MONITORING THE LHCb EXPERIMENT COMPUTING INFRASTRUCTURE WITH NAGIOS

E. Bonaccorsi, Niko Neufeld, CERN, Geneva, Switzerland

Abstract

LHCb has a large and complex infrastructure consisting of thousands of servers and embedded computers, hundreds of network devices and a lot of common infrastructure services such as shared storage, login and time services, databases and many others. All aspects that are operatively critic are integrated into the standard Experiment Control System (ECS) based on PVSSII.

This enables non-expert operators to do first-line reactions. At the lower level and in particular for monitoring the infrastructure, the Control System itself depends on a secondary infrastructure, whose monitoring is based on NAGIOS. We present the design and implementation of the fabric management based on NAGIOS. Care has been taken to complement rather than duplicate functionality available in the Experiment Control System.

INTRODUCTION

The infrastructure of networks and servers deployed in order to filter, move and store the data produced by LHCb are critical vitals for the success of the experiment.

While for the control and monitoring of detectors, PLCs, and readout boards an industry standard monitoring system based on PVSSII & SCADA has been put in production, a lower level the network infrastructure and resources used by each server in the LHCb Cluster needs to be monitored. This is because the PVSSII in ECS depends already on a lot of systems such as the shared storage, the network, dns and others. For this reason, selecting between open source projects, we have tested numerous monitoring software including Pandora, Lemon and NAGIOS.

Even if Pandora promises great features, the software was not mature enough for our needs while the implementation of Lemon would require skills and time not appropriate to our immediate needs: it depends on several services developed by the CERN IT department that we would need to fully replicate because the LHCb network is completely isolated; we ended up choosing NAGIOS, a scalable system where we already had a certain grade of expertise.

NAGIOS has much fewer dependencies then PVSSII and can so monitor the core infrastructure of the ECS itself, so that alarms can be sent, even when a critical failure makes the ECS partially “blind”.

The aim of the software is to quickly inform System Manager about questionable (WARNING) or critical conditions (CRITICAL). What is regarded as “questionable” or “critical” is defined in the configuration. A Web page summary then informs the administrator of normally working systems and services, which NAGIOS displays in green, of questionable conditions (yellow), and of critical situations (red).

There is also the possibility to inform the administrators on duty —depending on specific services or systems— selectively by e-mail but also by paging a mobile phone. By concentrating on traffic light states (green, yellow, red), NAGIOS is distinct from network tools that display elapsed time graphically (for example in the load of a WAN interface or a CPU throughout an entire day) or that record and measure network traffic (how high was the proportion of HTTP on a particular interface?). NAGIOS is involved plainly and simply with the issue of whether everything is on a green light [1] while the trend monitoring or storing data for statistical purpose is more the domain of PVSSII.

HOW NAGIOS WORKS

NAGIOS runs on a server as a daemon and periodically runs plugins in order to understand the status of hosts and services. The main tasks are the plugins executions scheduling, the dependency calculations and the alerting through notifications.

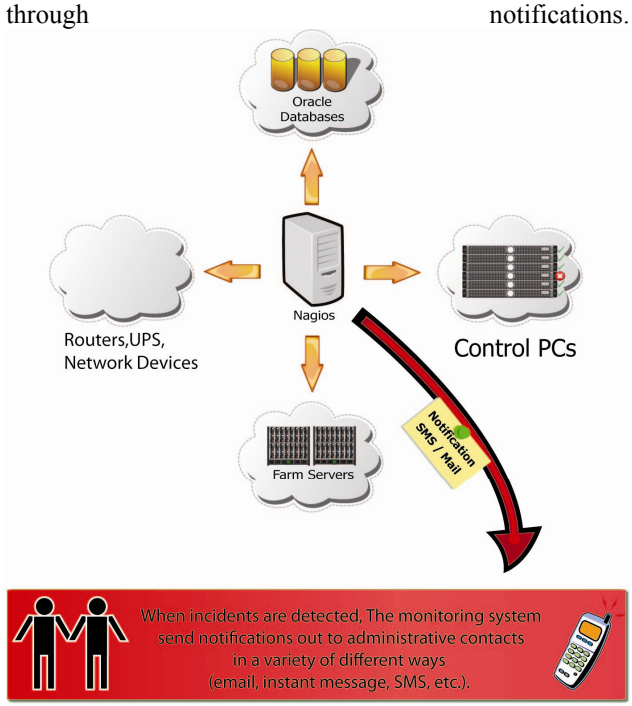

Figure 1: Descriptive diagram of the monitoring system.

Due the modularity of the software every action is delegated to an external program (such as specific scripts and binaries for plugins).

For example in the specific case of notifications the software could be programmed to execute, when a service or a host status change occurs, any kind of external program allowing the administrators to setup notifications

by mail, instant messaging, sms or by a phone call using Voice over IP and a PBX like asterisk.

PLUGINS

NAGIOS exports service checking logic into tiny single-purpose programs called plugins. A plugin is a simple program—often just a script (Bash, Perl etc.)—that gives out one of the four possible conditions OK, WARNING, CRITICAL, or (with operating errors, for example) UNKNOWN.

This means that in principle NAGIOS can test everything that can be measured or counted electronically: the temperature and humidity in the server room, the amount of rainfall, the presence of persons in a certain room at a time when nobody should enter it.

There are no limits to this, provided that you can find a way of providing measurement data or events as information that can be evaluated by computer [2].

LOCAL VERSUS REMOTE PROCESSING

Plugins make it possible to quickly end easily add checks for new types of services. The modularity of this approach makes it possible to execute the plugins themselves, either locally on the monitoring server or remotely on the monitored hosts.

Centralized execution is generally preferable whenever possible because the monitored hosts bear less of a resource burden. However, remote processing may be unavoidable, or even preferred, in some situations.

For large environments with thousands of hosts, centralized execution may be too much for a single monitoring server to handle. In this case, the monitoring system may need to rely on the clients to run their own service checks and report back the results.

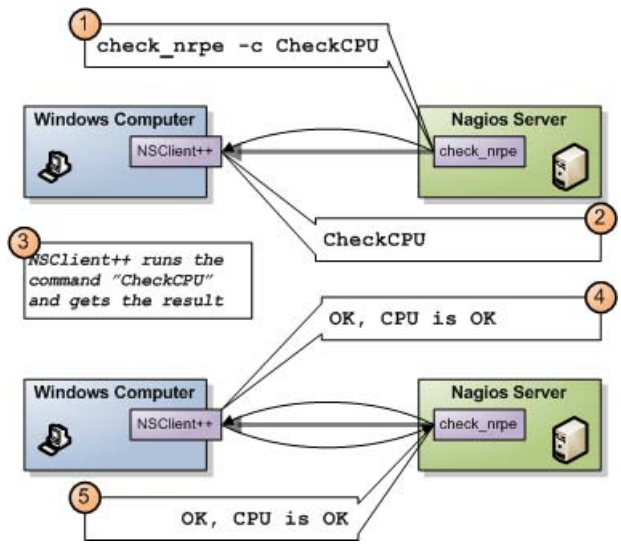

Figure 2: Implementation of remote plugin processing.

Some types of checks may be impossible to run from the central server. For example, plugins that check the amount of free memory may require remote execution [3].

The execution of remote plugins requires an agent installed in the target machine, this could be NRPE (NAGIOS Remote Plugin Executor), NSCA (Nagios Service Check Acceptor), NSClient++ [4] or the SSH daemon.

LHCB AGENTS IMPLEMENTATION

In order to have a fully and easily manageable monitoring system, the agent deployment and configuration is implemented using quattor [5] for the Linux machines and using System Management Server [6] for the Windows machines.

New servers installed in the experiment will automatically inherit the common agent configuration and at the first boot they will be ready to be monitored.

HOSTS & SERVICES UNDER MONITORING

We monitor the basic health of almost all our services, this means at the moment 1044 hosts and 8931 services and those numbers are still growing.

Servers

Combining local and remote executions of plugins our system monitors CPU usage, load average, local disks, memory and swap/pagefile utilisations, the reachability of the LDAP servers, status of the NFS mounts, SSH daemon, uptime, quattor daemons for the Linux based servers and all the services configured in automatic startup for the Windows servers.

In case of failure the system will try to restart automatically the services.

DNS and DHCP

NAGIOS query every 10 minutes the status of DNS and DHCP services simulating real requests.

Network Devices

Every network device in LHCb implement the Simple Network Management Protocol (SNMP) [7], allowing us to monitor not only if a certain device is up but also CPU utilisation, ports and trunk status, uplinks and event logs.

WAN links are also under monitoring.

Additionally we check the ability to process jumbo frames on the network devices that provide the network infrastructure from the TELL1 boards to the High Level Trigger (HLT).

Uninterruptable Power Supplies

The UPS battery charge, output amperes, frequency in and out, current load, status temperature, test date and result, volts in and out are being monitored.

Backups

LHCb uses Amanda as a backup solution, every night the backup consistency is checked by NAGIOS.

Web Services

All the web services are under monitoring; our implementation checks both the status of the TCP ports and additionally the dynamic creation of web pages.

Databases

The experiment itself and in particular the data acquisition part depends on a number of oracle databases: listener and db consistency are checked constantly.

Storage

Storage controllers are constantly monitored by SNMP looking for hardware disk failures. The storage is connected via optical links through fibre channel switches to a cluster of servers that publish the disks using NFS and CIFS protocols: status of the daemons and the free space on the storage is under monitoring.

fcswitch03		
Fan 1 Status	OK	OK - speed is 'nominal', 6250 rpm
Fan 2 Status	OK	OK - speed is 'nominal', 6250 rpm
Fan 3 Status	OK	OK - speed is 'nominal', 6250 rpm
PING	OK	OK - Packet loss 0%, RTA 0.30 ms
PSU Status	OK	OK - power supply unit is 'nominal'
SSH	OK	SSH OK - OpenSSH_3.8.1p1
Temp 3	OK	OK - 25 deg C

Figure 3: On Every fibre channel switch we monitor the network reachability, temperature and fans.

WEB INTERFACE AND NAGVIS

The web interface is word wide reachable through reverse proxy at the following URL: https://lbnagios.cern.ch.

A complementary interface that displays the LHCb network map has been implemented using NagVis [8], this map is displayed in the control room giving an instant overview of the infrastructures status.

NOTIFICATIONS AND DEPENDENCIES

The Systems Managers are kept informed about problems discovered via mail or short message on the mobile by the monitoring system.

When a service which has dependencies fails NAGIOS will disable the notifications for this depending service until the problem with the hierarchically highest service is solved (for example in case of problems on the main routers) avoiding the receiving of a large amount of mail or SMS and informing us only about the main problem[9].

In case of failure of the main switch connected to our monitoring server, NAGIOS is able to enable a backup network interface directly connected to the CERN network, and send notifications through this link.

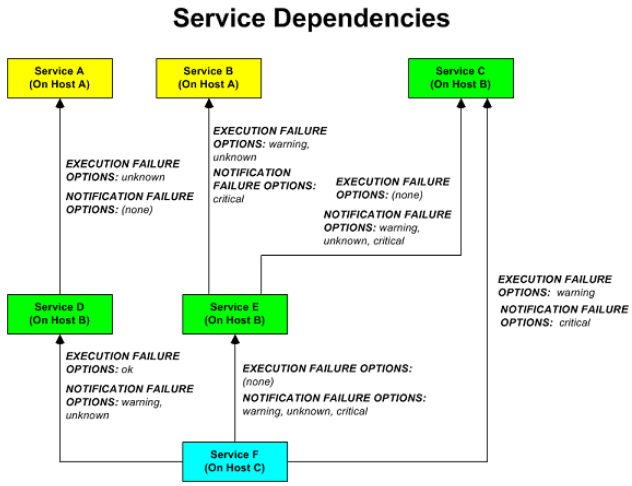

Figure 4: Logical layout of service notification and execution dependencies.

CONCLUSIONS

We have deployed our NAGIOS monitoring infrastructure six months ago and have it run successfully since.

The software provides us with an essential insight into our network- and server-availability and gives us the possibility to solve problems ideally even before anyone notices them. The modularity of the system has allowed us to set up fine-grained notifications: only the critical events are forwarded to PVSSII which are seen by non-experts operators.

We have successfully implemented a light independent system to monitor the key infrastructure of the LHCb ECS itself, this give us an important second level of monitoring in case of serious problems.

REFERENCES

[1] W. Barth "NAGIOS, Systems and Network Monitoring" p. 16 (2006).
[2] W. Barth "NAGIOS, Systems and Network Monitoring" p. 17 (2006).
[3] D. Josephsen "Building a monitoring infrastructure with NAGIOS" p. 4 (2007).
[4] http://nsclient.org.
[5] http://www.quattor.org.
[6] http://www.microsoft.com/SMServer/default.mspx.
[7] http://www.ietf.org/rfc/rfc1157.txt.
[8] http://www.nagvis.org.
[9] http://NAGIOS.sf.net/docs/3_0/dependencies.html.