ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Real Time Meta Level Performance Analytics with Big Data System

[I]Gurubalan T, [II]Pushpa H G
[I]Dept. of CSE, School of Engg. & Tech., Jain University, Kanakapura Road, Bangalore, India
[II]Professor & H.O.D, Dept. of CSE, School of Engg. & Tech., Jain University, Kanakapura Road, Bangalore, India

Abstract
Big Data represents an enormous data-set that may include information from every possible source that can produce data that can provide further business opportunity. Big Data is characterized by 4 v's such as Volume, Variety, Velocity and Veracity. Big Data analytics adds value to the business only when analyzing data happens in near real-time. The need for monitoring the resources of the systems and taking proactive actions is essential. In this paper, the method helps in identifying any bottleneck in cpu, memory, disk and network resources by monitoring them and gathering statistics in a central repository.

Keywords
Systems Behavior, Centralized Stats repository, Performance Monitoring.

I. Introduction

Big Data is meant to create new opportunities for organizations to create business value by saving money, increasing the growth in revenue.

In this fast moving world, there is always a demand from end user to query data from dataset and get response faster. The time elapsed to get the results has come down from month, days, or hours to minutes, seconds or fractions of seconds.

If there is a delay in a response to end user, it is due to several layers in between from data to end user. The system resources play an important role to get the data that user needs. It is essential to monitor the system resources and identify any bottleneck. Once it is identified, the system administrator should take a corrective action to bring the system to a normal state.

II. Related Work

Investigating a performance issue is a complex task, such that simply providing some system level OS statistics is not enough. Analysis of that data requires a context to understand how they may relate to the issue.

The performance bottleneck can be either on CPU, memory, disk, network or the application. In case of having multiple systems in a network, instead of monitoring each system independently, why not have a central repository which has the entire statistics from all the systems in the network. The tool that I am developing is used to get the statistics of system resources from multiple systems using built-in statistics tools available in the Solaris 11 system. I list the most important tools and its purpose as follows.

vmstat: reports statistics about cpu, disk, kernel thread and virtual memory. It averages the number of CPUs in case of multiprocessor system

mpstat: reports processor statistics for each processor thread independently

iostat: provides io busy, service time and io wait time statistics.

netstat: reports each network interface statistics like packet collision, packet erros and routing table.

prstat: examines all active processes on the system and reports statistics such as context switching, locking, microstate process accounting.

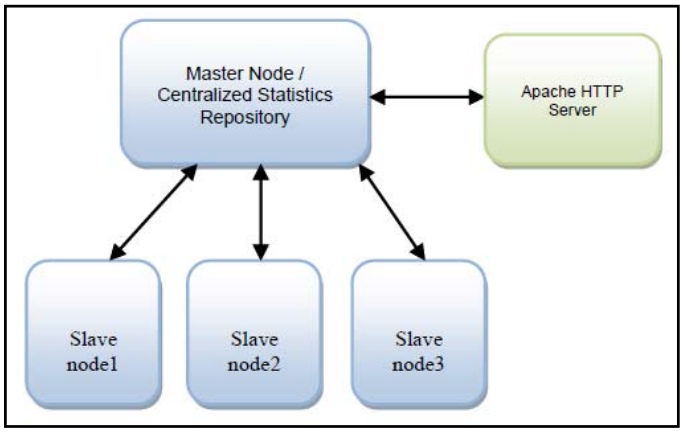

Fig1: Centralized statistics repository

PerfHalos: The tool is written using shell script. It is configured with one master system and the remaining servers as slaves or clients. The master system is going to have all the statistics received from the client nodes.

The tool includes the attributes of statistics that is going to be collected from all the system.

What data the tool collects?

The tool collects two kind of data such as static and dynamic.

Static Data: represents the data that doesn't change over a period of time. This includes the OS version, network configuration, system configuration

Dynamic Data: represents the data that change over a period of time. This includes statistics of cpu, memory, network packets, input output statistics of storage luns

A. How does the tool work?

The tool has static and dynamic configuration. One system is selected as master node where all statistics of group of systems will be stored. The rest of systems is known as client nodes. The password less ssh is setup between master and the client nodes.

From master node, init stat is executed and it will execute all the stats in master node as well as the client nodes. [Ref. Fig1]. Based on the interval set, it collects the stats in the local system till I run fini stat. Once I run fini stat, it terminates the stats on all the system. Then, run get stat which gathers the statistics from all the system and stores in the central repository of the master system.

B. What additional features this tool has?

This tool can also support plugins so that third party commands that are supported on this platform can be incorporated with this tool for gathering statistics.
Additionally, the output of this tool can be fed to graph generation tool called fenxi for plotting graphs.

