Scaling NMIS polling - how NMIS handles long running processes

Scaling NMIS polling is something many large enterprise and service provider customers have to deal with; One of the biggest issues with polling devices is when devices themselves start responding too slowly, either from network latency or congestion or because the devices themselves are overloaded. Depending on your configuration, a single slow or unresponsive device can substantially impede NMIS' progress.

NMIS has configurable features to ensure that the polling operations keep up and complete timely while not overloading the polling server.

TABLE OF CONTENTS

RELATED ARTICLES

- Scaling NMIS Polling
- Configuration Options for Server Performance Tuning

NMIS Poll Cycle Overview

NMIS performs two main operations periodically, collect and update. An update is really a node discovery, and determines what a node can do and how it should be managed. