ATL-DAQ-PROC-2012-014
12 June 2012

Tools and strategies to monitor the ATLAS online computing farm

S. Ballestrero[1,2], F. Brasolin[3], G.–L. Dârlea[1,4], I. Dumitru[4], D. A. Scannicchio[5], M. S. Twomey[1,6], M. L. Vâlsan[4] and A. Zaytsev[7]

[1]CERN CH-1211 Genève 23, Switzerland
[2]University of Johannesburg Department of Physics, PO Box 524 Auckland Park 2006, South Africa
[3]INFN Sezione di Bologna, Viale Berti Pichat 6/2, 40127 Bologna, Italy
[4]Politehnica University of Bucharest, Splaiul Independenţei 313, cod 060042, sector 6, Bucharest, Romania
[5]University of California, Irvine, CA 92697, USA
[6]University of Washington Department of Physics, Box 351560 Seattle WA 98195-1560, USA
[7]Russian Acad. Sci., Siberian Div., Budker Inst. of Nuclear Physics (BINP), 11 Academician Lavrentiev prospect, Novosibirsk, 630090, Russia

E-mail: `diana.scannicchio@cern.ch`

Abstract. In the ATLAS experiment the collection, processing, selection and conveyance of event data from the detector front-end electronics to mass storage is performed by the ATLAS online farm consisting of nearly 3000 PCs with various characteristics. To assure the correct and optimal working conditions the whole online system must be constantly monitored. The monitoring system should be able to check up to 100000 health parameters and provide alerts on a selected subset.

In this paper we present the assessment of a new monitoring and alerting system based on Icinga. This is an open source monitoring system derived from Nagios, granting backward compatibility with already known configurations, plugins and add-ons, while providing new features. We also report on the evaluation of different data gathering systems and visualization interfaces.

1. Introduction

The ATLAS [1] Online Farm, consisting of nearly 3000 PCs, must be continuously monitored to ensure the optimal working conditions. The monitoring system should be able to check up to 100000 health parameters and provide alerts on a selected subset: the health status of the OS, hardware, critical services and network components. The monitoring system is not critical for the ATLAS data taking, but it is very useful for promptly reacting to potentially fatal issues that may arise in the system.

Nagios v2.5 [2] was chosen in 2007 to design and implement the monitoring system and it is still being used. It has proven to be robust and scalable with the increase in size of the farm; to cope with the large amount of checks and the consequently high work load, the checks have been distributed across many Nagios instances on separate servers (up to ∼80 now).

Therefore from a configuration point of view the system is too complex to be managed manually. The many configuration files must be instead generated in an automated way; the resulting files consequently have a simplified, standardized structure, and cannot take advantage of the full flexibility native to Nagios.

As new tools have recently become available (e.g. Gearman [3] and mod_gearman [4]) to nicely distribute the work load on worker nodes, we have begun to evaluate the possibility of updating the current system to take advantage of these new features and at the same time simplify the current schema.

2. Nagios

Nagios is the monitoring and alerting system adopted and used since 2007 to implement the monitoring of the ATLAS online farm. The main requirements were to have a robust and reliable system capable of scaling up together with the planned growth of the farm.

Information and alert notifications are successfully provided by monitoring many different services, for example:

- ping and SSH connectivity
- NTP synchronization
- kernel version
- temperature
- HDD state (if present) and ramdisk usage
- automount status
- filesystem status
- CPU load
- memory usage

For events related to critical services, which are of crucial importance for the proper functioning of the ATLAS Trigger and Data Acquisition (TDAQ) infrastructure, Nagios provides e-mail and/or SMS alerts to the concerned experts.

2.1. Current implementation

Due to the large amount of checks and hosts ($\sim$3000 hosts with up to $\sim$40 checks each) it was necessary to distribute the work load on many servers. Consequently, a Nagios server has been installed on each of the $\sim$80 Local File Servers (in the ATLAS Online Farm architecture [5], an LFS is a server which provides DHCP, PXE, NFS and other services to a defined subset of clients, typically 32 or 40 hosts).

Since the number of servers (and nodes to be monitored by each server) is too large to be handled manually, the configuration files, describing the hosts and services to be checked, have to be generated automatically by ConfDBv2 [6]. This is a tool developed by TDAQ SysAdmin Team to manage network, host and Nagios configuration in the Online Farm. The automatically generated configuration files have a simplified and standardized structure; as a consequence the system cannot use all the features and flexibility native to Nagios (e.g. host and service checks dependencies).

