Observing the Processes in the High-performance Computing Programs

Kostadin Mishev[1], Sonja Filiposka[2], Anastas Mishev[3]
[1]The Faculty of Computer Science and Engineering at SS
Cyril and Methodius University in Skopje
Rudjer Boshkovikj 16
Skopje 1000, Macedonia
Kostadin.mishev@Finki.ukim.mk
[2]Faculty Of Computer Science and Engineering at SS
Cyril and Methodius University in Skopje
Rudjer Boshkovikj 16, Skopje 1000, Macedonia
Sonja.filiposka@Finki.ukim.mk
[3]The Faculty Of Computer Science and Engineering at SS
Cyril And Methodius University In Skopje, Rudjer
Boshkovikj 16, Skopje 1000, Macedonia
Anastas.mishev@Finki.ukim.mk

ABSTRACT: *High performance computing largely depends on the enhanced utility, use and efficiency in processing of data. Hence, the high-performance computing systems should be regularly monitored to ensure understanding and high utility. These systems can be given high power sources and supported by good and well-designed programs.*

Keywords: HPC, Monitoring, Efficient usage

Received: 18 September 2021, Revised 10 December 2021, Accepted 22 December 2021

DOI: 10.6025/pca/2022/11/1/1-7

1. Introduction

High Performance Computers (HPC) represent one of the the main building blocks of the todays e-Infrastructures [1]. Together with cloud-based solutions and efficient distributed large scale storage on top of ultrafast agile network, they are a crucial element that provides the processing power needed to tackle with big data problems.

However, achieving their optimal usage comes with a whole plethora of new problems and challenges. With the most important resource being processing power achieved with a huge number of cores (CPUs and accelerator cores like GPGPUs and Xeon Phy's), the goal of maximum performance needs to be reached under a complex set of constraints like efficient core usage, minimum inter process communication, but also minimized power consumption. Thus, the effective usage of these computing giants needs thorough monitoring that will reveal the usage of the available resources and the way they respond to different loads and tasks [2].

Most of the current monitoring solutions are vendor dependant proprietary solutions like IBM Platform HPC [3], HP Insight Cluster Management Utility [4], Fujitsu HPC Cluster Suite [5]. They provide various types of monitoring of HPC systems, mostly offering chassis and operating system (OS) level analysis and visualization tools. Notable open source monitoring solutions include Ganglia [6], Nagios [7], and OVIS [8]. These tools are usually based on probes running on the hosts themselves and different ways of collecting and representing the collected data.

Important questions that arise regarding the HPC monitoring [9] include the understanding of the HPC monitoring process, its evolution, technologies and vendor lock-ins vs neutrality. Vendor specific proprietary tools usually offer higher levels of details, while the open source solutions give more flexibility.

There are several perspectives of the HPC monitoring that should be taken into account when implementing a monitoring solution: OS level monitoring, chassis level monitoring, power consumption monitoring etc. The integration of different monitoring data sources gives a more complete and wider perspective [10]. Using integrated data, such as host or chassis level power consumption combined with the CPU and memory usage can help in developing power aware algorithms, schedulers etc. One of the main challenges when integrating monitored data from different sources is the time based association of the data that is gathered using different sampling rates, i.e. monitoring data with different resolution. HPC monitoring systems can be implemented as real-time monitors or log analysis tools [11]. Although real time monitoring provides valuable data for the current state of the HPC cluster, monitoring based on log analysis offers more indepth view of the overall system parameters and can lead to important conclusions regarding the correlation of the values of measured parameters obtained from different sources and levels.

In this paper we present a HPC data-log monitoring system acting as a collection server for monitoring data that stems from different sources. Upon collection, the system adapts the data into one common format, aligning the different sampling resolutions of the available monitored data. The summarized output of the system can be used for various purposes such as performance analysis, power consumption examination etc. The rest of the paper is organized as follows. Part 2 describes the sources of data aggregated by the proposed system. Part 3 explains the system architecture, while Part 4 discusses the possible applications of the system. Part 5 concludes the paper and points toward the future work.

2. Monitoring Sources

HPC monitoring data can be integrated from various sources that will provided information concerning different aspects of the system like node level monitoring, enclosures parameters, and out-of-band system management interfaces.

In order to obtain monitoring data from the OS perspective, a cluster management software (CMS) [12] can be used. Upon deployment, each computing node runs an agent of the cluster management utility that sends data to the appointed management node. Usually, the utility solution provides a tool for real time monitoring and visualization of the collected data. This data includes: CPU load and frequency, memory usage, OS version, disk space usage, etc. In order to obtain logged data for a specific time period, a command line tool can be used. This tool enables the user to select a subset of nodes and the parameters that will be exported to a file with a flat structure (timestamp, nodeID, parameter, value). This dump can then be combined and aligned with the other collected datasets.

