Purpose

This document describes how to use a RAM disk to boost performance on a Nagios XI server.

Target Audience

This document is intended for use by Nagios XI Administrators who need a boost in I/O performance.

Summary

Although having enough CPU power on a Nagios server is important, the biggest hardware limitation to a Nagios system is disk I\O. A large Nagios installation creates an enormous amount of disk activity, and if the hard disk can't keep up with the constant traffic flow that needs to happen, even a large number of CPU's are simply going to wait for the disk in order to write new information to the disk. This can cause check latencies to soar even though the CPU usage appears within a safe range. Some users have solved this by using creative file mounts on separate partitions, or purchasing extremely fast disks for their servers.

The information below will cover how to free up an enormous amount of disk activity, particularly on large systems by simply adding RAM disks on the local filesystem.

The following document will cover the topics below:

- Key Nagios Files That Affect Disk I\O
- Automatic RAM Disk Installation
- Manual RAM Disk Installation
    - Create Directories
    - Editing The Configuration Files
    - Update Performance Data Commands
    - Restart Services

Key Files That Affect Disk I\O

Lets look at some of the key files that create disk activity on a Nagios install. The examples below match a typical source install of Nagios Core, and any default Nagios XI install.

- `/usr/local/nagios/var/status.dat` – This is the bread and butter file of all of the "live" information on the monitoring environment. This file gets updated every 10-20 seconds (as specified in nagios.cfg) with all current status information.
- `/usr/local/nagios/var/objects.cache` – This file stores all of the object configuration data for Nagios. This file only gets updated upon a restart of the Nagios process.
- `/usr/local/nagios/var/host-perfdata & service-perfdata` – These files may be in a different location for Core install, but these files function as an intermediary file for PNP's NPCD daemon that processes performance data results. These files get updated about every 10-15 seconds.
- `/usr/local/nagios/var/spool` – This directory tree acts as a dropbox for all incoming check results. The disk activity in this directory is almost constant, since both Nagios and NPCD are continually creating result files, and then reaping the results every X number of seconds.
- `/usr/local/nagios/var/*.log` – These log files were worth mentioning. Logging takes both CPU and disk activity, so minimize all unnecessary logging if you need to scale, particularly with performance data processing. Logging can be minimized by editing: /usr/local/nagios/etc/pnp/npcd.cfg and `/usr/local/nagios/etc/pnp/process_perfdata.cfg`, and setting the log level to 0.

Automatic RAM Disk Installation

The easiest way to install RAM disk on the Nagios XI is use our `install_ramdisk.sh` script. Establish a terminal session to your Nagios XI server as the root user and execute the following commands:

```
cd /tmp
wget https://assets.nagios.com/downloads/nagiosxi/scripts/install_ramdisk.sh
chmod +x install_ramdisk.sh
./install_ramdisk.sh
```

The script will prompt you for the size of the RAM Disk. Refer to the table on the following page for information on this.

The `install_ramdisk.sh` script works for the following operating systems (OS):

- Red Hat Enterprise Linux (RHEL), CentOS, CentOS Stream, Oracle Linux
  - CentOS 7.x
  - CentOS Stream 8.x, 9.x
  - RHEL 7, 8, 9
- Debian
  - 10.x
  - 11.x
- Ubuntu
  - All active Long Term Support (TLS), these are all the even numbered versions (20, 22)

If you have an older distribution, you will need to follow the manual install instructions below. The same is valid in cases when you have a "non-standard" Nagios XI instance (custom paths, custom config locations, etc.) or you want to set up the RAM disk in a "non-default" location.

Manual RAM Disk Installation

These steps should only be followed if using the `install_ramdisk.sh` script is not possible.

In these steps you will be required to edit files. This documentation will use the vi text editor. When using the vi editor:

- To make changes press **`i`** on the keyboard first to enter insert mode
- Press **`Esc`** to exit insert mode
- When you have finished, save the changes in vi by typing **`:wq`** and press Enter

You need to determine the size of the RAM disk that you will be using. You can determine the recommended size of the RAM disk based on the total number and hosts and services that you are monitoring. You will be need to define this value in the steps below.

Total Number of Hosts + Services	RAM Disk Size
Less than 1000	100 MB
1000 to 5000	300 MB
More than 5000	500 MB

Tip: It is always a good idea to use a higher number than the "recommended size" in order to give some leeway and allow for growth. Keep in mind that this will only make an improvement if you have enough available memory, otherwise this will mount the RAM disk and use swap memory for excess RAM allocated.

SYSTEMD

Create RAM Disk Directory

Execute these commands to create the base directory:

```
mkdir /var/nagiosramdisk/
chown -R nagios.nagios /var/nagiosramdisk
```

Create SYSTEMD Service

The following steps will create a `ramdisk.service` file, the location of this file differs depending on your OS. Change into the correct directory as per:

- RHEL, CentOS, Oracle Linux
  - `cd /usr/lib/systemd/system/`
- Ubuntu, Debian

-
  ```
  cd /lib/systemd/system/
  ```

Create the `ramdisk.service` file and set permissions:

```
touch ramdisk.service
chown nagios:nagios ramdisk.service
chmod 775 ramdisk.service
```

