Monitoring a HPC Cluster with Nagios

Piero Calucci

Scuola Internazionale Superiore di Studi Avanzati
Trieste

2009-04-01
2009-04-03

1 Try again... Fail better.
Outline

1. Nagios Concept
2. Nagios Web Interface
3. Nagios Installation for HPC Monitoring @SISSA
What is Nagios?

“NAGIOS® is a system and network monitoring application. It watches hosts and services that you specify, alerting you when things go bad and when they get better».

— Nagios documentation
All the hard work is done by plugins, the nagios daemon «only» schedules them to be executed at the right time with the right parameters and collect results. The cgi interface is entirely optional, but highly useful.
All the hard work is done by plugins, the nagios daemon «only» schedules them to be executed at the right time with the right parameters and collect results.

The cgi interface is entirely optional, but highly useful.
Nagios Operation

The nagios daemon

- schedules and executes **active** host and service **checks**
- accepts asynchronous **passive checks**
- sends out **notifications** on host or service state change
- executes **event handlers** on host or service state change
- writes and rotates **log** and **state files**
Nagios Operation

The nagios daemon

- schedules and executes **active** host and service checks
- accepts asynchrononous **passive checks**
- sends out **notifications** on host or service state change
- executes **event handlers** on host or service state change
- writes and rotates **log and state files**
Nagios Operation

The nagios daemon

- schedules and executes **active** host and service checks
- accepts asynchronous **passive** checks
- sends out **notifications** on host or service state change
- executes **event handlers** on host or service state change
- writes and rotates **log** and **state** files
Nagios Operation

The nagios daemon

- schedules and executes **active** host and service checks
- accepts asynchronous **passive** checks
- sends out **notifications** on host or service state change
- executes **event handlers** on host or service state change
- writes and rotates **log** and **state** files
Nagios Operation

The nagios daemon

- schedules and executes *active* host and service checks
- accepts asynchronous *passive* checks
- sends out *notifications* on host or service state change
- executes *event handlers* on host or service state change
- writes and rotates *log* and *state* files
All **active checks** involve the local execution of some plugin.
Locally executed plugins can just check for some local service...
...or then can go to the network and check some remote host or service.
NRPE allows execution of plugins on remote hosts. Remote plugin results are reported to nagios by the locally executed `check_nrpe` plugin.
The NSCA daemon relays to Nagios asynchronous notifications sent by `send_nsca` (this is how passive checks work).
Monitoring a HPC Cluster with Nagios

Piero Calucci

Nagios Concept
Web Interface
HPC

Host Checks

- a **host** is basically anything that can be given a name and an address
- hosts can be **UP**, **DOWN**, **UNREACHABLE** (the host may well be up and running, but something in the network in between is broken)
- host checks are executed:
  - at regular intervals
  - on-demand when a service on the host changes state
  - on-demand when required by reachability or dependency logic
Host Checks

- a **host** is basically anything that can be given a name and an address
- hosts can be **UP**, **DOWN**, **UNREACHABLE** (the host may well be up and running, but something in the network in between is broken)
- host checks are executed
  - at regular intervals
  - on-demand when a service on the host changes state
  - on-demand when required by reachability or dependency logic
Host Checks

- a host is basically anything that can be given a name and an address
- hosts can be **UP** **DOWN** **UNREACHABLE** (the host may well be up and running, but something in the network in between is broken)
- host checks are executed
  - at regular intervals
  - on-demand when a service on the host changes state
  - on-demand when required by reachability or dependency logic
Service Checks

- a service is any monitored «thing» associated with a host
- service state can be OK, WARNING, CRITICAL, UNKNOWN
- service checks are executed
  - at regular intervals
  - on-demand when required by dependency logic
Service Checks

- a **service** is any monitored «thing» associated with a host
- service state can be
  - **OK**
  - **WARNING**
  - **CRITICAL**
  - **UNKNOWN**
- service checks are executed
  - at regular intervals
  - on-demand when required by dependency logic
Service Checks

- a service is any monitored «thing» associated with a host
- service state can be
  - OK
  - WARNING
  - CRITICAL
  - UNKNOWN
- service checks are executed
  - at regular intervals
  - on-demand when required by dependency logic
State Types

Each host or service state can be of **SOFT** or **HARD** type. SOFT type states are considered «uncertain» or «transitioning» and are checked with a different (usually higher) frequency until a specified maximum retry count is reached – they then become HARD states.
Flap Detection

- When a host or service changes its state «too frequently» it is detected as being **flapping**
- Flapping hosts do not trigger notifications in order to avoid filling up mailboxes
- Flap detection threshold is configurable, and flap detection can be disabled entirely. However, the defaults are good enough
Notifications

- notifications can be sent out whenever a HARD state transition occurs and when a host or service remains in a non-OK hard state for a specified time
- notification can enabled or disabled for each host or service
- notification times and contact groups can be set up so that only the right person is contacted, only when he is on duty
Event Handlers

- event handlers are executed (if defined) when a host or service changes state and for each retry in SOFT states
- they are given all state information: state, type, retry count
- they can do basically anything, as long as they are given sufficient permissions, including
  - restarting a failed service or host
  - changing nagios configuration by writing to the command pipe (adaptive monitoring)
Tactical Overview