Another common monitoring data source is the so-called Light-out management system (LOM) [13]. LOM uses a dedicated management channel to monitor and remotely manage nodes regardless of whether they are powered on, or if an operating system is installed or functional, i.e. compared to the cluster management utility, it provides an external monitoring view of the nodes and their enclosures. The data acquired from this system includes chassis information, power usage and distribution by chassis elements, aggregated thermal monitoring data, etc.

Although the LOM solutions are specialized implementations of the Intelligent Platform Management Interface (IPMI) [14], the direct use of IPMI enables further enrichment of the gathered data. IPMI is in fact a specification for interfaces used for out-of-band management of computer systems and monitoring of their operation. Using IPMI one can gather more detailed data including current power consumption, various temperature sensor readings and fan speed on node level, compared to the LOM's chassis level. Unlike CMUs that logs the monitoring data autonomously, using IPMI the data must be manually pulled in real-time with a given resolution. This can be achieved using the ipmitool command.

3. System Overview

In order to obtain a holistic, yet, comprehensive, overview of the monitored system, the data gathered from the three available monitoring sources described in the previous section must be composed in a single collection. For this reasons, wepropose an integrated data-logging monitoring system that will act as a collection server for the monitoring sources and thus provide a single point of reference when monitoring the usage of the HPC resources and related components.

In this section we describe the proposed system architecture and its implementation for a given HPC.

3.1. System Architecture

Data gathered from the sources described above, are merged and saved into a database as a raw data. The captured data series are organized in reservations enabling the start and the end of data capturing. The data captured by the LOM tool is the basis for establishing the time flow and data resolution. On top of this data, the captured measurements from the other two sources are added. All data imported in the monitoring software is automatically merged and aligned using the provided timestamps.

The complete raw data is organized in different reservations characterized with reservation start and end time. Inside each reservation, the raw data that belongs to the reservation is ligned according to the LOM's data resolution. Each reservation can be visualized using a chart panel, or it can downloaded as a set of flat files for external evaluation. The deployed example system for integrated HPC monitoring, given on Figure 1, is implemented using Java Spring 4.0 framework using MySQL database for data logging.

3.2. Use-case scenario

The described system is used to monitor a Hewlett-Packard HPC system located at the Faculty of Computer Science and Engineering in Skopje. The system is based on HP Cluster Platform 7000 enclosures with 42 blades BL2x220c G7 with dual motherboard and dual Intel Xeon L5640 @ 2.26Ghz (total 1008 cores). The blades are grouped in three enclosures. Using HP's

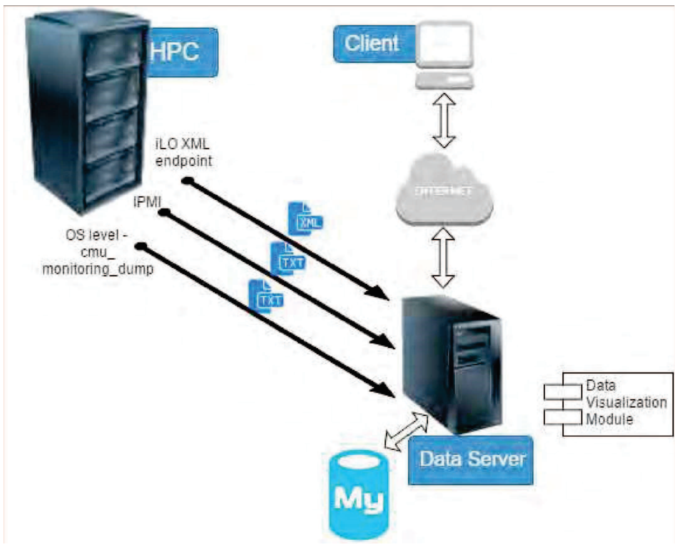


Figure 1. System architecture

Insight CMU [4] each blade can have its own monitoring station which measures several parameters such as temperature, CPU frequency, CPU load, memory used and process memory, which determine the current state of the blade depending of the current load handled by the appropriate blade. The data from the CMU is gathered using the cmu_monitoring_dump CLI tool.

