Monitoring of the infrastructure and services used to handle and automatically produce Alignment and Calibration conditions at CMS

Roland Sipos for the CMS Collaboration

Abstract

The Compact Muon Solenoid (CMS) experiment makes a vast use of alignment and calibration measurements in several crucial workflows in the event selection at the High Level Trigger (HLT), in the processing of the recorded collisions and in the production of simulated events. A suite of services addresses the key requirements for the handling of the alignment and calibration conditions such as recording the status of the experiment and of the ongoing data taking, accepting conditions data updates provided by the detector experts, aggregating and navigating the calibration scenarios, and distributing conditions for consumption by the collaborators. Since a large fraction of such services is critical for the data taking and event filtering in the HLT, a comprehensive monitoring and alarm generating system had to be developed. Such monitoring system has been developed based on the open source industry standard for monitoring and alerting services (Nagios) to monitor the database back-end, the hosting nodes and key heart-beat functionalities for all the services involved. This paper describes the design, implementation and operational experience with the monitoring system developed and deployed at CMS in 2016.

Presented at CHEP 2016 22nd International Conference on Computing in High Energy and Nuclear Physics
Monitoring of the infrastructure and services used to handle and automatically produce Alignment and Calibration conditions at CMS

Roland Sipos¹, Giacomo Govi², Giovanni Franzoni⁴, Salvatore Di Guida³, Andreas Pfeiffer⁴

¹ MTA-ELTE Momentum CMS Particle and Nuclear Physics Group, ² Fermi National Accelerator Laboratory, ³ Universita e INFN, Napoli, ⁴ European Organization for Nuclear Research - CERN

Abstract. The CMS experiment at Cern LHC has a dedicated infrastructure to handle the alignment and calibration data. This infrastructure is composed of several services, which take on various data management tasks required for the consumption of the non-event data (also called as condition data) in the experiment activities.

The criticality of these tasks imposes tight requirements for the availability and the reliability of the services executing them. In this scope, a comprehensive monitoring and alarm generating system has been developed.

The system has been implemented based on the Nagios open source industry standard for monitoring and alerting services, and monitors the database back-end, the hosting nodes and key heart-beat functionalities for all the services involved. This paper describes the design, implementation and operational experience with the monitoring system developed and deployed at CMS in 2016.

1. Introduction
The Compact Muon Solenoid (CMS) experiment[1] at the CERN Large Hadron Collider in Geneva, Switzerland[2], entails the production of two classes of sizable datasets both essential to meet the goals of its physics program:

- it acquires 1 kHz of events (1 MB each) for a few months a year; each raw event is processed to enable data analysis by the world-wide collaboration;
- it generates simulated events comparable in number and size to the datasets collected with the experimental apparatus.

To ensure the availability and optimal performance in the exploitation of such datasets, CMS has a complex infrastructure to handle the non event data which describe the alignment and calibration of all its sensitive elements. In the following we refer to such non event data also as condition data. Such infrastructure is composed by several services which take on various tasks such as:

- recording the status of the experiment and of the ongoing data taking,
- updating condition data as provided by the detector experts,
- aggregating and navigating the different set of calibration and alignment constants,
• and distributing condition data for consumption in the data processing and analyzing work-flows.

Such services are crucial in the execution of the main data processing work-flows of the CMS experiment, such as: the event selection at the High Level Trigger (HLT), the reconstruction of the recorded collisions, and the production of simulated events.

Because of such criticality, the Alignment Calibration and Database team in CMS (AlCa/DB) developed a comprehensive monitoring and alarm generating system as described in this paper. Such monitoring system was developed based on the Nagios open source industry standard for monitoring and alerting services, and assesses the status of the database back-end, of the hosting nodes and of key heart-beat functionalities for all the services involved.

2. Monitored entities
The aim of the monitoring infrastructure is probing and notifying the status of the services the CMS condition data rely upon, as well as the virtual machine hosting such services. Following the standard approach of solutions based on the Nagios technology, we divide our target entities into **hosts** and **services**.

