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

Roland Sipos (MTA-ELTE Momentum CMS Particle and Nuclear Physics Group),
Andreas Pfeiffer (European Organization for Nuclear Research, CERN), Giacomo Govi (Fermi National Accelerator Laboratory), Salvatore Di Guida (Universita e INFN, Napoli)

On behalf of the CMS Collaboration

INTRODUCTION

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**.

HOSTS AND SERVICES BEING MONITORED

Our monitoring configuration contains the following **hosts-groups**:

- CMS **ARM cluster** (2 hosts)
- CMS **MAC cluster** (7 hosts)
- CMS **CondDB Web Production cluster** (10 hosts)

The most important services that are responsible for the safe and sound operation of the calibration workflows are running under the CondDB Web Production host-group. The hosts of this group are the following:

- **Frontend machines,** that serve as proxies to access the applications from outside CERN.
- **Backend machines** where the applications are deployed.

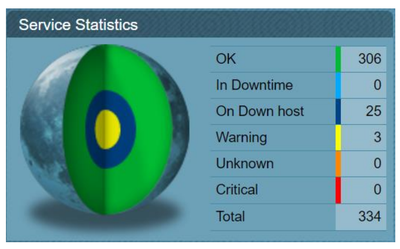
For both sub-groups, there are **development, integration and production** machines for the services, following the standard way of handling development cycles.

Machines of CondDB Web Production

2 Development	2 Integration	2 Production

We defined service-groups to make collections of interrelated services. These groups are defined as follows:

- **Host health checks**: This group contains general host-related checks: CPU, RAM and network utilization, disk I/O information, filesystem status, etc.
- **Infrastructural**: This group has checks that are monitoring architectural dependencies for other service's operation.



We have a special group of standalone applications, called **local checks**[1]:

- *Python executables*, that run by the monitoring agents through auto-discovery, grouped by their related source;
- This is the way how we query the status of CERN IT provided hosts, databases and services through the *meter-client*;
- Additional data also fetched from the online *Icinga2* monitoring system for services, that are running on hosts that are located in the online environment;
- Finally we *aggregate* the output of every local checks' output.

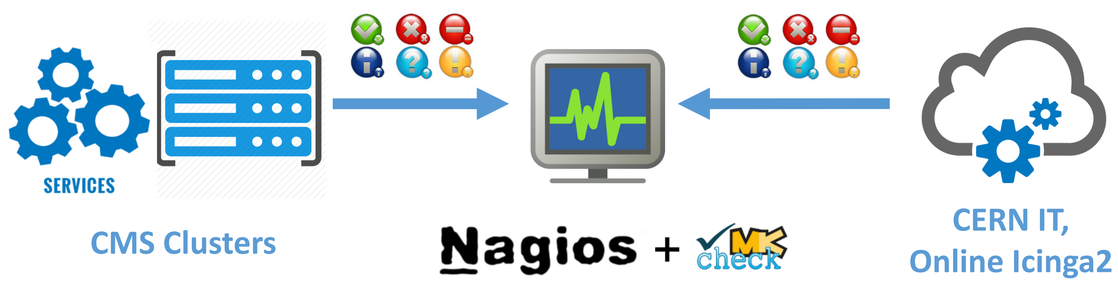
Other monitored, *special workflows*:

- **Frontier services**, in order to ensure that the caching mechanism works properly in both the online and offline environment.
- **Online to Offline (O2O)** is a mechanism that is responsible for the replication for a specified, and operation critical data set.
- **DB Access** is a crucial service that provides a RESTful API for accessing the online conditions database from the offline environment.

Databases	Online Database	Offline Databases
Frontier services	Online Frontier	Offline Frontier
O2Os	O2O hosts	O2O jobs
CondDB apps	App Host Payload Inspector host Load	DB Upload DB Access

USED TECHNOLOGIES

With the **Open Monitoring Distribution (OMD)**[2] we can avoid the tedious work of manual installation and configuration of different Nagios plugins. The package bundles Nagios together with the most important addons that can be installed with ease on every major Linux distribution.

