THE EVOLVING ROLE OF MONITORING IN A LARGE-SCALE COMPUTING FACILITY

A. Chan*, B. Gibbard, C. Hollowell, J. Hover, Z. Liu, R. Petkus,
O. Rind, J. Smith, T. Throwe, A. Withers, T. Wlodek, X. Zhao
Brookhaven National Laboratory, Upton, NY 11973, USA

Abstract

Monitoring a large-scale computing facility is evolving from a passive to a more active role in the LHC era, from monitoring the health, availability and performance of the facility to taking a more active and automated role in restoring availability, updating software and becoming a meta-scheduler for batch systems. This talk will discuss the experiences of the RHIC and ATLAS U.S. Tier 1 Computing Facility at Brookhaven National Lab in evaluating different monitoring software packages and how monitoring is being used to improve efficiency and to integrate the facility with the Grid environment. A monitoring model to link geographically dispersed, regional computer facilities which can be used to improve efficiency and throughput will be presented as well.

INTRODUCTION

The RHIC Computing Facility (RCF) is a large scale data processing facility at Brookhaven National Laboratory (BNL) for the Relativistic Heavy Ion Collider (RHIC), a collider dedicated to high-energy nuclear physics experiments. It provides for the computational needs of the RHIC experiments, including batch, mail, printing and data storage. As the U.S. Tier 1 Center for ATLAS computing, the ATLAS Computing Facility (ACF) provides for the computational needs of the U.S. collaborators in ATLAS. The rapid growth of the RCF/ACF in the past few years (see Fig. 1 and Fig. 2) has resulted in a 14 TFLOPS Linux Farm (over 4000 processors), a 1 PB distributed and network-attached disk storage system, a 7 PB robotic tape storage system, over 50 Intel/UNIX-based servers for general computing support, over 3000 active Gigabit-capable network ports distributed over several subnets, and increasing support for Grid-like technologies. The size and mission of the facility is transforming the RCF/ACF from a local into a globally available computing resource, with growing design and operational complexity that requires more sophisticated management & monitoring software for better operational oversight.

MONITORING – THE EARLY YEARS

The concept of remote monitoring of servers and services is a relatively new concept. Up to the mid-90's, only simple open-source or proprietary monitoring software were available. Only a handful of large companies or scientific institutions could afford high-priced, proprietary UNIX clusters so scalable, open-source monitoring software was not deemed necessary. By the late 90's, however, increasingly affordable clusters made up of commodity hardware became more common, creating a need for more scalable monitoring software for easier management and operation. Early open-source packages evaluated by the RCF/ACF were **Mon** [1] and **Spong** [2], both highly-configurable, general-purpose alert management tools used for monitoring service availability and triggering alerts upon detecting failure. The Mon package was designed to be extensible in the sense that it supports arbitrary monitoring facilities and alert methods via a common interface, all of which are easily implemented with programs in C, Perl, shell, SNMP traps and special Mon traps. An early implementation of Mon at the RCF/ACF is shown in Figure 3.

Spong is a simple system-monitoring package written in

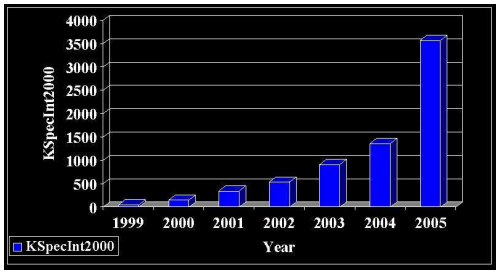

Figure 1: The increase in the RCF/ACF Linux Farm processing capacity.

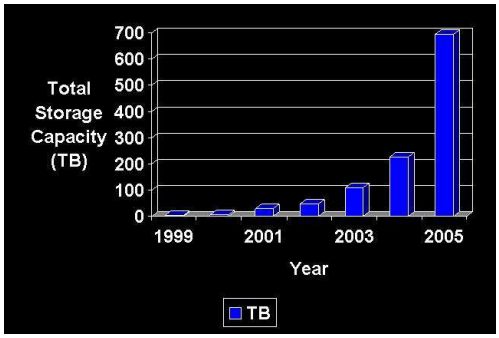

Figure 2: The growth of the RCF/ACF distributed storage capacity.

