Survey of execution monitoring tools for computer clusters

Espen S. Johnsen
Otto J. Anshus
John Markus Bjørndalen
Lars Ailo Bongo
Department of Computer Science, University of Tromsø

September 29, 2003

1 Introduction

Effective management and utilization of large computer clusters requires suitable management tools. This includes tools for monitoring execution, both in real-time (online) and analysis of traces after execution (off-line). Execution monitoring also involves measuring usage of system resources, such as CPU, memory and network. As cluster monitoring is tightly related to administration, some of the tools included in this survey is actually general cluster administration tools.

2 Tools in use by NOTUR today

This section gives an overview of the tools used for execution monitoring, by NOTUR participants today.

2.1 Ganglia – distributed monitoring and execution system

Ganglia[8] is an open-source project originating from the Millennium Project at *University of California, Berkeley.* It is a scalable distributed real-time monitoring tool for high performance clusters. Every monitored node is running an instance of a daemon call `gmond` which is responsible for monitoring and broadcast changes of various metrics, listen for state changes from other nodes and answer state queries. Each node keeps the state of all other nodes, therefor it is only necessary to query a single node to get the state of the entire cluster. The user may also extend the built-in list of metrics to be monitored and broadcasted by adding custom metrics.

The Ganglia Meta Daemon (`gmetad`) allows different monitored clusters to be connected to each other through unicast links. This daemon also act as the back-end for the Web interface, which provides a graphical view of collected data.

Vendor:	Ganglia Development Team at SourceForge
Platforms:	Linux
License:	BSD
Used by:	UiTø, “over 500 clusters around the world”

2.2 LoadLeveler

LoadLeveler is a batch oriented tool for managing serial and parallel jobs running on clusters. The user submits jobs to be executed and the scheduler decides when and one which nodes the job is to be executed on. *LoadLeveler* contains both an extensible GUI and command line tools for monitoring job execution.

Several sophisticated third party monitoring applications is also available:

- NERD – displays job execution status as web pages
- LoadView – java applet
- LLAMA – advanced monitor application

Vendor:	IBM
Platforms:	Linux/Unix
License:	commercial
Used by:	University of Bergen

2.3 Portable Batch System

PBS Pro and OpenPBS are extensible systems for workload managment in clusters. As with *LoadLeverer*, the focus seems to be on batch queuing and job management, but a command line tool (`qstat`) is provided to query the system about various execution parameters.

Vendor:	Altair Engineering, Inc.
Platforms:	Linux/Unix
License:	commercial
Used by:	University of Tromsø and University of Oslo

3 Other tools available

This section gives an overview of some of the other available tools for execution monitoring of clusters. It is by no means a complete list of all available tools, but should contain at least the most common. As monitoring and cluster administration is highly related, most of these tools also include mechanisms for administration.

3.1 openMosix/openMosixview

openMosix[14] is a set of extensions to the Linux kernel to support adaptive load-balancing and transparent process migration in a single-image clusters. The idea behind *openMosix* is that applications not written especially for parallel execution should take advantage of clusters technology. *openMosix* is an open source fork of the original *MOSIX*[10] project.

openMosixview[15] is a monitoring and administration tool suit for *openMosix* clusters. The package contains tools for monitoring and logging resource usage, analyzing collected data, process management, controlling load-balancing and monitoring process migration.

Vendor:	SourceForge project
Platforms:	Linux
License:	GPL

3.2 NetLogger

NetLogger[12] provides detailed end-to-end application and system level monitoring of high performance distributed systems. The package contains tools for instrumenting different parts of the system to be monitored and for visualization of collected data. Both applications and components of the operating system may be instrumented to do time-stamping and logging of interesting events. *NetLogger* is designed to help identify bottlenecks, assist in performance tuning and measuring of network performance.

The main components of *Netlogger* consist of an API and a library for application level instrumentation, tools for managing log files, tools for host and network monitoring and tools for visualization and analyzing of collected data.

NetLogger is claimed to be extremely useful for debugging and tuning of distributed applications.

Vendor:	Lawrence Berkeley National Laboratory
Platforms:	Linux, Solaris
License:	BSD

3.3 Vampir

Vampir[20] is a well known commercial tool for performance visualization and analysis of MPI programs. Instrumentation code is hook into the application to be monitored through the MPI profiling interface simply by linking in a library called *Vampirtrace*. Only if application specific events is to be recorded, is it necessary to do any modifications the application.

With *Vampir* it is possible to visualize a large variety of aspects about the runtime behavior of the application, such as time-line view of events and communication.

