

CMS Comments on document titled
“SCF/FEF Evaluation of Nagios and Zabbix Monitoring Systems”
(Docdb #3277)

Authored by
Paul Tader, Catalin Dumitrescu, Lisa Giacchetti, Jon Bakken
CT1 Department

Introduction

In 2007, CMS-T1 started to investigate other monitoring solutions when our infrastructure started to expand beyond what NGOP could easily scale to. NGOP was showing difficulty when the number of checks reached approximately 11,000 different items. When considering other software packages, we focused on active, well documented open source projects. Initially we choose Nagios as a possible solution. After a small rollout to a subset of our nodes, it became clear that Nagios would need some additional configuration and setup changes in order to scale to our environment at that time. We also contacted Brookhaven National Laboratory as we heard that they configured Nagios with a MySQL database backend. We'd figured that this could address scaling issues we were seeing during our initial test. As it turns out, BNL was only using the Nagios client to collect the data and record it into the database using custom written wrapper scripts. The “server” part of Nagios was never used.

We also researched Zabbix, a relatively new open source project that has a growing user and developer community. Zabbix stood out because it was already using a database backend and was tested with thousands of devices. The initial test roll out went very smooth with minimal resources consumed on the Zabbix server. Soon after this initial rollout, Zabbix was installed on all the nodes running in parallel with NGOP, utilizing NGOPs scripts and its own built-in checks. Since entering production the number of monitored devices has increased by 40% and the number of checks has reached over 126,000. Zabbix has not only enabled the CMS-T1 facilities group to quickly and easily roll-out new monitoring checks, but brings with it historical data and flexible alerts overall improving our monitoring capabilities.

The rest of this note addresses a few of the comments in FEF's report on Nagios that can be found in CD DocDB Document 3277-v1. We fully support FEF running Nagios on their systems; our comments are just meant to clarify CMS's position on several of their points so it is clear why CMS chose Zabbix as its monitoring solution instead of Nagios. We do not believe we would have been able to perform all the monitoring checks we've deployed with Nagios, at least not as easily, and consider Zabbix a key component in our CMS infrastructure.

Analysis Comments

- *Zabbix has a much more attractive and mature user interface. Navigating it can be quite confusing, but the entire system can (must) be configured from the web browser.*

Because interfacing to Zabbix is mostly done through the web, it may initially seem difficult to navigate at first. In reality, most sysadmins use only a few key pages such as viewing the current Triggers and marking down/up hosts. CMS has documented the most common task on a twiki that a sysadmin might need to use. In addition, having a single interface to monitor and configure your site instead of a collection of scripts simplifies the entire monitoring service.

- *Zabbix has better reporting built in, particularly regarding performance data and trending. Nagios can produce many of the same types of graphs, but only via third-party add-ons and external tools.*

Agreed. Every item being checked by Zabbix has a graph that can be referenced for baseline purposes or trending analysis. Any number of custom graphs can be created from multiple host components or multiple hosts together. Trends are available for the past year on all critical and dCache nodes and 90 days for worker nodes. This default behavior has proven to be beneficial when reviewing historical data for items that we would normally not consider for charting. For example, if you needed to review network traffic for a non-critical server.

- *Zabbix, by default, monitors a much larger number of metrics from each server, but that can have an effect on the monitoring server's performance and disk requirements. The CMS group actually trimmed down many of these metrics because of the overhead involved in collecting them and storing them, particularly because the Zabbix MySQL database was growing much faster than they could accommodate. We see some advantages to having all these performance metrics available, but in our testing we also found it necessary to eliminate many of them to limit the resources required to run Zabbix.*

Zabbix provides a wide breadth of user community predefined metrics to choose from. We initially deployed them all, and based on our experience, we've determined that we can disable some of the little used ones without loss of monitoring integrity. Further, we do not anticipate going back more than 90 days on the worker nodes so we reduced the retention period for these nodes. Systems classified as servers and dCache nodes still retain one year worth of data. Even though we did have the disk space available these minor changes have stabilized the database size (chart available).

- *Zabbix is harder to configure, particularly if adding new, custom checks. It takes more steps to build a check in Zabbix and the process must be done through the GUI.*

Configuring a new check doesn't take that much effort at all. Everyone in the CT1 staff adds and configures additional checks. Many times checks are already in the default Templates that come with Zabbix. For example, for basic checks (ie. memory, disk space, network utilization) start to be monitored as soon as the node is online and running a Zabbix client. There isn't a need to push any monitoring script to accomplish this basic monitoring. Once in-place, Zabbix properties such as frequency, history/trends and disabling/enabling also managed from the Zabbix server. Changes are automatically propagated out to nodes. This is especially useful for nodes that are unreachable. There is also only one configuration file for all clients. Except for the hostname, it's identical site-wide making client setup very simple. This configuration file contains all the custom checks CT1 sysadmins have written. The server controls whether or not a particular node has the check enabled.