III. Results

The stats are gathered for master node as well as the client nodes. The graphs for each node are created. The snapshot of stats gathering and graph generation are given below.

```
INITIATING STATS FROM MASTER NODE
root@namenode1:/PerfHalos/scripts# ./initstat
root@namenode1:/PerfHalos/scripts# initializing stats in namenode1
initializing stats in datanode1
initializing stats in datanode3
initializing stats in datanode2

root@namenode1:/PerfHalos/scripts#
```

Fig. 2: Initializing stats on all nodes

```
STOPPING STATS FROM MASTER NODE
root@namenode1:/PerfHalos/scripts# ./finistat
root@namenode1:/PerfHalos/scripts# stopping stats in namenode1
stopping stats in datanode1
stopping stats in datanode3
stopping stats in datanode2
```

Fig. 3: Stopping stats on all nodes

```
root@namenode1:/PerfHalos/scripts# echo "GATHERING STATS AND GENERA
GATHERING STATS AND GENERATING GRAPHS
root@namenode1:/PerfHalos/scripts# ./getstat
Enter the load name with date:
DemoTest
Gathering stats from namenode1
Gathering stats from datanode1
Gathering stats from datanode2
Gathering stats from datanode3
Collected stats are under: /PerfHalos/output/DemoTest
Generating Graphs for Collected stat at /PerfHalos/output/DemoTest
processing fenxi output for machine datanode1
Creating database ...                          6.78s
Loading create statements from txt
Loading netsum                                 ...+..+..+..OK
Loading prstat                                 ...+..+..+..OK
Loading vmstat                                 ...+..+..+..OK
Loading mpstat                                 ...+..+..+..OK
Loading iostat                                 ...+..+..+..OK
Processing profile default_profile             ...OK
- Generating section   53/53       0 errors
Query Execute took                             18.27s
```