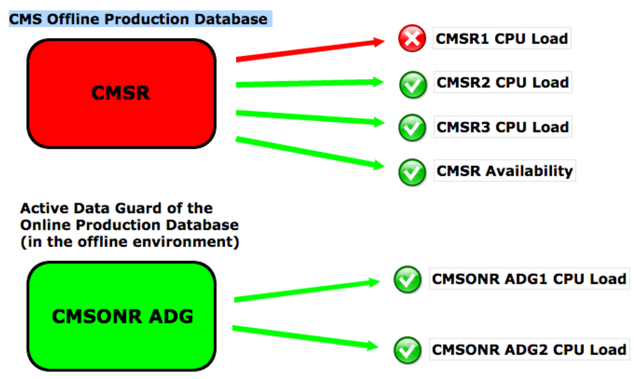


The most important monitoring sub-systems that we are actively using, are the following:

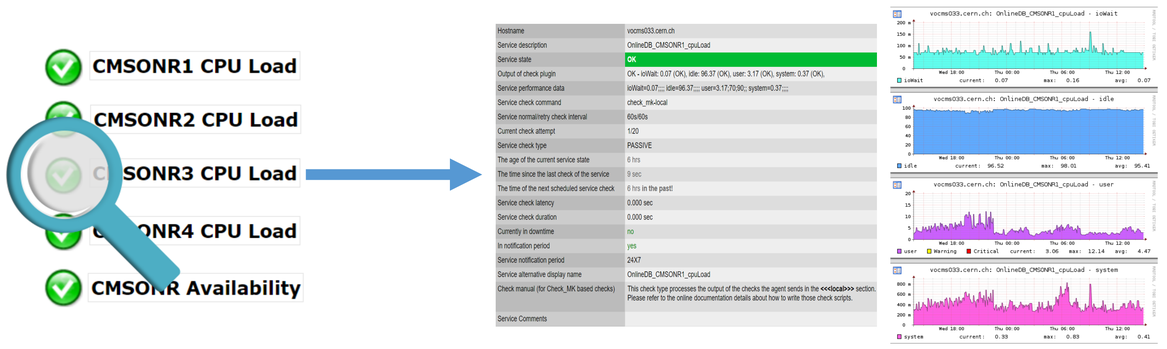
- **Check_MK** – Automatic service recognition, instead of manual Nagios configurations.
- **MK Livestatus** – Nagios-Broker-Module that provides direct connection to status data on hosts and services, using UNIX socket.
- **WATO** – A Web Administration Tool that supports the complete administration of the OMD sites over a browser.
- **Notify** – Simple and flexible notification configurations with rule-based aproach.

For visulization we use the **NagVis** plugin, with we use to create maps for showing a global state of different service-groups.

These maps are also used in the online environment for *real-time monitoring during the data taking*, ensuring that the conditions databse related services are up and running.



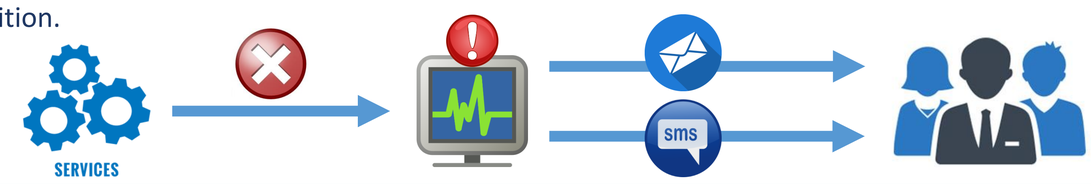
For each service-group there is a standalone map for easier navigation to the related services. Thanks to the WATO managed authentication mechanism, guest users can also open the *Check_MK view for the given service, for additional data and better insight*.



NOTIFICATIONS

On top of the real-time monitoring aspect of our setup, we had another key requirement: a flexible and straightforward way to create a notification and alarming sub-system.

For this we use the rule-based notification mechanism to send **SMS to the on-call experts** and the offline experts are **notified by e-mail**, that contains detailed information about the state transition.



CONCLUSIONS

With the upgraded monitoring infrastracture we can ensure that our operations in the conditions database web production are working safe and sound. Thanks to the third party monitoring sources, we can also aggregate the state of our dependency services and a global state of our services, and notify the responsible parties in the case of failiures and critical problems.

REFERENCES

[1] Local checks of CondDB-Monitoring - https://gitlab.cern.ch/cms-ppdweb/cmsDBCondMon (2016 October)
[2] The Open Monitoring Distribution - http://omdistro.org/ (2016 October)

ACKNOWLEDGEMENT

I would like to express my gratitude to Giovanni Franzoni, Giacomo Govi, Marco Musich and Andreas Pfeiffer for their feedback and support, and making the monitoring upgrade possible, also for Katarzyna Maria Dziedziniewicz-Wojcik and Georgiana Lavinia Darlea for the help with the 3rd party data sources.