]# wsrf-query -s https://osweb.teipir.gr:8443/wsrf/services/DefaultIndexServ
[root@osweb ~]# /opt/globus/bin/wsrf-query -s https://osweb.teipir.gr:8443/wsrf/services/
```

DRAFT

XPath

XPath is used to parse an XML document and get a part of it using an address scheme. For XPath, the XML document is a tree consisting of nodes, and its purpose as a language is to get the nodes that are addressed using the XPath query from that document.

Its syntax is compact, non-XML and much like the filesystem addressing, so it facilitates the use of XPath within URIs.

Example queries used in this project are:

The following is used in the PHP code that queries the WebMDS for all nodes of the XML of the WSRF containing nodes with name *Host*:

```
//*[local-name()='Host']
```

Another example is a more complex query that asks the WSRF for all nodes with name *Host* that contains a sub-node named *ProcessorLoad* and its *Last*15*Min* attribute has value larger than 20:

```
//glue:Host[glue:ProcessorLoad[@glue:Last15Min>20]]
```

Finally the following example may return only the *ProcessorLoad* node of the *Host* that has the attribute Name set to *xenia.oslab.teipir.gr*:

```
//glue:Host[@glue:Name='xenia.oslab.teipir.gr']/glue:ProcessorLoad
```

XSLT

my note: WSRF is GLUE 2.0 schema CE compatible

```
file /opt/globus/etc/globus_wsrf_mds_usefulrp/ganglia_to_glue.xslt
```

Listing 3.2: WSRF XSLT for Ganglia Information Provider

```
<glue:ProcessorLoad>

<xsl:attribute name="glue:Last1Min">
```


DRAFT

```
  <xsl:call-template name="emitProperNumeric">
    <xsl:with-param name="numeric"
     select="floor(100 * METRIC[@NAME='load_one']/@VAL)"/>
  </xsl:call-template>
</xsl:attribute>

<xsl:attribute name="glue:Last5Min">
  <xsl:call-template name="emitProperNumeric">
    <xsl:with-param name="numeric"
     select="floor(100 * METRIC[@NAME='load_five']/@VAL)"/>
  </xsl:call-template>
</xsl:attribute>

<xsl:attribute name="glue:Last15Min">
  <xsl:call-template name="emitProperNumeric">
    <xsl:with-param name="numeric"
     select="floor(100 * METRIC[@NAME='load_fifteen']/@VAL)"/>
  </xsl:call-template>
</xsl:attribute>

</glue:ProcessorLoad>
```

DRAFT

WebMDS

WebMDS is a web interface to query WSRF resource property information. It consists of forms and views of raw XML or organized in tables of results. This user friendly frontend comes as a part of Globus Toolkit version 4 and it can deployed in any application server. Behind this application reside the data that the WSRF aggregation framework provides through the Index Service.

For this project an Apache Tomcat server was installed in the box that globus toolkit was running, and the **webmds application** from the GT4 home was deployed. In webmds configuration file, the globa option to allow user specified queries using XPath was enabled.

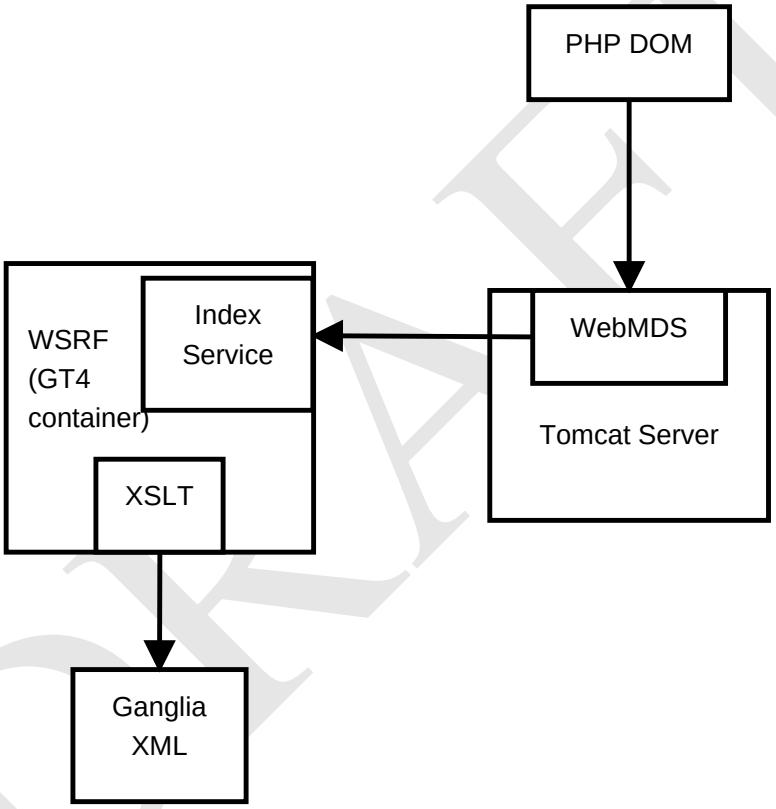


Figure 3.8: WebMDS application

DRAFT

Chapter 4

Results

4.1 Tables and Plots

4.1.1 Unix stuff

As described in subsection Metrics in the previous chapter, linux provides through the **proc pseudo-filesystem** a simple file interface to the metrics taken from the scheduler of processes that are queued in the processor. The three metrics about cpu load on 1, 5 and 15 minutes average is displayed as follows:

```
[root@gr03 ~]# cat /proc/loadavg
2.29 0.73 0.32 1/230 3584
```

which may also be displayed using the **uptime command**:

```
[root@gr03 ~]# uptime
 00:01:20 up  1:41,  3 users,  load average: 2.29, 0.73, 0.32
```

Looking in the Linux kernel source, there is the CALC_LOAD macro command which takes the options that have been discused and returns the result of the metric. The definition of the macro can be seen in file **/usr/src/kernels/‘uname -r‘/include/linux/sched.h**:

Listing 4.1: Linux kernel CALC_LOAD macro

DRAFT

```
extern unsigned long avenrun[];         /* Load averages */
extern void get_avenrun(unsigned long *loads,
                        unsigned long offset, int shift);

#define FSHIFT      11          /* nr of bits of precision */
#define FIXED_1     (1<<FSHIFT) /* 1.0 as fixed-point */
#define LOAD_FREQ   (5*HZ+1)    /* 5 sec intervals */
#define EXP_1       1884        /* 1/exp(5sec/1min) as fixed-point */
#define EXP_5       2014        /* 1/exp(5sec/5min) */
#define EXP_15      2037        /* 1/exp(5sec/15min) */

#define CALC_LOAD(load,exp,n) \
        load *= exp; \
        load += n*(FIXED_1-exp); \
        load >>= FSHIFT;

extern unsigned long total_forks;
extern int nr_threads;
DECLARE_PER_CPU(unsigned long, process_counts);
extern int nr_processes(void);
extern unsigned long nr_running(void);
extern unsigned long nr_uninterruptible(void);
extern unsigned long nr_iowait(void);
extern unsigned long nr_iowait_cpu(int cpu);
extern unsigned long this_cpu_load(void);
```

DRAFT

4.1.2 Ganglia

Installation and configuration

Listing 4.2: Gmetad installation

```
# yum install rrdtool perl-rrdtool rrdtool-devel apr-devel ↩
   libconfuse libconfuse-devel expat expat-devel pcre pcre-devel
# GANGLIA_ACK_SYSCONFDIR=1 ./configure --with-gmetad
# make
# make install
# mkdir -p /var/lib/ganglia/rrds
# chown -R nobody /var/lib/ganglia
```

Listing 4.3: Gmond installation

```
# yum install apr-devel libconfuse libconfuse-devel expat expat-↩
   devel pcre pcre-devel
# GANGLIA_ACK_SYSCONFDIR=1 ./configure
# make
# make install
# iptables -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p ↩
   tcp --dport 8649 -j ACCEPT
```

configure with multicast or unicast. jitter problem environments may use unicast, as long as amazon ec2 environments that are not support multicast.

Transport and sample

Gmond code uses the ganglia libmetrics library which in case of Linux operating system parses the */proc/loadavg* pseudo-file to get linux kernel calculated system load average.

Listing 4.4: libmetrics code to get load average

