

The Industry Standard in IT Infrastructure Monitoring

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 simple adding RAM disks on the local filesystem. The following document will cover the topics below:

- **Key Nagios Files That Affect Disk I/O**
- **Creating the RAM Disk**
- **Editing the nagios.cfg to use the RAM Disk**
- **Making the RAM Disk Permanent**
- **Moving Check Results Onto the RAM Disk**
- **Moving Performance Data Onto the RAM Disk**

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 this file functions as an intermediary file for PNP's NPCD daemon that processes performance data results. This file gets 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.

Creating the Disk

First thing that should be done is the creation of the directory that we'll be mounting the RAM disk to. For the purposes of this write-up, we will simply be using `/var/nagiosramdisk`. Execute the following from the command-line.

```
mkdir /var/nagiosramdisk/
```

Now we have to actually mount a RAM disk to that location. This is the point where we need to determine the size of the RAM disk that will be set aside. This can be determined by taking a look at your `status.dat` and `objects.cache` which default to being in the `/usr/local/nagios/var/` directory. On my test machine, these two files added up to be around 13MB, so I will set up the RAM disk to be 50MB to give some leeway and allow for growth. 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.

```
mount -t tmpfs none /var/nagiosramdisk -o size=50m
```

Now we have to allow nagios to access the whole directory:

```
chown nagios /var/nagiosramdisk
```

Editing the Nagios.cfg to Utilize the Disk

Ok, so now that that's mounted we actually have to tell Nagios that it needs to reference those locations. Open up the `nagios.cfg` file which is usually located at `/usr/local/nagios/etc/nagios.cfg`. There are two lines that need updating:

```
objects_cache_file=/var/nagiosramdisk/objects.cache
status_file=/var/nagiosramdisk/status.dat
```

Restart the nagios service:

```
service nagios restart
```

Now check to see if everything went well by seeing if `status.dat` and `objects.cache` were created in your ramdisk directory:

```
ls /var/nagiosramdisk/
```

Should show both of these files. If it does not, review the above steps again and make sure the nagios user has RW permissions on the ramdisk.

Making the RAM Disk Permanent

Now if you wish to make this permanent you'll need to edit the `fstab` so that this partition automounts. Open up the `fstab` that's in the `/etc/` directory usually under `/etc/fstab` and add the following line:

```
tmpfs          /var/nagiosramdisk tmpfs size=50m    0    0
```

The tabs don't matter too much, there just has to be whitespace, but keeping it orderly is a good idea. Now save it and reboot (if you want) to see if it automounts. Use the command `df -i` to see if it automounted to `tmpfs`.

Moving Check Results to the RAM Disk

In order to move check results to a RAM disk, we first need to create the necessary directory structure:

```
mkdir /var/nagiosramdisk/spool
mkdir /var/nagiosramdisk/spool/checkresults
chown -R nagios.nagios /var/nagiosramdisk
chmod -R 775 /var/nagiosramdisk
```

Now that the necessary directories are created on the RAM disk, the new location will need to be specified in the `/usr/local/nagios/etc/nagios.cfg` file:

```
check_result_path=/var/nagiosramdisk/spool/checkresults
```

Moving Performance Data to the RAM Disk

Moving the performance data spool to the RAM disk takes a little more modification to the system, and should be set up in a test environment before implementing in a production system. These modifications also require Nagios XI 2011R2.3 or later. First, we'll need to create the necessary directories for the performance data spool.

```
mkdir /var/nagiosramdisk/spool/perfdata
mkdir /var/nagiosramdisk/spool/xidpe
```

Update or add the following lines in `/usr/local/nagiosxi/html/config.inc.php`.

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

Additionally, the following files will need to be modified:

- **`/usr/local/nagios/etc/nagios.cfg`:**

```
host_perfdata_file=/var/nagiosramdisk/host-perfdata
service_perfdata_file=/var/nagiosramdisk/service-perfdata
```

- **`/usr/local/nagios/etc/pnp/npcd.cfg`:**

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

Additionally, the following command definitions will need to be updated in the Nagios XI->Core Config Manager->Commands.

```
command_name process-host-perfdata-file-bulk
command_line /bin/mv /var/nagiosramdisk/host-perfdata /var/nagiosramdisk/spool/xidpe/${TIMET$}.perfdata.host

command_name process-host-perfdata-file-pnp-bulk
command_line /bin/mv /var/nagiosramdisk/host-perfdata /var/nagiosramdisk/spool/perfdata/host-perfdata.${TIMET$}

command_name process-service-perfdata-file-bulk
command_line /bin/mv /var/nagiosramdisk/service-perfdata /var/nagiosramdisk/spool/xidpe/${TIMET$}.perfdata.service

command_name process-service-perfdata-file-pnp-bulk
command_line /bin/mv /var/nagiosramdisk/service-perfdata /var/nagiosramdisk/spool/perfdata/service-perfdata.${TIMET$}
```

Conclusion

Making use of local RAM disks can cause 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. If you experience any problems during setup, feel free to submit your questions to the the Nagios online support forum: <http://support.nagios.com/forum>.