The Light-out management system used for monitoring data is based on the HP's implementation of the LOM, the HP integrated Lights Out (iLO) [15]. iLO uses a custom build scripting language, based on XML, to provide detailed information about the HPC enclosure and its elements. Each iLO-based enclosure serves as an endpoint that returns XML files presenting the values of the sensors. The enclosure which wraps a set of blades, has its own control unit. The control unit monitors the overall system health:

• The rotations per minute of cooling fans

• The enclosure manager provides information about hardware state, firmware versions and temperature of monitoring unit

• Temperature of switches

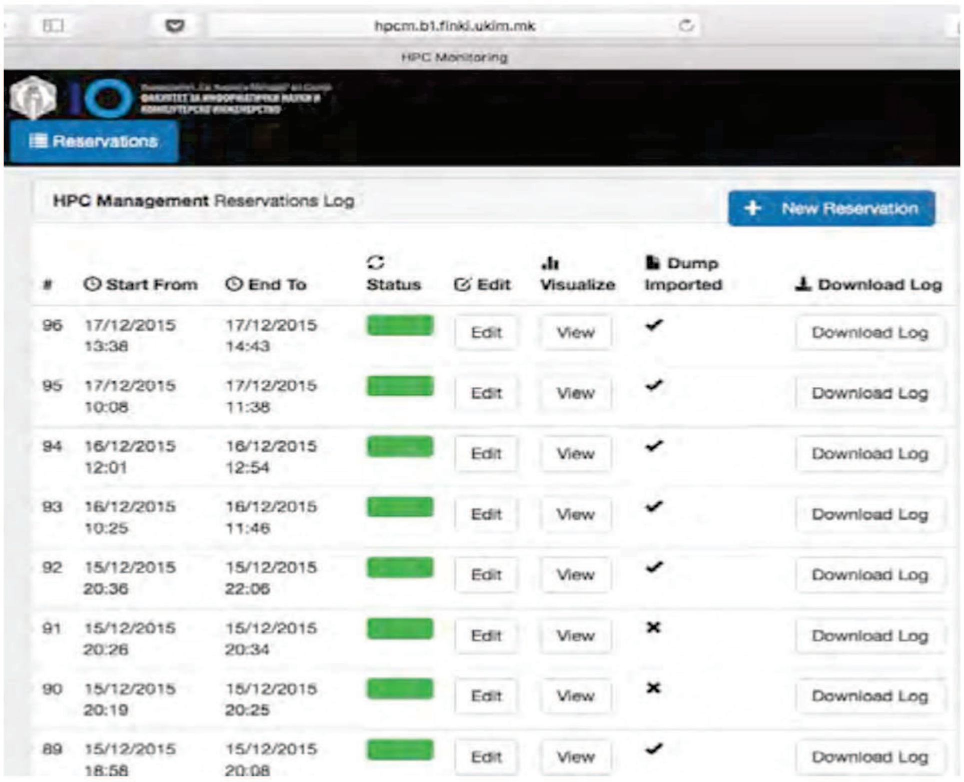
• Power consumption and power output

• Ambient temperature

Additionally, using IPMI fine-grain monitoring data is gathered on the node level, specifically node power consumption and node level CPU temperature.

More detailed information on the various available measurement parameters that can be obtained using the merged data log monitoring approach is given in Table I that can be found at the end of this paper.

Within the developed solution (available at [16]), each monitoring session is presented as a reservation. In order to conserve storage, the system is not running all the time, but is acting upon scheduled user created reservations. To enable the log during specified time period, the user must create a new reservation and specify the time interval of the active session together with the enclosures that are to be monitored. Upon creating the new reservation, using the main page, the user can monitor the current status of the reservation presented via a green slider (see Figure 2).