```
timely_file proc_loadavg = { {0,0} , 5., "/proc/loadavg" };
/* ... */
g_val_t
load_one_func ( void )
```

DRAFT

```
{
    g_val_t val;
    val.f = strtod( update_file(&proc_loadavg), (char **)NULL);
    return val;
}
```

code from gmond client

example output

When gmond starts, it listens on port 8649/TCP by default, to accept TCP connections and throw XML report for the whole cluster. It also binds to the multicast address on port 8649/UDP to get other hosts messages for metrics changes, and also multicast its own metrics.

Listing 4.5: Gmond networking

```
[root@gr01 ~]# lsof -i 4 -a -p `pidof gmond`
COMMAND   PID   USER   FD   TYPE DEVICE SIZE NODE NAME
gmond   11900 nobody    4u  IPv4  33699       UDP 239.2.11.71:8649
gmond   11900 nobody    5u  IPv4  33701       TCP *:8649 (LISTEN)
gmond   11900 nobody    6u  IPv4  33703       UDP gr01.oslab.teipir↩
    .gr:39991->239.2.11.71:8649
```

port and XML staff

Listing 4.6: Gmond XML cluster report

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<GANGLIA_XML VERSION="3.1.7" SOURCE="gmond">
  <CLUSTER NAME="RDLAB" LOCALTIME="1297198943" OWNER="TEIPIR" ↩
      LATLONG="unspecified" URL="unspecified">
    <HOST NAME="gr02.oslab.teipir.gr" IP="10.0.0.32" REPORTED="↩
        1297198934" TN="8" TMAX="20" DMAX="0" LOCATION="unspecified"↩
        GMOND_STARTED="1296569542">
```

DRAFT

```
        <METRIC NAME="load_one" VAL="0.01" TYPE="float" UNITS=" " TN=↩
            "50" TMAX="70" DMAX="0" SLOPE="both">
        <EXTRA_DATA>
        <EXTRA_ELEMENT NAME="GROUP" VAL="load"/>
        <EXTRA_ELEMENT NAME="DESC" VAL="One minute load average"/>
        <EXTRA_ELEMENT NAME="TITLE" VAL="One Minute Load Average"/>
        </EXTRA_DATA>
        </METRIC>
        ...
    </HOST>
    ...
  </CLUSTER>
</GANGLIA_XML>
```

multicast examples

Gmetad

XDR output

Listing 4.7: XDR sample

```
[root@gr01 ~]# tcpdump -A -i eth2 dst host 239.2.11.71
22:38:26.062266 IP gr01.oslab.teipir.gr.39991 > 239.2.11.71.8649: ↩
    UDP, length 56
E..T..@.....G.7!..@.X.......gr01.oslab.teipir.gr....load_one↩
    ........%.2f..
```

gstat -al1

Listing 4.8: Gstat output

```
[root@gr01 ~]# gstat -al1
```

DRAFT

```
gr03.oslab.teipir.gr      2 (    0/   87) [  0.00,  0.00,  0.00] [ ↩
     0.0,   0.0,   0.0,  99.9,   0.1] OFF
gr01.oslab.teipir.gr      1 (    0/   75) [  0.00,  0.00,  0.00] [ ↩
     0.0,   0.0,   0.0,  99.9,   0.0] OFF
gr02.oslab.teipir.gr      1 (    0/   99) [  0.00,  0.00,  0.00] [ ↩
     0.0,   0.0,   0.1,  99.9,   0.0] OFF
```

4.1.3 Visualization

Drill down and levels of visualization

1. local resource
2. site
3. regional
4. grid

ganglia GUI

rrdtool

4.2 Methods of Presentation

4.2.1 WSRF

```
/opt/globus/bin/wsrf-query \
-s https://osweb.teipir.gr:8443/wsrf/services/DefaultIndexService \
"//*[local-name()='Host']"
```

Listing 4.9: WSRF query output

```
<ns1:GLUECE xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
 <ns1:Cluster ns1:Name="OSLAB" ns1:UniqueID="OSLAB">
```


