Fermilab FERMILAB-Pub-04/122-CD July 2004

Configuration Monitoring Tool for Large-scale Distributed Computing

Y. Wu[a], G. Graham[a] X. Lu[b], A. Afaq[a] B. J. Kim[c], and I. Fisk[a] for the CMS collaboration

[a]CD/CMS, MS 234, Fermilab, PO BOX 500, Batavia, IL 60504, USA

[b]Department of Computer Science, 14 MacLean Hall, The University of Iowa, Iowa City, IA 52242, USA

[c]P.O. Box 118440, Department of Physics, University of Florida, Gainesville, FL 32611, USA

The CMS (Compact Muon Solenoid) experiment at the Large Hadron Collider (LHC) at CERN will likely use a grid system to achieve much of its offline processing need. Given the heterogeneous and dynamic nature of grid systems, it is desirable to have in place a configuration monitor. The configuration monitoring tool is built using the Globus toolkit and web services. It consists of an information provider for the Globus MDS, a relational database for keeping track of the current and old configurations, and client interfaces to query and administer the configuration system. The Grid Security Infrastructure (GSI), together with EDG Java Security packages, are used for secure authentication and transparent access to the configuration information across the CMS grid. This work has been prototyped and tested using US-CMS grid resources.

1. Introduction

The offline processing of the CMS experiment will be distributed globally due to the size of the collaboration and the complexity of the computing tasks. Grid technology, which is built to enable large-scale coordinated sharing of heterogeneous and dynamic computing resources, provides excellent tools to satisfy the need of CMS distributed computing. To enable high-performance grid distributed computing, it is very important to have a monitoring and discovery system in place for resource discovery, selection, optimization, job scheduling, and resource status monitoring, etc. [1].

Traditionally, monitoring systems have been focused on system status and performance monitoring. There are a lot of existing tools to monitor the system status and performance of distributed computing systems, for example, Big Brother [2], Ganglia [3], MonaLisa [4], Nagios [5], Netlogger [6], etc. However, during our utilization of shared grids resources in the distributed CMS production and analysis, we found it is very useful to know the configuration information on the remote grid resources, for example, a job generator need to know a list of configurations on a computer resources (CMS software locations, scratch area and paths, etc) before generating and submitting jobs; users need to know the basic resource configurations (like gatekeeper host name, port number and available job managers) before utilizing the resources. It is critical to have a tool to address these issues.

In this paper, we describe some of our prototyped development efforts on a configuration-monitoring tool. After briefly describing the design consideration, we describe the architecture and various components of the configuration monitoring tool. Later we describe the current status and initial application of the configuration monitoring tool in the USCMS testbed, and its future direction.

2. Architecture

2.1. Design consideration

The configuration monitoring tool is designed to provide information services for large-scale distributed computing resources, especially to fit the

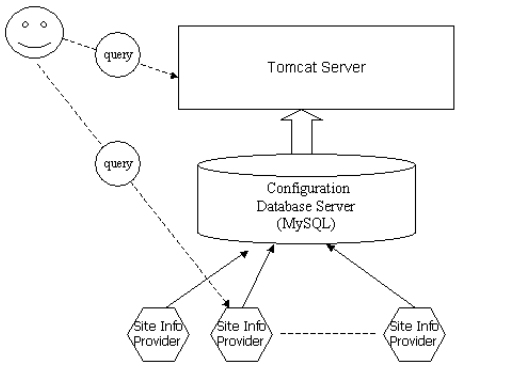

Figure 1. A prototyped architecture of the configuration monitoring system

need of the CMS production and analysis. It utilizes the existing Globus MDS [7], Tomcat web service [8] and a relational database system to provide the needed configuration information for CMS applications. It strives to offer flexibility on what configuration information to provide and archives the history of configuration information for future utilization. It also utilizes the authorization mechanism on authorizing which user, group and/or role to access what kind of information.

2.2. Components

A layered structure is to design the whole architecture. This has the advantage of replacing different layers without interfering with other layers. A prototyped architecture is shown in Figure 1. In this architecture, the whole system is divided into the following layers:

Site info provider layer: This layer is responsible for collecting and publishing site configuration information at each resource. The module in this layer is distributed at each computing resource. Currently, it is implemented as an information provider in the Globus MDS with the standard GLUE schema (Grid Laboratory Uniform Environment) [9]. The information provider can collect the service configuration information from text files, command lines or other plug-in scripts written by users. Then, the collected information is published into MDS through the standard MDS creating new information provider mechanism [7].

Configuration database server layer: This layer tracks the hosts and services to be monitored, and stores all the collected configuration information. This layer consists of a relational database server and cron job scripts to collect and update the information in the database. It is one of the core components of the whole configuration monitoring architecture. It collects the site configuration information by querying the site information providers, tracks the availability of the services, and archives the old configuration information.

