Development of the active monitoring system for the computer center at IHEP

V. Kotliar[a], V. Anshukov[b], V. Ezhova[c], V. Gusev[d], A. Kotliar[e], G. Latyshev[f], A. Shishov[g]

National Research Center "Kurchatov Institute" State Research Center of Russian Federation Institute for High Energy Physics, Protvino, Russia

E-mail: [a] Viktor.Kotliar@ihep.ru, [b] Vladimir.Anshukov@ihep.ru, [c] Victoria.Ezhova@ihep.ru, [d] Victor.Gusev@ihep.ru, [e] Anna.Kotliar@ihep.ru, [f] Grigory.Latyshev@ihep.ru, [g] Artur.Shishov@ihep.ru

Computer center at IHEP is a complex system of many technologies gathered together. Among them are distributed computing, high throughput networking, high reliable uninterruptable power systems, precision cooling systems. Monitoring and control of such complex is a very difficult task. Even more difficult is to create self-optimization, self-healing and self-defense systems on top of the monitoring. As a first step it might be a creation of several databases to accumulate all information about center infrastructure, events, logs, statuses and then as the second step a creation of an active monitoring system which could be able to perform simple tasks itself or make advice for the human interventions. The current status of the development of such system for the IHEP computer center described in this work.

Keywords: elastic search, kibana, APC symmetra, monitoring, computer cluster

Introduction

Computer center at IHEP is a complex system of many technologies gathered together. Among them are distributed computing, high throughput networking, high reliable uninterruptable power systems, precision cooling systems. Monitoring and control such of complex is a very difficult task. Even more difficult is to create self-optimization, self-healing and self-defense [Kephart, Chess, 2003] systems on top of the monitoring. As a first step it might be a creation of several databases to accumulate all information about center infrastructure, events, logs, statuses and then as the second step a creation of an active monitoring system which could be able to perform simple tasks itself or make advice for the human interventions. The current status of the development of such system for the IHEP computer center described in this work.

This work consists of two parts. In the first part it is described a current status of the monitoring systems in the computer center. And in the second part it is described a development of the new system with achievements and plans. As soon as it is not only software monitoring but a real engineering hardware it is really important work and has a big value for the whole data center infrastructure.

What needs to be monitored

Computer center at IHEP starts his history with the installation of Minsk-2 computing system in 1965. Over all these fifty years the installed computer systems just grow with computer power and storage capacities. And significant increase was made with introducing grid-computing technology where a computer center at IHEP became a part of the distributed all over the world computing system like WLCG (World-wide LHC computing grid [WLCG homepage]).

At the moment the computer infrastructure consist of the following components spread over two independent working zones:

- around 3000 CPUs which are split over 150 computer nodes;
- near 2PB disk storage on 50 servers;
- power hardware as two UPS APC Symmetra plus 30 small UPSes and 26 PDUs;
- 6 Emerson Liebert cooling systems;
- cluster network with 1000 of 1Gbs connections.

Not only hardware has to be monitored but also software installed and used on the cluster. As a matter of fact it is a system for distributed computing and a failure of one subsystem could not be a problem and even will not be seen by programs. But degraded system need to be repaired as soon as possible to go back to a full production mode.

Used monitoring tools

Depending on needs there are several monitoring systems for the data center and in particular for the computing cluster at IHEP:

- Nagios to check almost 3500 computer center services;
- Splunk and central syslog facility to store and analyze all logs (400 thousands events per day);
- Collectl to provide a real-time monitoring for 160 servers on one screen;
- Elasticsearch plus kibana for engineering infrastructure monitoring;
- Munin rdd monitoring for IPMI sensors on all servers in the computer cluster;
- pmact + cacti for network traffic monitoring;
- Self-guard local monitoring on server for IMPI events, temperature limits, UPS events;
- Self-build accounting system;
- Big Red Button for safety switching off all servers by human – operators monitoring.

These are only monitoring systems for hardware and operating system or service level software. But for the monitoring of the distributed grid environment additional monitoring systems are used: regional nagios for Russian grid sites; CERNs check_mk monitoring for all LHC experiments and their software; operational portal security dashboard for security monitoring; Grid middleware monitoring for grid services; four monitoring system per experiments like CMS dashboard, Alice site monitoring, Atlas panda monitoring, LHCb Dirac monitoring; Grid accounting system.