* awchan@bnl.gov

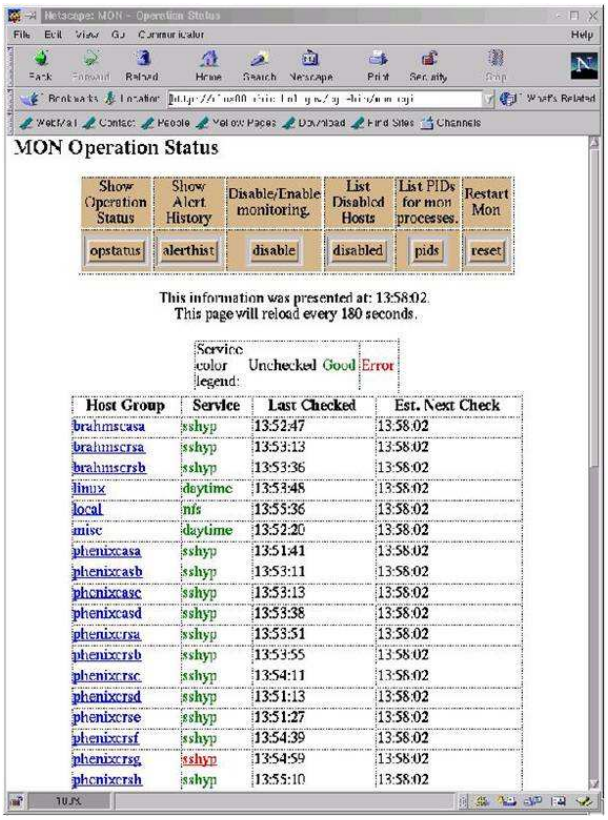

Figure 3: A screenshot of Mon at the RCF/ACF.

Perl. It features client based monitoring, monitoring of network services, results displayed via the Web or console, history of problems, and flexible messaging when problems occur. Both packages worked well when the number of monitored systems numbered in the hundreds. However, server-side load issues began to challenge the scalability of the tools once the systems approached the thousands. The simultaneous tasks of collecting information, organizing and displaying it became too complex for the tools to handle.

THE SEARCH FOR SCALABILITY

In 2000, the RCF/ACF began investigating replacing Mon with a more robust package. A pilot project by staff members to write a database-based, scalable monitoring tool with a graphical, Web interface evolved as a replacement for Mon (see Fig. 4).

The introduction of a MySQL database back-end to the monitoring tool allows one to keep historical records, which is useful for diagnostic purposes and provides a scalable and organized way to separate the tasks of collecting, organizing and displaying the monitored information. The RCF/ACF monitoring tool performed the tasks asynchronously, thereby diminishing the server-side load. To make monitoring even more scalable, multiple read/write threads from the client servers directly to MySQL were employed to optimize database access. As a result, this tool

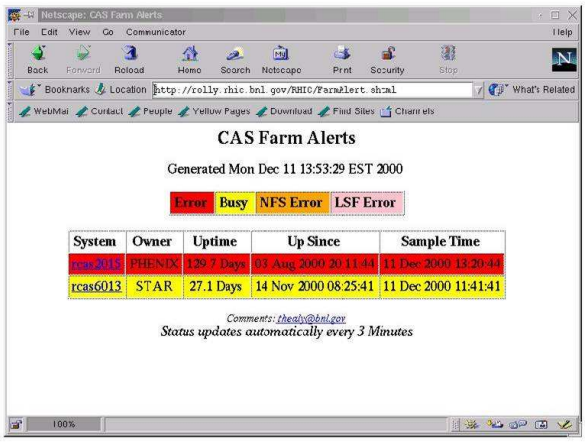

Figure 4: The graphical Web interface of the RCF/ACF monitoring tool.

has scaled well up to $\sim$ 1,300 systems.

