

opCharts 3 Performance Tuning



Version

This relates to opCharts version 3

- [Introduction](#)
- [DB Caching](#)
 - [Node list Sorting Health](#)
 - [Primary Poller Caveat](#)
- [Subnet Caching](#)
 - [Background Subnet Caching](#)
- [Multiple web users](#)

Introduction

DB Caching

As of opCharts version 3.2.6 Node list performance has been improved significantly and will be noticeable on systems with higher node counts, particularly on the node list page and on and scheduled outage page.

There are two basic modes that can be run: db off and db on. db off has cache running in memory. db on uses the db for caching. When db is set to on, the nodesum files are cached into the db and filters are run against the db instead of doing the filters manually in memory. NMISx db caching is now the faster option. By default this config item is turned off, to enable this edit your opCommon.nmis file located at /usr/local/omk/conf/opCommon.nmis and set nmisx_db_cache_enable to 1. For systems with a low number of nodes, using the db may not show any performance improvement

```
'nmisx_db_cache_enable' => 1
```

There are two more configuration items that control how the cache works:

```
'nmisx_cache_min_age_before_refresh' => 60
```

This setting tells the app how long to wait before checking to see if the cache needs refreshing. In order to know if the cache needs refreshing we check the time the nodesum files are written and compare that against the time the cache was built. This is a simple optimisation. Since we access the cache quite often it gets a little expensive to check it several times a second, this prevents those checks for at least X seconds. The idea is that NMIS only writes the nodesum files at most once a minute so don't bother looking more than that often.

```
'nmisx_cache_time' => 900
```

This setting controls how long the data will stay in the cache before it's removed. This is only for db mode, in memory mode this will have no affect.

Node list Sorting Health

If you have configured your node list to show the 8h health metric and using DB caching by default you cannot sort by this metric. This is because the node list is an aggregation of node configuration, node summary and node summary 8h.

To enable sorting of 8h health please edit opCommon.nmis and change:

```
'nmisx_db_cache_node_health_sorting' => 1
```

Primary Poller Caveat

When using DB caching on a Primary server, 8h and 16h summary metrics for nodes not polled by the Primary are unable to be cached. In the node list you will find the 8h node summary metric missing.

If this data is important we recommend disabling db caching for your environment.

Subnet Caching

In environments with high node counts subnet generation can be expensive to process. To make sure the application remains fast we introduced a cache for this data and by default it has a 5min expiry.

Generation of this cache takes part while viewing a subnet map and the cache mtime is greater than the `nmisx_subnet_cache_time`. In instances with large node counts this can slow down the users web request each time the cache needs to be regenerated, if needed you can increase the cache time and lessen the impact on users.

```
'nmisx_subnet_cache_time' => 300
```

Background Subnet Caching

Recommended for larger node counts.

To perform subnet caching in the background you can use a cron job to call the opCharts CLI and perform subnet caching manually, the period in which this is run is balance between stale data and processing time taken to calculate subnets. We would recommend to run the cron job every hour. If you have implemented this method you should change `nmisx_subnet_cache_time` to 86400 which will stop the web thread from calculating subnets . An example cron job provided

/cron.d/opcharts-example

```
59 * * * * /usr/local/omk/bin/opcharts-cli.exe act=refresh-subnet-cache >/dev/null 2>&1
```

Multiple web users

If you have multiple users of the platform you can increase the worker count, default is 10

```
'omkd_workers' => 10
```