As it might be seen such many monitoring tools has a big impact on how effective they can be used. It is extremely difficult to maintain many tools (support, upgrade, backup) and difficult to use them as soon as the classic monitoring assumes a creation of a computer center control room with many monitors and at least two people to check all screens and to react if needed. It is also difficults to use programs for analyzing data and making decisions as there is no single API and single DB for monitored data. All systems have their own data format like: rrd, sql tables, nosql indexes, text, JSON and others. Used monitoring systems do not provide «clever» alarms and they have only simple threshold limits mechanism to trigger alarm events. And at last it is difficult to add new monitored parameters if they are not implemented by the system.

Architecture of the new system

The primary goal for the monitoring system is not only to see nice graphs or alarms but rather to be a part of the self-management system [White paper, 2006]. There are four following aspects for such systems:

- self-configuration where systems can configure themselves automatically depending on the high level policies;
- self-optimization where systems have hundreds of tunable parameters and continually seek ways to improve their operation;
- self-healing where systems analyze information from log files and monitors for healing themselves;
- self-protection where systems detect malicious attacks and prevent themselves from cascading failures.

Monitor together with Analyze, Plan, Execute, Knowledge is a core part of the Autonomic manager inside of autonomic element for self-management systems.

Considering IHEP cluster environment a new system has to fulfill some requirements. First of all it has to collect information from all available sensors inside computer center for offline or online analysis. Collected data must be easily reachable for programs, scripts and humans. It should allow to develop the logic for the self-defense of the computing center from power cuts, cooling problems, fire and also implement automatic detection of anomalies in hardware. And at the end such system has to be used for the self-optimization based on job efficiency, cooling efficiency, power consumption efficiency.

Architecture of the developing monitoring system is shown on figure 1. It consist of the three major parts:

- indexing cluster – it is a cluster of nodes which are able to receive data;
- data nodes – they are nodes for storing real data;
- search cluster – it is a cluster of nodes to perform search on stored data.

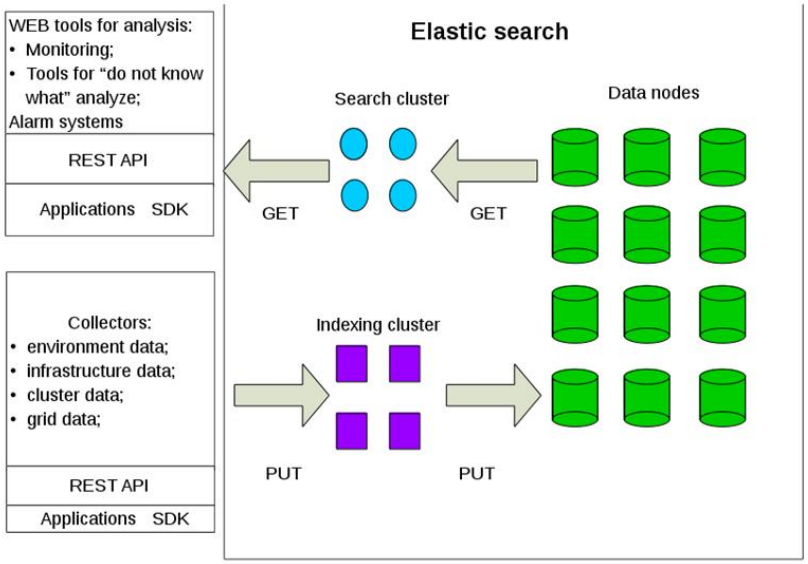

Figure 1. Architecture of the monitoring system

Each of these components might be easily extended horizontally if more performance is needed. In the very beginning everything could be started from a single node and grows up by needs. To store data such system has collectors – programs which collect different kinds of data like environment metrics, infrastructure data, cluster metrics, grid metrics and send these data by well documented REST API to the core of the monitoring. Also applications SDK available to be used directly in the collector programs. For analysis and monitoring data there is a build-in simple web interface which allows to create custom dashboards for groups of data or it allows just navigate through raw data with a convenient interface. Search cluster also provides REST API and applications SDK.

The main point in the architecture is adding a new monitoring data which do not require to change the core of monitoring (indexing cluster, data nodes, search cluster) and it just need to create or deploy appropriate collector and to create a dashboard for monitored data. It simplify the whole process of extension of the monitoring system.

Achievements and plans

Following the architecture, the core of the monitoring system was installed based on Debian GNU Linux 7.1 with NoSQL database on ElasticSearch (ES) 1.7.3. Additional ElasticSearch plugin elasticsearch-xml installed to allow store XML data directly to ES. As a primary tool for visualization and data analysis with web-interface Kibana 4.1.1 was installed.

Access to the web-interface is done by login/password and a proxy server based on Apache is used for that. All data are stored on the RAID6 storage system on the hosting Xen-hypervisor. There is an everyday incremental backup for ES indexes.

