ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 4, Issue 2 (Apr. - Jun. 2016)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2016)

Enhancing Nagios Capability by Plugin Development

[I]Karthik.B, [II]Prajakta Madankar
[I]M.Tech Student, [II]Assistant Professor
[I,II]Dept. of Information Science & Engineering, The National Institute of Engg., Mysuru, India

Abstract
Network monitoring is a key component in making sure that network is running smoothly. There are many open source monitoring tools available for network monitoring. Nagios is an open source Network monitoring tool. Nagios can be used to monitor host and services. Nagios provide only limited information when the host or service is down. If the administrator wants to know why the particular service or host went down, then administrator must manually check the log files of the particular host or services to understand the problem and rectify it. This is a very tedious and time consuming process. To overcome this problem a plugin is developed that can check the log file of particular service or host that is down and will display the cause for the failure of the service or host on the Nagios UI itself. So, the administrator can take corrective action immediately and saves time.

Keywords
Nagios, NRPE, Plugin, Hostcheck, Servicecheck.

I. Introduction

The purpose of Network Monitoring is collecting the useful information from various parts of the network so that the network can be managed and controlled using the collected information. As more people communicate using networks, networks have become bigger and more complex. The proliferation of the internet has increased the pace of network expansions. At this age of big and complex networks, network monitoring applications need to use effective ways of checking the status of the networks so that network management applications can fully control the network and provide economical and high quality networking services to the user. It is very important to know what goals to achieve in network monitoring. By knowing the goals of network monitoring, network monitoring application can choose among network monitoring techniques that will be best to help monitor the networks. [13]

There are generally three basic goals for network monitoring.

- Performance Monitoring
- Fault Monitoring
- Account Monitoring

1. Performance Monitoring

Performance monitoring deals with measuring the performance of the network. There are two important issues in performance monitoring. First, performance monitoring information is usually used to plan future network expansion and locate current network usage problems. Second, choosing what to measure is important. There are too many measureable things in a network. But the list of items to be measured should be meaningful and cost effective. This list of items to be measured is called network indicators because they indicate attributes of the network. [13]

2. Fault Monitoring

Fault monitoring deals with measuring the problems in the network. There are two important issues in fault monitoring. First, fault monitoring deals with various layers of the network. When a problem occurs, it can be at different layers of the network. Thus it is important to know which layer is having problem. Second, fault monitoring requires establishing normal characteristics of the network in an extended period of time. There are always errors in the network but when there are errors, it does not mean the network is having persistent problems. Some of these errors are expected to occur. For example, noise in a network link can cause transmission errors. The network only has problem when the number of errors has suddenly increased above its normal behavior. Thus, a record of normal behavior is important. [13]

3. Account Monitoring

Account monitoring deals with how users use the network. The network keeps a record of what devices of the network are used by users and how often they are used. This type of information is used for billing user for network usage, and for predicting future network usage. [13]

II. Literature Survey

A. Related Work

Networks are getting bigger, complex and heterogeneous due to increased dependency of latest computing technologies. These networks based technologies and services are part of daily routine like e-health, online banking, online ticketing and so on.

The users don't want to compromise on the quality and availability of the services they are using. Therefore network management requires a great deal of attention. Network management refers to activities associated with running a network, and a big part of which is monitoring. There are many things which can cause a network to fail, only two things can save the system from downtime: redundancy and monitoring. Network monitoring is the important part of network management which provides necessary data about the network; this data reveals the information about the network's infrastructure, health and is also used for most of the network management tasks.

Network monitoring is the use of a system that constantly monitors a computer network for slow or failing components and that notifies the network administrator via email, SMS or other alarms in case of outages.

The ideal network monitoring system should have the following properties:

- It should be automatic and continuously monitor the network.
- It should quickly inform the administrator about the problem as soon as it arises.
- It should be intelligent enough to point out the problem and its exact location in the network topology. It should also be able to identify the problem effects on the rest of the network and the services that will become unavailable.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

- It should keep a record of the changes in the network which makes easier to find the cause of the problem due to configuration change.
- It should provide remote authentication and authorization for the administrator to get access to the monitoring system from everywhere.

It is very important to select a system that is most suitable for one's requirements.