DRAFT

```
<ns1:SubCluster ns1:Name="main" ns1:UniqueID="main">
 <ns1:Host ns1:Name="gr03.oslab.teipir.gr"
 ns1:UniqueID="gr03.oslab.teipir.gr"
 xmlns:ns1="http://mds.globus.org/glue/ce/1.1">
  <ns1:Processor ns1:CacheL1="0" ns1:CacheL1D="0"
  ns1:CacheL1I="0" ns1:CacheL2="0" ns1:ClockSpeed="2392"
  ns1:InstructionSet="x86"/>
  <ns1:MainMemory ns1:RAMAvailable="299" ns1:RAMSize="1010"
  ns1:VirtualAvailable="2403" ns1:VirtualSize="3132"/>
  <ns1:OperatingSystem ns1:Name="Linux"
  ns1:Release="2.6.18-194.26.1.el5"/>
  <ns1:Architecture ns1:SMPSize="2"/>
  <ns1:FileSystem ns1:AvailableSpace="201850"
  ns1:Name="entire-system" ns1:ReadOnly="false"
  ns1:Root="/" ns1:Size="214584"/>
  <ns1:NetworkAdapter ns1:IPAddress="10.0.0.33"
  ns1:InboundIP="true" ns1:MTU="0"
  ns1:Name="gr03.oslab.teipir.gr" ns1:OutboundIP="true"/>
  <ns1:ProcessorLoad ns1:Last15Min="45" ns1:Last1Min="337"
  ns1:Last5Min="126"/>
 </ns1:Host>
</ns1:SubCluster>
</ns1:Cluster>
</ns1:GLUECE>
```

XSLT

WebMDS and XPath

query examples

interface and forms description XPath query:

```
//glue:Host[@glue:Name='ltsp.oslab.teipir.gr']
```

DRAFT

Result:

Listing 4.10: WebMDS results from XPath query

```
<WebmdsResults>
  <ns1:Host ns1:Name="ltsp.oslab.teipir.gr" ns1:UniqueID="ltsp.↩
     oslab.teipir.gr">
    <ns1:Processor ns1:CacheL1="0" ns1:CacheL1D="0" ns1:CacheL1I="0↩
       " ns1:CacheL2="0" ns1:ClockSpeed="1600" ns1:InstructionSet="↩
       x86_64"/>
    <ns1:MainMemory ns1:RAMAvailable="17806" ns1:RAMSize="20121" ↩
       ns1:VirtualAvailable="22137" ns1:VirtualSize="24508"/>
    <ns1:OperatingSystem ns1:Name="Linux" ns1:Release="2.6.32-24-↩
       server"/>
    <ns1:Architecture ns1:SMPSize="8"/>
    <ns1:FileSystem ns1:AvailableSpace="34243" ns1:Name="entire-↩
       system" ns1:ReadOnly="false" ns1:Root="/" ns1:Size="251687"/↩
       >
    <ns1:NetworkAdapter ns1:IPAddress="192.168.0.101" ns1:InboundIP↩
       ="true" ns1:MTU="0" ns1:Name="ltsp.oslab.teipir.gr" ↩
       ns1:OutboundIP="true"/>
    <ns1:ProcessorLoad ns1:Last15Min="9" ns1:Last1Min="1" ↩
       ns1:Last5Min="9"/>
  </ns1:Host>
</WebmdsResults>
```

raw XML output from WebMDS

4.2.2 BDII

/opt/glite/etc/gip/provider/glite-info-provider-service-xxx create a wrapper to present ganglia url and status to the BDII

Ganglia official python client:

Listing 4.11: Python Ganglia client MDS export

```
[root@mon ~]# /opt/ganglia/bin/ganglia --format=MDS | grep -A 30 ↩
    host=gr03

dn: host=gr03.oslab.teipir.gr, scl=sub2, cl=datatag-CNAF, \
 mds-vo-name=local, o=grid
objectclass: GlueHost
GlueHostName: gr03.oslab.teipir.gr
GlueHostUniqueID: RDLAB-TEIPIR-gr03.oslab.teipir.gr
objectclass: GlueHostArchitecture
GlueHostArchitecturePlatformType: x86-Linux
GlueHostArchitectureSMPSize: 2
objectclass: GlueHostProcessor
GlueHostProcessorClockSpeed: 2392
objectclass: GlueHostMainMemory
GlueHostMainMemoryRAMSize: 1035104
GlueHostMainMemoryRAMAvailable: 306280
objectclass: GlueHostNetworkAdapter
GlueHostNetworkAdapterName: gr03.oslab.teipir.gr
GlueHostNetworkAdapterIPAddress: 10.0.0.33
GlueHostNetworkAdapterMTU: unknown
GlueHostNetworkAdapterOutboundIP: 1
GlueHostNetworkAdapterInboundIP: 1
objectclass: GlueHostProcessorLoad
GlueHostProcessorLoadLast1Min: 2.57
GlueHostProcessorLoadLast5Min: 1.48
GlueHostProcessorLoadLast15Min: 0.58
objectclass: GlueHostSMPLoad
GlueHostSMPLoadLast1Min: 2.57
GlueHostSMPLoadLast5Min: 1.48
GlueHostSMPLoadLast15Min: 0.58
objectclass: GlueHostStorageDevice
```

DRAFT

```
GlueHostStorageDeviceSize: 209555000
GlueHostStorageDeviceAvailableSpace: 197120000
GlueHostStorageDeviceType: disk
```

Perl script to export MDS format:

Listing 4.12: Perl Ganglia Information Provider for MDS

```
[root@mon ~]# ./ganglia_ip -h mon -p 8649 -o mds | grep -A 22 host=↩
   gr03

dn: host=gr03.oslab.teipir.gr, cl=RDLAB, \
 mds-vo-name=local, o=grid
objectclass: GlueHost
GlueHostName: gr03.oslab.teipir.gr
GlueHostUniqueID: RDLAB-TEIPIR-gr03.oslab.teipir.gr
objectclass: GlueHostProcessorLoad
GlueHostProcessorLoadLast1Min: 2.57
GlueHostProcessorLoadLast5Min: 1.48
GlueHostProcessorLoadLast15Min: 0.58
objectclass: GlueHostSMPLoad
GlueHostSMPLoadLast1Min: 2.57
GlueHostSMPLoadLast5Min: 1.48
GlueHostSMPLoadLast15Min: 0.58
objectclass: GlueHostArchitecture
GlueHostArchitectureSMPSize: 2
objectclass: GlueHostProcessor
GlueHostProcessorClockSpeed: 2392
objectclass: GlueHostNetworkAdapter
GlueHostNetworkAdapterName: gr03.oslab.teipir.gr
GlueHostNetworkAdapterIPAddress: 10.0.0.33
objectclass: GlueHostMainMemory
GlueHostMainMemoryRAMSize: 1035104
```

DRAFT

```
GlueHostMainMemoryRAMAvailable: 306280
```

Listing 4.13: BDII LDAP search for Glue CE ProcessorLoad attributes

```
# ldapsearch -H ldap://osweb.teipir.gr:2170 -x \
-b GlueHostName=ainex.local,Mds-Vo-name=local,o=grid \
GlueHostProcessorLoadLast1Min GlueHostProcessorLoadLast5Min \
GlueHostProcessorLoadLast15Min

# ainex.local, local, grid
dn: GlueHostName=ainex.local,Mds-Vo-name=local,o=grid
GlueHostProcessorLoadLast1Min: 27
GlueHostProcessorLoadLast15Min: 22
GlueHostProcessorLoadLast5Min: 20
```

4.3 Description of Information

4.3.1 DOM

Listing 4.14: PHP DOM call to WebMDS

```
$url="http://osweb.teipir.gr:8080/webmds/webmds?info=indexinfo&xsl↩
   =&xmlSource.indexinfo.param.xpathQuery=%2F%2F*[local-name↩
   %28%29%3D%27Host%27]";
$file  = file_get_contents($url);
$dom = DOMDocument::loadXML($file);
$host = $dom->getElementsByTagName('Host');
$procload = $host->item($k)->getElementsByTagName('ProcessorLoad');
echo $procload->item($i)->getAttribute('Last1Min');
```