Zabbix enables CMS to write simpler scripts as the logic whether to raise a trigger is builtin to the Zabbix trigger. For example, Zabbix can look for an absolute value change or a delta over a configured time in the data being returned. Scripts don't need to just return 1 or a 0. They can return a range of numbers or text. Different triggers can then be setup to monitor one check and act accordingly to the returned value.

- *Zabbix is much more cumbersome to script than Nagios, and interactions with the system must be done through the web-based GUI. Nagios' named pipe interface and simple config files are much easier to script.*

Zabbix uses a DB engine and schema - Interacting with the database could be considered much easier and less prone to error than maintaining text files. CMS has deployed several alerts that interface with the database for its data and have written scripts to mine values such as the running kernel for a node to collect data for other monitoring operations.

- *Zabbix currently does not support scheduled maintenance at the server level. The entire Zabbix server can be set offline, or alerts can be disabled by hand during maintenance, but scheduled maintenance cannot be set for specific time periods on specific hosts or services. In Nagios, scheduled maintenance can be assigned to host groups (i.e., clusters), individual hosts, or specific services either through the GUI, or via the named pipe interface. Zabbix has promised some form of scheduled maintenance in their next release, but we don't know what capabilities the feature will provide.*

Zabbix can disable monitoring for individual systems, groups or the entire server. Although scheduling is not yet a feature in Zabbix, it is promised with the next version. For now we can easily write a script to disable hosts and items at a certain date for a certain time by simply changing one field in the database.

- *Zabbix does not allow work logs or comments to be stored with a username and time stamp. It has a single free-form comment field, but does not log individual comments which we like to use as a work log for server issues. Nagios assigns a username and time stamp to each comment, so a better history is available for each host. Having this capability may allow us to entirely eliminate “Faultlog”, an internally developed application that does basic monitoring of server availability and work histories.*

This is incorrect. Acknowledgement functions are present in Zabbix today.

- *Mass changes in Zabbix are very slow and usually require clicking through the interface numerous times. They have added some new capabilities for making mass changes to some attributes, but the interface is clumsy and confusing. Nagios, because of its simple text file configurations, allows for mass changes with simple shell scripts (via perl, sed, etc.).*

Mass changes made to checks and triggers are completed from with the Template system Zabbix uses as mentioned above. Item and trigger changes occur within a few seconds for several hundred nodes.

- *We have more internal experience with Nagios. Staff who have used both Nagios and Zabbix believe Nagios is more stable than Zabbix. Zabbix, in our testing, has been unstable and requires restarting occasionally.*

December '09 will mark the 2 year anniversary using Zabbix. During this time the Zabbix server has remained online (except scheduled downtimes). The Zabbix software has been upgraded 4 times, moved to new hardware twice and the OS and mySQL software have been upgrade to new major releases. As of this writing we are now monitoring a 2,102 nodes with 126,488 items being monitored all from one Zabbix server.

- *Nagios has a very large user community, with over 1500 plugins (checks) for many common and not-so-common devices and services. Zabbix has a growing community as well, though not as large as Nagios, but far fewer third-party plugins and utilities.*

Zabbix is a relatively new Opensource monitoring solution. Initial release in 2001. Community support has been growing ever since. The forums are very well visited and users contributions have been growing.

Other Comments

- We recently took delivery of 448 nodes. Adding them involved creating an XML file and importing it into Zabbix. After a few minutes all the new nodes were in Zabbix, with the proper group, items and triggers. They were also added to Zabbix in a “Not monitored” until the burnin completes.
- Zabbix is written from the the beginning as a tool to monitor large installations (1000+ nodes). Obviously, this is a good thing for most farm installations at Fermilab.
- Any changes are immediately recorded in an audit log.
- Installation was very easy. We had a test server up and running with 170 clients within a couple hours and monitoring approximately 4,000 items. During the production rollout the remaining nodes were easily imported using the XML import utilities.
- The Template system is very powerful, provides a large set of pre-configured item for servers and greatly reduced the time needed to roll out basic checks.
- Zabbix can receive SNMP traps.
- A network “weather” map can be configured that will visually identify problem nodes.
- With Zabbix you have the ability control user access and permissions. For example, we gave other computing groups read access to our system so that they could browse the interface and reporting features.