Evolution, Design, Management and Support for the CMS Online computing cluster

Jean-Marc Andre[5], Ulf Behrens[1], James Branson[4], Philipp Brummer[2], Olivier Chaze[2], Sergio Cittolin[4], Cristian Contescu[5], Benjamin G. Craigs[2], Diego Da Silva Gomes[2], Georgiana-Lavinia Darlea[6], Christian Deldicque[2], Zeynep Demiragli[6], **Marc Dobson**[2], Nicolas Doualot[5], Samim Erhan[3], Jonathan Richard Fulcher[2], Dominique Gigi[2], Maciej Gładki[2], Frank Glege[2], Guillelmo Gomez-Ceballos[6], Jeroen Hegeman[2], Andre Holzner[4], Mindaugas Janulis[2a], Raúl Jimenez Estupiñán[2], Lorenzo Masetti[2], Frans Meijers[2], Emilio Meschi[2], Remigius K. Mommsen[5], Srecko Morovic[2], Vivian O'Dell[5], Luciano Orsini[2], Christoph Paus[6], Petia Petrova[2], Marco Pieri[4], Attila Racz[2], Thomas Reis[2], Hannes Sakulin[2], Christoph Schwick[2], Dainius Simelevicius[2a], Petr Zejdl[5b]

CHEP 2016, 10 – 14 Oct 2016, San Francisco, CA, USA

CMS Online cluster

The CMS Online computing cluster is composed of approximately 2000 computers, responsible for the control of the detector, the sub-detector electronics, the data collection and selection.

In order to support this cluster, core services are required.

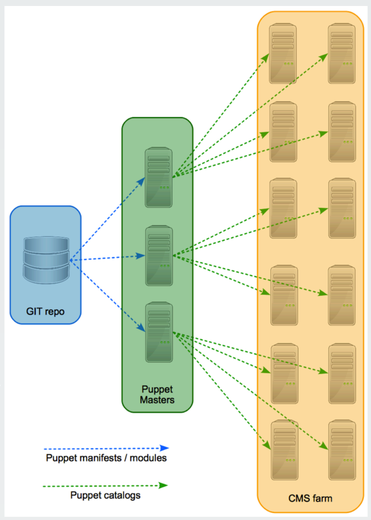
- There is a fully redundant, high performance distributed network for the control, configured and managed by CERN IT for CMS.
- Core system administration services, such as DNS, DHCP, LDAP, Kerberos, etc… are provided as redundant services, with own mechanisms, proxy, High Availability, or virtualization.
- To increase reliability, core services are redundant to rack level failure (failover to neighboring rack) or power failure (multiple sources, of which UPS; multiple PSUs).
- A fully redundant NetApp NAS provides the file storage needs (user home directories, project areas, software repositories) to all computers (Linux or Windows).
- A performant and flexible configuration management system, based on Puppet and Foreman, allows full support for the various sub-detectors and their evolution over time.

Configuration Management

Puppet

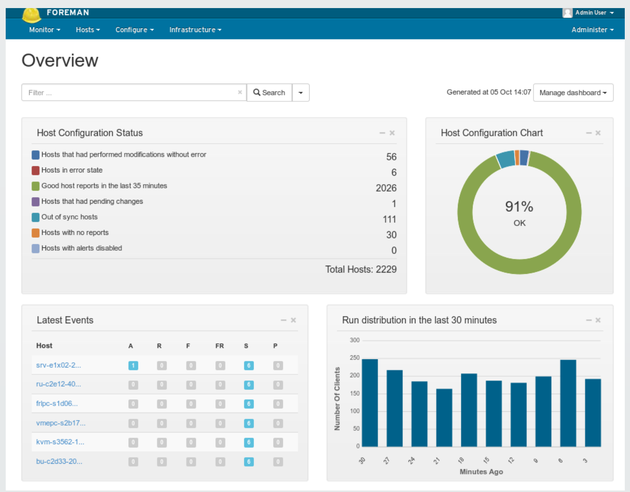
Puppet is a configuration management tool. It provides a performant and flexible management and configuration infrastructure to install, update operating systems in CMS computing farm.
Puppet reliably installs OS and makes sure the configuration is up to date.
A puppet node profile is built on top of single purpose configuration modules put together to create the node's manifest. From that node manifest, the puppet master will build a catalog of checks and actions to perform to match the described configuration. The catalog is then sent to the client which runs the checks and performs the corrective actions if necessary.
Here is the CMS puppet server infrastructure:

- puppet node manifests and modules definition are managed by GIT
- a DNS RR alias to the puppet masters spreads the load
- puppet uses YUM snapshotted repositories to install/update CMS software suits
- a puppet agent runs on every CMS node every 30mn and keeps the machines configuration up to date
- a report is sent by mail when a puppet run fails



Foreman

Foreman is a lifecycle management tool for physical and virtual servers. The CMS SysAdmin team uses it to install or re-install machines and collect the reports from puppet runs.
The current OS (SLC6 based on RHEL6) and future OS (CC7 based on CentOS 7) used on most of the cluster use kickstart files for installation. Foreman offers a powerful templating mechanism which is used to build custom kickstart files based on the hardware type and machine purpose. Therefore there is only one kickstart template (per OS) which considerably reduces maintenance time and errors.
Together with puppet, they make the CMS online cluster lifecycle management much easier.