Eventually the data resulting from the checks is stored on a MySQL database which represents the current status; the database is hosted on a MySQL Cluster for performance and reliability reasons. The same data is also accumulated in RRD (Round-Robin Database, [7]) format and stored on a centralized file storage, to be used later to produce graphs showing the historical evolution of the monitored parameters. However, because of Nagios' design, the log files, used by its web interface, can only be stored locally.

This implementation, sketched in Figure 1, has the obvious disadvantage of having access to the information scattered across all the LFSes: the standard Nagios GUI of each server can

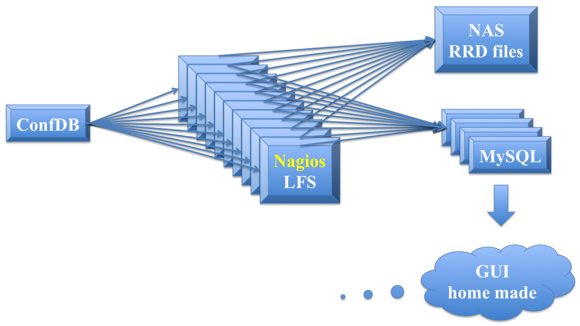


Figure 1. Schema of the current implementation of the monitoring system.

only display the data collected locally. To overcome this limitation, a web interface has been developed in-house to group all the information in a single page: the interface displays summaries and detailed data of the checks performed by each of the $\sim$ 80 servers, and provides, for each monitored host, quick links to the GUI of its Nagios server.

Figures 2 and 3 show screenshots of the main summary page and of the in-house Nagios web page showing a particular group of hosts being monitored. This custom interface has proven effective, but it does not offer all the advanced functionalities that are available in other Web GUIs developed for different, more standard Nagios-based monitoring systems, and requires maintenance to follow the evolution of the system.

Besides the checks provided via the standard Nagios plugins, a few customizations have also been introduced:

- some of the existing plugins have been adapted to suit the needs of the monitoring system. For example, the plugin used to check the status and the traffic of various ethernet interfaces of a node has been adapted to be able to simultaneously monitor all interfaces, regardless of their names.
- a full system has been put in place to monitor all the significant IPMI [8] sensor information, which is provided by the Baseboard Management Controllers (BMC) [1] of the PCs in the farm; in general more than 20 hardware sensors are monitored for each node. Amongst the most important are CPU and system temperatures, fan speeds, power supplies statuses, various currents and voltages from the system. These checks run as an independent service which then exports the result in a format which is easily interpreted by Nagios.

Despite the described disadvantages of the complex, custom distributed implementation, the system has been working well since the start-up of LHC operation in 2008.

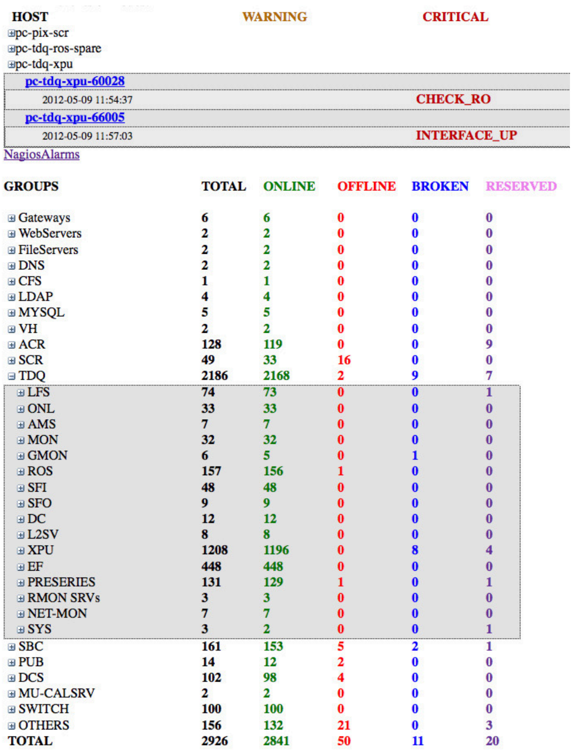
3. Ganglia

As a first step to improve the monitoring system, we have recently introduced Ganglia [9].