Vendor:	Pallas GmbH
Platforms:	any standard-conforming MPI implementation on a large number of Linux/Unix platforms
License:	proprietary, free download of demo
Used by:	misc Govt. and Research Labs, Universities, hardware vendors and ISVs

3.4 The Network Weather Service

NWS[13] is a distributed system that monitors various resources (eg. network and CPU) and periodically broadcasts a short-term forecast of available resources.

The system currently contains sensors for TCP/IP performance, CPU usage and available memory. New sensors may be added by the user through a configuration interface.

The forecasting uses various statistical methods on time series of gathered data, to estimate the future availability of a given resource. The methods initially used by NWS is mean-based methods, median-based methods and autoregressive methods. The method best suited for a resource is dynamically selected based on the accuracy of previous forecasts.

Platforms:	Linux/Unix
License:	BSD

3.5 EnFuzion

EnFuzion[7] is primarily a tool to automatically spread a large number of smaller jobs across a cluster. Several methods is provided for monitoring execution – log files, web based monitoring, command line monitoring and monitoring from custom programs.

Vendor:	Axceleon Inc
Platforms:	Windows, Linux, Solaris, AIX, HP-UX, Irix, Tru64
License:	commercial and evaluation license

3.6 Cluster Systems Management

CSM is a distributed system for managing and monitoring clusters running on IBM hardware. Supported features includes remote software installation, node group management, configuration file management, hardware/software monitoring and diagnostic probes. CSM works in conjunction with Reliable Scalable Cluster Technology (RSCT) which is a set of software components to form cluster environments with Linux.

Vendor:	IBM
Platforms:	Red Hat Linux
License:	commercial

3.7 Dogsled

Dogsled[4] is open source tool for monitoring and managing large Linux clusters from a central point. Monitored parameters are displayed with GUI or text interfaces. Alarms and shutdown condition can also be set. Administration and monitoring may be done out-of-band through serial interfaces.

One of the design goals of *Dogsled* is to make a minimal system, where only the most essential information is normally present to the user. But at the same time be robust enough to provide all information that is required when necessary. The user may define contexts (groups of machines) to be managed as a single entity. *Cluster* is a predefined context containing all machines in the cluster. New nodes and nodes going down are automatically detected.

Vendor:	Paralogic, Inc
Platforms:	linux
License:	GPL

3.8 C3 and M3C

The *Cluster Command and Control* (C3) [2] and *Monitoring and Managing Multiple Clusters* (M3C)[1] are tools originally developed by *Oak Ridge National Laboratory* to manage their own clusters.

The C3 package contains a number of command line tools for administrating large clusters. This includes tools to distribute software updates, manage processes and shuting down the system.

M3C is a tool built on C3, with a web based GUI which allows the user to view and administrate a cluster as a single entity. The package also contains tools for monitor real-time parameters such as CPU load and for cluster reservation.

Vendor:	Computer Science and Math Division, Oak Ridge National Laboratory
Platforms:	linux/unix
License:	free software

3.9 Paradyn

Paradyn[17] is a tool for performance monitoring and analysis of parallel programs using dynamic instrumentation. Dynamic instrumentation is a technique where the decision of which performance data to collect is made based on dynamic control. The user is either assisted in deciding which bottlenecks to search for, or the system can automatically detect performance bottlenecks.

Vendor:	University of Wisconsin
Platforms:	Solaris, AIX, Linux, Windows 2000
License:	free for research use, no redistribution allowed

3.10 PARMON

PARMON[3] is a commercial[18] cluster monitoring tool similar to *Ganglia*. Each node to be monitored is running a data collecting daemon, and a GUI client is used to gather an present monitored data from the nodes nodes running the daemon. Some of the features in *PARAMON* are:

- support for monitoring at node, group and cluster level
- monitoring of processes, system log, kernel, CPU, memory, disk and network
- listing of system information and configuration
- Web interface

Vendor:	Centre for Development of Advanced Computing (C-DAC), India
Platforms:	AIX, Solaris, Linux
License:	Proprietary

3.11 XMPI and Upshot/ Jumpshot

XMPI [23] is a performance monitoring, debugging, and visualization tool for MPI programs. It was originally developed at the Ohio Supercomputer Center and is currently being developed by Open Systems Laboratory at Indiana University, the laboratory developing LAM/MPI. Online and post-mortem analysis are supported.

Upshot, nupshot, and jumpshot [9] are included with the MPICH implementation. The MPI profiling interface is used.

Platforms:	Many
License:	Free

3.12 PAPI and VTune

PAPI [16] provides an interface for accessing the performance counters on most modern CPUs. The perfometer tool, developed as part of the PAPI project, can be used to visualize the information gathered from the counters.