Foreman main features:

- Bare metal installation of HW and virtual machines
- Nice web GUI
- Powerful CLI
- Kickstart templating
- Puppet ENC (External Node Classifier) (not yet used inside CMS)
- Global dashboard with indicators on Puppet's health

Software Repositories, Snapshots and dropbox

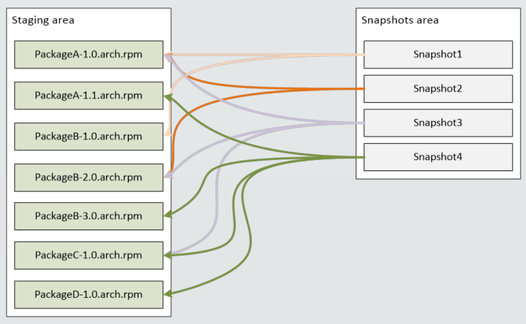
A unique mechanism to version yum repositories was developed to be able to select a specific version, and go forward or backwards between versions.

Snapshoting

- Using yum repositories, support rollback of installed software
- Yum distro-sync used (instead of the usual yum update) to force the package version from a repository snapshot
- Each repository has its own snapshots
- Hard-links used to limit disk usage of snapshots

Dropbox

- New packages added to the dropbox by SysAdmins
- Users can update existing packages in dropbox without SysAdmin intervention
- Built upon the snapshoting mechanism and mcollective
- Users can revert changes on their machines using the rollback capability of snapshots



Author Information

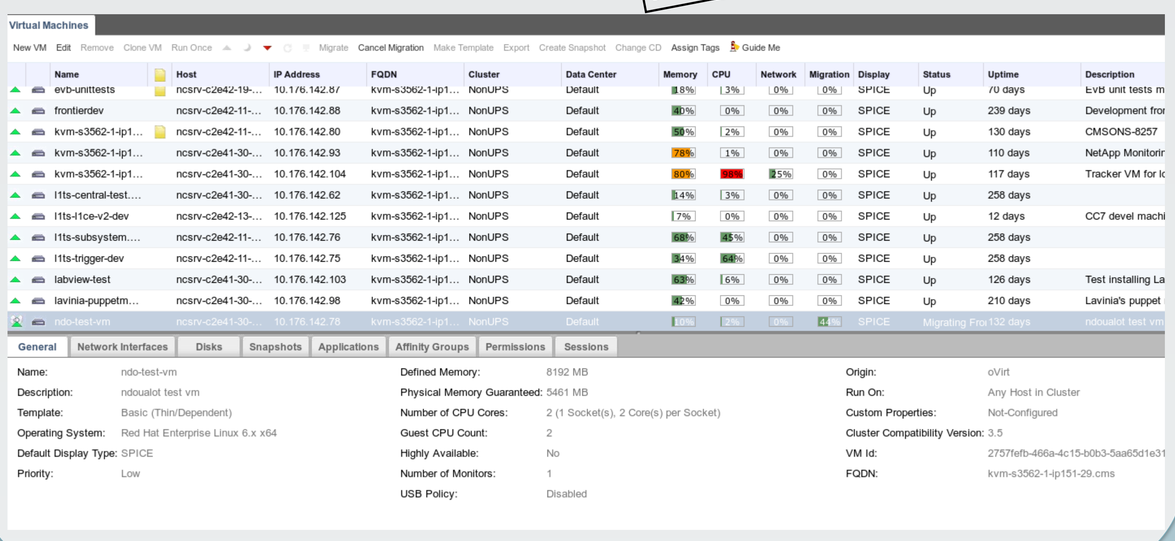
[1]DESY, Hamburg, Germany
[2]CERN, Geneva, Switzerland
[3]UCLA, Los Angeles, California, USA
[4]UCSD, San Diego, California, USA
[5]FNAL, Chicago, Illinois, USA
[6]MIT, Cambridge, Massachusetts, USA
[a]also at Vilnius University, Vilnius, Lithuania
[b]also at CERN, Geneva, Switzerland

Virtualization

oVirt is a virtual machines, storage and virtualized networks management tool. It is OpenSource and powered by KVM on Linux and runs on RHEL like operating systems.
CMS uses oVirt to run SLC6, MS Windows (Win7, Win8, Win2008, Win2012), and a few CC7 (for testing full cluster migration) virtual machines. Virtualization is more and more popular because it provides more flexibility and better utilization of the hardware machines.
oVirt helps CMS SysAdmin team to better manage CMS farm virtual machines:

- Virtual machines and hypervisors are managed at a cluster level
- Virtual machines disk images are stored on the NetApp NFS filer
- Virtual machines can be moved from one hypervisor to another in the same cluster
  - Allows maintenance operations on the hypervisor without interruption of service
- Several networks can be assigned to a cluster (Control Network + GPN)
- Virtual machines can be configured as HA
  - The VM will be restarted on another hypervisor is the one it is running on dies
- Very complete rights management system
  - Users can be given only the needed rights
- Full Admin web GUI
- User portal with console access
- HDD thin provisioning
- Network QoS
- Para-virtualized drivers (virtio)

Machines that do not require big HDD IO/s are likely to be virtualized.
CMS datacenter services (DNS, DHCP, LDAP, Kerberos, …), puppet masters, squids, web servers are running on VMs.



