

Groundwork monitoring for HP Lefthand P4000 SANs

revision 1.0

October 19, 2010

P4000 Monitoring part 3: Creating Alert Thresholds and Graphing Performance

presented by:



Business Partner



Introduction:

This document attempts to explain how to configure a groundwork monitor nagios server to monitor HP Lefthand P4000 series SAN modules and virtual appliances. The concepts should be applicable to any Nagios implementation; however the specific instructions and screenshots included were using [Groundwork Community Edition](#). These methods were specifically tested against GWCE versions 5.6.1 CentOS is used in the examples. The version of ESX used was vSphere 4 Update1



Prerequisites:

- Completed PART 1, initial configuration of your GroundWork Monitor appliance.
- Completed PART 2 configuring check commands, services, and storage hosts.
- Working knowledge of the console and configuration interfaces for Linux.
- Properly installed, configured, HP Lefthand P4000 series Cluster
- Ability to work with vi/nano or other Linux editor

Goals:

Our end-result should be monitoring capabilities of multiple HP Lefthand P4000 series storage modules, Virtual Storage Appliances (VSA), clusters, management groups, hardware resources, volume statistics and utilization. Trending and historical graphing of the utilization and alerting based on warning and critical thresholds supplied.

About the Author:

Paul Drangeid is a senior systems architect and partner at TeleData Consulting, Inc.
 Areas of competence focused on the following technologies:
 Virtualization (VMware ESX/vSphere, Capacity Planning, SRM, XenServer, HyperV)
 Storage (SCSI, iSCSI, SAN, DAS, NAS, DRP and replication)
 Microsoft (SQL Server, Exchange, Active Directory, Terminal Services, general infrastructure)
 Citrix (Winframe – Xenapp; Web Interface, Secure Gateways)

Resources and Tools used:

Nagios	GroundWork OpenSource Community Edition 6.1	http://sourceforge.net/projects/gwmos/files/installable (BIN) and Vmware appliance versions available.
SCP	Veeam FastSCP	http://www.veeam.com/vmware-esxi-fastscp.html
Check comma nds	Various shell and perl check commands	Links supplied within document

Alerts and Notification

Now that we have working monitoring hosts and services there are a few you will want to configure some alerting thresholds for:

Typically the most telling storage bottleneck will be Disk Queuing. This is identifying how many IO requests are stacking up and waiting in line. To remedy queuing you generally will need to increase cache, spindle count, and/or lower contention. (ie: running large backup jobs during heavy database report generation).

Generally a healthy storage system will not be (sustaining) queuing at or above 2 times the physical number of disks in the storage system. So I present these guidelines. These are not definitive thresholds. Transactional systems like Exchange, SQL etc are much more sensitive to disk queuing whereas file share/backups are much more tolerant, so use the following numbers as a guide:

# disks in system	Warning Threshold	Critical Threshold
8 (P4300 Models)	20	26
12 (P4500 Models)	24	39
16 (Starter SAN)	32	52
24 (Virtualization SAN)	48	78
32 (2 Starter SANs)	64	104
48 (Multi-Site SANs)	96	156
96 (2 Multi-Site SANs)	192	300

You will want to configure thresholds for both individual modules and the entire cluster. Why? This can help you identify a faulty module. What if disk queue levels in the entire cluster rise, but you are seeing the queuing grow substantially on just one particular storage module, that may be an indication of a problem with that storage node, and not a cluster-wide throughput limitation.

To Configure the Thresholds for your Storage Module:

Configuration → Hosts → Storage → Your Module → nsm_queue
 Click the Service Check tab

Service name:	nsm_queue
<input type="checkbox"/> Inherit check from template	
Check command:	check_nsm_queuedepth
Command definition:	\$USER1\$/check_snmpx \$HOSTADDRESS\$ \$USER7\$ LEFT MIB::clusModuleStatsQDepthTotal.1 \$ARG1\$ \$ARG2\$ Queue
Usage:	check_nsm_queuedepth!ARG1!ARG2
Command line:	? check_nsm_queuedepth!30!50
Test:	<div style="border: 1px solid gray; padding: 5px;"> Host: <input type="text" value="tdnsm01"/> <pre> /usr/local/groundwork/nagios/libexec/check_snmpx 10.0.5.5 tdsnmp LEFTHAND-NETWORKS-MIB::clusModuleStatsQDepthTotal.1 30 50 QueueDepth SNMP OK - 0 QueueDepth=0;30;50; Command returned exit status 0 </pre> </div>

Note the Arguments 30 and 50 (I choose more aggressive thresholds, as this module should have relatively low load).

Click **Test** You should see SNMP OK – ## | QueueDepth=0;30;50;
 If the test results look normal then click **Save**

Note: QueueDepth is a TOTAL for modules. Individual Read and Write metrics are available at the Cluster level.