Currently, we are using MySQL, an open source product, as the relational database server. It can fit our current need when the number of hosts and services to be monitored is relatively small.

Tomcat service layer: Tomcat servlet technology is used to provide a web interface for users to browse/query available hosts, services and their configurations. It also provides the mechanism to authorize an administrator to perform the administration tasks securely, like updating the resources/services to be monitored, modifying the availability of a service.

User interface layer: The user interface layer provides a set of scripts for users to query the configuration database and/or query the individual MDS server to get the needed information directly. They are provided for the convenience of users.

2.3. Security consideration

Keeping the configuration information only accessible by authorized users is on the top of our priorities when designing the system. We paid special attention on the security and utilized the latest existing grid security technology to accomplish this.

For the site information provider, its security is handled by Globus GSI as it is part of the Globus MDS.

For the Tomcat service layer, strong authentication and authorization is enforced using the

digital certificates: on the server side, all the web pages and servlets are put behind an authorization servlet filter developed by EDG [10]. The authorization filter examines every incoming request and tries to extract the client certificate from the request. It then passes the extracted client DN to an authorization manager for verification. If the authorization manager can verify the client DN, it gives permission for the user to view the web info; otherwise, it just terminates the request and informs the user of authorisation failure. There are several possible ways to configure the authorization manager regarding which user, group and/or role can access the configuration information. Currently, the authorization manager is configured to examine a standard grid-mapfile, a text file listing all users DN and their local mapping accounts, to see if a request user DN can be found in it.

3. Current status and utilization of the configuration monitoring tool

We have finished the initial prototype development on major components of the configuration monitoring tool and deployed it on the USCMS grid resources. We have tested the configuration monitoring tool in the USCMS distributed Monte Carlo MOP production [11]. In the new setting, instead of passing the site configuration parameters by e-mails between MOP users and site grid system administrators, parameters can be pulled from sites directly using the configuration monitoring tool. This has the big advantage for both MOP users and local grid site administrators: less harassment for site administrators and fewer errors for MOP users when there is a change of the site configuration.

We are also monitoring a set of critical services deployed on the USCMS testbed. Users and grid administrators can check the service status and configurations. Other applications, such as US-MOP Portal [12] under development at University of Florida, also show interest in using the information published by the Configuration Monitoring tool in their applications.

4. Conclusions and future work

Knowing the site configuration is very important for users to utilize the grid resources. A configuration monitoring tool has been prototyped on top of the Globus technology and Tomcat web service to allow users/sites to publish the site configuration info, archive the collected info and query them. The Grid Security Infrastructure, together with EDG Java Security packages, is used for secure authentication and transparent access to the configuration information across the USCMS grid. The configuration monitoring tool has been installed on the USCMS Grid testbeds and tested in the USCMS grid production jobs.

We plan to focus on providing web services for the configuration monitoring tool in the near future. Users will be able to query the configuration information in the database through web services. And system administrators will be able to update the critical site configuration information through it. Further development is also needed to collect configuration information from other monitoring tools, like MonaLisa, Ganglia, etc. And it is also nice to have a web interface to view archived history data.

Acknowledgement

We are grateful to many colleagues for the suggestions and discussions on a configuration monitoring system. We want to thank everybody's help in the USCMS testbed, especially Richard J. Cavanaugh, Maarten Ballintijn, James Letts, Natalia Ratnikova, Jorge L. Rodriguez, Suresh Singh, Alan De Smet, Shaowen Wang for insightful suggestions, discussions and deployments. We also thank the reviewers whose critical review is sincerely appreciated.

REFERENCES

1. K. Czajkowski, S. Fitzgerald, I. Foster and C. Kesselman: Grid Information Sharing Services for Distributed Resource Sharing. In Proc. of the 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, 2001.
2. Big Brother System and Network Monitor.

Available at: http://www.bb4.org.
3. Ganglia distributed monitoring and execution system. Available at: http://ganglia.sourceforge.net.
4. MonaLISA: Monitoring Agents in A Large Integrated Services Architecture. Available at: http://monalisa.cacr.caltech.edu.
5. Nagios monitoring system. Available at: http://www.nagios.org.
6. Netlogger Toolkit. Available at: http://www-didc.lbl.gov/NetLogger.
7. Globus Monitoring and Discovery Service (MDS). Available at: http://www.globus.org/mds/mds2.
8. Tomcat web service. Available at: http://jakarta.apache.org/tomcat.
9. GLUE schema: Grid Laboratory Uniform Environment. Available at: http://www.cnaf.infn.it/~sergio/datatag/glue.
10. EDG WP2 Security Task. Available at: http://edg-wp2.web.cern.ch/edg-wp2/security.
11. MOP: Monte Carlo Production system. Available at: http://grid.fnal.gov/test_beds/MOP_install.htm.
12. USMOP Portal. Available at: https://gdsuf.phys.ufl.edu:8443/gridmon/gridserv/gridserv/dpeclient.