Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018), Dubna, Moscow region, Russia, September 10 - 14, 2018

MULTICOMPONENT CLUSTER MANAGEMENT SYSTEM FOR THE COMPUTING CENTER AT IHEP

V. Ezhova [a], A. Kotliar [b], V. Kotliar [c], E. Popova [d]

Institute for High Energy Physics named by A.A. Logunov of National Research Center "Kurchatov Institute", Nauki Square 1, Protvino, Moscow region, Russia, 142281

E-mail: [a] Victoria.Ezhova@ihep.ru, [b] Anna.Kotliar@ihep.ru, [c] Viktor.Kotliar@ihep.ru, [d] Ekaterina.Popova@ihep.ru

Cluster management system is a core part of any computing infrastructure. Such system includes components for allocating and controlling over resources for different computing tasks, components for configuration management and software distribution on the computing hardware, components for monitoring and management software for the whole distributed infrastructure. The main goals of such system are to create autonomic computing system with functional areas such as self-configuration, self-healing, self-optimization and self-protection or to help to reduce the overall cost and complexity of IT management by simplifying the tasks of installing, configuring, operating, and maintaining clusters. In the presented work current implementation of the multicomponent cluster management system for IHEP computing center will be shown. For the moment this system consists of event-driven management system, configuration management system, monitoring and accounting system and a ChatOps technology which is used for the administration tasks.

Keywords: self-management, distributed computing, event-driven automation, ChatOps, cluster management system

1. Introduction

Computing infrastructure especially for distributed computing has a complex structure from many components and services distributed across a whole data center. A core part of such system is a cluster management system (CMS). It includes the following elements:

- hardware management system;
- services provisioning system;
- configuration and software package management system;
- monitoring system;
- operation management system.

The main goal of such system is to implement self-management principals from autonomic computing theory [1]. They are self-configuration, self-healing, self-optimization and self-protection. Where self-configuration is covered by configuration and software package management system and it allows to configure the cluster by itself automatically based on the high-level policies. Self-healing is covered by monitoring, configuration and operation management systems and it allows to automatically recover system or services in distributed environment from failures. Self-optimization also uses all these systems as a self-healing but it has a goal to continually seek ways to improve operation where hundreds of tunable parameters in the distributed system need to be adjusted for the best system effectiveness. The last aspects is self-protections. It is covered by all elements from cluster management system and protect software and hardware components of the computing infrastructure from cascading failures which could damage the infrastructure.

The current implementation of the multicomponent cluster management system for IHEP computing center described in this work. For the moment this system consists of hardware management system, provisioning system, event-driven management system, configuration management system, monitoring and accounting system. A ChatOps technology is used for the administration tasks and for the communications.

2. IHEP computing center overview

To have a reason for using a cluster management system the scale and complexity of the cluster need to meet some conditions which were hit at IHEP. Computer center at IHEP starts its history with the installation of Minsk-2 computing system in 1965. Over all these fifty-five 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 [2]).

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.

This center has a status of Tier-2 in WLCG with network connectivity by two 10Gb/s links to LHONE network and contribute on the third place to the whole grid-computing in Russia.

3. Prerequisites to the cluster managements system

Being of such size is not the only one condition for implementing the cluster management system. There are several other factors which cause its necessity. One of them is the human factor. Over the years of operating and growing, the number of system administrators has being decreased. This is shown on figure 1.

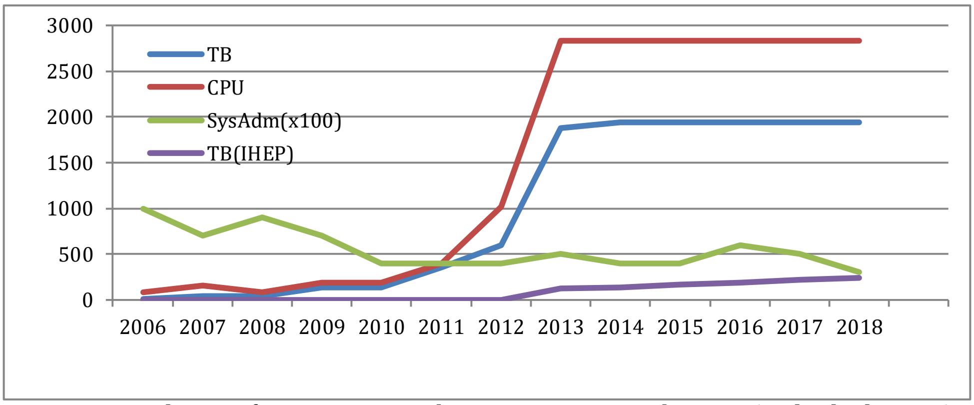

Figure 1. Evolution of IHEP resources by year in TB, CPU, humans (multiplied to 100)

