International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 8, August 2013)

PERFORMANCE ANALYSIS OF A GRID MONITORING SYSTEM - GANGLIA

Rakesh Bhatnagar[1], Dr. Jayesh Patel[2]

[1] Asst. Prof. & Coordinator (MCA Department), S. K. Patel Institute of Management & Computer Studies, Gandhinagar-382023, India.
contactbhatnagar@yahoo.com
[2] Associate Prof., Acharya Motibhai Patel Institute of Computer Studies, Ganpat University, Kherva-382711, India.
jayeshpatel_mca@yahoo.com

***Abstract*—Monitoring is critical for providing robust, high performance Grid environment. Without monitoring mechanisms, it will be impossible to determine the status or health of environment. And so it will be difficult to use it effectively & efficiently. Monitoring data can be used for performance analysis, performance diagnosis, performance tuning, fault detection and scheduling.**

Due to the complexity and dynamics of grid, a strong monitoring system is needed which should have features like Grid Monitoring Architecture compliance, Scalability, Resource Performance Analysis and Forecasting, Directory service for events subscription & notification etc. In this paper, we have reviewed the grid monitoring system – 'Ganglia'. 'Ganglia' is a distributive monitoring system for high performance computing systems such as clusters and the Grid. It is based on a hierarchical design targeted at federations of clusters.

This paper discusses the various performance parameters and criteria of the grid monitoring system – 'Ganglia' and provides some useful information as far as Grid Monitoring Architecture is concerned.

***Keywords*— Ganglia Monitoring System, gmond daemon, gmetad daemon, Scalability, CPU overhead, Bandwidth, Memory Overhead**

I. INTRODUCTION

A complex globally distributed system that involves large sets of diverse and geographically distributed components that can be used for a number of applications, is called a Grid environment [1]. These components include all the software and hardware services and resources needed by the applications.

The problem with the GRID environment is its diversity of these components and their large number of users. These can make them vulnerable to faults, failure and excessive loads. Hence, some suitable mechanisms are needed to monitor the components, and their use, which can hopefully detect conditions that may lead to bottlenecks, faults and failures. Grid monitoring is a critical facet for providing a robust, reliable and efficient environment [2].

The goal of grid monitoring is to measure and publish the state of resources at a particular point in time. To be effective, monitoring must be end-to-end, meaning that all components in an environment must be monitored. This includes software (ex. Applications, services, processes and operating systems), host hardware (ex. CPUs, disks, memory and sensors) and network components like routers, switches, bandwidth and latency.

Monitoring data is needed to understand performance, identify problems and to tune a system for better overall performance. Fault detections and recovery mechanisms need the monitoring data to help determine if parts of an environment are not functioning correctly, or whether to start a component or redirect service request elsewhere. A service that can forecast performance might use monitoring data as input for a prediction model which in turn be used by a scheduler to determine which components to use. In this paper we have reviewed architecture of Ganglia Monitoring System and have found that Ganglia support many features but also has many limitations due to its architecture.

II. Ganglia – Related Literature

'Ganglia' is a distributive monitoring system for high performance computing systems such as clusters and the Grid [4][5][6]. It is based on a hierarchical design targeted at federations of clusters [4][5][6].

Ganglia relies on a multicast based listen/announce protocol to monitor state within clusters and uses a tree of point-to-point connections amongst representative cluster nodes to federate clusters and aggregate their state.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 8, August 2013)

Ganglia use XML for data representation, XDR for data transport and RRD tool for data storage and visualization [4][5]. It attempts to minimize per node system overheads by using specialized data structures and algorithms.

General Architecture -GANGLIA

The architecture of Ganglia consists of Ganglia Monitoring Daemon (gmond) and Ganglia Meta Daemon (gmetad). The architecture is given below-

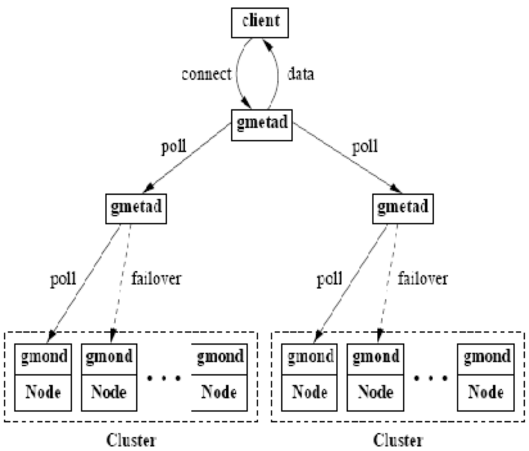

Figure 2: Architecture of Ganglia [4]

The Ganglia monitoring Daemon (gmond) is a multi threaded daemon which runs on each cluster node to be monitored. It has four main responsibilities -

- To monitor changes in host state
- To multicast relevant changes
- To listen to the state of all other Ganglia nodes via a multicast channel
- To answer requests for an XML description of the cluster state.

