

NMIS - Optimize nmis8/var For Speed - RAM Disk

- [Overview](#)
- [Procedure](#)
 - [Step #1 - Determine Space Required](#)
 - [Step #2 - Create the RAM disk](#)
 - [Step #3 - Disable NMIS](#)
 - [Step #4 - Point nmis8/var at the RAM Disk](#)
 - [Step #5 - Re-enable NMIS](#)
 - [Step #6 - Verify](#)

Overview

Some customers may benefit by moving files that NMIS reads/writes to very often to a RAM Disk. We have seen this improve performance on pollers that have shared storage and 2,000+ nodes provisioned. The concept is that NMIS will be able to read/write to RAM faster than the shared storage. Most of the files that NMIS touches very often are found in the nmis8/var directory.



Any files written to RAM disk will not survive a reboot. After the server reboots the RAM disk will be empty.

Please review this Wikipedia article that addresses tmpfs:

<https://en.wikipedia.org/wiki/Tmpfs>



This is an advanced configuration technique for large scale deployments with experienced sysadmin. The benefits for small scale deployments would be negligible.

Procedure

Step #1 - Determine Space Required

There are two questions that quickly come to mind:

- How big should the RAM disk be?
- How much RAM is available?

Check how much space nmis8/var is currently using; minus the nmis8/var/events directory. We will keep the nmis8/var/events directory on a hard drive because it contains stateful event history; we don't want to lose these if the server is rebooted.

```
[root@TEST var]# pwd
/usr/local/nmis8/var

[root@TEST var]# du -hs
2.6G    .

[root@TEST events]# pwd
/usr/local/nmis8/var/events

[root@TEST events]# du -hs
1.7G    .
```

In this example we notice that the nmis8/var directory minus the nmis8/var/events directory is approximately 900MB. It appears that a RAM disk of 1GB would be sufficient.

How much RAM is available?

```
[root@TEST events]# free
              total        used          free      shared    buffers     cached
Mem:      37016616   25990916   11025700         1796     861860   15379752
-/+ buffers/cache:    9749304   27267312
Swap:      4169724     558624     3611100
```

In the example above there appears to be 11GB of RAM available.



The NMIS Self Test settings may need to be adjusted if the nmis8/var directory is running at a high rate of usage. Look for the following attributes in nmis8/conf/Config.nmis:

- 'selftest_min_diskfree_percent' => 10
- 'selftest_min_diskfree_mb' => 25

Step #2 - Create the RAM disk

The RAM disk mount point will need to be created. In this example we are using /media/ramdisk; be sure to mind the permissions.

The following commands will create the RAM disk, keep events on the hard drive, and ensure permissions are not lost. These commands should also need to be added to a script that runs at boot time. On Centos 6 /etc/rc.d/rc.local works well.

```
mount -t tmpfs -o size=1000M tmpfs /media/ramdisk/  
mkdir -p /media/ramdisk/nmis8/var  
ln -s /data/nmis8/var/events /media/ramdisk/nmis8/var  
/usr/local/nmis8/admin/fixperms.pl
```



When creating the RAM disk the size parameter may be a number expressed with the suffix k, m or g (kilo, mega, and giga). It may also be expressed with the % sign. For example if you'd like to allow the ramdisk to consume up to 25 percent of the available RAM the following technique could be used.

```
mount -t tmpfs -o size=25% tmpfs /media/ramdisk
```

Keep in mind that the ramdisk will not immediately consume the amount of RAM specified. This is the most it will be allowed to consume should it need to. Also keep in mind that if the RAM is oversubscribed the ramdisk could be pushed into swap. If the ramdisk finds itself in swap space it defeats the purpose of this expedition, so do try to avoid this situation.



If you find yourself in a situation where you need to resize the ramdisk on the fly the following method may be useful.

```
# mount -o remount,size=4096M /media/ramdisk
```

This does not require a reboot and will preserve the files that reside on the ramdisk.

Use the df command to confirm that the ramdisk was created.

```
[root@TEST]# df -h  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/vg_nmis64-lv_root  
                16G   11G  4.1G   73% /  
tmpfs           5.9G    0  5.9G    0% /dev/shm  
/dev/sda1       477M    75M  377M   17% /boot  
/dev/mapper/vg_nmis64_data-lv_data  
                246G   95G  139G   41% /data  
/dev/mapper/vg_nmis64-lv_var  
                20G   10G   8.3G   55% /var  
tmpfs           1000M   30M  971M    3% /media/ramdisk
```

Step #3 - Disable NMIS

We need to disable NMIS so that it will not attempt to modify the files as we point nmis8/var to the new RAM disk. This is easily accomplished by placing a file called 'NMIS_IS_LOCKED' in the nmis8/conf directory as in the following example.

```
[root@TEST]# touch /usr/local/nmis8/conf/NMIS_IS_LOCKED
```

Once the lock file is in place check the server for any nmis.pl processes.

```
[root@TEST]# ps -ef | grep nmis.pl
```

When none are found continue

Step #4 - Point nmis8/var at the RAM Disk

Since we are planning to put a sym link in /usr/local/nmis8/ for /var pointing at the new RAM disk; verify that in /usr/local/nmis8/conf/Config.nmis nmis8/var points to /usr/local/nmis8/var. Here's a sample of the directories configuration in that file.

/usr/local/nmis8/conf/Config.nmis

```
'directories' => {
  '<menu_base>' => '<nmis_base>/menu',
  '<nmis_admin>' => '<nmis_base>/admin',
  '<nmis_backups>' => '<nmis_data>/backups',
  '<nmis_base>' => '/usr/local/nmis8',
  '<nmis_bin>' => '<nmis_base>/bin',
  '<nmis_cgi>' => '<nmis_base>/cgi-bin',
  '<nmis_conf>' => '<nmis_base>/conf',
  '<nmis_data>' => '<nmis_base>',
  '<nmis_logs>' => '<nmis_base>/logs',
  '<nmis_menu>' => '<nmis_base>/menu',
  '<nmis_models>' => '<nmis_base>/models',
  '<nmis_var>' => '<nmis_data>/var',
  'config_logs' => '<nmis_logs>/config',
  'database_root' => '<nmis_data>/database',
  'json_logs' => '<nmis_logs>/json',
  'log_root' => '<nmis_logs>',
  'mib_root' => '<nmis_base>/mibs',
  'plugin_root' => '<nmis_conf>/plugins',
  'report_root' => '<nmis_base>/htdocs/reports',
  'script_root' => '<nmis_conf>/scripts',
  'web_root' => '<nmis_base>/htdocs'
},
```

Create the sym link in /usr/local/nmis8/ sending /var to the RAM disk.

```
ln -s /media/ramdisk/nmis8/var/ /usr/local/nmis8
```

Step #5 - Re-enable NMIS

Remove the previously created NMIS lock file.

```
[root@TEST]# rm /usr/local/nmis8/conf/NMIS_IS_LOCKED
```

Step #6 - Verify

- move to the /usr/local/nmis8/var directory and execute a 'df -h .' command. This command should reveal that we are indeed on the ramdisk.

```
[root@TEST ~]# cd /usr/local/nmis8/var
[root@TEST var]# df -h .
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           1000M    30M   971M   3% /media/ramdisk
[root@TEST var]#
```

- Take a look in the /usr/local/nmis8/var directory and confirm that node.json and view.json files have been created.
- Verify that the NMIS GUI is functioning normally.
- Review the nmis8/logs/nmis.log file
- Check the root user mailbox