So there are less humans and more servers in the computer center throught the time. All hardware that is used in the data center become cheap a generic purpose hardware. It causes increasing of the number of hardware parameters for monitoring and tuning for smooth and effective operation. And with evolution of data center software and software for the clusters it is appeared a new way of using infrastructure – software-defined infrastructure (SDI). SDI consists of software-defined infrastructure services, software-defined storages, software-defined networks. In such environment an ordinary system administrator becomes a site reliability engineer [3] where he has to use programming languages and automatization techniques to operate the distributed system.

4. IHEP requirements for the CMS

In IHEP environment the term "cluster management" defined as the set of common tasks required to run and maintain the distributed cluster. As mentioned before, these tasks are: infrastructure management, resource provisioning, configuration and software package management, monitoring, operation management. The main idea for CMS is to create self-management system by using already available open source software and developing only a few small pieces of system which are unique for IHEP data center and its operation model. From theory of autonomic computing such system should consist of hierarchy of blocks of autonomic managers and managed resources [4]. In their turn autonomic managers have monitoring, analyze, plan, execute components clued by knowledge base and managed resource has sensor and effectors in its touchpoint what is shown on figure 2.

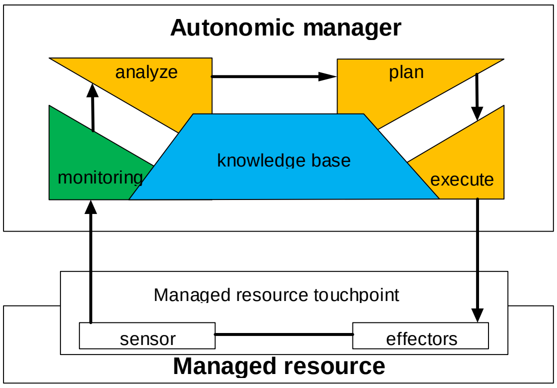

Figure 2. Autonomic manager and managed resource schema

In terms of CMS "managed recourse" is the computing cluster and "autonomic manager" is the multicomponent cluster management system itself.

Cluster management software already available as open source and as proprietary software. There are Apache Mesos, Kubernetes, OpenHPC, Rocks cluster Distribution, xCAT, Foreman, IBM Platform Cluster Manager and others systems which already are used by many clusters. But all this software mostly is built for specific types of clusters like high performance clusters or container based clusters. Not all of required features for IHEP are implemented and mostly mentioned software is created for specific Linux distributions that is also restrict its usage at IHEP. So it was decided to use software for the cluster management system which is satisfy the following conditions:

- tradeoff for functionality over complexity must meet IHEP environment;
- it has to support all operating systems which are used on IHEP cluster (Debian Linux, CentOS, Scientific Linux);
- it has to be open source with big community and has to be stable (not experimental);
- it has to be implemented in such way that different components could be installed independently on each other step by step;
- it has to be pluggable that it is possible to customize functionality for IHEP needs.

Based on all these criterias a multicomponent cluster management system was built at IHEP computer center.

5. IHEP cluster management system overview

Current implementation for the cluster management system is shown on figure 3. It should be mentioned that it still has two control loops: one through event-driven software based on StackStorm that automate all operations and second one is a classic way system with system administrator console.