Click to the Service Detail Tab, and make sure that Notifications are enabled, and that the Notification Options match your desired behavior:

<input checked="" type="checkbox"/> Is volatile:	?	<input type="checkbox"/>
<input checked="" type="checkbox"/> Check period:	?	24x7
<input checked="" type="checkbox"/> Max check attempts:	?	3
<input checked="" type="checkbox"/> Normal check interval:	?	10
<input checked="" type="checkbox"/> Retry check interval:	?	1
<input checked="" type="checkbox"/> Active checks enabled:	?	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Passive checks enabled:	?	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Obsess over service:	?	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Check freshness:	?	<input type="checkbox"/>
<input checked="" type="checkbox"/> Freshness threshold:	?	
<input checked="" type="checkbox"/> Notifications enabled:	?	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Notification interval:	?	60
<input checked="" type="checkbox"/> Notification period:	?	24x7
<input checked="" type="checkbox"/> Notification options:	?	<input checked="" type="checkbox"/> Unknown <input checked="" type="checkbox"/> Critical <input checked="" type="checkbox"/> Warning <input checked="" type="checkbox"/> Recovery <input type="checkbox"/> Flapping <input type="checkbox"/> Downtime <input type="checkbox"/> None
<input checked="" type="checkbox"/> Event handler enabled:	?	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Event handler:	?	

Note the left checkbox tells Nagios if you want to Inherit the settings (default).
 If you wish to override the inheritance UNCHECK the left checkbox, and then you can change the settings at the right. If you leave the checkbox on the left checked, no changes to settings on the right will be performed. It is simply SHOWING you the inherited settings.

You may not have yet configured notification settings at the Nagios Server level.

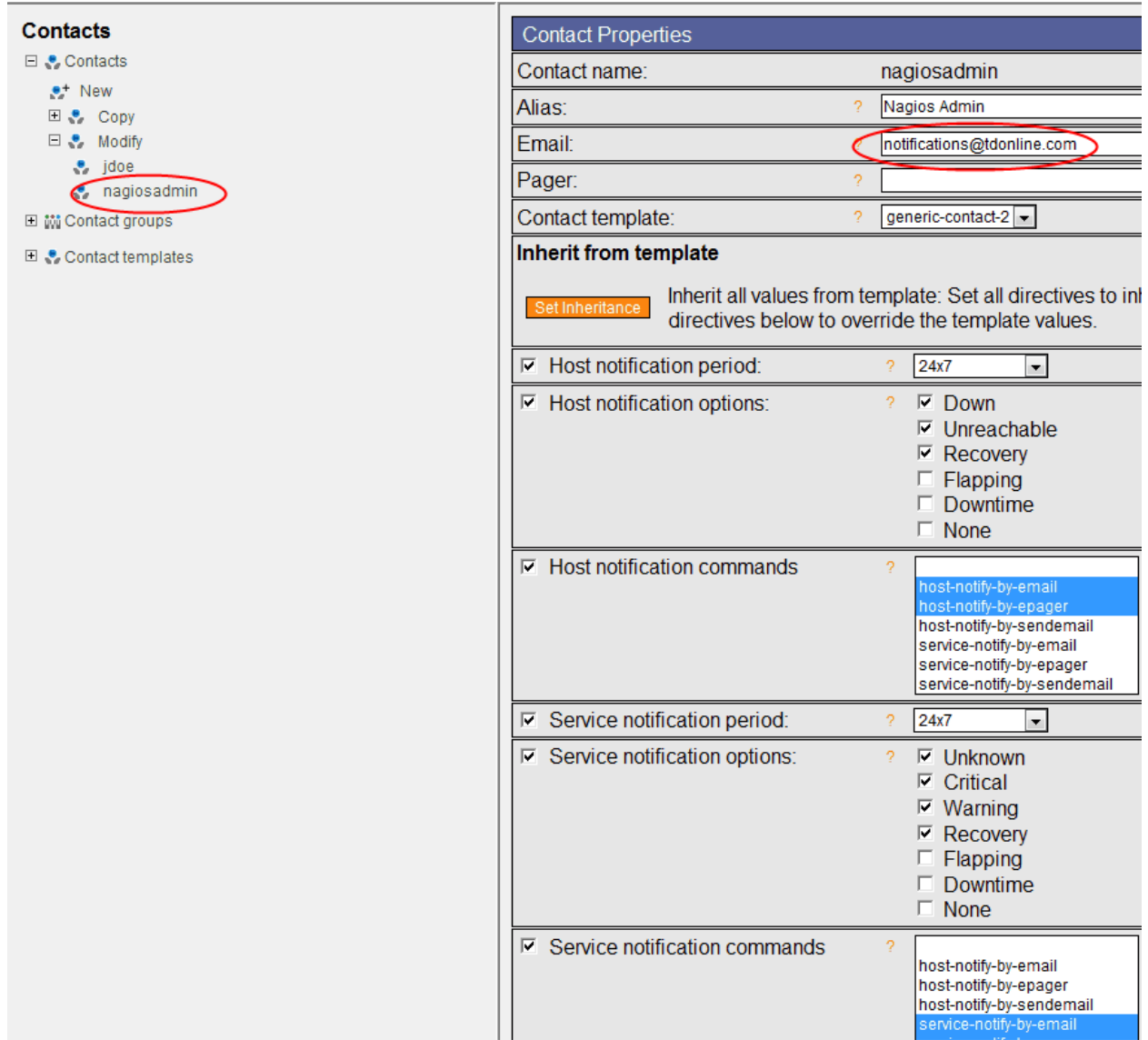
To do that go to **Configuration → Contol → Nagios main configuration:**