B. Why Nagios

Although there are many proprietary monitoring tools out there to select from depending upon the requirement, no proprietary tool can provide the source code modification that an open source tool provides. Nagios allows development of custom plugins as per the user requirements.

Nagios is open source and web based software used for network monitoring [3]. It monitors network nodes and services applied on them and inform the network administrator when any change happens in the network [2]. Nagios is well suited application for linux environment but it can also run on other platforms as well. Nagios is a secure and easy manageable application which provides nice web interface, automatic alerts if condition changes and various notification options [5]. When any node or service in the network gets problem, Nagios generates notification to the network administrator in the form of email or SMS. Nagios is developed under GNU general public license and supports different services like HTTP, Ping, SMTP, etc.

Nagios decide about the condition of nodes and services with two factors: "status" and "type of state". The status can be either up, down, critical or unreachable while the type of state can be either soft state or hard state. The type of state has great importance for alerting process. It decides about the final status before a notification is sent out.

In order to avoid false notifications, Nagios check the nodes and services for pre-defined number of times before declaring them to have real problem [5]. The number of attempts can be controlled by "max_check_attempts" option in the host and service definitions.

III. System Analysis

A. Existing System

Nagios Core was originally designed to run under Linux, now it has been further developed to work on windows also. System monitoring in Nagios is split into two categories of objects: hosts and services. Hosts represent a physical or virtual device on the network (servers, routers, workstations, printers, and so on). Services are particular functionalities, for example, a Secure Shell (SSH) can be defined as a service to be monitored. Each service is associated with a host on which it is running. A major benefit of Nagios is that it only uses four distinct states—Ok, Warning, Critical, and Unknown. Nagios is based on plugins—this means if user want to check something that's not yet handled, then user just need to write a simple piece of code.

B. Problem Statement

Nagios provides only limited notification information when a host or services goes down. If the administrator wants to know why the particular host or service went down, then administrator must manually check the log files of the particular host or service to understand the problem and rectify it. This is very tedious and time consuming process.

C. Proposed System

The above problem can be overcome by developing a plugin that can check the log file of host or service that went down and display the errors which caused the host or services to go down on the Nagios user interface itself. This will allow getting faster feedback and administrator can take corrective action immediately.

IV. System Design

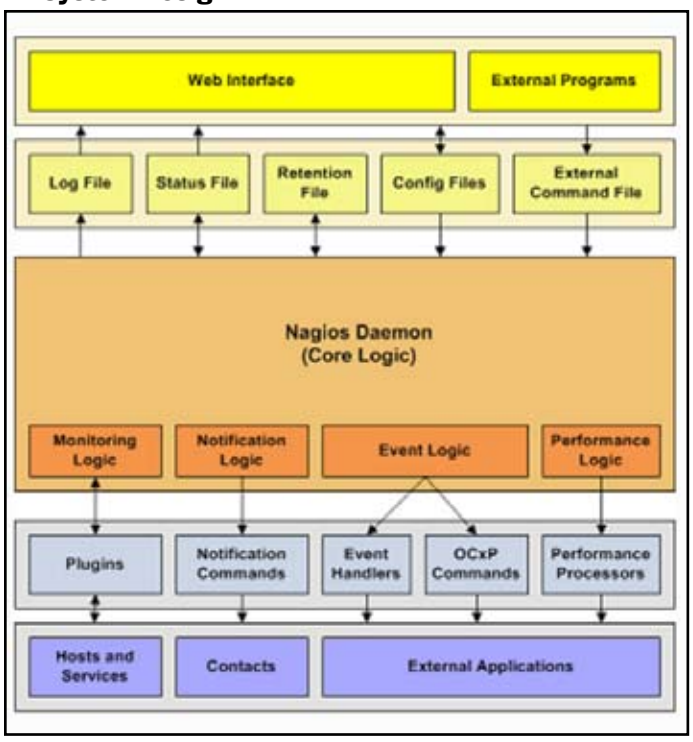

Fig. 1: Layered Architecture of Nagios [12]

1. Plugins

Nagios does not include any internal mechanisms for checking the status of hosts and services in the network. Nagios relies on external programs called plugins to do this work.

Plugins are compiled executable or scripts (Perl scripts, shell scripts, etc.) that can be run from a command line to check the status of host or service. Nagios uses the results from plugins to determine the current status of hosts and services on a network. Nagios will execute a plugin whenever there is a need to check the status of a service or host.