Ganglia is a software package designed for monitoring the workload and performance of multiple large, and possibly geographically distributed, high performance computing clusters; contrary to Nagios, it does not have advanced alerting capabilities.

We use it to monitor about 300 hosts, mainly servers, for which it provides detailed information on CPU, disk and network performance; the advanced functionalities of its Web User Interface help in performance analysis and forecast. We are also evaluating the option of using it as a data source for the Nagios alerting system [10].

[1] The Baseboard Management Controller is described in the IPMI standard, see [8].



HOST	WARNING	CRITICAL
pc-pix-scr		
pc-tdq-ros-spare		
pc-tdq-xpu		
pc-tdq-xpu-60028		
2012-05-09 11:54:37		CHECK_RO
pc-tdq-xpu-66005		
2012-05-09 11:57:03		INTERFACE_UP

NagiosAlarms

GROUPS	TOTAL	ONLINE	OFFLINE	BROKEN	RESERVED
Gateways	6	6	0	0	0
WebServers	2	2	0	0	0
FileServers	2	2	0	0	0
DNS	2	2	0	0	0
CFS	1	1	0	0	0
LDAP	4	4	0	0	0
MYSQL	5	5	0	0	0
VH	2	2	0	0	0
ACR	128	119	0	0	9
SCR	49	33	16	0	0
TDQ	2186	2168	2	9	7
LFS	74	73	0	0	1
ONL	33	33	0	0	0
AMS	7	7	0	0	0
MON	32	32	0	0	0
GMON	6	5	0	1	0
ROS	157	156	1	0	0
SFI	48	48	0	0	0
SFO	9	9	0	0	0
DC	12	12	0	0	0
L2SV	8	8	0	0	0
XPU	1208	1196	0	8	4
EF	448	448	0	0	0
PRESERIES	131	129	1	0	1
RMON SRVs	3	3	0	0	0
NET-MON	7	7	0	0	0
SYS	3	2	0	0	1
SBC	161	153	5	2	1
PUB	14	12	2	0	0
DCS	102	98	4	0	0
MU-CALSRV	2	2	0	0	0
SWITCH	100	100	0	0	0
OTHERS	156	132	21	0	3
TOTAL	2926	2841	50	11	20

Figure 2. Example of the status summary page.

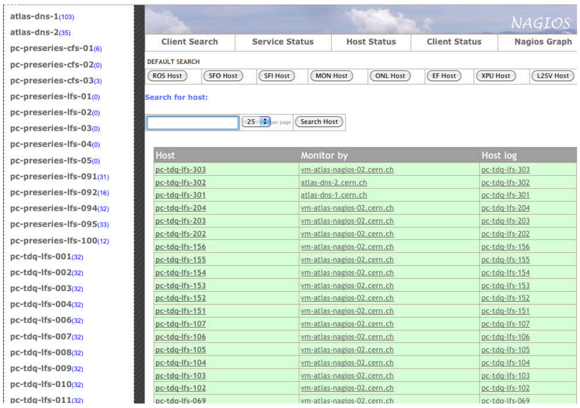


Figure 3. Example of the in-house Nagios GUI.

4. Icinga

Icinga [11] is a fork of Nagios v3 and is backward compatible: Nagios configurations, plugins and addons can all be used with Icinga. Though Icinga retains all the existing features of its predecessor, it builds on them to add many patches and features requested by the user community as described on its web site. Moreover a very active community provides regular major releases, bug fixes and new features. All these characteristics, and the backward compatibility with Nagios configurations, convinced us to test it.

The main goals in updating the current system are to simplify the current setup by reducing the number of distributed servers, while maintaining the same performances, and to increase the flexibility of the configuration.

4.1. Tests performed

The first evaluations have been performed installing Icinga on a single server: $\sim$ 1100 hosts and $\sim$ 12000 services have been configured to be monitored; this load corresponds to about one third of the online farm.

For this test, the configuration files have been prepared manually by copying the existing ones from the multiple Nagios servers. This was a very important proof of the backward compatibility. Therefore the same system currently in use (ConfDB [6]) can be easily adapted to generate configurations for a system based on a single Icinga server.

A local MySQL server has been used to store the data produced by Icinga.

Icinga, being a more recent and actively updated software package for system monitoring, behaves well with a high number of hosts and checks and provides new and useful options. For example one may benefit from the *use_large_installation_tweaks* configuration option that allows

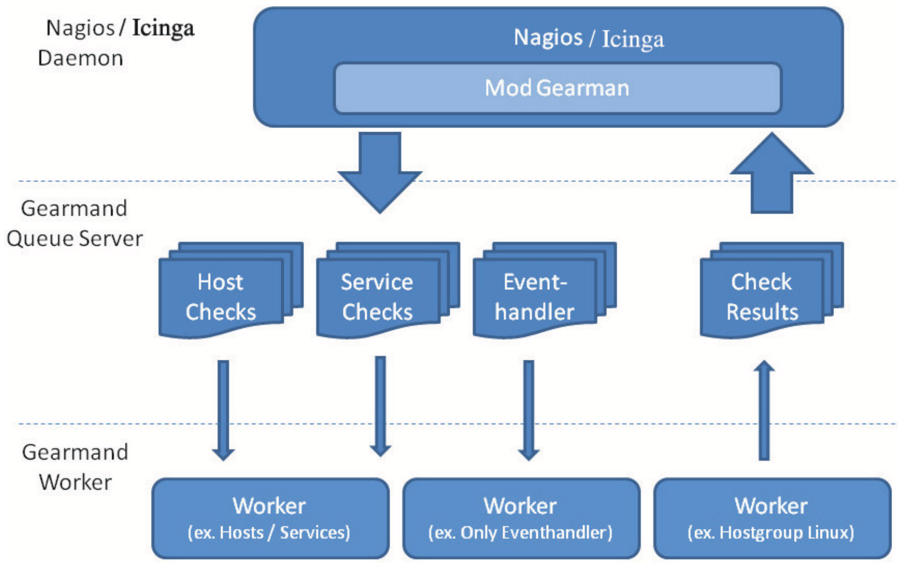


Figure 4. Schema representing how mod_gearman integrates with Nagios/Icinga.

the Icinga daemon to take certain shortcuts (e.g. better memory management and usage of forks), which result in a reduced system load and increased performance.

The results obtained show that Icinga copes better with the high number of checks performed than the currently used Nagios v2.5 implementation: the work load on a single node shows an average check execution time and latency below 1 second.

4.2. Gearman and mod_gearman

Gearman [3] is a generic application framework that distributes work from a server to other machines.

Mod_gearman [4] is an easy way of distributing active Nagios/Icinga checks across the network and of increasing scalability. It can even help to reduce the load on a single monitoring server, because it is much smaller and more efficient in executing checks. It consists of three parts (see Figure 4):

- a module for the Nagios or Icinga core which sends service and host checks to a Gearman queue
- one or more worker nodes to execute the checks
- at least one Gearman Job Server

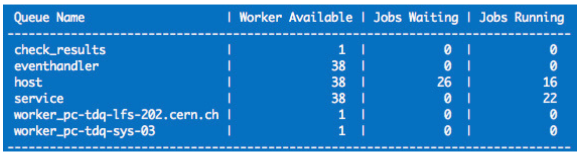
4.3. Test performed using Gearman

Some tests have also been performed using Gearman and mod_gearman (see section 4.2) to distribute the work load of the thousands of checks on multiple servers.

In the test setup we have defined two workers: one on the Icinga server itself and the second on another node.

Figure 5 is a snapshot from the Gearman status tool showing the number of workers available, jobs waiting and jobs running.

For the time being the customized scripts (described in Section 2) have not been used. As they make use of files saved locally to generate the check results, they are not suitable to be used in a distributed environment like the one provided with Gearman/mod_gearman.



Queue Name	Worker Available	Jobs Waiting	Jobs Running
check_results	1	0	0
eventhandler	38	0	0
host	38	26	16
service	38	0	22
worker_pc-tdq-lfs-202.cern.ch	1	0	0
worker_pc-tdq-sys-03	1	0	0

Figure 5. Snapshot showing gearman status tool with two workers nodes.

Figures 6 and 7 respectively show the CPU load of the worker node and of the server (which runs also a worker process).

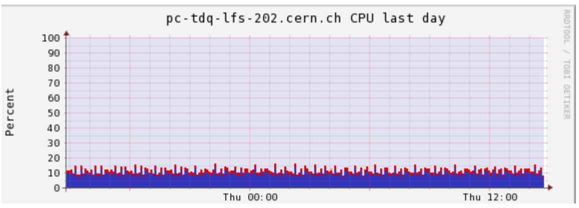


Figure 6. Snapshot of the CPU usage on a Gearman worker node.

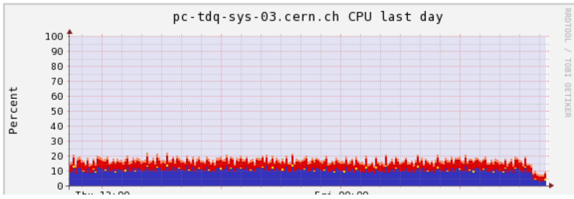


Figure 7. Snapshot of the CPU usage on a server running Icinga, MySQL and a Gearman worker.

5. Migration strategy

A smooth migration from the current to the new monitoring system is highly desirable in order to maintain the current level of reliability and robustness and avoid disruption to the system during the LHC operation.

The schema of the foreseen monitoring system implementation based on Icinga is shown in Figure 8.

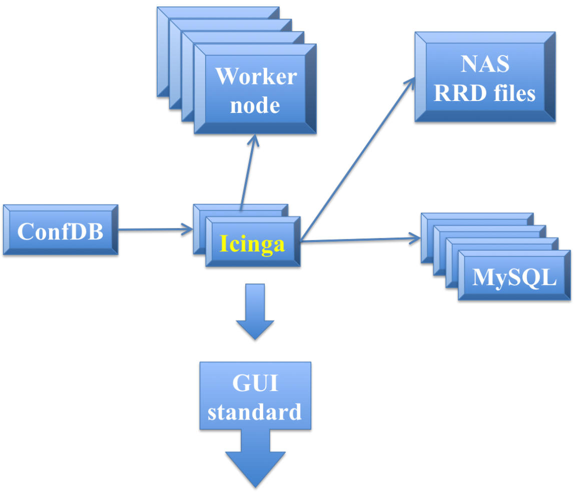


Figure 8. Sketch of the new possible implementation of the monitoring system based on Icinga.

Different upgrade strategies have been considered and all foresee having Icinga running in parallel with Nagios until the end of the current year (2012). The final step of putting the new monitoring system into production can safely be performed next year, taking advantage of the LHC Long Shutdown (2013).

For this upgrade ConfDBv2 will have to be adapted to generate the configuration file on a single server. Moreover different tools will still need access to the centralized MySQL database and RRD storage and will likely need to be adapted for the new system environment. The centralized Icinga server will provide a unified web user interface which may replace most of the functionality of our dedicated in-house Nagios GUI.

One drawback of an Icinga implementation with a single central server is that losing the central server results in no access to the monitoring information. In theory it is possible to build a high availability system, but the increased complexity may not be justified since the monitoring is not a critical system for the ATLAS data taking.

6. Conclusions

The current monitoring and alerting system for the ATLAS Online Farm is based on Nagios v2.5, with the addition of in-house developed web interface and specialised plugins. It has proven its reliability and effectiveness in production, providing alerting and basic performance monitoring.

To better support the evolution of the Farm in the long term we are evaluating an upgrade of the system, based on Icinga and mod_gearman. Initial tests performed using a standalone Icinga server have shown good performance in monitoring ∼1100 hosts with ∼12000 services. Adding mod_gearman, with 2 workers, provides the expected increase in performance, pointing to a very good scalability of the Icinga–Gearman combination.

The coming months will see the completion of the scalability tests, the porting of certain data collection plugins, and the choice of one of the possible upgrade scenarios.

References

[1] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08003

[2] Nagios: http://nagios.org

[3] Gearman: http://gearman.org/

[4] mod_gearman: http://labs.consol.de/nagios/mod-gearman/

[5] ATLAS Collaboration, ATLAS High Level Trigger, Data Acquisition and Controls: Technical Design Report, CERN/LHCC/2003-022 (2003), ISBN 92-9083-205-3

[6] ATLAS TDAQ Sysadmin, Centralized configuration system for a large scale farm of network booted computers, to be published in proceedings of CHEP 2012

[7] RRDTool: http://oss.oetiker.ch/rrdtool/

[8] Intelligent Platform Management Interface (IPMI): http://www.intel.com/design/servers/ipmi/

[9] Ganglia: http://ganglia.sourceforge.net/

[10] ATLAS TDAQ Sysadmin, Upgrade and integration of the configuration and monitoring tools for the ATLAS Online farm, to be published in proceedings of CHEP 2012

[11] Icinga: https://www.icinga.org