DRAFT

4.3.2 LDAP

Listing 4.15: PHP LDAP call to BDII

```
$ds=ldap_connect("osweb.teipir.gr","2170");
if ($ds)
{
    $r=ldap_bind($ds);
    $sr=ldap_search($ds, "mds-vo-name=local,o=grid", "(&(↩
        objectClass=GlueHostProcessorLoad))");
    if ($sr)
    {
        $info = ldap_get_entries($ds, $sr) or die("could not fetch↩
            entries");
        echo ($info[0][gluehostprocessorloadlast1min][0])/100;
    }
ldap_close($ds);
}
```

Hostname	1min	5min	15min
ltsp.oslab.teipir.gr	0.01	0.09	0.09
xenia.oslab.teipir.gr	0.00	0.06	0.06
gr201.oslab.teipir.gr	0.52	0.59	0.42
gr130.oslab.teipir.gr	0.00	0.06	0.16
gr131.oslab.teipir.gr	1.22	0.79	0.40
gr212.oslab.teipir.gr	0.06	0.09	0.09
gr180.oslab.teipir.gr	2.06	1.49	0.59
gr181.oslab.teipir.gr	0.71	0.57	0.32

Table 4.1: Sample output from both calls with DOM or LDAP

DRAFT

4.3.3 Nagios

Host	Service	Status	Last Check	Status Information
gr129	load_fifteen	OK	02-08-2011 20:17:23	CHECKGANGLIA OK: load_fifteen is 0.17
	load_five	OK	02-08-2011 20:18:12	CHECKGANGLIA OK: load_five is 0.27
	load_one	OK	02-08-2011 20:17:43	CHECKGANGLIA OK: load_one is 0.02
gr130	load_fifteen	OK	02-08-2011 20:14:23	CHECKGANGLIA OK: load_fifteen is 1.77
	load_five	WARNING	02-08-2011 20:14:15	CHECKGANGLIA OK: load_five is 4.75
	load_one	CRITICAL	02-08-2011 20:14:43	CHECKGANGLIA OK: load_one is 11.60

Table 4.2: Example Nagios service status details for ganglia check

screenshot of output (green/yellow/red) colored

pages from `http://people.brunel.ac.uk/~dc09ttp`

DRAFT

Chapter 5

Analysis

5.1 Methods Adopted

In complex distributed systems such as grids, performance bottlenecks may be located using monitoring data. From the processor usage on a single node of a computing element to the total usage of processed jobs in a large cluster, performance data help to focus on the problem that impacts the overall performance.

In order to succeed in grid monitoring, some requirements should be considered. A very large amount of data should be delivered real-time, from many heterogeneous sources on different networks or even countries. These data must be accurate and consistent. There should be synchronized timestamps on the generation of each metric, to the measurement value that should be comparable between different architectures. The time synchronization of the hosts of each cluster may be done using network time protocol, so all metrics are taken on the time that they actually report. Metrics should have error bounds to preserve accuracy, and the consistency issue is solved using coordination of that activity, so the impact of a metric to other sensors is controlled.

The flow of the monitoring process initialization is described from the GMA standard. The application-consumer queries the directory service in order to declare its interest to get metrics for a specific host/cluster. The sensors of the elements that is equivalent to the specific query generates the metrics that will be given to the consumer from the producer, which in turn queries the directory service to find the consumer. The producer is the one that initializes the

DRAFT

connection to the consumer in order to deliver the measurements, even if the consumer had asked the directory service for this.

5.1.1 Performance Metrics

CALC_LOAD - load average

extra

5.1.2 Information Systems

BDII

WSRF

5.2 Interpretation of Results

Discussion about performance results based on load average.

some UNIX internals, processes, scheduler not a percentage counter of CPU usage

Difference between these metrics and the availability of a grid based on the queue of jobs have been submitted.