As with Mon, the RCF/ACF monitoring tool includes client software that allows for service monitoring with custom scripts, although both are still limited to service (NFS, AFS, batch) monitoring. Performance monitoring and monitoring of remote sites were not goals in the limited scope of the project, thus making the RCF/ACF monitoring tool unsuitable for the distributed computing environment.

MONITORING & DISTRIBUTED COMPUTING

Distributed computing in the LHC era has brought new requirements, including monitoring of performance metrics (which was absent from the RCF/ACF monitoring tool), monitoring of remote resources and mirroring of database information at remote sites.

In 2003, the RCF/ACF began deploying **Ganglia** [3] to address the need for performance monitoring in a distributed computing environment. Ganglia is an open-source project and a scalable distributed monitoring system for large computing clusters. It is based on a hierarchical design for federations of clusters, and it uses XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization. The implementation is robust and requires low-overhead per node. It has been used to link clusters around the world and can scale to handle clusters with thousands of systems. The graphical, Web-based interface for Ganglia is shown in Figure 5.

To complement Ganglia's performance monitoring, the RCF/ACF in 2005 has also deployed **Nagios** [4] as a scalable and better instrumented replacement for the RCF/ACF service monitoring tool. Nagios is a host and service monitor optimized to run under the Linux operating system, but it operates under other UNIX variants as well. A monitoring daemon on the client host runs intermittent checks on the host and services specified by externally-supplied "plu-

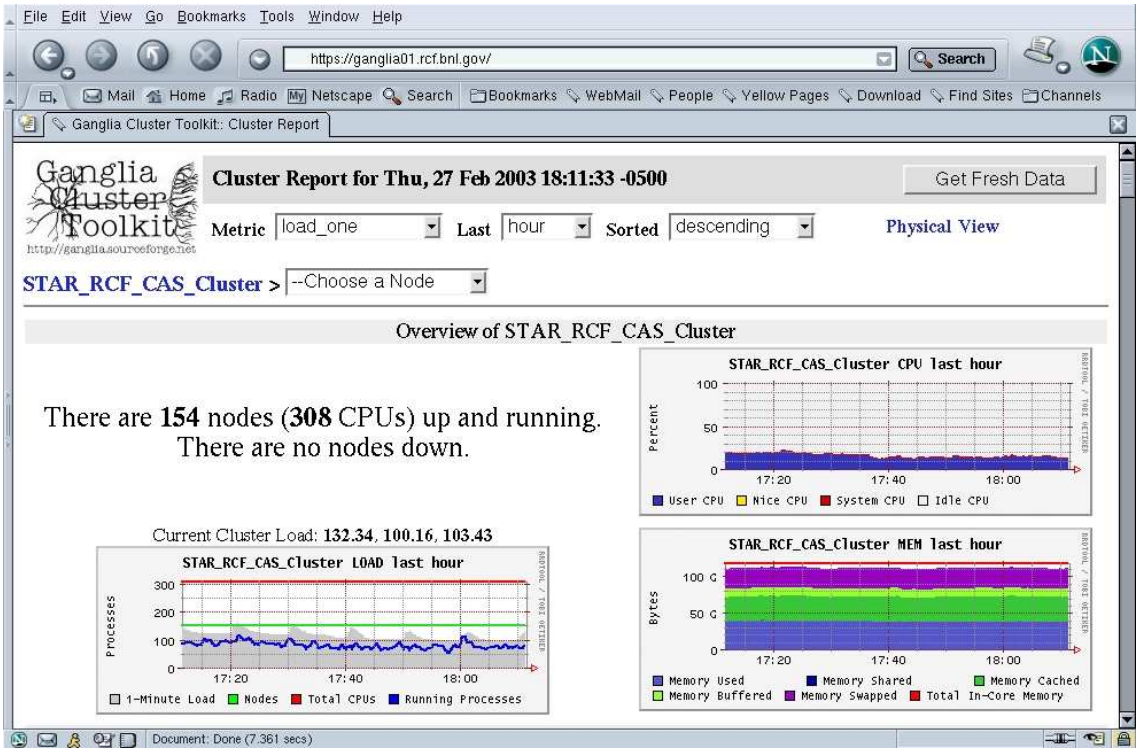

Figure 5: The Ganglia monitoring tool.

gins" which return status information. The daemon can be configured to take specific actions when problems are encountered (email notification, system reboot, etc). Current status information, historical logs, and reports can all be accessed via a Web interface (shown in Figure 6). Nagios has been deployed on over 2,000 systems as of early 2006.