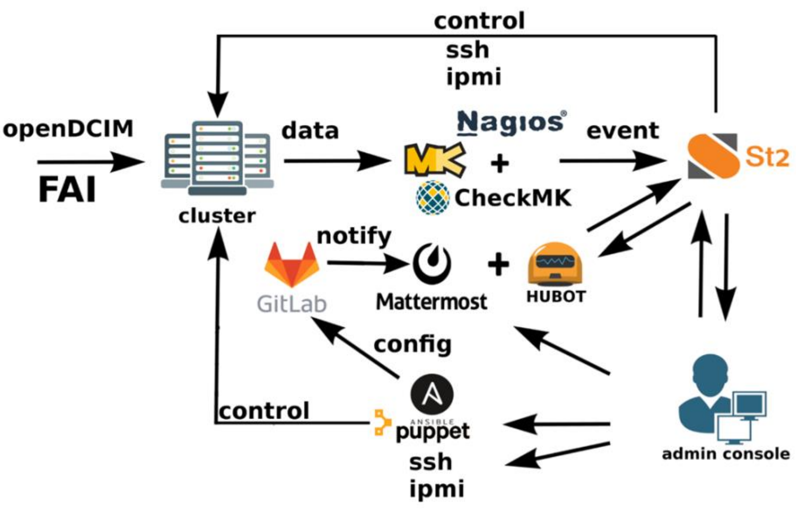

Figure 3. CMS overview

5.1. Infrastructure management system

To manage infrastructure openDCIM (data center infrastructure manager) software is used. That system hold the information about computing hardware, storage hardware, network hardware, management hardware, engineering infrastructure, physical location of the hardware and communication between each components. It is a cluster description database where everything that is known about the cluster components is stored. To manage cluster hardware a set of IPMI (Intelligent Platform Management Interface) tools are used. Linux ipmitool package or Supermicro IPMItool software allows to get full control over the hardware through baseboard management controller on each server. This control is used for management purposes and for monitoring. To control power distribution units (PDU) and uninterruptable power supplies (UPS) build-in software and SNMP (simple network management protocol) are used.

All these software components allow to implement several monitoring and control systems for engineering components [5]. Among them software to control power utilization by the hardware on the computing center, cooling parameters control software, power supplies parameters control. Implementation of the big red button (BRB) could be used as an example of using cluster description database. BRB is an IHEP data center feature that allow to power off the whole infrastructure as soft as possible in case of emergency. Usually these are cooling problems or also could it be a fire in the data center. A snippet of code for BRB action is shown on figure 4.

```
logger BUTTON PRESSED sending power off to WNs
echo "select label,DeviceType from fac_Device where DeviceType='server' AND
Cabinet in (select CabinetID from fac_Cabinet where DataCenterID='3' and
ZoneID='1');" | mysql -h dcim.ihep.su -u dcim  -u dcim --password=*** dcim |
awk '{ if ($2=="Server") { sizeAr=split($1,ar,","); for (i=1;i<=sizeAr;i++) print
ar[i];}}' | grep WN | xargs -iSERVER ipmitool -U ***-P ***-H SERVER chassis
power off
```

Figure 4. Code for powering off all servers

5.2. Provisioning

For services provisioning PXE with FAI (Fully Automatic Installation) system is used [6]. This system allows to install a compute node or a node for the storage system and has several features which are used in IHEP environment. They are: installation of Debian GNU/Linux, Ubuntu, CentOS, Scientific Linux; class concepts that supports heterogeneous configuration and hardware; central configuration repository for all install clients; full remote control via ssh during installation process; fast automatic installation for Beowulf clusters; hooks that can extend or customize the normal behavior. For unique custom build servers and for containers Ansible software is used as provisioning tool. It is simple and easy to learn, agentless, has easy understandable playbooks and big Ansible galaxy with many playbooks ready to use.

5.3. Configuration and software package management

Configuration and software package management system usually consists of tools that help to manage system configuration (files, dirs, permissions, etc.) software packages and software behavior. To manage configuration a content management systems is used. There is a software which does not have packages and installed on cluster in a software storage area (Ansys, Mathematica, etc.). Software distribution system based on CernVM-FS [7] is used for this purpose.

Puppet software manages configuration of the servers and the installed packages at IHEP cluster. Puppet is an extensible system with declarative language for a configuration description, it understands dependencies among packages, has a good reporting system about applied changes, has built-in feature for auto generation for documentation. Agent-server setups could have different versions and even different operating systems (really big benefit in IHEP cluster environment), puppet is easy to start using.

On the cluster one puppet server manages configuration for all computing and disk nodes. Whole configuration consists of 300 directories and around 500 files. It is used for Scientific Linux on computing nodes, Debian GNU/Linux 7,8,9 on disk nodes, CentOS7 on user interactive nodes. There are two configurations on the server for production use and pre-production testing system. They are both managed by git. That allows to use several branches for pre-production development. At IHEP cluster self-healing functionality of puppet heavily used (force daemon to run, force mount to be present and so on) and monitoring system checks that puppet agents always run and working.

5.4. Monitoring

Monitoring is one of the main building blocks for the cluster manager system. Depending on needs there are several monitoring systems which are used in production. At IHEP following software is used:

- Nagios with Check_MK to check computer center services;
- Splunk and central syslog to get and analyze all logs;
- Collectl – realtime monitoring;
- Elasticsearch and Kibana for engineering infrastructure;
- Pmacct and cacti for network traffic monitoring on the cluster;
- accounting system which is developed in IHEP.