Nagios Main Configuration Page 1	
Log file:	? <input type="text" value="/usr/local/groundworl"/>
Object cache file:	? <input type="text" value="/usr/local/groundworl"/>
Precached object file:	? <input type="text" value="/usr/local/groundworl"/>
Resource file:	? <input type="text" value="/usr/local/groundworl"/>
Temp file:	? <input type="text" value="/usr/local/groundworl"/>
Status file:	? <input type="text" value="/usr/local/groundworl"/>
Aggregated status data update interval:	? <input type="text" value="15"/>
Nagios user:	? <input type="text" value="nagios"/>
Nagios group:	? <input type="text" value="nagios"/>
Enable notifications:	? <input checked="" type="checkbox"/>
Execute service checks:	? <input checked="" type="checkbox"/>

Verify that Enable notifications is checked

If you have not setup a Contact you will also need to do that:

Configuration → Contacts → Contacts → Your Contact (you may need to create New)



The screenshot shows the Nagios Groundwork interface for configuring a contact. On the left, a sidebar lists 'Contacts' with sub-items: 'New', 'Copy', 'Modify', 'jdoe', 'nagiosadmin' (circled in red), 'Contact groups', and 'Contact templates'. The main area is titled 'Contact Properties' and contains the following fields:

- Contact name:** nagiosadmin
- Alias:** Nagios Admin
- Email:** notifications@tdonline.com (circled in red)
- Pager:** (empty)
- Contact template:** generic-contact-2

Below these fields is the 'Inherit from template' section, which includes a 'Set Inheritance' button and the text: 'Inherit all values from template: Set all directives to inherit from the template values. Directives below to override the template values.'

The configuration is divided into three sections, each with a checked checkbox and a dropdown menu:

- Host notification period:** 24x7
- Host notification options:**
 - Down
 - Unreachable
 - Recovery
 - Flapping
 - Downtime
 - None
- Host notification commands:**
 - host-notify-by-email
 - host-notify-by-epager
 - host-notify-by-sendemail
 - service-notify-by-email
 - service-notify-by-epager
 - service-notify-by-sendemail

The 'Service notification' section is identical in structure to the 'Host notification' section, with the following options:

- Service notification period:** 24x7
- Service notification options:**
 - Unknown
 - Critical
 - Warning
 - Recovery
 - Flapping
 - Downtime
 - None
- Service notification commands:**
 - host-notify-by-email
 - host-notify-by-epager
 - host-notify-by-sendemail
 - service-notify-by-email

Note the various notification options: Host Notification Period, Notification Options, Service Notification Period, Service Notification Options.

Click **Save** and then Preflight Test and **Commit**. If you choose you can have different email addresses for Email and Pager. (This way you could use your mail server as an SMS gateway to send more critical notifications as a text message for example).

As a general rule I setup notification distribution groups on my corporate mail server, that way you don't have to come back and reconfigure Nagios Groundwork to change who should receive notifications, simply change membership in a distribution group.

If you have configured everything correctly you should start getting harassed by your Nagios Server if something goes awry:

Since we have thresholds that we hope our SAN is not currently reaching, you may want to create an artificial problem to verify that notifications are working. This document assumes that your sendmail configuration is properly configured. (Briefly shown in Part 1) Configuring and troubleshooting sendmail functionality is beyond the scope of our document. Please Google for “sendmail configuration” for help with that.

Go to **Configuration → Hosts → Storage → Your Storage Module → nsm_netstatus**
 Select the Service Check Tab, and change the expected results to something different:

Service instance name suffix: <input type="text"/>		or enter a number
<input type="button" value="Add Instance(s)"/>		
Instance Name Suffix	Status	Arguments
<input type="checkbox"/> <input type="text" value="_bond0"/>	<input checked="" type="checkbox"/> Active	<input type="text" value="!1!fakeerror"/>
<input type="checkbox"/> <input type="text" value="_port1"/>	<input checked="" type="checkbox"/> Active	<input type="text" value="!2!Active"/>
<input type="checkbox"/> <input type="text" value="_port2"/>	<input checked="" type="checkbox"/> Active	<input type="text" value="!3!Active"/>
<input type="button" value="Remove Instance(s)"/> <input type="button" value="Check All"/> <input type="button" value="Uncheck All"/>		
<input type="button" value="Save"/>		

In this case I changed the expected response to “fakeerror” which should generate some fireworks for us. Click and then Preflight Test and .

Within 10 minutes you should receive an email that looks something like this:

```

***** Nagios *****

Notification Type: PROBLEM

Service: nsm_netstatus_bond0
Host: NSM 2120 G1 Module01
Address: 10.0.5.5
State: CRITICAL

Date/Time: Thu Mar 25 18:17:38 CDT 2010

Additional Info:

SNMP CRITICAL - *Active*
    
```

If not check all the requirements. Is sendmail working? Did you allow SMTP relay on your mail server FROM the IP address of your GroundWork server? Are notifications enabled?