2.1. Hosts
By hosts we mean the virtual machines where services required by the CMS condition work-flows are executed. The configuration of the monitoring infrastructure holds one main host-group of 10 items, including all the virtual machines which are part of a cluster used by the AlCa/Db team - we refer to such cluster as CMS CondDB Web Production cluster. Within the CondDB Web Production cluster, hosts are divided into two main categories:

• **Frontend** machines, that serve as web proxies to access the applications from outside of the CERN network
• **Backend** machines where the actual service applications are deployed

For both categories, there are four instances for the services: test, development, integration, and production; each level is deployed on a separate virtual machine. In addition, there are a few special machines providing common, dedicated services:

• **DBAccess** - running the dbaccess service, which is used by all other applications as interface to write and update contents in the oracle master database holding the payloads of the CMS non event data
• **CondDB-Mon** - hosting the monitoring service subject of this paper,
• **CondDB-CC7-Test** - for general testing exercises, with a CERN CentOS7 Linux installation

The monitoring of the **hosts** is limited to checking whether the corresponding virtual machine is reachable and responds to the queries of the monitoring infrastructure.

2.2. Services
The CMS condition work-flows require the availability of multiple services, either part of the CERN IT infrastructure or provided by other projects in CMS. Such **Services** present a wider variety compared to the hosts, and may have dependencies on each other. Since some services also need to rely on sources of monitoring information other than those generated by tools of the AlCa/Db team, we defined a number of “service-groups” to aggregate collections of related services. The following service-groups are defined:

• **Host health checks** - foreseeing general host related checks such as CPU, Memory and network utilization, disk I/O information, file-system, and other status information about the machines.
• **Infrastructure health checks** - foreseeing checks that are depend on the specific architecture of each external system. Examples for this are the checks for the EOS file system mount points and their availability on the relevant hosts, or the load on the Oracle database servers (both examples are of services centrally managed by CERN’s IT department).

For the services whose status is not directly available to the CMS monitoring, standalone applications pulling information have been developed, called “local checks”. These are executables, developed in Python, are run by the monitoring agents. We aggregate the output of every “local checks” output and categorize them as follows:

- hosts provided by the CERN IT, for instance the databases; information for these services is acquired via the meter-client.
- additional monitorables retrieved from the CMS online infrastructure, for services running on hosts located in local area network at the physical site of the CMS experiment

We also categorized some special workflows of the CMS condition data operations into the following three service groups:

- **Frontier services** monitoring the infrastructure which takes care of caching the CMS condition data for consumption by the clients; Frontier has two separate instances serving the online and offline consumers
- **Online to Offline (O2O)** monitoring the operation-critical work-flows responsible for the replication of detector- or run- specific configurations
- **DB Access**, the set of services providing a RESTful API for accessing the online condition database from the offline environment.

### 3. Technologies employed

The monitoring service subject of this paper is deployed on a CERN CentOS 7 node, where the OMD (Open Monitoring Distribution) package is installed; OMD significantly simplifies the otherwise rather tedious work of manual installation and configuration of different Nagios plugins. OMD bundles Nagios together with the most important add-on’s that can be installed with ease on every major Linux distribution. The most important monitoring components from the OMD distribution which we are actively using, are:

- **Check-MK** for automatic service recognition
- **MK Livestatus**, the Nagios-Broker-Module that provides direct connection to status data on hosts and services, using UNIX sockets
- **WATO**, the Web Administration Tool that supports the complete administration of the OMD sites over a browser
- **Notify**, a generic and flexible notification engine that runs with a rule-based approach

The general scheme of the setup is seen in figure [1].

The various checks are set-up within Check-MK in the following way:

- **Inventory checks** - Thanks to Check-MK agent plugins, every monitored host is capable of sending a wide range of check results. These checks are activated through WATO on a per-host level, where we select which metrics we would like to monitor. The most important services are grouped up to different service-groups for easier handling:
  - **Host-health checks** General information about the host: CPU Load and utilization, file-system and disk usage, kernel process creations, context switches, major page faults, etc.
Process counts These particular checks are monitoring if some services are running as a process on a given host. Currently we have 2 service groups using this feature: the EOS daemon process check on the backend machines, and the DBAccess related process checks.