Open the `ramdisk.service` file in vi:

```
vi ramdisk.service
```

Add lines on the following page:

```
[Unit]
Description=Ramdisk
Requires=local-fs.target
After=local-fs.target
Before=nagios.service
[Service]
Type=simple
RemainAfterExit=yes
Restart=always
ExecStartPre=/usr/bin/mkdir -p -m 775 /var/nagiosramdisk
ExecStartPre=/usr/bin/mount -t tmpfs -o size=100m tmpfs /var/nagiosramdisk
ExecStartPre=/usr/bin/mkdir -p -m 775 /var/nagiosramdisk/tmp
/var/nagiosramdisk/spool /var/nagiosramdisk/spool/checkresults
/var/nagiosramdisk/spool/xidpe /var/nagiosramdisk/spool/perfdata
ExecStart=/usr/bin/chown -R nagios:nagios /var/nagiosramdisk
```

```
[Install]
WantedBy=multi-user.target
```

Note:

- The third instance of the **`ExecStartPre`** line ending with **`perfdata`** (line 7) is all on one line

When you have finished, save and close the file.

You then need to issue this command for the OS to detect the new service:

```
systemctl daemon-reload
systemctl enable ramdisk.service
```

You can now proceed to the Editing Configuration Files section of this document.

Editing Configuration Files

/usr/local/nagios/etc/nagios.cfg

Open up the `/usr/local/nagios/etc/nagios.cfg` file in vi:

```
vi /usr/local/nagios/etc/nagios.cfg
```

Update the following lines:

```
service_perfdata_file=/var/nagiosramdisk/service-perfdata
host_perfdata_file=/var/nagiosramdisk/host-perfdata
check_result_path=/var/nagiosramdisk/spool/checkresults
object_cache_file=/var/nagiosramdisk/objects.cache
status_file=/var/nagiosramdisk/status.dat
temp_path=/var/nagiosramdisk/tmp
```

When you have finished, save and close the file.

/usr/local/nrdp/server/config.inc.php

Open `/usr/local/nrdp/server/config.inc.php` in vi:

```
vi /usr/local/nrdp/server/config.inc.php
```

Update the following line:

```
cfg["check_results_dir"]="/var/nagiosramdisk/spool/checkresults";
```

When you have finished, save and close the file.

/usr/local/nagiosxi/html/config.inc.php

Open `/usr/local/nagiosxi/html/config.inc.php` in vi:

```
vi /usr/local/nagiosxi/html/config.inc.php
```

Update the following lines:

```
$cfg['xidpe_dir'] = '/var/nagiosramdisk/spool/xidpe/';
$cfg['perfdata_spool'] = '/var/nagiosramdisk/spool/perfdata/';
```

When you have finished, save and close the file.

/usr/local/nagios/etc/pnp/npcd.cfg

Open up `/usr/local/nagios/etc/pnp/npcd.cfg` in vi:

```
vi /usr/local/nagios/etc/pnp/npcd.cfg
```

Update the following line:

```
perfdata_spool_dir = /var/nagiosramdisk/spool/perfdata/
```

When you have finished, save and close the file.

Note: If you have Nagios XI version 5.6 or older, you will need to edit a file in the nagiosmobile directory. If you are using Nagios XI version 5.7 or higher, you will skip this step.

/usr/local/nagiosmobile/include.inc.php

Open up `/usr/local/nagiosmobile/include.inc.php` in vi:

```
vi /usr/local/nagiosmobile/include.inc.php
```

Update the following lines:

```
$STATUS_FILE = "/var/nagiosramdisk/status.dat";
$OBJECTS_FILE = "/var/nagiosramdisk/objects.cache";
```

When you have finished, save and close the file.

Update Performance Data Commands

Login to Nagios XI and navigate to **Configure** > **Core Config Manager** > **Commands** > **>_ Commands.**

In the search field type `file-bulk` and press **Enter**.

The screen will update with the two commands `process-host-perfdata-file-bulk` and `process-service-perfdata-file-bulk`.

Tip: Make a copy (backup) of the existing `process-host-perfdata-file-bulk` and `process-service-perfdata-file-bulk` commands, prior to modifying them. Leave the "old" commands (copies) inactive. This can be helpful if you need to go back.

Both of these commands need to be updated with the new directory locations as follows:

```
process-host-perfdata-file-bulk
/bin/mv /var/nagiosramdisk/host-perfdata /var/nagiosramdisk/spool/xidpe/$TIMET$.perfdata.host
process-service-perfdata-file-bulk
/bin/mv /var/nagiosramdisk/service-perfdata /var/nagiosramdisk/spool/xidpe/$TIMET$.perfdata.service
```

After making these changes click the **Apply Configuration** button.

It is possible that the Apply Configuration will fail the first time. Simply click the **Try Again** button and it will succeed the second time.

Restart Services

After making all of these changes, execute the following commands in your terminal session:

SYSTEMD

```
    systemctl restart nagios.service
    systemctl restart npcd.service
```

- RHEL, CentOS, Oracle Linux

  -
    ```
    systemctl restart httpd.service
    ```
- Ubuntu, Debian

  -
    ```
    systemctl restart apache2.service
    ```

Conclusion

Making use of a RAM disk can provide huge performance improvements on larger systems, or any system where check latencies are greater than 2 seconds. We recommend testing these settings in a non-production environment before implementing into production.

Finishing Up

This completes the documentation on how to use a RAM disk to boost performance on a Nagios XI server. If you have additional questions or other support related questions, please visit us at our Nagios Support Forums:

https://support.nagios.com/forum

The Nagios Support Knowledgebase is also a great support resource:

https://support.nagios.com/kb