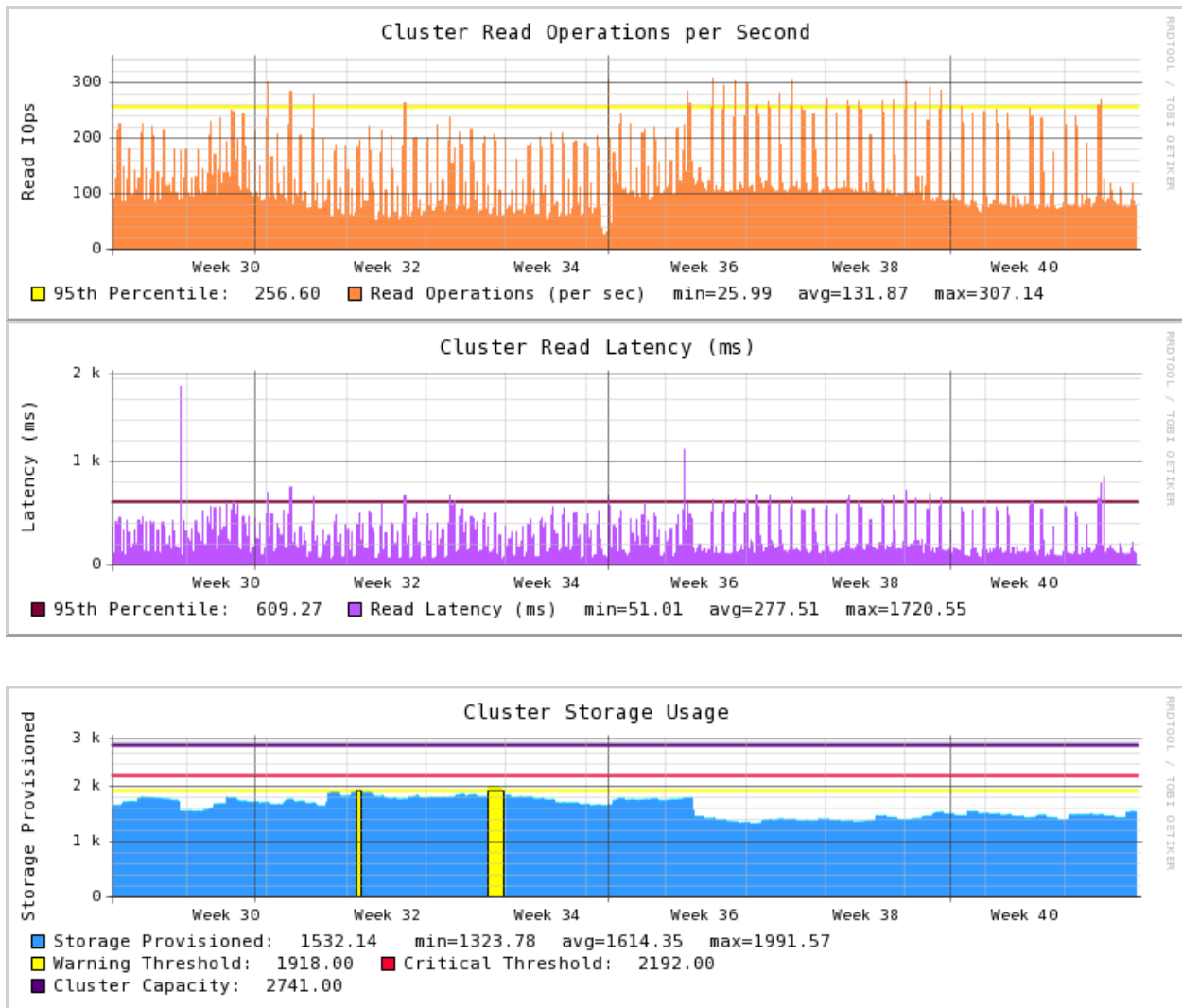
I have found it particularly useful to create rules on your email client to categorize your nagios alerts. For example it will look for email from nagios with the word "CRITICAL" in the subject, assign it to the "critical/RED" category, flag it with high importance and require follow-up.

I recommend that you verify notification for the following monitors:

- Nsm_Batterycache
- Ping
- Nsm_Drivebay
- Nsm_Fanstatus
- Nsm_Queue
- Nsm_Raid
- Nsm_Storage
- Lhc_usage
- Lhc_writequeue
- Lhc_readqueue

Graphing and Performance Tracking

By tracking usage, performance metrics, and throughput with graphs, we can easily view long-term usage patterns, and identify changes in load, capacity, and get ahead of the curve on when it is time to add storage capacity or throughput. For Example:



We can create graphs at various levels:

Cluster

Throughput, usage, and performance metrics for the entire cluster.

Management Group

Throughput, usage, and performance metrics for volumes within a particular management group. These include Bytes, IOPs, latency, and usage statistics.

Storage Module (node level)

Throughput, health monitoring, and performance metrics for individual storage nodes. These include Bytes, IOPs, latency, hardware monitoring (network, PSU, drivebays, fans etc). **Note: some of the legacy modules don't properly monitor node level performance metrics via SNMP.**

You should see green, verifying you are getting results for the checks.