Primary data format for ES is JSON and database clients can directly index their data where access is controlled by iptables.

According to the architecture above it was already implemented several collectors and dashboards mainly for engineering infrastructure as a fundamental layer of the whole cluster. One of such systems is presented on figure 2.

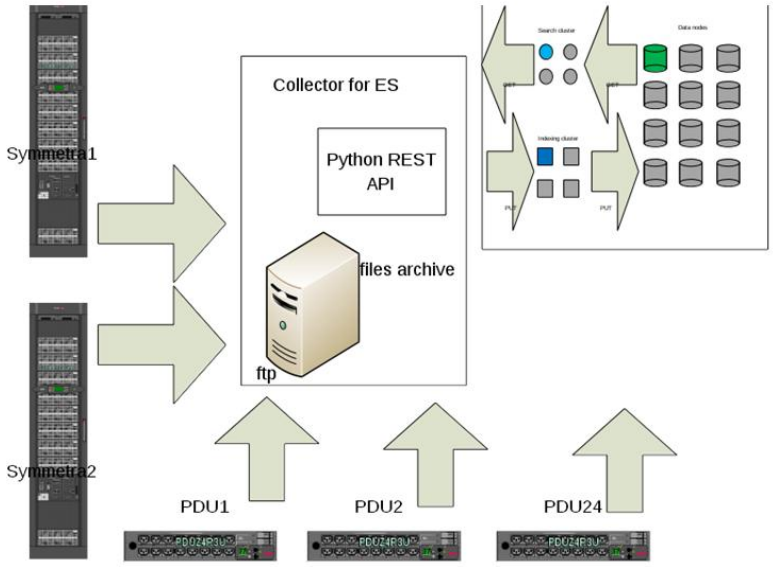

Figure 2. Collector for UPS system of the cluster

It gathers UPS metrics from two APC Symmetra PX 160kW and more than 20 APC PDU through internal feature of APC to store data on a remote ftp server. Then all these data are parsed with python programs and are put through REST API to the ElasticSearch cluster. To monitor that data an engineer dashboard was created in the Kibana system. Implemented monitoring already prevented a big UPS batteries damage: where high amperage on the batteries triggered alarm for battery replacement.

Using similar schema it is implemented a monitoring for cooling system of the cluster where the cooling metrics from Libert PDX are collected by python program and then pushed to the ES cluster and Kibana dashboard shows it.

For the monitoring there also were added collectors for GPU-specific data from computing nodes, data from standalone UPS'es, metrics considering XEN virtual machines in the computer center, accounting data for the batch jobs on the computer cluster.

For future developing it is going to be added more and more monitoring information to ES from servers, accounting system, storages, antiviruses. More and more threshold based alarms are needed to be created. But main point is to start to implement a smart alarms and active monitoring. It means that by using statistical functions, mathematical models and even machine learning techniques it will be possible to detect anomalies on the data and trigger alarms for humans even if these data in their threshold limits or if there are no limits at all. When self-healing systems is deployed on cluster all these alarms will be used for triggering self-healing events for the cluster [Gaudin, Hinchey, 2011].

Conclusion

The architecture of the new monitoring system for IHEP cluster and some implemented parts of the monitoring was presented in the described work. Developed architecture allows to easily extend always growing monitoring system horizontally so it is elastic and scalable. This architecture may store all kinds of unstructured data which is important when it is not known what is needed to be monitored.

The procedure of adding new monitoring elements simply means creating collector parts and parts for visualization and analysis. They are independent from the core infrastructure which is a big advantage of the created system. The whole system uses only open software which also a main point in the advantages.

As a proof of concept a few systems for monitoring were already added to such architecture and allowed to achieve practical results in a short period of time.

As future works it is planned to use programming languages for detecting anomalies in data patterns with statistical, mathematical and machine learning techniques which could help to implement active and smart monitoring to generate alarms or trigger self-healing events for the whole cluster infrastructure which will allow to implement some principles of autonomic computing [White paper, 2006] such as self-optimization, self-healing, self-protection for the IHEP computer cluster.

References

White Paper. An architectural blueprint for autonomic computing // IBM. — 2006.

Gaudin B., Hinchey M. FastFIX: An approach to self-healing // Proceedings of the Computer Science and Information Systems conference. — 2011. — ISBN: 978-1-4577-0041-5.

Kephart J.O., Chess D.M. The vision of autonomic computing // Computer. — 2003. Vol. 36. — P. 41-52. — doi:10.1109/MC.2003.1160055.

WLCG homepage [Электронный ресурс]: http://wlcg.web.cern.ch/