Fig. 4: Gathering stats and generating Graphs

Snapshots of Graphs

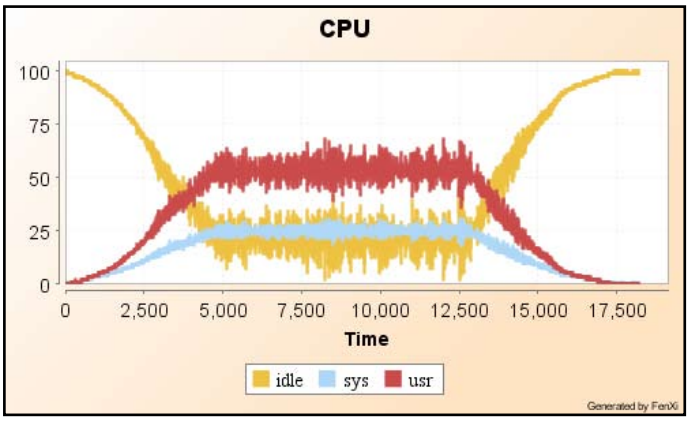

Fig. 5: CPU utilization

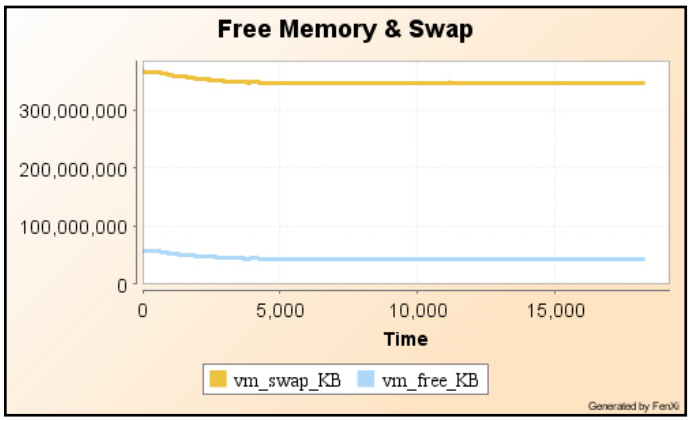

Fig. 6: Memory Utilization

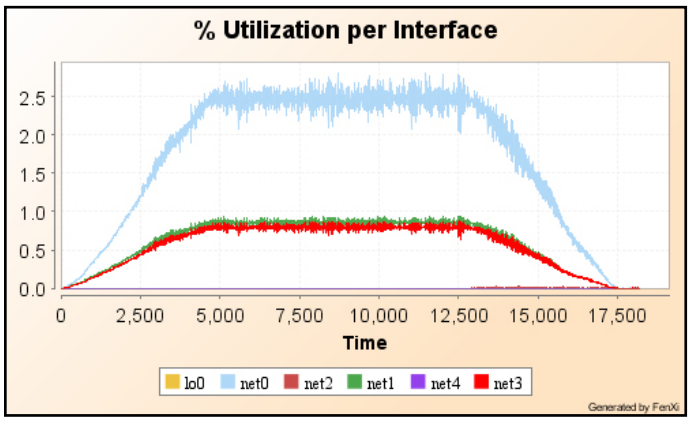

Fig.7: Network Utilization

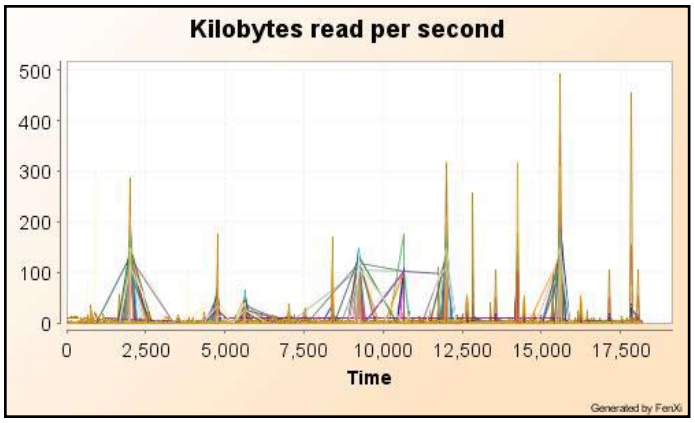

Fig. 8 : IO read Utilization

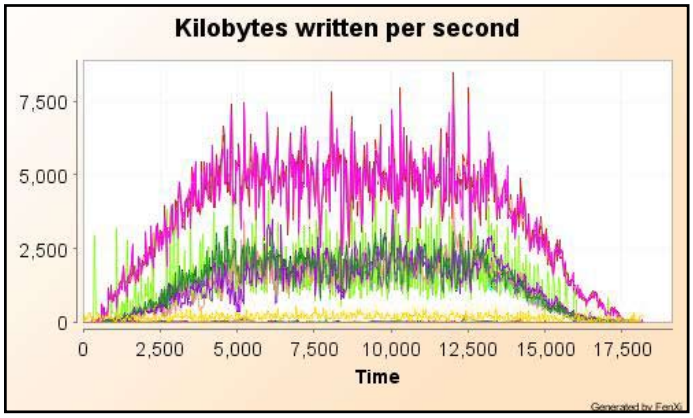

Fig. 9: IO write Utilization

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

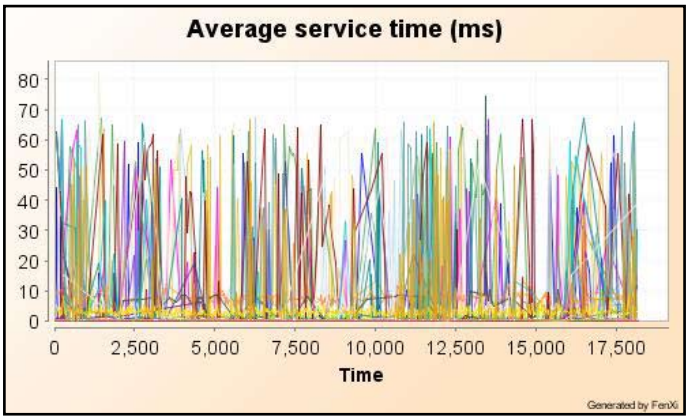

Fig.10 : IO Avg. Service Time

Fig.11: HTTP Server authentication

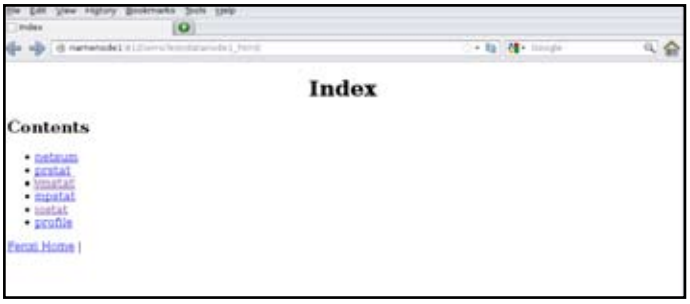

Fig.12 : Graphs link for Master node

IV. Conclusion

The tool is used to gather statistics from multiple systems and store them in the central repository. It is also integrated to graph generation tool that creates graph for easier understanding of the pattern of cpu, memory, io, network. The generated graphs are accessible from apache web server configured on master node using the url http://<master-node>:<port-no>.

V. Future Work

The current project can be enhanced by integrating with creating alert message based on threshold configured and the system administrator gets a notification remotely.

References

[1] *Xiaohua Zhang, Zhirou Zhang. Data center integrated monitoring based on performance monitoring of server and application system*
[2] *About Monitoring System Performance http://docs.oracle.com/cd/E23824_01/html/821-1451/spconcepts-38776.html.*
[3] *Rules-of-thumb for monitoring Sun Solaris statistics http://www.ibm.com/developerworks/lotus/library/ls-Rules_SunSol/*
[4] *Solaris Performance Monitoring & Tuning – iostat, vmstat, netstat http://www.adminschoice.com/iostat-vmstat-netstat*
[5] *Configure the Apache HTTP server http://httpd.apache.org/docs/2.2/configuring.html*
[6] *Solaris Troubleshooting and Performance Tuning http://www.princeton.edu/~unix/Solaris/troubleshoot/*
[7] *Solaris Performance Checklist http://www.brendangregg.com/USEmethod/use-solaris.html*
[8] *Solaris Tunable Parameters Reference Manual http://docs.oracle.com/cd/E19455-01/816-0607/816-0607.pdf*