Check a particular service, like lhc_readkbs and verify it is returning valid data

- icmp_ping_alive
- lhc_readios
- lhc_readkbs
- lhc_readlatency
- lhc_readqueue
- lhc_usage
- lhc_writeios
- lhc_writekbs
- lhc_writelatency
- lhc_writequeue

Status Information

- Status** SNMP OK - 4853688770560
- Acknowledged** N/A
- In Downtime** No [Schedule](#)
- Last Notification** N/A (notification 0) [Disable Notifications](#)

Check Information

- Next Check** Sun Oct 17 11:17:19 CDT 2010 (Active) [Schedule](#)
- Check Attempt** 1 of 3 (Soft) [Disable Check](#)
- Latency / Duration** 0.03 / 0.09
- % State Change** 0.0%

Now we go to configure a graph for lhc_readkbs as follows:

Configuration → Performance → Select icmp_ping and click Copy

Dashboards
My GroundWork
Event Console
Status
Reports
Configuration
Auto Discovery
Nagios

Services
Profiles
Hosts
Contacts
Escalations
Commands
Time Periods
Groups
Control
Tools
Performance

Performance Configuration

Performance Configuration Administration

Create New Entry
Export All

Select Service-Host entry: icmp_ping - *

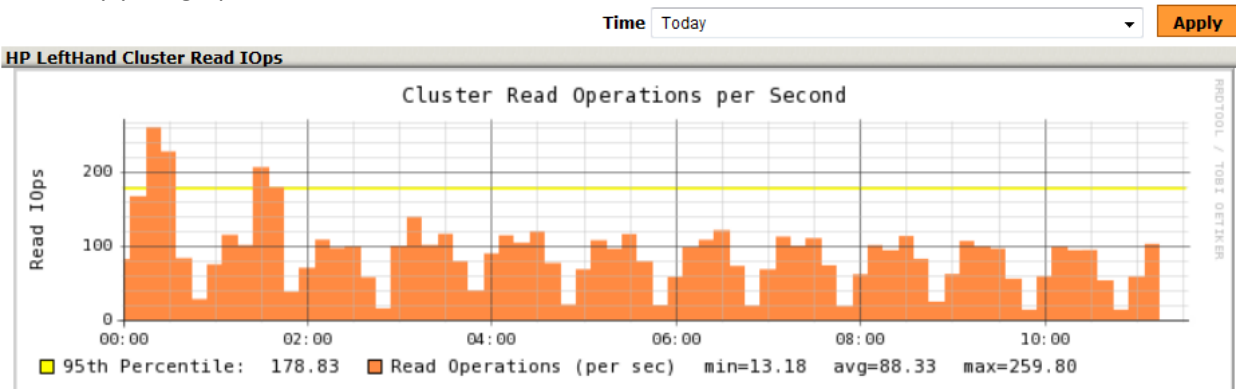
Graph Label:	ICMP Ping Response Time
Service:	icmp_ping
Use Service as a Regular Expression:	ON
Host:	*
Status Text Parsing Regular Expression:	
Use Status Text Parsing instead of Performance Data:	OFF
RRD Name:	/usr/local/groundwork/rrd/\$HOST_\$SERVICE\$.rrd
RRD Create Command:	\$RRDTOOL\$ create \$RRDNAME\$ --step 300 --start n-1yr DS:rta:GAUGE:1800:U:U DS:pl:GAUGE:1800:U:U RRA:AVERAGE
RRD Update Command:	\$RRDTOOL\$ update \$RRDNAME\$ \$LASTCHECKS:\$VALUE1:\$VALUE2\$ 2>&1
Custom RRDtool Graph Command:	'rrdtool graph - --imgformat=PNG --title="ICMP Performance" --rigid --base=1000 --height=120 --width=700 --alt-aut DEF:a="rrd_source";ds_source_1:AVERAGE DEF:b="rrd_source";ds_source_0:AVERAGE CDEF:cdefa=b CDEF:cdefb=GPRINT:cdefa:AVERAGE:"Average\:%8.2lf %s" GPRINT:cdefa:MAX:"Maximum\:%8.2lf %s\n" LINE1:cdefb#307D7E:P %s" GPRINT:cdefb:MAX:"Maximum\:%8.2lf %s"'
Enable:	ON

Modify
Copy
Delete
Export

Modify the fields to match the information in the following table, then click Create Copy	
Graph Label:	HP LeftHand Cluster Read Mbps
Service:	lhc_readkbs
Use Service as a Regular Expression:	OFF
Host:	*
Status Text Parsing Regular Expression:	<i>unchecked</i>
Use Status Text Parsing instead of Performance Data:	OFF
RRD Name	/usr/local/groundwork/rrd/\$HOST\$_\$SERVICE\$.rrd
RRD Create Command	\$RRDTOOL\$ create \$RRDNAME\$ --step 300 --start n-1yr DS:readkbs:DERIVE:1800:0:20000000 RRA:AVERAGE:0.5:1:8640 RRA:AVERAGE:0.5:12:9480
RRD Update Command	\$RRDTOOL\$ update \$RRDNAME\$ \$LASTCHECK\$: \$VALUE1\$ 2>&1
Custom RRDtool Graph Command	'rrdtool graph - DEF:a="rrd_source":ds_source_0:AVERAGE CDEF:bps=a,8,* CDEF:kbs=bps,1024,/ / CDEF:mbs=kbs,1024,/ VDEF:ds95pct=mbs,95,PERCENT LINE2:ds95pct#7F0037:"95th Percentile\:" GPRINT:ds95pct:"%.2lf" AREA:mbs#41892D:"Mbps (read)" GPRINT:mbs:MIN:min=%.2lf GPRINT:mbs:AVERAGE:avg=%.2lf GPRINT:mbs:MAX:max=%.2lf -c BACK#FFFFFF -c CANVAS#FFFFFF -c GRID#C0C0C0 -c MGRID#404040 -c ARROW#FFFFFF -Y --height 120 --vertical-label="Mbps" -- title="Cluster Megabits per second (Read)" --alt- autoscale-max -l 0 --rigid'EnableON
Enable:	ON

