Scaling NMIS Polling

Opmantek gets many questions on how to scale NMIS8 for very large networks. There are many factors impacting polling performance and the upper limits of polling is really only limited by the number of processor cores, available memory and disk IO performance. We have customers managing 10's of 1000's of nodes using NMIS.

- Related Articles

Related Articles

- Scaling NMIS polling - how NMIS handles long running processes
- Configuration Options for Server Performance Tuning

Server Specifications

The following server specifications are guidelines for NMIS installations. Storage performance is one of the greatest factors for scaling polling, you should consider using highly performant storage like SSD.
The ideal way to determine specifications for your server is to baseline some nodes on the server and determine what resources are required.

	Small	Medium	Large	Massive
OS Storage	20GB	20GB	20GB	20GB
Data Storage	40GB	60GB	140GB	280GB
Memory	4GB	8GB	16GB	32GB+
CPU	2 x vCPU	2 to 4 x vCPU	8 x vCPU	16+ vCPU
Device Count	< 500 devices	< 1500 devices	< 2500 devices	A very large number of devices
Element Count	2000 elements	8000 elements	14000 elements	A very large number of elements

Elements are additional data being collected, an interface is an element, a CBQoS class is an element.

An element requires additional SNMP polling to collect the values and then storage on the disk to save the data.

Baseline Performance

Once you know device counts and have an idea of the virtual server (yes you can use physical servers) specifications, it is a good idea to get an idea of baseline performance.

The baseline is to establish how NMIS and the virtual server are performing, add ~50 nodes and see how it is performing. We have a customer polling ~200 nodes and the average poll cycle is about 20 seconds. So if you are able to poll 50 nodes in less than 20 seconds your performance should be OK, if it is longer than this, then you might need to look at CPU and Disk performance. This will also give you an idea of memory footprint.

Adding Nodes to NMIS

If you try to add 1000's of nodes to NMIS at once it is going to take a while to process the nodes for the first time. You two main choices, add nodes in smaller batches, or stop polling while adding large numbers of nodes.

If you want to stop the polling and then run an update cycle, then a collect cycle manually, then start the polling.

To stop polling modify the configuration option global_collect to false or stopping polling in the crontab, comment out this line:

```
#*/5 * * * * /usr/local/nmis8/bin/nmis.pl type=collect mthread=true maxthreads=8
```

Then add nodes, all of them at once or in batches. Restart fpingd so that it reloads all the nodes you added.

```
/usr/local/nmis8/bin/fpingd.pl restart=true
```

Then run an update manually with nohup, if you have 12GB of memory you can give it lots of threads, probably 20 should do it, but watch your memory usage you can probably get to 30 threads.

```
cd ~
nohup /usr/local/nmis8/bin/nmis.pl type=update mthread=true maxthreads=20&
```

This will take a while to run the first time, when it finishes, run a collect cycle the same way

```
nohup /usr/local/nmis8/bin/nmis.pl type=collect mthread=true maxthreads=20&
```

Now all the big disk activity is done and you should be able to start NMIS polling by letting the poller go again.

```
*/5 * * * * /usr/local/nmis8/bin/nmis.pl type=collect mthread=true maxthreads=<MAX THREADS BASED ON YOU
BASELINE>
```

Configuration Considerations

You can also control the way NMIS does its thing by moving the summary and thresholding to cron, I would suggest this as a good practice for larger installations.
In CRON:

```
*/2 * * * * /usr/local/nmis8/bin/nmis.pl type=summary
4-59/5 * * * * /usr/local/nmis8/bin/nmis.pl type=threshold
```

In Config.nmis:

```
'threshold_poll_cycle' => 'false',
'nmis_summary_poll_cycle' => 'false',
'disable_interfaces_summary' => 'true',
```

The other BIG consideration is what is your polling policy, the more interfaces you collect on, the more disk, cpu and memory you will consume, just collecting more data may not help you operationally, collect the right data, which is how NMIS has been configured.

If you hare having problems scaling your NMIS installation, you could contact Opmantek for assistance.

Using JSON for NMIS Database

To optimise how NMIS files are saved, you can use the JSON database, this will require NMIS 8.4.8g or greater. The following needs to be run on every Primary and poller server in an NMIS cluster and this should be co-ordinated to run very close together.

```
/usr/local/nmis8/admin/convert_nmis_db.pl
```

This script will stop NMIS polling, convert the database files, update the NMIS configuration to use the new database format, then start the polling again.