Monitoring a HPC Cluster with Nagios

Piero Calucci

Nagios Concept

Web Interface

HPC
Monitoring a HPC Cluster with Nagios

Piero Calucci

Nagios Concept
Web Interface
HPC

Host State

Host Information
Last Updated: Fri Mar 20 12:51:53 CET 2009
Updated every 90 seconds
Nagios® 3.0.6 - www.nagios.org
Logged in as nagiosadmin

View Status Detail For This Host
View Alert History For This Host
View Trends For This Host
View Alert Histogram For This Host
View Availability Report For This Host
View Notifications This Host

Host State Information
Host Status: **UP** (for 1d 0h 48m 9s)
Status Information: PING OK - Packet loss = 0%, RTA = 0.21 ms
Performance Data: rta=0.207000ms;3000.000000;5000.000000;0.000000 pl=0%;80;100;0
Current Attempt: 1/15 (HARD state)
Last Check Time: 03-20-2009 12:51:34
Check Type: ACTIVE
Check Latency / Duration: 0.590 / 4.276 seconds
Next Scheduled Active Check: 03-20-2009 12:56:44
Last State Change: 03-19-2009 12:03:44
Last Notification: N/A (notification 0)
Is This Host Flapping? **NO** (0.00% state change)
In Scheduled Downtime? **NO**
Last Update: 03-20-2009 12:51:44 (0d 0h 0m 9s ago)
Service Status Detail

Current Network Status
Last Updated: Fri Mar 20 12:51:28 CET 2009
Updated every 90 seconds
Nagios® 3.0.6 - www.nagios.org
Logged in as nagosadmin

View History For This Host
View Notifications For This Host
View Service Status Email For All Hosts

Current Service Status For Host 'c007'

<table>
<thead>
<tr>
<th>Host</th>
<th>Service</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempt</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>c007</td>
<td>EDAC memory errors</td>
<td>WARNING</td>
<td>03-20-2009 12:42:12</td>
<td>0d 14h 9m 16s</td>
<td>3/3</td>
<td>WARNING - several recoverable memory errors</td>
</tr>
<tr>
<td>c007</td>
<td>NFS mounts on local files</td>
<td>OK</td>
<td>03-20-2009 12:41:38</td>
<td>1d 0h 46m 53s</td>
<td>1/3</td>
<td>NFS mounts OK</td>
</tr>
<tr>
<td>c007</td>
<td>NTP server</td>
<td>OK</td>
<td>03-20-2009 12:46:58</td>
<td>1d 0h 22m 31s</td>
<td>1/3</td>
<td>NTP OK Offset -0.0013928002921 secs</td>
</tr>
<tr>
<td>c007</td>
<td>PING</td>
<td>OK</td>
<td>03-20-2009 12:46:31</td>
<td>1d 0h 14m 57s</td>
<td>1/3</td>
<td>PING OK - Packet loss = 0%, RTA = 0.24 ms</td>
</tr>
<tr>
<td>c007</td>
<td>SSH</td>
<td>OK</td>
<td>03-20-2009 12:42:48</td>
<td>1d 0h 28m 41s</td>
<td>1/3</td>
<td>SSH OK - OpenSSH 4.3 (protocol 2.0)</td>
</tr>
<tr>
<td>c007</td>
<td>all events</td>
<td>OK</td>
<td>03-19-2009 11:15:51</td>
<td>1d 16h 16m 0s</td>
<td>1/3</td>
<td>job 22072 by amgrunov</td>
</tr>
</tbody>
</table>
Service Problems

Monitoring a HPC Cluster with Nagios

Piero Calucci

Nagios Concept

Web Interface

HPC

![Service Status Details For All Hosts](image)

<table>
<thead>
<tr>
<th>Host ID</th>
<th>Service Type</th>
<th>Status</th>
<th>Last Check</th>
<th>Duration</th>
<th>Attempt</th>
<th>Status Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>e191</td>
<td>DAC memory err</td>
<td>CRITICAL</td>
<td>03-20-2009 12:42:09</td>
<td>10h 1h 28m 46s</td>
<td>2/9</td>
<td>CRITICAL - many correctable memory errors</td>
</tr>
<tr>
<td>0007</td>
<td>DAC memory err</td>
<td>WARNING</td>
<td>03-20-2009 12:42:12</td>
<td>0d 14h 8m 36s</td>
<td>3/3</td>
<td>WARNING - several correctable memory errors</td>
</tr>
<tr>
<td>0011</td>
<td>DAC memory err</td>
<td>WARNING</td>
<td>03-20-2009 12:44:44</td>
<td>10h 1h 26m 10s</td>
<td>2/3</td>
<td>WARNING - several correctable memory errors</td>
</tr>
<tr>
<td>0045</td>
<td>DAC memory err</td>
<td>WARNING</td>
<td>03-20-2009 12:43:54</td>
<td>10h 1h 27m 26s</td>
<td>3/3</td>
<td>WARNING - several correctable memory errors</td>
</tr>
</tbody>
</table>
What to Monitor