Plugins can run locally and remotely.

Local Execution: In local execution the Nagios daemon will trigger the plugin at intervals specified in the check_interval field. The plugin will perform check on local host/service and return the status of the host/service to Nagios daemon.

Remote Execution: In remote execution, Nagios Remote Plugin Execution (NRPE) will come into picture. Nagios daemon will trigger the check_nrpe plugin locally at the intervals specified in check_interval field. Check_nrpe will establish connection with NRPE daemon on remote server. NRPE daemon on the remote server executes the required Nagios plugin locally, and passes the results back to the Nagios daemon. More about NRPE is explained in Implementation section.

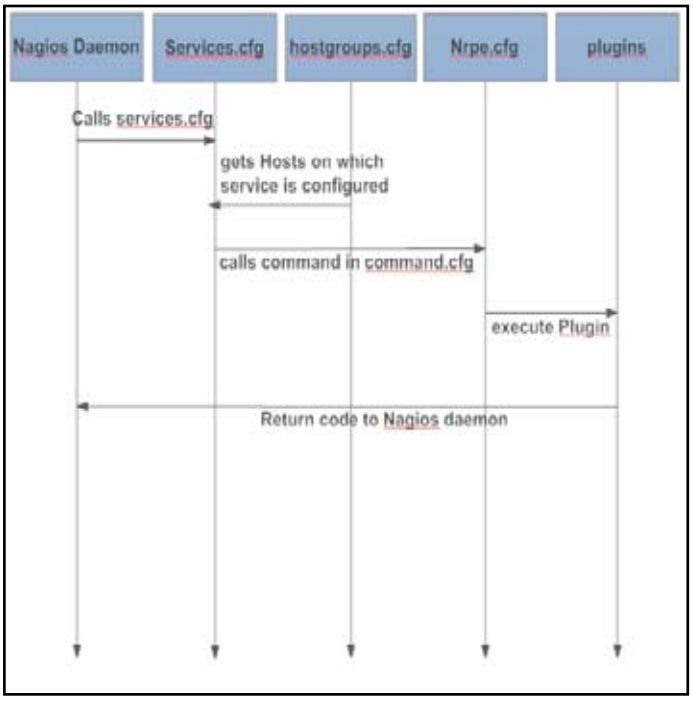

Fig: 2: Sequence Diagram of Plugin Execution

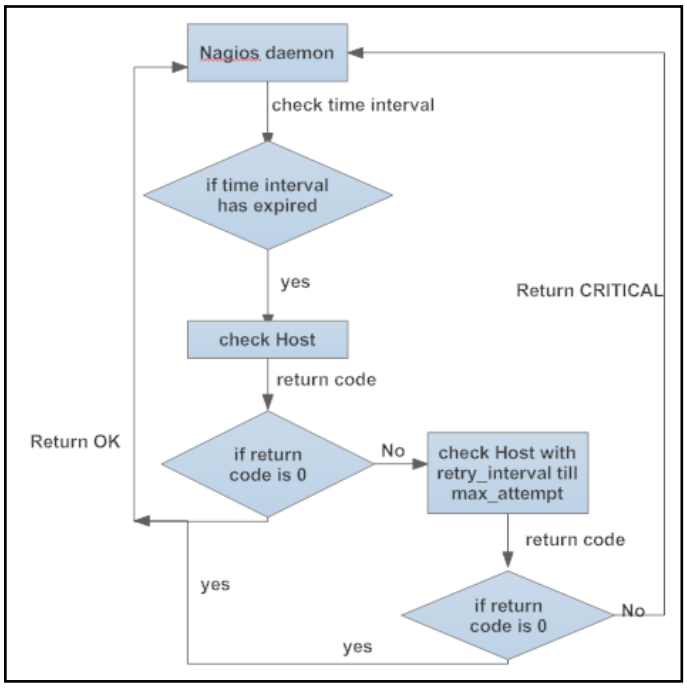

Fig 3: Flowchart of Host Check Workflow.

Based on the result provided by the plugins the Nagios daemon will take appropriate action i.e. to run event handler or to send notifications to the administrator. Decision of whether to run event handler or to send notification depends on the state type of host/ service.