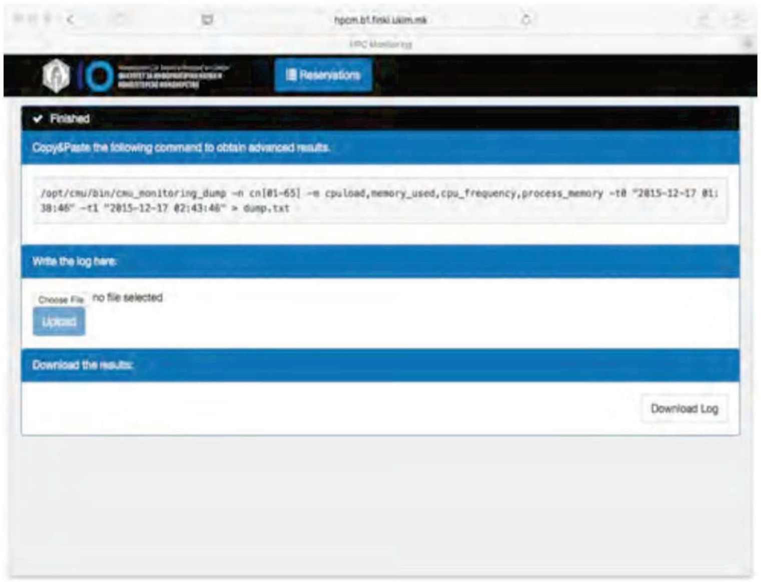


HPC Management Reservations Log

#	Start From	End To	Status	Edit	Visualize	Dump Imported	Download Log
96	17/12/2015 13:38	17/12/2015 14:43	🟩	Edit	View	✓	Download Log
95	17/12/2015 10:08	17/12/2015 11:38	🟩	Edit	View	✓	Download Log
94	16/12/2015 12:01	16/12/2015 12:54	🟩	Edit	View	✓	Download Log
93	16/12/2015 10:25	16/12/2015 11:46	🟩	Edit	View	✓	Download Log
92	15/12/2015 20:36	15/12/2015 22:06	🟩	Edit	View	✓	Download Log
91	15/12/2015 20:26	15/12/2015 20:34	🟩	Edit	View	✗	Download Log
90	15/12/2015 20:19	15/12/2015 20:25	🟩	Edit	View	✗	Download Log
89	15/12/2015 18:58	15/12/2015 20:08	🟩	Edit	View	✓	Download Log

Figure 2. Monitoring reservations

Afterwards, the user can import the dump files obtained from the other sources by navigating to the Edit section of the reservation. This page assists the user to execute the appropriate commands on the native interfaces to obtain the right dump file (see Figure 3). The file should be uploaded by using the file upload option on this page. The system calls background data merge services which matches the appropriate data parameters from different sources. The product of such merge, can be observed by navigating to the Visualize section by clicking the View button.



✓ Finished

Copy&Paste the following command to obtain advanced results.

```
/opt/cmu/bin/cmu_monitoring_dump -n cn[01-65] -m cpuload,memory_used,cpu_frequency,process_memory -t0 "2015-12-17 01:
38:46" -t1 "2015-12-17 02:43:46" > dump.txt
```

Write the log here:

Choose File no file selected

Upload

Download the results:

Download Log

Figure 3. Import of the data from other sources

On the navigation menu, the user chooses the parameters that he will observe. The result is a chart series which visualize the parameters changes over time (see Figure 4). Also, there exists an option for downloading a zip file of a set of flat files for all parameters by reservation for additional processing.

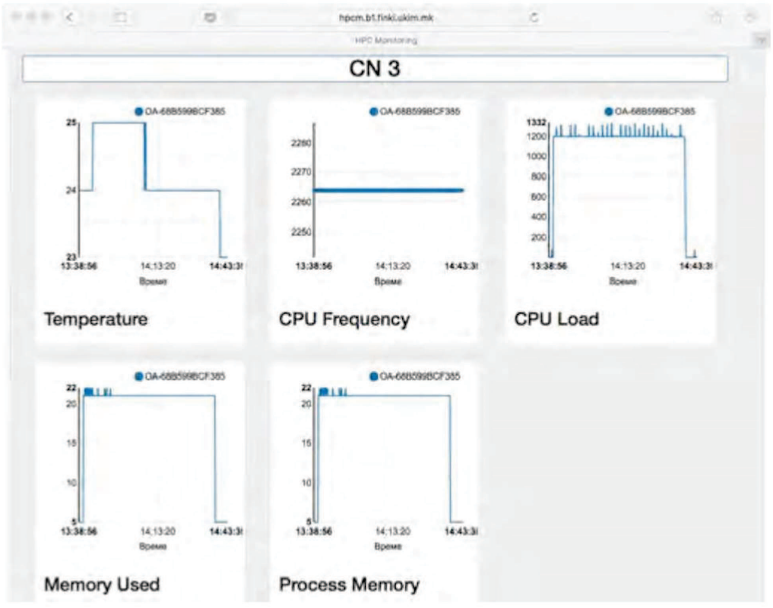


Figure 4. Visualization of the data

4. Applications

The proposed system offers a wide spectre of applications. Analysis and visualization of the execution of specific applications can help the developers optimize their solutions toward better resource utilization.

If we take power efficiency as an example, the collected data can be used for investigating power aware algorithms or schedulers, providing optimal power consumption.