Hosts

- a masternode
- 160+ computing nodes
- several NFS servers
  - including a HA NFS cluster
- lustre servers
What to Monitor

Services

- «generic» services (SSH, NTP, …)
- HPC-specific services (maui, pbs_server, pbs_mom)
- computing node health (load average, hardware errors, …)
What to Do

- send notifications when things go unrecoverably bad
  - avoid sending out notifications every hour (Nagios default) – how often is too often?
- restart services and hosts when possible
  - how much does it take to declare a service crashed or a computing node dead?
  - do we trust Nagios to detect correctly?
  - do we accept the risk of rebooting a node that was just responding late?
What to Do

- send notifications when things go unrecoverably bad
  avoid sending out notifications every hour (Nagios default) – how often is too often?
- restart services and hosts when possible
  - how much does it take to declare a service crashed or a computing node dead?
  - do we trust Nagios to detect correctly?
  - do we accept the risk of rebooting a node that was just responding late?
Open Issues

- «undead» nodes
  - what are we going to do with nodes that reply to ping but nothing else? (they are UP from Nagios PoV)
  - how are we going to reliably detect the undead state?
- host reachability
  - cluster network topology is too simple to make Nagios reachability logic useful (no pingable gateways)
  - we have no way to check single switch port (do we?)
- multihomed nodes
  - Nagios has minimal support for multihomed hosts (or maybe I didn’t understand it...)
  - we have no clear way to know all addresses associated with a node FQDN
Open Issues

- «undead» nodes
  - what are we going to do with nodes that reply to ping but nothing else? (they are UP from Nagios PoV)
  - how are we going to reliably detect the undead state?

- host reachability
  - cluster network topology is too simple to make Nagios reachability logic useful (no pingable gateways)
  - we have no way to check single switch port (do we?)

- multihomed nodes
  - Nagios has minimal support for multihomed hosts (or maybe I didn’t understand it...?)
  - we have no clear way to know all addresses associated with a node FQDN
Open Issues

- «undead» nodes
  - what are we going to do with nodes that reply to ping but nothing else? (they are UP from Nagios PoV)
  - how are we going to reliably detect the undead state?
- host reachability
  - cluster network topology is too simple to make Nagios reachability logic useful (no pingable gateways)
  - we have no way to check single switch port (do we?)
- multihomed nodes
  - Nagios has minimal support for multihomed hosts (or maybe I didn’t understand it...)
  - we have no clear way to know all addresses associated with a node FQDN
After 3 weeks we are over 30% CPU usage on a 2x dual core opteron...
Open Issues
File Access Permissions Mess

On monitoring server:

- nagios needs to read its own configuration files and write log and status files
- web server needs to read some nagios config files and write to nagios command pipe
- nagios event handlers need to write to nagios command pipe
- nsca needs to read its own configuration and write to nagios command pipe

On monitored hosts:

- nrpe needs to read its own configuration and execute nagios plugins
- send_nsca needs to read its own configuration
Open Issues
File Access Permissions Mess

On monitoring server:

- nagios needs to read its own configuration files and write log and status files
- web server needs to read some nagios config files and write to nagios command pipe
- nagios event handlers need to write to nagios command pipe
- nsca needs to read its own configuration and write to nagios command pipe

On monitored hosts:

- nrpe needs to read its own configuration and execute nagios plugins
- send_nsca needs to read its own configuration
Example
Check pbs_mom /1

```plaintext
define host {
    use          linux-server
    host_name    p001
    address      10.2.13.1
}

define hostgroup {
    hostgroup_name p-nodes
    alias         planck nodes
    members       p001,p002,p003,...
}
```
define service {
    use generic-service
    hostgroup_name m-nodes, c-nodes, p-nodes, ...
    service_description pbs_mom
    check_command check_pbs_mom
    max_check_attempts 4
    notifications_enabled 1
    event_handler restart_pbs_mom
    servicegroups sg_node_batch
}

Example
Check pbs_mom /2
Example
Check pbs_mom /3

define command {
    command_name    check_pbs_mom
    command_line    $USER1$/check_tcp
                    -H $HOSTADDRESS$
                    -p 15002
}

define command {
    command_name    restart_pbs_mom
    command_line    /.../eventhandlers/restart_pbs_mom.sh
                    $HOSTNAME$ $SERVICESTATE$
                    $SERVICESTATETYPE$
                    $SERVICEATTEMPT$
}

Monitoring a HPC Cluster with Nagios
Piero Calucci
Nagios Concept
Web Interface
HPC
Example
Check pbs_mom /4

```
restart_pbs_mom.sh is:

#!/bin/sh

SSH="ssh -i /var/lib/nagios/.ssh/id_dsa -t -t"
RESTART="sudo /sbin/service pbs_mom restart"

case $2 in
  CRITICAL)
    case $3 in
      SOFT)
        case $4 in
          3)
            $SSH $1 $RESTART
          ;;
        esac
      ;;
    esac
    ;;
  HARD)
    $SSH $1 $RESTART
    ;;
esac
```

Piero Calucci
Nagios
Concept
Web Interface
HPC