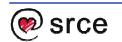
Monitoring systems: Concepts and tools

Zdenko Škiljan Branimir Radić Department of Computer Systems, University Computing Centre, Croatia {zskiljan, bradic}@srce.hr

Zagreb, 4. September 2004



Introduction

Review of:

- Single system oriented monitoring facilities
- Complex tools for monitoring distributed systems (Cluster Monitoring Tools).

Guiding thought:

Systematical supervision of every computer system is necessary on account of recognition of critical circumstances that need

- troubleshooting,
- system/application tuning or, in the end,
 - system upgrade.



Single system oriented tools

Three groups of monitoring facilities are explored:

- log files, /proc pseudo file system and basic system commands as building blocks of advanced monitoring systems.
 - Selected monitoring tools from different areas of interest
- Visualization tools



Log files, /proc pseudo file system and basic system commands

• Log messages:

the most valuable sources of information for system administrators, on UNIX-like systems handled by syslog daemon

- basic system commands the very first help to a system administrator
- /proc pseudo file system –

interface to kernel data structures that provides a reasonably complete set of local monitoring data, including system load, memory utilization and per-process system resource utilization



Selected monitoring tools

- Security related tools
 - System Log Anomaly Monitors
 - Logcheck, Swatch
 - File integrity checkers
 - Tripwire, Aide, AIDE Repository Management Suite
 - IDS (Snort, Portscan)
- System performance monitors
 - SNMP, Orcallator/Procallator, Orca Services
- Service activity monitors
 - MON monitoring daemon, BigBrother



Visualization tools

- Very important role in monitoring systems has data visualization.
- The most popular tools used for that purpose are RRDTool and MRTG
- Perl GD::Graph is a perl5 module that
 - overcome RRDTools inability to create dynamic databases,
 - has poor graphing capabilities comparing to RRDTool, and
 - requires programming effort to create graphs



Cluster Monitoring Tools

 Cluster concept demands employment of monitored data for efficient job distribution

Monitoring function is inherent to JMS



Cluster Monitoring Tools

- We'll review cluster monitoring systems with consideration of
 - JMS built-in monitoring and
 - self-contained cluster monitoring systems.



JMS built-in Monitoring

Role of a JMS is to distribute jobs within a cluster

JMS consist typically of three components:

- Queuing Module,
- Scheduling Module and
- Resource Manager.



Self-contained Cluster Monitoring Systems

Self-contained Cluster Monitoring Systems typically consist of three major entities:

- daemons that reside on cluster nodes
- server that collects cluster state information from nodes
- GUI-based front-end, which provides system activity visualization



Independent Cluster Monitoring Systems

The most prominent cluster monitoring systems are **Ganglia**, **Supermon** and Hawkeye.





Ganglia includes following components:

- Gmond local monitoring system
- Gmeta wide-area monitoring system.
- Ganglia web front-end

Gmond operates on cluster level and uses UDP multicast to exchange data within a cluster





The Supermon is another distributed monitoring system which consists of three different components:

- A loadable kernel module providing data (through /proc entries in form of s-expressions)
- A single node data server (Mon) that serves data prepared by the kernel module,
- A data concentrator (Supermon), which composes samples from many nodes into a single data sample through a TCP port.

Advantages: Fast and efficient data-collector.

Disadvantages: New nodes cannot be included in

SICE MONITORING AUTOMATICALLY, POOR DOCUMENTATIONS

Hawkeye

Hawkeyes architecture consist of four major parts:

- Hawkeye pool, (cluster nodes, includes Hawkeye agents and manager)
- Hawkeye module that perform actual node monitoring (It is possible to add custom-made sensors.
- Hawkeye monitoring agent node monitoring daemons that send information to Hawkeye manager, and
- Hawkeye manager, (node information collector)
- It has a poor front-end, it is a system under development.



Other Systems

- NetLogger
- Paradyn
- Falcon
- /dproc



Conclusion

- Different monitoring tools for different purposes.
- Security related monitoring systems are inevitable on every computer system
- Monitoring tools useful, depending on user requirements and limitations as well as system-specific features



Conclusion

When planning cluster monitoring system, one should carefully think about cluster type to be used and its specific features.