Thermal related data (temperature sensors measurements, fan rotation speeds) can help avoid hot spots in the chassis by varying the scheduling and the distribution of the processes by nodes.

Analysis of the data from the networking elements can help develop intelligent parallel algorithms that will optimize interposes communication.

5. Conclusion

In this paper we presented an integrated monitoring solution for HPC systems. It is based on data collections from various sources including server management interfaces and OS level probes. The integrated data can be visualized or exported for further in-depth analysis. The acquired and integrated datasets can be used for better system understanding, development of novel HPC schedulers, increased resource usage under power restrictions, etc.

Future versions of the system will be aimed towards enabling finer resolution of the data sampling, real time import and alignment and wider integration and scalability.

Acknowledgement
This work is supported by the project Design and analysis of green ICT systems, financed by the Faculty of computer science and engineering, UKIM.

References

[1] Mustafee, N., JE Taylor S. (2013). High-performance simulation and simulation methodologies. 2013.

[2] Browne, J.C., DeLeon, R.L., Patra, A. K., Barth, W.L., Hammond, J., Jones, M. D., Furlani, T. R., Schneider, B. I., Gallo, S.M., Ghadersohi, A., Gentner, R.J. (2014). Comprehensive, open source resource usage measurement and analysis for HPC systems. *Concurrency and Computation: Practice and Experience*, 26 (13) 2191-2209.

[3] IBM Platform HPC, http://www- 03.ibm.com/systems/platformcomputing/products/hpc/, visited March 2016.

[4] HP Insight Cluster Management Utility, http://www8.hp.com/us/en/products/server-software/productdetail. html?oid=3296361, visited March 2016.

[5] Fujitsu Cluster Suite, http://www.fujitsu.com/fts/microsites/hpc/productsservices/index.html, visited March 2016.

[6] Ganglia Monitoring System, http://ganglia.info, visited March 2016

[7] Nagios, https://www.nagios.org, visited March 2016.

[8] Brandt, J. M., Gentile, A. C., Hale, D. J., Pebay, P. P. (2006). OVIS: a tool for intelligent, real-time monitoring of computational clusters, *Parallel and Distributed Processing Symposium*, IPDPS 2006. 20th International, Rhodes Island.

[9] Allcock, W., Felix, E., Lowe, M., Rheinheimer, R., Fullop, J. (2011). Challenges of HPC monitoring. In State of the Practice Reports (SC '11). ACM, New York, NY, USA.

[10] Bohm, S., Engelmann, C., Scott, S. L. (2010). Aggregation of Real-Time System Monitoring Data for Analyzing Large-Scale Parallel and Distributed Computing Environments, *High Performance Computing and Communications (HPCC)*, 2010 12th *IEEE International Conference on, Melbourne*, VIC, 72-78.

[11] Moore, C. L., et al. (2015). Monitoring High Performance Computing Systems for the End User. Cluster Computing (CLUSTER), 2015 *IEEE International Conference on. IEEE*.

[12] Bonvin, N., Vaudenay, S., Junod, P. (2003). Cluster Management Software. No. LSIR-STUDENT-2009-001.

[13] Bojinov, H., Bursztein, E., Lovett, E., Boneh, D. (2009). Embedded Management Interfaces: Emerging Massive Insecurity. Black Hat USA.

[14] IPMI – Intelligent Platform Management Interface, http://www.intel.com/content/www/us/en/servers/ipmi/ipmihome. html, visited March 2016.

[15] HPE Integrated Lights Out (iLo), http://www8.hp.com/us/en/products/servers/ilo/index.html, visited March 2016.

[16] Integrated HPC monitoring, http://hpcm.b1.finki.ukim.mk/.

Parameter	**Applies to**	**CMU**	**iLO**	**IPMI**
Temperature	Node		X	
CPU frequency	Node	X		
CPU load	Node	X		
Memory used	Node	X		
Process memory	Node	X		
Current fan RPM	Chassis		X	
Output power	Chassis		X	
Input power	Chassis		X	
Power supplies	Chassis		X	
Ambient temp	Chassis		X	
Power meter	Node			X
CPU temp	Node			X
Inlet ambient temp	Node			X
DIMM temp	Node			X
Memory zone temp	Node			X
Chipset temp	Node			X
NIC temp	Node			X
System exhaust temp	Node			X
Virtual fan	Node			X
Enclosure status	Node			X
Memory status	Node			X
Health LED	Node			X
Per switch temp	Chassis		X	
Manager temp	Chassis		X	
Power on switch	Chassis		X	
Power off switch	Chassis		X	

Table 1. Selected Measured Parameters By Source