Nagios with Check_MK is a primary monitoring system. At one monitoring server it is checked around 250 hosts and system has about 20000 service checks. The system has a live web interface with events history, all checks are configurable in one place. It is easy to add and configure checks for new servers or services. Besides Check_MK plugin system all Nagios plugins are available, it has good API for external access, as soon as Check_MK Python based it allows to write any new checks even with performance data in short time, powerful notification system and plugin architecture is used for seamless integration with other systems. At IHEP installation the system checks 340 services per second that means it needs only one minute to check all services for the cluster.

Second monitoring system is based on Elasticsearch and Kibana. This system mostly store unstructured data from the monitoring system for engineering infrastructure which is developed at IHEP. In cluster environment all possible data is stored for developing automate self-healing or self-protecting functions. The main benefit in the current setup is a wide set of features that Elasticsearch with Kibana brings to the user for data analyzes not only in real-time but also post factum. The easy way of external access for data, simple schema for data gathering systems, new technologies like machine learning and docker containers all make such system very powerful and promising. At IHEP this system gather all information about power units, cooling systems, SMART information for all hard drives on the cluster and more others [5].

5.2. Operation management

To perform cluster operation event-driven automation system StackStorm [8] is used at IHEP. This system allows automate all actions on IHEP cluster. Base tasks that are used are: automate remediation (self-healing and self-protection), create automated tasks for whole cluster, execute administrative commands on the cluster with full log support, create complex workflows from many tasks on the distributed system. Integration packs with many systems allow to bind Check_MK monitoring system with StackStorm and good API with micro services architecture let to develop all necessary custom components if needed.

StackStorm consists of four main logical parts like sensors, triggers, rules and actions. Sensors are python plugins for integration that receives or watches for events. When an event from external systems occurs and is processed by a sensor than an internal trigger emitted into the system. From event-driven system’s view triggers are representation of external events. At IHEP it is already used timers, webhooks and integrations triggers. Rules map the triggers to simple actions or to tasks as a set of actions (workflows) by applying matching criteria and by mapping trigger payload to action inputs. Actions are StackStorm outbound integrations. At IHEP mostly rules and actions are used for daily operations which are deployed in one pack. Some examples of the usage in production are using mistral workflows for kernel upgrade, self-healing the system if external software has bugs and mitigations procedures, data center crontab feature to run distributed data verification and some others.

To bring background work into a common place and synchronize all system administration works a ChatOps is used. Such operation paradigm allows to open fully IT daily operations and unify the communication about what should get done with action history of work being done.

6. Conclusion

At this work the multicomponent cluster management system at IHEP has been presented. All components based on open source software are in place and work for day-to-day operation. The whole infrastructure is going to the direction of fully autonomic self-managed system with minimum human intervention and prepared for the future with new hardware or software technologies that could be used at IHEP data center. All these make the computing center flexible and allow to use different kind of computations. By increasing availability and reliability parameters implemented system allowes to increase cluster effectiveness to the level of 75% where the effectiveness equal Availability multiplied by Reliability, Maintainability and Capability.

As a next step toward the fully autonomic system there are plans to add cluster states description database with allowed actions between the states by using finite-state machine mathematical model of computation theory applied to IHEP computing cluster. Also all self-features of current implementation need to be developed further.

References

[1] Kephart J.O., Chess D.M. The vision of autonomic computing // Computer 36. – 41-52 – 2003 – DOI:10.1109/MC.2003.1160055

[2] WLCG homepage [WLCG site]. Available at: http://wlcg.web.cern.ch/. (accessed 24.09.2018)
[3] Google site reliability engineering portal [Site reliability engineering] https://landing.google.com/sre/. (accessed 24.09.2018)

[4] White Paper. An architectural blueprint for autonomic computing // IBM. 2006

[5] Kotliar V., Anshukov V., Ezhova V., Gusev V., Kotliar A., Latyshev G., Shishov A. Development of the active monitoring system for the computer center at IHEP // CEUR Workshop Proceedings. February 2017: Vol. 1787. Pp. 317-322

[6] Kotlyar V., Popova E., Trushin K. Installation and setup of an IAAS private cloud based on OPENNEBULA with XEN hypervisor and LUSTRE file system. // Distributed computing and grid-technologies in science and education Proceedings. July 2012

[7] CernVM file system home page [CernVM-FS]. https://cernvm.cern.ch/portal/filesystem (accessed 24.09.2018)

[8] StackStorm documentation [StackStorm overview]. Available at: https://docs.stackstorm.com/overview.html. (accessed 24.09.2018)