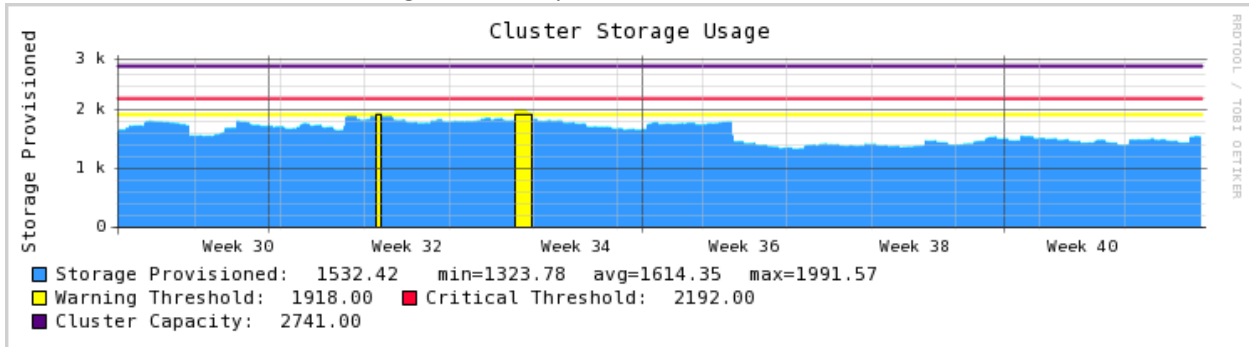
Once you save this graph it will take 20-30 minutes before you begin getting performance data displayed in the graph for the service. (The graph needs at least 2 datapoints to create a graph).

Eventually your graph should look like this:



Using the “Create Copy” method, you can create graphs for ALL of the cluster and Module services. See Appendix A for the remaining graph definitions.

You may have notices a check command you did not have previously: `lhc_usage`
 This returns the entire cluster usage and free space:



You need a custom check command for this functionality. This is available from my collection of P4000 checks: <http://www.tdonline.com/training/scripts/p4000-checks.zip>. The text for these check commands is listed in Appendix B.

Download, unzip and copy the commands into your `/usr/local/groundwork/nagios/libexec` folder. Be sure to grant execute permissions to the Nagios user.

Command Definition:

Command Name:	<code>check_lhc</code>
Type:	Check
Command Line:	<code>`\${USER1}`/check_lhc `\${HOSTADDRESS}` `\${USER7}` `\${ARG1}` `\${ARG2}` `\${ARG3}`</code>
Usage:	<code>check_lhc!ARG1!ARG2!ARG3</code>

Service Definition:

Service Name:	<code>Lhc_usage</code>
Check Command	<code>Check_lhc</code>
Command Definition	<code>`\${USER1}`/check_lhc `\${HOSTADDRESS}` `\${USER7}` `\${ARG1}` `\${ARG2}` `\${ARG3}`</code>
Usage:	<code>check_lhc!ARG1!ARG2!ARG3</code>

Example: `(check_lhc!warningthreshold!criticalthreshold!cluster_instance)`
`check_lhc!80!90!1`

Results:

```
/usr/local/groundwork/nagios/libexec/check_lhc2 10.0.5.75 public 80 90 1
OK - "nuvem-sata01" is 87.00 % free | clustersize=8009 available=7026
warning=6407 critical=7208 provisioned=983 nodes=2
```

Create the command, service, and add it to the service entries to your newly created cluster host entry.



Groundwork monitoring for HP Lefthand P4000 SANs Author: Paul Drangeid | <http://www.tdonline.com>



Groundwork monitoring for HP Lefthand P4000 SANs Author: Paul Drangeid | <http://www.tdonline.com>

So now you should be successfully collecting Module statistics with graphs, and Cluster statistics with graphs.