2. Host Check

Hosts are checked by the Nagios daemon:

- At regular intervals, as defined by the check_interval and retry_interval options in the host definition
- On-demand when a service associated with the host changes state.

On-demand checks are made when a service associated with the host changes state because Nagios needs to know whether the host has also changed state. Services that change state are often an indicator that the host may have also changed state. For example, if Nagios detects that the HTTP service associated with a host just changed from a CRITICAL to an OK state, it may indicate that the host just recovered from a reboot and is now back up and running.

3. Service Check

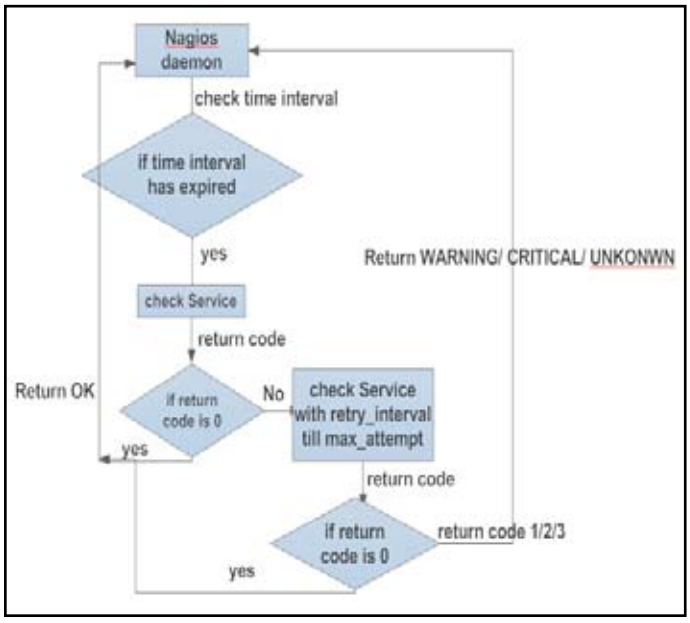

Fig 4: Flowchart of Service Check Workflow

Services are checked by the Nagios daemon:

- At regular intervals, as defined by the *check_interval* and *retry_interval* options in the service definition.
- On-demand as needed for predictive service dependency check.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Return code	Status of service	Status of host
0	OK	UP
1	WARNING	UP or DOWN
2	CRITICAL	DOWN/UNREACHABLE
3	UNKNOWN	DOWN/UNREACHABLE

Fig. 5: Status of Hosts and Services based on return code

V. Implementation

A. Defining NRPE Configuration File

Nagios Remote Plugin Executor (NRPE) is a daemon for running check commands on remote computers. It allows the central Nagios server to trigger checks on remote machines in a secure manner.

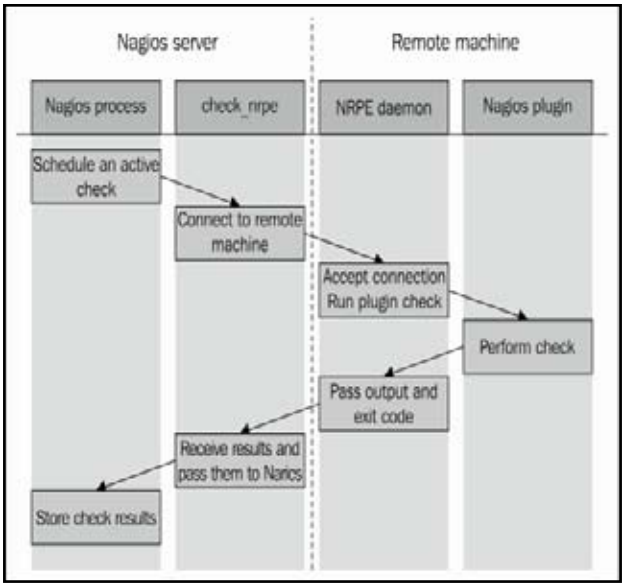

Fig 6: Workflow of NRPE [1]

Nagios daemon runs the check_nrpe plugin that connects to the remote host's NRPE daemon and check_nrpe sends the command to be run to the remote machine. NRPE daemon will accept the connection and run the command and pass the results back to check_nrpe on the machine hosting the Nagios daemon. This information is then passed back to the Nagios daemon.

