Monitoring Systems and Services

Alwin Brokmann
DESY-IT
March 24 – 28, 2003
CHEP 2003 San Diego
## Requirements

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Monitoring</td>
</tr>
<tr>
<td>Service Monitoring</td>
</tr>
<tr>
<td>Navigation</td>
</tr>
<tr>
<td>User specific Parameters</td>
</tr>
<tr>
<td>WEB Interface</td>
</tr>
<tr>
<td>Alarming</td>
</tr>
<tr>
<td>Escalation</td>
</tr>
<tr>
<td>Simple Configuration</td>
</tr>
<tr>
<td>Reporting</td>
</tr>
<tr>
<td>Interface to Trouble Ticket System</td>
</tr>
<tr>
<td>Fault Isolation</td>
</tr>
</tbody>
</table>
Why NAGIOS

There are several reasons for us to select NAGIOS:

a. Fulfills most of our requirements
b. Possibility to submit our own tests (Plug In Concept)
c. Scalability
d. Design (WEB Interface)
e. Out of the Box functionality
e. Price
History

Starting Oct. 2001
One PC running NetSaint -- Monitoring 50 Hosts

March 2002
3 PC`s running NetSaint -- Monitoring ~300 Hosts, several Proc`s

Jun 2002
Testing logsurfer

Nov 2002
NAGIOS 1.0 available
4 PC`s running NAGIOS ~ growing Number of Hosts and Services
LogHost is running /connecting to NAGIOS

Feb 2003
New Service Checks for AFS
Monitoring Policy

Every Host in the Computer Center will be monitored and also Centrally Supported Printer`s

<table>
<thead>
<tr>
<th>Host</th>
<th>Check by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Device</td>
<td>PING</td>
</tr>
<tr>
<td>Farm PC</td>
<td>PING</td>
</tr>
<tr>
<td>Printer</td>
<td>SNMP</td>
</tr>
<tr>
<td>Workgroup Server</td>
<td>Load, Disk, Process</td>
</tr>
<tr>
<td>Mail</td>
<td>POP, IMAP</td>
</tr>
<tr>
<td>WEB Server</td>
<td>HTTP</td>
</tr>
<tr>
<td>AFS Server</td>
<td>Service Monitoring</td>
</tr>
</tbody>
</table>
Monitoring Service for Clusters

**Hardware Cluster:**
Mail cluster, consisting of 2 computers.

**Service Cluster:**
YP cluster which consists of several computers for the YP Service

To make the check of a cluster possible we need a Check Cluster Plug In.

We can define for each cluster how many components may fail before an alarm is triggered.
Monitoring AFS

With the introduction of OpenAFS at DESY we experienced, that a simple process monitoring gives no reliable answers.

Therefore we added some new tests in NAGIOS to ensure the operation of the AFS Servers.

For these tests we use afs tools like rxdebug, vos etc..

The result is then transferred to NAGIOS.
**Host Statistics**

(Feb. 2003)

<table>
<thead>
<tr>
<th>Host</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>~630</td>
<td>~1300</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer</td>
<td>17</td>
</tr>
<tr>
<td>Network</td>
<td>37</td>
</tr>
<tr>
<td>UNIX</td>
<td>550</td>
</tr>
<tr>
<td>Windows</td>
<td>26</td>
</tr>
</tbody>
</table>
At present we use simple flat files to describe host and services. This means a very high effort, but makes it simply possible to distribute the checks.

For the future we plan to produce the tests automatically over our AMS.

For an automatic configuration of NAGIOS we will extract the informations such as computer name, interface and naturally which service runs on the computer from the AMS database.

Perhaps we will hold the data in a database.
Configuration Example

define hostgroup{
    name        night
    hostgroup_name   night
    alias        night
    contact_groups sgi-admins,night-admins
    members       netra8,test1,test2
}
define host{
    host_name       netra8
    alias            netra AFS Server
    address          131.169.40.109
    parents          route-194,route-40
    use              hostcheck
}
define service{
    use            fileserver
    host_name       netra8,test1,test2
    contact_groups  afs-admins,night-admins
}
## Configuration Example

```plaintext
define service{
    name fileserver
    service_description fileserver
    is_volatile 0
    active_checks_enabled 0
    passive_checks_enabled 1
    check_period 24x7
    max_check_attempts 10
    normal_check_interval 1
    retry_check_interval 5
    notification_interval 2200
    notification_period 24x7
    notification_options w,u,c,r
    check_command check_named_proc!'"'!fileserver
    register 0
}
```
Monitoring Setup @ DESY IT

Central Monitoring Server

SUSE Linux
Kernel: 2.4.18
2 x Pentium III
1.2 GHz
RAM 1 GB
DISK 2 x 40 GB

Log Host

SUSE Linux
Kernel: 2.4.18
2 x Pentium III
1.2 GHz
RAM 1 GB
DISK 4 x 40 GB

Distributed Monitoring Server

SUSE Linux
Kernel: 2.4.10
Pentium IV
1.7 GHz
RAM 256 MB
DISK 40 GB
Monitoring Server Setup
Operator Console
Problem Notification

**** Nagios 1.0 *****
Notification Type: PROBLEM Service: IT Web Server
Host: WWW Server WEB
Address: 131.169.40.38
State: CRITICAL
Date/Time: Tue Mar 19 08:35:59 MET 2003
Additional Info: Connection refused by host
Recovery Notification

***** Nagios 1.0 *****
Notification Type: RECOVERY
Service: IT Web Server
Host: WWW Server WEB
Address: 131.169.40.38
State: OK Date/Time: Wed Mar 19 08:37:46 MET 2003
Additional Info: HTTP ok: HTTP/1.1 200 OK - 0 second response time
* Works on any textfile (or text from standard input)
* Matching of lines is done by two regular expression (logline must match the first expression but must not match the optional second regular expression). So you are able to specify exceptions.
* Uses contexts (collection of messages) instead single lines
* Flexible but easy configuration
* Timeouts and resource limits included
* Handles "shifting" of logfiles
* Dynamic rules can change the actions associated with logmessages (something might happen that makes you interested in messages you would usually drop)
* Multiple reactions on one logline possible
References

NAGIOS
www.nagios.org

SNMP
www.net-snmp.org

logsurfer
www.dfn-cert.de/eng/logsurf/index.html