

JINR Tier-1 service monitoring system: Ideas and Design

I.S. Kadochnikov, I.S. Pelevanyuk^a

Joint Institute for Nuclear Research,
Joliot-Curie 6, 141980 Dubna, Moscow region, Russia

E-mail: ^apelevanyuk@jinr.ru

In 2015, a Tier-1 center for processing data from the LHC CMS detector was launched at JINR. After a year of operation it became the third among CMS Tier-1 centers considering Completed Jobs. The large and growing infrastructure, pledged QoS and complex architecture, all make support and maintenance very challenging. It is vital to detect signs of service failures as early as possible and have enough information to react properly. Apart from the infrastructure monitoring, which is done on the JINR Tier-1 with Nagios, there is a need for consolidated service monitoring. The top-level services that accept jobs and data from the Grid depend on lower-level storage and processing facilities that themselves rely on the underlying infrastructure. The sources of information about the state and activity of the Tier-1 services are diverse and isolated from each other. Several tools have been examined for the service monitoring role, including HappyFace and Nagios, but the decision was made to develop a new system. The goals are to retrieve a monitoring information from various sources, to process the data into events and statuses, and to react according to a set of rules, e.g. to notify service administrators. Another important part of the system is an interface visualizing data and a state of the systems. A prototype has been developed and evaluated at JINR. The architecture, current and planned functionality of the system are presented in this report.

Keywords: grid-computing, monitoring, Tier-1

© 2016 Ivan S. Kadochnikov, Igor S. Pelevanyuk

Introduction

The history of the JINR participation in grid-computing for LHC started back in 2001 [Tikhonenko, Korenkov, 2001]. By the 2004 the JINR computing center was fully integrated into the WLCG/EGEE environment as the grid site called "JINR-LCG2". It still serves as Tier-2 center for LHC experiments. In 2011 it was decided to create Tier-1 center for all LHC experiments in Russia. According to the plan [Astakhov, Baginyan, Belov, 2016] Tier-1 center for ATLAS, ALICE and LHCb experiments were implemented in NRC "Kurchatov institute" (NRC KI, Moscow), and Tier-1 for the CMS experiment was built in JINR. It became the 7th CMS Tier-1 center in the world. JINR Tier-1 has been already included in the first four ones as to its resources and indicators of reliability, which according to the QoS applied to Tier-1's, should be close to 100% of resource availability and reliability. Complex monitoring system is required in order to allow maximal uptime. Hardware monitoring system for the computing center is based on Nagios and satisfies its demands. But it does not guarantee availability of important services which are available in Tier-1. To cover them and to support the local admins the decision to develop a monitoring system of higher level was taken.

Service failures

The Tier-1 CMS infrastructure at JINR consists of the following elements (services): Data storage subsystem, Computing system (Computing Elements), Data transfer manager (Phedex), Data transfer subsystem (FTS), Management of data transfer and data storage (CMS VOBOX), Load distribution system, and CMS Tier-1 network infrastructure [Astakhov, Baginyan, Belov, 2015]. These services could fail due to many reasons: misconfiguration, network problems, bugs in software, problems with integration between systems of different levels. To fix any problem with services the admin needs first to notice that problem exists via notification or after manual check of dedicated web-page. Sometimes it is easy and sometimes it requires expertise to identify the problem. When a problem is detected it is necessary to understand the cause. Information for that can be in local logs and systems or it can be on external web-pages and monitoring systems.

Due to these specifics many errors are detected after some time and could cause violation of QoS. Detailed problem investigation sometimes requires to read through several different available information resources.

Service monitoring system

In order to simplify daily operations, problem detection and identification the decision to develop dedicated monitoring system was taken. The main requirements to this system are the following:

1. Collect monitoring data from different sources (local and external).
2. Automatically detect problems and issues.
3. Give notification to admins.
4. Provide a web-page as a single place to investigate reasons of problems and issues.

The building of such a system is a difficult task because there are many monitoring data sources, all have different data access methods (API, command line, HTML pages), visualisation of monitoring data require custom interface elements (plot, table, list, text), and detection of problems should be custom for all services. All these constraints impose modular design according to which particular module for monitoring of a service should be able to collect raw data, analyze it, react on problems and contain web-part in order to show particular monitoring results in the best way. So the whole monitoring system is based on many monitoring modules, core modules (database access, utils) and web-server.

In order to unify the process of reaction on problems, to allow a simpler status representation of the whole system, and to provide possibility to use machine learning methods for analysis and prognosing, a notion of event has been introduced to the system. Event is not necessarily negative, it could be any significant change of system status: or appearance/disappearance of error, big increase or decrease in transfers rate. This allows to look at the work of a services from higher level and go deeper into details only in case of errors. This also means that it is possible to apply Complex Event Processing methods to the flow of events [Etzion, Niblett, 2010].

Architecture

The system has a modular design. There are several modules dedicated to different information resources. A module can retrieve raw data, analyze data (generate events), and react to events. It also contains a list of methods which return data needed for web-interface (REST methods). To manage configuration, database and all the other modules there is a special module called "core". It contains all auxiliary methods and functions. Collection of data is initiated by a cron job. All modules and core are written in Python. They operate independently from web-interface. That means that first three requirements described in the previous section can be satisfied by installing and configuring monitoring core and modules. The schematic view of the modules is given on Figure 1.

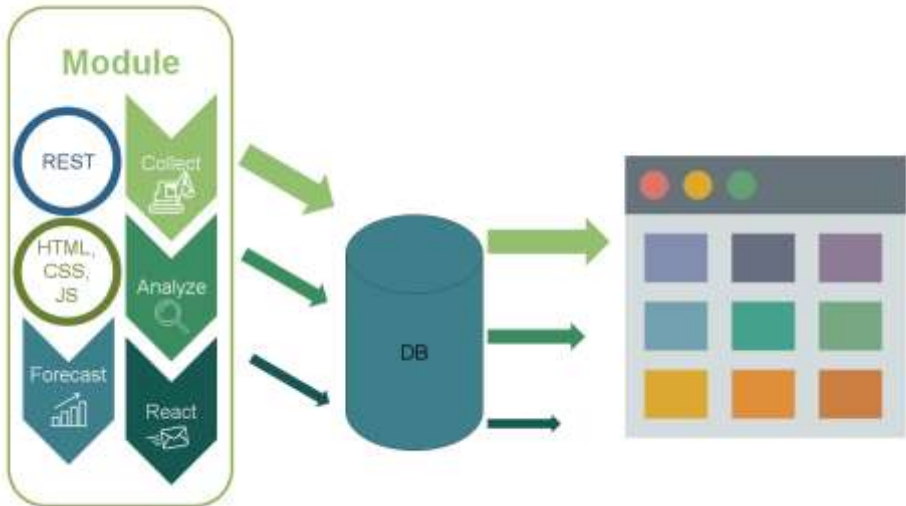


Fig.1. Monitoring module schema

In addition to that it is possible to configure a web-interface. It has two purposes: to give pure an HTML page to the users and to serve as a REST server by returning monitoring data in JSON format for HTML pages. The part of the server responsible for REST does not need to be changed when modules are added or changed since all possible REST urls are generated automatically from the monitoring modules themselves. The Javascript and HTML part responsible for one particular module should be placed in one directory with the monitoring module. Now the web-server is built on Django but it can be replaced later with any other suitable technology. As a web-framework Angular and AdminLTE2 were chosen. Angular simplify managing many small application on a single web page. AdminLTE2 provides modern page layout and design.

Conclusion

Development of a completely new system requires decent amount of time and a lot of expertise. Our service monitoring system is still in a development phase but concepts it is built on proved to be effective. The architecture of the system is completely modular. The most important task now is to implement modules (both server part and web-interface part) for all the services.

The monitoring system described here could be useful not only for Tier-1 center monitoring in JINR but also for other systems which should be monitored by collecting and analyzing raw data. The system could be useful as an additional monitoring and alarming tool for distributed computing system of the BESIII [Belov, Deng, Korenkov, 2016] experiment or JINR Cloud infrastructure [Baranov, Balashov, Kutovskiy, 2016].

References

- Tikhonenko E. Korenkov V.* The grid concept and computer technologies during an lhc era. Part. Nucl., 32:6, 2001.
- Astakhov N.S., A.S. Baginyan, S.D. Belov, A.G. Dolbilov, A.O. Golunov, I.N. Gorbunov, N.I. Gromova, I.S. Kadochnikov, I.A. Kashunin, V.V. Korenkov, V.V. Mitsyn, I.S. Pelevanyuk, S.V. Shmatov, T.A. Strizh, E.A. Tikhonenko, V.V. Trofimov, N.N. Voitishin, and V.E. Zhiltsov.* JINR Tier-1 Centre for the CMS Experiment at LHC. Physics of Particles and Nuclei Letters, 13(5):714–717, 2016.
- Astakhov N.S., A.S. Baginyan, S.D. Belov, A.G. Dolbilov, A.O. Golunov, I.N. Gorbunov, N.I. Gromova, I.A. Kashunin, V.V. Korenkov, V.V. Mitsyn, S.V. Shmatov, T.A. Strizh, E.A. Tikhonenko, V.V. Trofimov, N.N. Voitishin, and V.E. Zhiltsov.* JINR TIER-1-level computing system for the CMS experiment at LHC: status and perspectives. Computer Research and Modeling, 7(3):455–462, 2015.
- Etzion O. and P. Niblett.* Event Processing in Action. Manning Publications, 2010.
- Belov S.D., Z.Y. Deng, V.V. Korenkov, W.D. Li, T. Lin, Z.T. Ma, C. Nicholson, I.S. Pelevanyuk, V.V. Trofimov B. Suo, A.U. Tsaregorodtsev, A.V. Uzhinskiy, T. Yan, X.M. Zhang X.F. Yan, , and A.S. Zhemchugov.* BES-III distributed computing status. Physics of Particles and Nuclei Letters, 13(5):700–703, 2016.
- Baranov A.V., N. A. Balashov, N. A. Kutovskiy, and R. N. Semenov.* JINR cloud infrastructure evolution. Physics of Particles and Nuclei Letters, 13(5):672–675, 2016.