For NRPE to work user need to specify the host that is allowed to execute check_nrpe plugin in nrpe.cfg file i.e. user needs to specify the IP of the host in "allowed_hosts" field.

Next, user need to specify the command name that needs to be executed on remote host in the nrpe.cfg file along with its arguments. In this project a plugin "check_logs.sh" has been written. So, user specifies the command name "check_storm_logs" along with the plugin "check_logs.sh" and its argument.

B. Defining the Basic Configuration Files

Nagios will not be knowing which host/service to monitor, which plugin to use, time interval for (re)check, time interval for notifications to send, number of times to check, etc. user need to specify these parameters in the appropriate configuration files for Nagios to start monitoring. In templates.cfg user can get the templates for contact, hosts and services configuration files for linux and windows systems. In this project some of the fields and its values in templates.cfg have been retained and some fields have been customized as per the requirements.

- For Nagios to monitor any host/service first it need to be configured in the hosts.cfg and services.cfg configuration files. hosts.cfg file is used to define a physical server, workstation, device, etc. that resides on network. Services.cfg file is used to identify a service that runs on a host.
- Nagios allows grouping the hosts/services that are related, so that the hosts/services that are related to each other can be visualized easily and monitored. This is done using hostgroup.cfg and servicegroup.cfg configuration files.
- Nagios can send notifications to administrator in case of host/service failure. For Nagios to do this, user need to define contacts.cfg configuration file.
- After configuring hosts.cfg, services.cfg, hostgroup.cfg, servicegroup.cfg and contacts.cfg configuration files user need to restart nrpe service on all the nodes in the cluster and Nagios service need to be restarted on monitoring node i.e. on the node where Nagios is installed.

The following commands are used to restart nrpe and Nagios services.

[root@nactive objects]# service nrpe restart
[root@nactive objects]# service nagios restart

VI. Conclusion

Previously Nagios was giving the status of the host and service with only limited notification information. After implementing the chek_logs plugin the cause of the host or service failure will be displayed on the Nagios user interface itself. Manual checking of the log files by the administrator to identify the cause of failure is avoided. Time required to take corrective action by the administrator has been reduced.

VII. Future Enhancement

The plugin can be further enhanced so that it can take the corrective actions automatically without the administrator's intervention. Whenever a service or host goes down the plugin can be designed to restart the host or service automatically by rectifying the error which was found.

References

[1] *"Learning Nagios 4" by Wojciech Kocjan, Packt Publication, 2nd edition, 2014*

[2] *Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Muhammad Inayatullah Babar, "An Efficient Network Monitoring and Management System", presented at International Journal of Information and Electronics Engineering (IEEE), Vol. 3, No. 1, January 2013*

[3] *Ahmed D. KORA, Moussa Moindze SOIDRIDINE, "Nagios Based Enhanced IT Management System" presented at International Journal of Engineering Science and Technology (IJEST), Vol. 4 No.03 March 2012*

[4] *D. Oliveira, T. Vasques, F. Vieira, G. de Deus et al., "A management system for PLC networks using SNMP Protocol," presented at IEEE International Symposium on Power Line Communications and Its Applications (ISPLC), ,,10, Goias, Brazil, June 2010.*

[5] *M. Schubert, A. Hay, D. Bennett et al., "Nagios3 Enterprise Network Monitoring," Designing Configurations for Large Organizations, Chap:2, pp.25-84, 2008.*

[6] *A. Gomez, C. Dafonte, and B. Arcay, "3D Visualization for system and networks monitoring support," presented at 3rd IEEE Conference on Human System Interactions (HSI-10), A Coruna, Spain, July 2010.*
[7] *http://assets.nagios.com/download/nagioscore/docs /nagioscore/3/en/toc.html*
[8] *http://www.techrepublic.com/blog/linux-and-open-source/ nagios-monitoring-with-nrpe-allows-better-tracking-of-remote-systems/*
[9] *https://geekaider.wordpress.com/2012/11/16/add-remote-host-to-nagios-monitoring/*
[10] *http://nagios.manubulon.com/traduction/docs14en / plugintheory*
[11] *http://blog.roozbehk.com/post/25059446631/nrpe-monitoring-linux-remote-hosts-nagios*
[12] *https://wiki.eclipse.org/COSMOS_Design_188390*
[13] *https://en.wikipedia.org/wiki/Network_monitoring*