Each daemon transmits information in two different ways- multicasting host state in XDR format or sending XML over a TCP connection.

Ganglia Meta Daemons [4][5] are used to provide a federated view. At each node in the tree, a gmetad periodically polls a collection of child data sources, parses the collected XML, saves all numeric, volatile metrics to round robin databases and exports the aggregated XML over a TCP socket to clients. Data sources may be either gmond daemons representing specific clusters or other gmetad daemons representing sets of clusters. Data sources use source IP addresses for access control and can be specified using multiple IP addresses for failure. The latter capability is natural for aggregating data from clusters since each gmond daemon contains the entire state of its cluster.

Ganglia PHP web user interface provides a view of the gathered information via real time dynamic web pages. Most importantly, it displays Ganglia data in a meaningful way for system administrators and computer users.

III. FEATURES PROVIDED BY GANGLIA

Following are the features supported in Ganglia –

- Checking of heartbeat – gmond daemons sends a periodic heartbeat (every 20 seconds by default) [5]. If the heartbeat is missing it is fair to assume host is down. This alerts to potential down time much quicker [5].
- Checking of multiple metrics – allows using a single check to multiple metrics on the same host.
- Checking single metric across multiple hosts – use a single check to check low disk space on a set of hosts defined by a regular expression [5].
- While viewing aggregate graphs with more than 6-7 items, colors will start to blend together and it may be hard to distinguish what on graph is what. This feature allows decomposing a graph by taking every item on the aggregate graph and putting it on a separate graph. [5]
- Provision of utilization heat maps instead of the old style pie charts.
- Adding trend lines to graphs.
- It allows to compare hosts across all the matching metrics (this can mean hundreds of graphs)
- If a regular expression is supplied that matches a set of hosts, it will aggregate all the hosts. This is useful in finding why particular hosts are performing differently.

Following features are now provided in recent versions of Ganglia -

- Prevention of memory leak when receive channel is not configured or not hearing any data.
- It can be started in Solaris containers.
- It can be started when Solaris CPU is in FAILED state (segmentation fault).
- It can be started when there is no address on the network interface.
- Improvements are provided to the Live Dashboard.
- Aggregate graphs metric auto complete is provided.
- More reliable notification of changed files.

- Multiple CD drives are supported for bringing up a frontend.
- Ganglia metrics are now saved on frontend reboot. After a reboot, all Ganglia history will be restored from the previous boot.
- XML escape characters (e.g., &, <, >) are supported in the installation screens (e.g., the Cluster Information screen and the Root Password screen) [8].
- All the Intel compiler libraries are now copied to the compute nodes.

Following features are enhanced in Ganglia -

- Addition of the Condor Roll. This brings the distributed high-throughput features from the Condor project to clusters.
- Addition of the Area51 Roll. This roll contains security tools and services to check the integrity of the files and operating system on cluster.
- Ganglia RSS news event service.
- Improved network handling for compute nodes: any interface may be used for the cluster private network.
- Better support for cross-architecture clusters containing x86 and x86_64 machines.
- Device driver now builds and loads on compute nodes that have a custom kernel.
- RAID for custom compute node partitioning is supported.
- Addition of variables for root and swap partition. If the size of root is to be changed and/or swapped, only two XML variables have to be reassigned.
- The default root partition size has been increased to 6 GB from 4 GB.
- Use of ssh instead of telnet for security.
- Updated JDK available.
- Latest software updates recompiled for architectures.
- Automatic MySQL Cluster database backup.
- MAC addresses are included for each node in the "Cluster Labels" output.
- State of the frontend can be examined by typing "frontend rescue" at the boot prompt.