The Intel VTune Performance Analyzer [21] can also be used to monitor the performance counters. In addition to sampling, the tool also provides call graph monitoring. The gathered data can be visualized, and the tool also provides code tuning adivse. VTune is a Windows tool with limited Linux support. Also a commercial license is required.

4 Resarch on execution monitoring tools

This section gives an overview of some of the research related to cluster monitoring, which have tanken place in recent years.

4.1 Mirror Object Model

The Mirror Objet Model[5] is a approach to program monitoring and steering based on higher-level object abstraction. The idea is that application-level entities are treated as objects with methods and state and that an interactive system (monitoring and steering system) extends these objects with addition methods and state variables. These extensions is not added to application it selves, but contained in the interactive system as mirror objects. Communication between the application and the interactive system is done through remote method invocation.

An experimental implementation of the model using CORBA-like objects, called *Mirror Object Steering System*[11] is also described.

4.2 OCM – a monitoring system for interoperable tools

The paper[22] addresses the problems of not having a standard interface or protocol for tools supporting distributed applications. It describes an protocol called *On-Line Monitoring Interface Specifications* for a universal interface between tools and distributed applications. An reference implementation of an OMIS compliant monitoring system is also described.

4.3 An Agent-Based Architecture for Tuning Parallel and Distributed Applications Performance

In[6] an agent based model for execution monitoring and real-time tuning of performance parameters is described. The agents are arranged in hierarchical order, where communication takes place only between agents on a different layers. All agents share the same architecture but agents on different layer have different domain of control. Data are collected mainly from the running processes (by the leaf agents) and propagated up through the hierarchy and eventually reaching the master. Each agent may do it's own tuning decisions, which is propagated down the hierarchy and finally to the process. User interaction is performed through the master agent.

4.4 JAMM

JAMM[19] is a monitoring sensor management system for Grid environments. Focus is on automating the execution of monitoring sensors and the collection of data. In JAMM sensors generate events that can be collected, filtered and summarized by consumers using event gateways.

References

[1] Brim, M., Geist, A., Luethke, B., Schwidder, J., and Scott, S. L. M3c: Managing and monitoring multiple clusters. In *Proceedings of the 1st International Symposium on Cluster Computing and the Grid* (2001), IEEE Computer Society, p. 386.

[2] Brim, M., R. Flanery, A. G., Luethke, B., and Scott, S. Cluster command & control (c3) tool suite.

[3] Buyya, R. Parmon: A portable and scalable monitoring system for clusters. *International Journal on Software: Practice & Experience (SPE)* (Jun 2000).

[4] Dogsled. http://www.plogic.com/dogsled/index.html.

[5] Eisenhauer, G., and Schwan, K. An object-based infrastructure for program monitoring and steering. In *Proceedings of the SIGMETRICS symposium on Parallel and distributed tools* (1998), ACM Press, pp. 10–20.

[6] Elfayoumy, S. A., and Graham, J. H. An agent-based architecture for tuning parallel and distributed applications performance. http://www.crhc.uiuc.edu/ steve/wcbc00/wcbc-00-elg.pdf.

[7] EnFuzion. http://www.axceleon.com.

[8] Ganglia. http://ganglia.sourceforge.net.

[9] Gropp, W., and Lusk, E. Installation Guide to MPICH, a Portable Implementation of MPI, Version 1.2.4.

[10] MOSIX. http://www.mosix.org.

[11] Mirror Object Steering System. http://www.cc.gatech.edu/systems/projects/MOSS.

[12] Netlogger. http://www-didc.lbl.gov/NetLogger.

[13] The network weather service.
http://nws.cs.ucsb.edu.

[14] openMosix. http://openmosix.sourceforge.net.

[15] openMosixview.
http://http://www.openmosixview.com.

[16] PAPI. http://icl.cs.utk.edu/projects/papi/.

[17] Paradyn. http://www.cs.wisc.edu/paradyn/.

[18] PARMON.
http://www.cdacindia.com/html/ssdgblr/parmon.asp.

[19] Tierney, B., Crowley, B., Gunter, D., Lee, J., and Thompson, M. A monitoring sensor management system for grid environments. *Cluster Computing 4*, 1 (2001), 19–28.

[20] Vampir.
http://www.pallas.com/e/products/vampir.

[21] Vtune.
http://www.intel.com/software/products/vtune/vpa/.

[22] Wismüller, R., Trinitis, J., and Ludwig, T. Ocm – a monitoring system for interoperable tools. In *Proceedings of the SIGMETRICS symposium on Parallel and distributed tools* (1998), ACM Press, pp. 1–9.

[23] XMPI.
http://www.lam-mpi.org/software/xmpi/.