Groundwork monitoring for HP Lefthand P4000 SANs Author: Paul Drangeid | <http://www.tdonline.com>

Appendix A

Appendix B

check_lhc

```
#!/bin/sh
#
STATE_OK=$(expr 0)
STATE_WARNING=$(expr 1)
STATE_CRITICAL=$(expr 2)
STATE_UNKNOWN=$(expr 3)
CLUSTERINSTANCE=$5

# figure out the name of the cluster for instance supplied on command line
clustername=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C
$2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusClusterName.$CLUSTERINSTANCE|cut -d" " -f4)
#echo "Clustername=$clustername"

# figure out Available Space
clusteravail=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C
$2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusClusterSpaceAvailable.$CLUSTERINSTANCE|cut -d" " -f4)

# total number of modules in management group
totalmodules=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C
$2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusModuleCount.0|cut -d" " -f4)
#echo "totalmodules=$totalmodules"

# total number of modules in this particular cluster
clustermodules=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1
-C $2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusClusterModuleCount.$CLUSTERINSTANCE|cut -d" " -f4)
#echo "clustermodules=$clustermodules"

checkcount=0

#cycle through each module.  Get the totalsize, check only those who match
our clustername, then find the smallest sized module in our cluster, multiply
it by
# the number of modules in our cluster - that should give us our total
cluster capacity.
while [ "$checkcount" -lt $totalmodules ]
do
ck=$(echo $checkcount + 1 | bc)

#echo "checking $ck"
ibelongto=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C $2
-o LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusModuleClusterName.$ck|cut -d" "
-f4)
#echo "ibelongto is $ibelongto"

#echo "does $ibelongto equal $clustername ? "
```



```
if [ "$ibelongto" = "$clustername" ] ; then

modtotal=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C $2
-o LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusModuleTotalSize.$ck|cut -d" " -
f4)
#echo "modtotal is $modtotal"
modtotal=$(echo "$modtotal" | bc)
if [ checkmodtotal=0 ] ; then
checkmodtotal=$(echo "$modtotal" | bc)
fi

# echo "Check to see if $modtotal is -le $checkmodtotal"
if [ $modtotal -le $checkmodtotal ] ; then
    checkmodtotal=$(echo "$modtotal" | bc)
# echo "setting total to $checkmodtotal"
fi

fi

#echo "checkmodtotal is $checkmodtotal"
checkcount=$((checkcount+1))
done

clustertotal=$(echo "$checkmodtotal * $clustermodules" | bc)

#convert to MB
clustertotal=$(echo "$clustertotal / 1024" | bc)
clusteravail=$(echo "$clusteravail / 1024" | bc)

#convert to GB
clustertotal=$(echo "$clustertotal / 1024" | bc)
clusteravail=$(echo "$clusteravail / 1024" | bc)
totalused=$(echo "$clustertotal - $clusteravail" |bc)

# calculate the current percentage free of the cluster
percentfree=$(echo "scale=2; $clusteravail/$clustertotal " |bc)
#echo "percentfree=$percentfree"

wp=$(echo "scale=2; .01*$3" | bc)
cp=$(echo "scale=2; .01*$4" | bc)

# calculate the GB that equals the warning threshold:
wt=$(echo "scale=10; $clustertotal * $wp" | bc)
# get rid of the decimal places that bash can't deal with
wt=$(echo "$wt /1" | bc)
#echo "the wt is $wt"

# calculate the GB that equals the critical threshold:
ct=$(echo "scale=2; $clustertotal * $cp" | bc)
# get rid of the decimal places that bash can't deal with
ct=$(echo "$ct /1" | bc)
#echo "the ct is $ct"
```



```
#echo "wp=$wp"
#echo "ct=$ct"
#echo "wt= $wt"

cu=$(expr $clustertotal - $clusteravail)

percentfree=$(echo "$percentfree*100" | bc)

if [ $cu -ge $ct ] ; then
    echo "CRITICAL - *$clustername is $percentfree % free* |
clustersize=$clustertotal available=$clusteravail warning=$wt critical=$ct
provisioned=$totalused nodes=$clustermodules"
    exit $STATE_CRITICAL

elif [ $cu -le $wt ] ; then
    echo "OK - $clustername is $percentfree % free |
clustersize=$clustertotal available=$clusteravail warning=$wt critical=$ct
provisioned=$totalused nodes=$clustermodules"
    exit $STATE_OK

elif [ $cu -gt $wt ] ; then
    echo "WARNING - *$clustername is $percentfree % free* |
clustersize=$clustertotal available=$clusteravail warning=$wt critical=$ct
provisioned=$totalused nodes=$clustermodules"
    exit $STATE_WARNING

else
    echo "problem - No data received from host"
    exit $STATE_UNKNOWN
fi
```



check_lhvolstats

```
#!/bin/sh
#

STATE_OK=$(expr 0)
STATE_UNKNOWN=$(expr 3)

volumecount=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C
$2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusVolumeCount.0|cut -d" " -f4)
volumecount=$(echo "$volumecount+1" |bc)

volinstance=$(/usr/local/groundwork/common/bin/snmpbulkwalk -c $2 -v2c $1
LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusVolumeName | grep -E "$3" | cut -
d"." -f2 | awk '{ print $1 }')

if [ "$volinstance" = "" ] ; then
echo "problem - volume '$3' is not a volume known to the management group."
exit $STATE_UNKNOWN
fi

checkreadstat=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -
C $2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusVolumeStats$4Read.$volinstance|cut -d" " -f4|cut -d'"'"' -f2)
checkwritestat=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1
-C $2 -o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusVolumeStats$4Write.$volinstance|cut -d" " -f4|cut -d'"'"' -f2)

echo "OK - $3 $checkreadstat(R) $checkwritestat(W) | $4Read=$checkreadstat
$4Write=$checkwritestat"
exit $STATE_OK
```