IV. PERFORMANCE CRITERIA

Following performance parameters and criteria are reviewed:

- CPU Overhead
  Obtain process information by running 'ps' & 'vmstat' command many times and then take the average. This leads to CPU overhead. Average CPU overhead comes to 0.3%. This is very less resulting in negligible CPU overhead.
- Memory Overhead
  Physical memory required is very less and virtual memory required is moderate.
- Bandwidth provided
  Obtained by running 'tcpdump' command and monitoring the traffic i.e packets sent. There are two types of bandwidth – The monitoring bandwidth that denotes average per node bandwidth for monitoring in a single cluster (i.e., bandwidth per gmond) and the federation bandwidth that denotes total bandwidth for aggregating data from a set of clusters (i.e., bandwidth for gmetad). The overall bandwidth is small as compared to the bandwidth provided by LAN.
- I/O Overhead
  There is significant overhead of I/O in interactive jobs as Ganglia has to write the data in the database and has to interact extensively to peripheral devices. This is because of the gmetad.
- Scalability (Number of nodes within a cluster)
  Gmond daemon is disabled to measure performance overhead and the number of nodes within a cluster. Gmetad is used to find the number of sites federated. It was found that the scaling was linear in all the factors like bandwidth, packet rates, and memory.

V. CONCLUSION

Ganglia – a popular grid monitoring system currently runs on over 500 clusters around the world, has been ported to nine different operating systems and six CPU architectures, and has even seen use on classes of distributed systems that it was never intended for. Along the way, the architecture of the system has had to evolve, features needed to be introduced, and the implementation has been continually refined to keep it fast and robust.

The simple architecture, that is providing a fast and robust implementation led to good scalability, robustness, and low per-node overheads.

Management overhead got substantially reduced and manual configuration is removed by the use of multicast listen/announce protocol for automatic discovery of nodes.

Ganglia is simple and robust because of the use of simple widely used technologies such as XML for data representation and XDR for data transport. Integrating Ganglia into other information services which add query languages and indexing and building new front-ends to

IJETAE
EXPLORING RESEARCH AND INNOVATIONS

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 8, August 2013)

export Ganglia's monitoring information became very simple by exporting XML.

With all these advantages, It also has some drawbacks. It does not support following features-

- IP SLA (Service Level Agreement) Reports. IP SLAs uses active monitoring to generate traffic in a continuous, reliable, and predictable manner, thus enabling the measurement of network performance and health [7]. No service level agreement reports are provided by Ganglia.
- Monitoring data is published using a flat namespace of monitor metric names. As a result, monitoring of naturally hierarchically data becomes awkward.
- Ganglia lack access control mechanisms on the metric namespace.
- Huge amounts of I/O activity. As a result, nodes running gmetad experience poor performance, especially for interactive jobs.
- Metrics published in Ganglia did not originally have timeouts associated with them. The result was that the size of Ganglia's monitoring data would simply grow over time.
- Within a single cluster, it is well-known that the quadratic message loads incurred by a multicast-based listen / announce protocol is not going to scale well to thousands of nodes. As a result, supporting emerging clusters of this scale will likely require losing some amount of symmetry at the lowest level.
- For federation of multiple clusters, monitoring through straightforward aggregation of data also presents scaling problems.
- Self-configuration while federating multiple clusters will also require substantial changes to the architecture since manual specification of the federation graph will not scale.

REFERENCES

[1] www.inetmon.com/index.php/products/inet-grid

[2] Global Grid Forum (GGF), Grid Monitoring Architecture Working Group, GMA White paper,

http://www-didc.lbl.gov/GGF-PERF/GMA-WG/

[3] Ruth Aydt-University of Illinois, Urbana-Champaign, Dan Gunter-Lawrence Berkeley National Laboratory, Warren Smith-NASA, "A Grid Monitoring Architecture", July 2001, citeseerx.ist.psu.edu/viewdoc/donload?doi=10.1.1.18.6602...pdf

[4] Matthew L. Massie, Brent N. Chun, David E. Culler, "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience", www.sciencedirect.com, June 2004, Published in Elsevier

[5] http://ganglia.info

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

[7] http://www.cisco.com/en/US/tech/tk920/tsd_technology_support_sub-protocol_home.html

[8] Release 3.2.0 - changes from 3.1.0 - Rocks, www.rocksclusters.org/roll-documentation/base/5.5/x8783.html