Local checks These checks only exist on the monitoring master node, and we use them to pull information from 3rd party monitoring sources.

- O2O The O2O jobs are monitored through a logging mechanism based on the condition database. To check the safe and sound operation for each job, there is an SQL query that looks for errors for each job. Based on the query result, the OMD site evaluates the state of the job.
- Online Environment To gather information from services running in the CMS online environment, which is protected by a firewall, a special access to the Online monitoring system was set up. Through this, we pull information via a REST-API about the following metrics:
  - OnlineFrontier checks weather the Frontier based caching layer in the online environment is working properly.
  - O2O Virtual Machines get monitored for their system load. These machines are deployed in the online environment.
  - The Meter-client provides access to several services provided by the CERN IT. We use the command line tool, called meter-client, provided by the CERN IT team, to query these metrics relevant for us:
    * Online DB Load: The CPU load of the four-node of the Oracle RAC that holds the CMSONR instance.
    * Online DB Availability: The availability information about the CMSONR instance.
    * Offline DB Load: The CPU load of the three-node Oracle RAC that holds the CMSR and CMSONR-ADG instances.
    * Offline DB Availability: The availability information about the CMSR and CMSONR-ADG instance.
    * Offline Frontier Availability: The availability information of the Frontier instance used by the CMS Tier-0 system and the Offline Frontier Launchpads.
    * cmsdev03 cpuload: Checking the load on the “Payload Inspector” back-end, a service to the users giving access to visualisation of selected conditions data.

4. State transitions and their severity
State transitions can occur for a host or a service. Once a state transition is detected by the monitoring system, the corresponding service is flagged, and a notification (e-mail or SMS) is sent to the experts associated to that service.
4.1. Grace period
Check-MK differentiates between SOFT and HARD states. Once a transition of a service or host into a faulty state has been detected, the service or host is flagged as being in a “soft” failed state. If after a consecutive number of checks the service or host didn’t recover from the soft failed state, the service or host is flagged as being in a “hard” failed state. This feature allows to limit the number of notifications for minor issues where the services recover quickly, as they may happen due to short network issues/glitches. In the case of these short-time fluctuations in states, we have defined a single global, and a few service-specific, grace periods, in which the monitored entities can “recover”, i.e. transition back to a “healthy” state, without sending an explicit notification to the experts. If the service or host doesn’t recover over the following five consecutive checks, its state is set to a “hard” failed state, and notifications are sent about the issue.

5. Alarm system
For visualization of the states of hosts and services we use the NagVis plugin, which allows to create maps to show the overall state of the defined service-groups. These maps are also used in the control room of the CMS experiment, for real-time monitoring during the data taking.

![Figure 2. Overview of the OMD setup for the CMS Conditions Web Production.](image)

Each service-group has a standalone map for easier navigation to the related services. Thanks to the WATO managed authentication mechanism, guest users can also open the Check-MK view for the given service, for additional data and better insight. This allows any member of the collaboration to check and verify the status of the CMS condition data services.

In addition to the real-time monitoring aspect of our setup, we had another key requirement: a flexible and straightforward way to create and customize a notification and alarming sub-system. For this we use the rule-based notification mechanism implemented in Nagios, allowing us to send SMS to alert the on-call experts, while other members of the team are notified by e-mail, which contain more detailed information about the issues.

6. Summary
With the monitoring infrastructure described in this paper, we can ensure that the operation of the CMS non event data work-flows proceed fully and correctly. Thanks to the third party monitoring sources, we can also aggregate the state of external services which the Condition data services depend on. We provide a view of the global state of the relevant services, and notify the responsible parties in the case of failures and critical problems.

7. Acknowledgment
We would like to express our gratitude to Giovanni Franzoni and Marco Musich for their feedback and support, and making the monitoring upgrade possible, also for Katarzyna Maria Dziedziniwicz-Wojcik and Georgiana Lavinia Darlea for the help with the 3rd party data sources.
References

   JINST 3 S08004, doi:10.1088/1748-0221/3/08/S08004

   JINST 3 S08001

   http://stacks.iop.org/1742-6596/664/i=4/a=042024

[4] https://support.nagios.com