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

Roland Sipos\textsuperscript{1}, Giacomo Govi\textsuperscript{2}, Giovanni Franzoni\textsuperscript{4}, Salvatore Di Guida\textsuperscript{3}, Andreas Pfeiffer\textsuperscript{4}

\textsuperscript{1} MTA-ELTE Momentum CMS Particle and Nuclear Physics Group, \textsuperscript{2} Fermi National Accelerator Laboratory, \textsuperscript{3} Universita e INFN, Napoli, \textsuperscript{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\cite{ref1} at the CERN Large Hadron Collider in Geneva, Switzerland\cite{ref2}, produces two classes of sizable datasets that are essential to meet the goals of its physics program:

- \textit{raw data} - CMS acquires 1 kHz of events (\sim 1 MB each) for a few months a year; each raw event is processed to enable data analysis by the world-wide collaboration;
- \textit{simulated data} - CMS 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 that describe the alignment and calibration of all its sensitive elements, the so-called condition data. The 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 after validation by the detector experts;
- aggregating and navigating the different set of calibration and alignment constants;
- distributing condition data for consumption in the data processing and analyzing workflows.
These 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. As a result, the Alignment Calibration and Database team in CMS (AlCa/DB) developed a comprehensive monitoring and alarm generating system as described in this paper. The monitoring system uses the Nagios open source industry standard for monitoring and alerting services.

2. Monitored entities
The monitoring infrastructure is set up to assess both the status of the database back-end nodes and the machines running the services, and selected information about the status of key processes of the conditions work-flows (e.g. if periodic updates of selected conditions succeed or fail). Following the standard approach of solutions based on the Nagios technology, these are classified as hosts and services, respectively.

2.1. Hosts
The configuration of the monitoring infrastructure holds one main host-group of 10 items, including all the (virtual) machines which are part of the AlCa/Db infrastructure. Within this cluster, hosts are divided into two main categories:

- **Frontend** machines serving as web proxies to access the applications from outside of the CERN network;
- **Backend** machines where the actual service applications are deployed: they are firewalled inside the CERN network.

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 the interface to insert and update contents in the Oracle master database hosting the CMS conditions data;
- **CondDB-Mon** - providing the monitoring service;
- **CondDB-CC7-Test** - for testing the services on the next version of the operating system (CERN CentOS7 Linux).

The monitoring of the hosts checks whether the corresponding virtual machine is reachable and responds properly 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. Services may depend on each other, and therefore can rely on sources of information other than the tools managed by the AlCa/Db team. Consequently, a number of “service-groups” that aggregate collections of related services are defined:

- **Host health checks** - general host related checks such as CPU, Memory and network utilization, disk and file-system information, and other status information about the machines;
- **Infrastructure health checks** - checks that depend on the specific architecture of each external system. For example, 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 services are centrally managed by CERN IT department).
For the services whose status is not directly available to the CMS monitoring, standalone applications have been developed, called “local checks”, which collect the required information. These are typically Python scripts run by the monitoring agents. We aggregate the output of every “local check” and categorize them as follows:

- information for services on hosts provided by the CERN IT (e.g. the database machines) is acquired via the meter-client;
- additional monitorable quantities for machines residing in the “online” local-area network of the CMS experiment are retrieved from gateways in the CMS online infrastructure.

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

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

3. Technologies employed
The monitoring service is deployed on a CERN CentOS 7 node, using the Open Monitoring Distribution (OMD) [5] package. OMD simplifies significantly the otherwise rather tedious work of manual installation and configuration of different Nagios plugins. OMD bundles Nagios together with a number of addons, which can be easily installed on every major Linux distribution. The most important monitoring components from the OMD distribution that we are actively using are:

- Check-MK - for automatic service recognition;
- MK Livestatus - the Nagios-Broker-Module providing 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.

![Figure 1. The OMD setup for the CMS CondDB Web Production.](image)

The various checks are set-up within Check-MK in the following way:
- Inventory checks - every monitored host is capable of sending, via Check-MK agent plugins, a wide range of check results. These checks are activated through WATO on a per-host level, where we select which metrics we 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 specific services are running as a process on a given host. Currently we have two service groups using this feature: the EOS daemon process check on the backend machines, and the DBAccess related process checks.
- Local checks - These checks are running on the monitoring master node, and are used to pull information from third party monitoring sources (e.g. the CMS online gateways).
  - O2O - The O2O jobs are monitored through a logging mechanism based on the condition database. In order 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 gateway access to the Online monitoring system was set up. Through this, we pull information via a REST-API about the following metrics:
    * Online Frontier - checks whether the Frontier based caching layer in the online environment is working properly.
    * O2O Virtual Machines - these get monitored for their system load. These machines are deployed in the online environment and therefore the monitoring uses the dedicated online gateway.
  - Meter-client - this provides access to several services provided by CERN IT. We use their command line tool, called meter-client, to query a number of metrics relevant for our monitoring.

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 responsible for 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 did not 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 by themselves, as 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 does not recover during the following five consecutive checks, its state is set to a “hard” failed state, and notifications are sent.
5. Alarm system
For the visualization of the states of hosts and services we use the NagVis plugin to create maps showing 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.

![Diagram of alarm system]

**Figure 2.** Overview of the OMD setup for the CMS Conditions Web Production.

Each service-group has a standalone map for easier navigation to its related services. The WATO managed authentication mechanism also allows guest users to access the Check-MK view for specific services. This allows any member of the collaboration to check and verify the status of the CMS conditions data services.

In addition to the real-time monitoring aspect of the AlCa/DB infrastructure, another key requirement is 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 alert the on-call experts of the affected service by sending them a SMS message with the basic information about the failure, while other members of the team are notified by e-mail that contains more detailed information about the issues.

6. Summary
The monitoring infrastructure described in this paper ensures the correct operation of the CMS conditions data work-flows. Using third party monitoring sources, the state of external services, which the Condition data services depend on, is aggregated and incorporated into the monitoring. A view of the global state of the relevant services is provided to all members of the collaboration, and responsible parties are notified in the case of failures and critical problems.

7. Acknowledgment
We would like to express our gratitude to Marco Musich for his feedback and support, and making the monitoring upgrade possible, also for Katarzyna Maria Dziedziniewicz-Wojcik and Georgiana Lavinia Darlea for the help with the third party data sources.

References
[1] CMS Collaboration “The CMS experiment at the CERN LHC” (2008) *JINST* 3 S08004, doi:10.1088/1748-0221/3/08/S08004
[2] L. Evans and P. Bryant “LHC Machine” (2008) *JINST* 3 S08001, doi:10.1088/1748-0221/3/08/S08001
[3] S. Di Guida, G. Govi, M. Ojeda, A. Pfeiffer, R Sipos “The CMS Condition Database System” (2015), *J. Phys.: Conf. Series* 664 042024. http://stacks.iop.org/1742-6596/664/i=4/a=042024
[4] https://support.nagios.com
[5] http://omdistro.org/start