Availability monitoring

-MyEGI, monitor visualization environment -django data models -MRS database, ATP based schema

5.3 Specific Interpretations

5.3.1 Scaling

LDAP

there is a paper about how information systems perform in large scale

DRAFT

WSRF

5.4 Enveloping Interpretations

DRAFT

Chapter 6

Conclusions

6.1 Conclusions

Conclusions should be based on an in depth critical analysis of the information presented in the dissertation and should be related to the objectives stated in the introduction.

do not simply summarize the dissertation

do not recapitulate the analysis or discussion

do not introduce new ideas

identify specific points that have been clarified or discovered, and specific actions to be taken

identify specific additional investigation that is required (and why)

It is important to remember that conclusions should only be drawn on the basis of the information presented in the dissertation. Generalized conclusions without supporting evidence are to be discouraged.

6.2 Further Work

Identify specific additional investigation that is required to be carried out.

DRAFT

Bibliography

[1] M. Li and M. Baker, *The grid: core technologies*. John Wiley & Sons Inc, 2005.

[2] S. Fisher, "DataGrid information and monitoring services architecture: design, requirements and evaluation criteria'," tech. rep., Technical Report, DataGrid, 2002.

[3] G. Taylor, M. Irving, P. Hobson, C. Huang, P. Kyberd, and R. Taylor, *Distributed monitoring and control of future power systems via grid computing*. IEEE, 2006.

[4] A. Kertész and P. Kacsuk, "A Taxonomy of Grid Resource Brokers," in *6th Austrian-Hungarian Workshop on Distributed and Parallel Systems (DAPSYS*, pp. 318–366, Basil Blackwell, 2006.

[5] B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, and R. Wolski, "A grid monitoring architecture," in *The Global Grid Forum GWD-GP-16-2*, Citeseer, 2002.

[6] X. Zhang, J. L. Freschl, and J. M. Schopf, "A performance study of monitoring and information services for distributed systems," in *12th IEEE International Symposium on High Performance Distributed Computing, 2003. Proceedings*, pp. 270–281, 2003.

[7] A. J. Wilson, R. Byrom, L. A. Cornwall, M. S. Craig, A. Djaoui, S. M. Fisher, S. Hicks, R. P. Middleton, J. A. Walk, A. Cooke, and Others, "Information and monitoring services within a grid environment," in *Computing in High Energy and Nuclear Physics (CHEP)*, Citeseer, 2004.

[8] S. Fisher, "Relational model for information and monitoring," in *Global Grid Forum, GWD-Perf-7*, vol. 1, 2001.

DRAFT

[9] R. Byrom, B. Coghlan, A. Cooke, R. Cordenonsi, L. Cornwall, M. Craig, A. Djaoui, S. Fisher, A. Gray, S. Hicks, and Others, "Production services for information and monitoring in the grid," *AHM2004, Nottingham, UK*, 2005.

[10] D. Bonacorsi, D. Colling, L. Field, S. Fisher, C. Grandi, P. Hobson, P. Kyberd, B. MacEvoy, J. Nebrensky, H. Tallini, and S. Traylen, "Scalability tests of R-GMA-based grid job monitoring system for CMS Monte Carlo data production," *IEEE Transactions on Nuclear Science*, vol. 51, pp. 3026–3029, December 2004.

[11] R. Byrom, D. Colling, S. Fisher, C. Grandi, P. Hobson, P. Kyberd, B. MacEvoy, J. Nebrensky, and S. Traylen, *Performance of R-GMA for Monitoring Grid Jobs for CMS Data Production*. IEEE, 2005.

[12] G. Von Laszewski and I. Foster, "Usage of LDAP in Globus," *Mathematics and Computer Science Division, Argonne National Laboratory*, 1998.

[13] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on*, 2001.

[14] X. Zhang and J. M. Schopf, "Performance analysis of the globus toolkit monitoring and discovery service, mds2," *Arxiv preprint cs/0407062*, 2004.

[15] Y. Liu and S. Gao, "Wsrf-based distributed visualization," in *Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on*, pp. 615 –619, May 2009.

[16] Luciano Gaido, "DSA1.2.2: Assessment of production service status," 2010.

[17] M. Gerndt, R. Wismuller, Z. Balaton, G. Gombás, P. Kacsuk, Z. Németh, N. Podhorszki, H. L. Truong, T. Fahringer, M. Bubak, and Others, "Performance Tools for the Grid: State of the Art and Future, APART White Paper," 2004.

[18] S. Zanikolas, *Importance-Aware Monitoring for Large-Scale Grid Information Services*. PhD thesis, the University of Manchester, 2007.

DRAFT

[19] S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone, and M. C. Vistoli, "Gridice: a monitoring service for grid systems," *Future Gener. Comput. Syst.*, vol. 21, pp. 559–571, April 2005.

[20] G. Tsouloupas and M. Dikaiakos, "Gridbench: a tool for benchmarking grids," in *Grid Computing, 2003. Proceedings. Fourth International Workshop on*, pp. 60 – 67, Nov. 2003.

[21] D. Guo, L. Hu, M. Zhang, and Z. Zhang, "Gcpsensor: a cpu performance tool for grid environments," in *Quality Software, 2005. (QSIC 2005). Fifth International Conference on*, pp. 273 – 278, 2005.

[22] S. Browne, J. Dongarra, N. Garner, G. Ho, and P. Mucci, "A portable programming interface for performance evaluation on modern processors," *International Journal of High Performance Computing Applications*, vol. 14, no. 3, p. 189, 2000.

[23] E. Imamagic and D. Dobrenic, "Grid infrastructure monitoring system based on Nagios," in *Proceedings of the 2007 workshop on Grid monitoring*, p. 28, ACM, 2007.

[24] P. R. Hobson, E. Frizziero, C. Huang, M. R. Irving, T. Kalganova, P. Kyberd, F. Lelli, A. Petrucci, R. Pugliese, G. Taylor, and R. Taylor, *Grid Computing Technologies for Renewable Electricity Generator Monitoring and Control*. IEEE, June 2007.

[25] C. Huang, P. R. Hobson, G. A. Taylor, and P. Kyberd, *A Study of Publish/Subscribe Systems for Real-Time Grid Monitoring*. IEEE, March 2007.

[26] S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke, "A directory service for configuring high-performance distributed computations," in *High Performance Distributed Computing, 1997. Proceedings. The Sixth IEEE International Symposium on*, pp. 365 –375, Aug. 1997.

[27] K. Goel and I. Chana, "Agent-Based Resource Monitoring for Grid Environment," *ICON 2008*, p. 62, 2008.

DRAFT

[28] J. Huo, L. Liu, L. Liu, Y. Yang, and L. Li, "A Study on Distributed Resource Information Service in Grid System," in *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, vol. 1, pp. 613–618, 2007.

[29] J. Novak, "Nagios SAM Portal for ROCs," 2009.

[30] S. Zanikolas and R. Sakellariou, "A taxonomy of grid monitoring systems," *Future Generation Computer Systems*, vol. 21, no. 1, pp. 163 – 188, 2005.

[31] B. K. Singh, A. Amintabar, A. Aggarwal, R. D. Kent, A. Rahman, F. Mirza, and Z. Rahman, "Secure grid monitoring, a web-based framework," in *Proceedings of the first international conference on Networks for grid applications*, GridNets '07, (ICST, Brussels, Belgium, Belgium), pp. 15:1–15:7, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[32] Z. Balaton, P. Kacsuk, N. Podhorszki, and F. Vajda, "Use cases and the proposed grid monitoring architecture," tech. rep., Tech. Rep. LDS-1/2001, Computer and Automation Research Institute of the Hungarian Academy of Sciences, 2001. http://www. lpds. sztaki. hu/publications/reports/lpds-1-2001. pdf, 2001.

[33] T. Kiss, P. Kacsuk, G. Terstyanszky, and S. Winter, "Workflow level interoperation of grid data resources," in *Cluster Computing and the Grid, 2008. CCGRID '08. 8th IEEE International Symposium on*, pp. 194 –201, May 2008.

DRAFT

School of Engineering & Design

Electronic & Computer Engineering

Industrial Mentor's Report on a MSc Dissertation

Student's name: Theofylaktos Papapanagiotou

Dissertation Title: Grid Monitoring

I confirm / cannot confirm* that the MSc dissertation submitted by the above student truly reflects the respective contributions of the student and others, with whom the student has worked during the MSc project.

∗ delete as appropriate

Comments:

Signed:

Name: Dr.Paul Kyberd

Date:

DRAFT