Monitoring

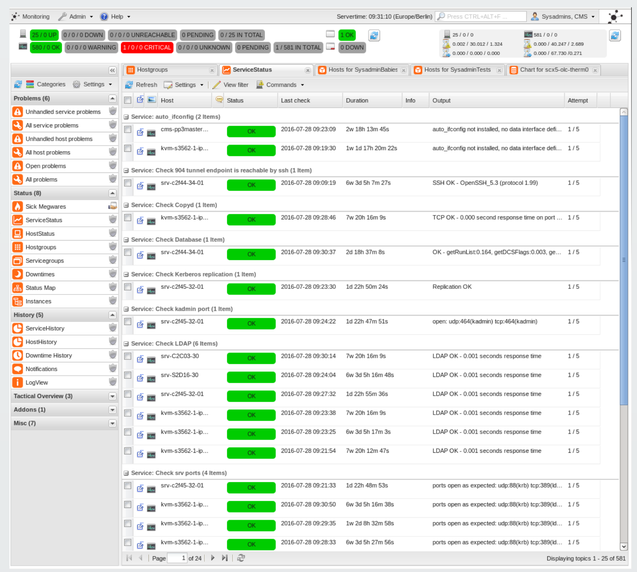
For this size of cluster, a reliable and comprehensive monitoring system is crucial to having a reliable and highly available computing system. To achieve this, our system and health monitoring is based on Icinga2 for all hosts and services, and Collectd with Grafana for performance metrics and more generic plotting.

Icinga2

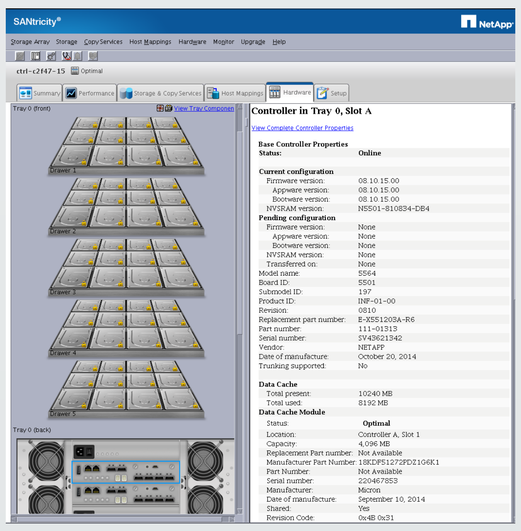
The large cluster with machines split over multiple sub-detectors and groups has taylored the way the system has been configured.

The features of our Icinga2 system are:

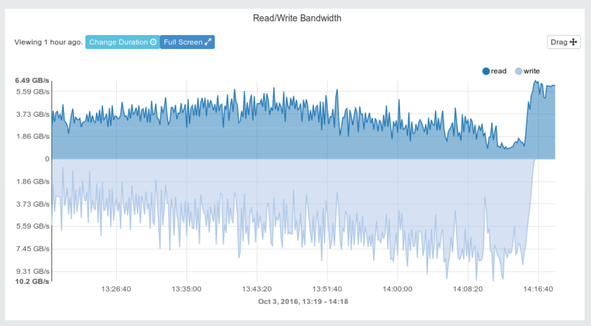
- Versions: icinga2-2.4.1-1, icinga-web-1.11.2-1
- 1 server: PowerEdge R620, 64GB RAM
- each sub-detector has their own user with access to their machines
- 2230 hosts, 54612 checks
- average host latency: 0, average service latency: 0.271
- alerting: per sub-detector



Specific HW monitoring



- Event data storage in CMS is on a SAN based on NetApp HW.
- Specific NetApp HW monitoring tool
- Assess health of the devices and send alarms



- Event data storage in CMS uses Lustre as a Global File System on top of the SAN.
- Intel monitoring and management tool
- Assess performance of the Lustre file system

Esper

Esper is an open source event series analysis and event correlation engine (CEP) in Java. It is used to parse incoming logs on the syslog server and find patterns of events, or correlations between events.
Simple rules have been implemented which find high load, blocked processes, repeated issues and alert the SysAdmin Team (E-mail, SMS).

Collectd + InfluxDB + Grafana

Collectd: agent running on client. Collects and sends metrics to InfluxDB.
InfluxDB: Time-series database (retention policy, down-sampling, replica, clustering)
Grafana: Dashboard, plots metrics from InfluxDB (accepts data from other sources, like elastic search, graphite, etc…)
Very useful for data correlation, e.g. post mortem analysis.