check_lhvolume

```
#!/bin/sh
#
start_time=$(date +%s)
STATE_OK=$(expr 0)
STATE_WARNING=$(expr 1)
STATE_CRITICAL=$(expr 2)
STATE_UNKNOWN=$(expr 3)
volname=$(echo "$3")

volinstance=$(/usr/local/groundwork/common/bin/snmpbulkwalk -c $2 -v2c $1
LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusVolumeName | grep -E "$3" | cut -
d"." -f2 | awk '{ print $1 }')

if [ "$volinstance" = "" ] ; then
echo "problem - volume '$3' is not a volume known to the management group."
exit $STATE_UNKNOWN
fi

volspace=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C $2
-o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusVolumeProvisionedSpace.$volinstance|cut -d" " -f4)

allocated=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C $2
-o LEFTHAND-NETWORKS-NSM-CLUSTERING-MIB::clusVolumeSize.$volinstance|cut -d"
" -f4)

snapcount=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C $2
-o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusVolumeSnapshotCount.$volinstance|cut -d" " -f4)

repllevel=$(/usr/local/groundwork/nagios/libexec/check_snmp -P 2c -H $1 -C $2
-o LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusVolumeReplicaCount.$volinstance|cut -d" " -f4)
repllevel=$(echo "$repllevel / 1" | bc)
#echo "Got a rep level of $repllevel"

if [ $snapcount -ge 1 ] ; then
#echo "WE GOT $snapcount SNAPSHOTS"
varsnap=0
totalspace=$volspace

snaptemp=$(/usr/local/groundwork/common/bin/snmpbulkwalk -c $2 -v2c $1
LEFTHAND-NETWORKS-NSM-CLUSTERING-
MIB::clusVolumeSnapshotProvisionedSpace.$volinstance | awk '{ print $4 }')

snapspace=$(echo "$snaptemp" | awk '{ for (i = 1; i <= NF; i++) s = s+$i };
END { print s+0 }')
lastsnap=$(echo "$snaptemp" | awk '{ field = $NF }; END { print field }')

totalspace=$(expr $totalspace + $snapspace)
volonly=$(echo "$lastsnap" | bc)
```



```
snaponly=$(expr $totalspace - $volonly)

else
volonly=$(echo "$volospace /1" |bc)
snaponly=$(echo "0" |bc)
totalspace=$(echo "$volospace /1" |bc)
fi

volonly=$(echo "$volonly / $repllevel" | bc)
snaponly=$(echo "$repllevel * $snaponly" |bc)
totalspace=$(echo "$repllevel * $totalspace" |bc)
totalspace=$(echo "$totalspace / $repllevel" |bc)

mbospace=$(echo "$totalspace / 1024" |bc)
gigspace=$(echo "$mbospace / 1024" |bc)
gigspace=$(echo "$gigspace / $repllevel" |bc)

volonly=$(echo "$volonly / 1024" |bc)
volonly=$(echo "$volonly / 1024" |bc)

if [ $snaponly -gt 0 ] ; then
snaponly=$(echo "$snaponly / 1024" |bc)
snaponly=$(echo "$snaponly / 1024" |bc)
snaponly=$(echo "$snaponly/ $repllevel" |bc)
fi

finish_time=$(date +%s)

if [ "$allocated" = 0 ] ; then
perct="N/A"
gigspace=$(echo "$gigspace * $repllevel" |bc)
volonly=$(echo "$volonly * $repllevel" |bc)
echo "OK - (remote snapshot) $gigspace GB | $volname=$volonly;0;0;
snapshots=$snaponly total=$gigspace snapcount=$snapcount"
else
warning=$(echo "scale=2; $4 * .01" | bc)
critical=$(echo "scale=2; $5 * .01" |bc)

warning=$(echo " (($warning * $allocated)/1024)/1024" |bc)
critical=$(echo " (($critical * $allocated)/1024)/1024" |bc)
allocated=$(echo "$allocated / 1024" |bc)
checksize=$(echo "$volonly * 1024" |bc)

perct=$(echo "scale=2; ($volonly*1024) / $allocated" |bc)
perct=$(echo "($perct * 100)/1" |bc)

if [ $volonly -ge $critical ] ; then
```



Groundwork monitoring for HP Lefthand P4000 SANs Author: Paul Drangeid | <http://www.tdonline.com>

```
    echo "CRITICAL - *$gigspace GB ($perct% used)* |
$volname=$volonly;$warning;$critical; snapshots=$snaonly total=$gigspace
snapcount=$snapcount"
    exit $STATE_CRITICAL

    elif [ $volonly -ge $warning ] ; then
        echo "WARNING - *$gigspace GB ($perct% used)* |
$volname=$volonly;$warning;$critical; snapshots=$snaonly total=$gigspace
snapcount=$snapcount"
        exit $STATE_WARNING

    elif [ $volonly -lt $warning ] ; then
        echo "OK - $gigspace GB ($perct% used) |
$volname=$volonly;$warning;$critical; snapshots=$snaonly total=$gigspace
snapcount=$snapcount"
        exit $STATE_OK

    else
        echo "problem - No data received from host"
        exit $STATE_UNKNOWN
    fi

fi
```

Updates and additions to this document series will be posted here:

<http://www.tdonline.com/training/lefthand/>