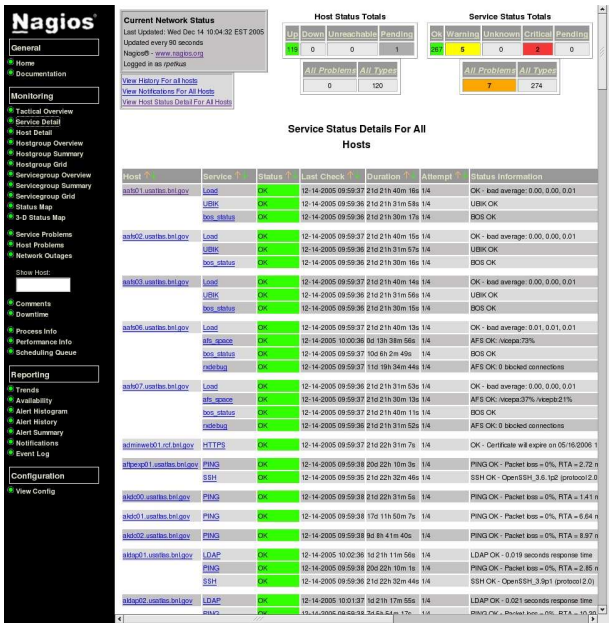

Figure 6: The Nagios graphical monitoring interface.

Another promising monitoring tool being evaluated at the RCF/ACF is **MonALISA** [5], a distributed monitoring service system using JINI/JAVA and WSDL/SOAP technologies. MonALISA is designed to provide monitoring information from large distributed systems to a set of "higher level services", as part of a loosely-coupled service architectural model to perform effective resource utilization in large, heterogeneous distributed centers. MonALISA has been deployed at the RCF/ACF as part of the Ultralight [6] project to promote network infrastructure as a dynamic, configurable and monitored resource. A graphical view of the MonALISA tool is show in Figure 7.

Perhaps the biggest appeal of MonALISA is that its framework can integrate other existing monitoring tools to collect parameters describing computational nodes, applications and network performance. MonALISA potentially provides the common fabric in which geographically dispersed, heterogeneous computer facilities can monitor each other, help increase efficiency and improve resource utilization in a distributed computing environment.

SUMMARY

Table 1 summarizes the evolution of monitoring tools at the RCF/ACF over the last few years.

The transformation of the RCF/ACF from a local to a globally available resource in the last few years is a significant driving force behind the evolving role of monitoring at our facility. The increasing relevance of the distributed computing model, and the rising importance of scalable, well-instrumented monitoring tools that provide real time

Figure 7: The Web interface of the MonALISA monitoring tool.

information are important considerations for the efficient operation and management of large scale computing facilities.

Table 1: Monitoring Tool Evolution

Old Package	New Package	Date
Mon	Mon	1997
	RCF/ACF	2000
RCF/ACF	Ganglia	2003
	Nagios	2005
	MonALISA	2005

ACKNOWLEDGEMENTS

The RCF/ACF would like to thank BNL's Physics Department, Information Technology Division (ITD) and the United States Department of Energy (DOE) for their support.

REFERENCES

[1] http://www.kernel.org/software/mon/

[2] http://sourceforge.net/projects/spong

[3] http://ganglia.sourceforge.net/

[4] http://www.nagios.org/

[5] http://monalisa2.cern.ch/

[6] http://ultralight.caltech.edu/web-site/ultralight/html/index.html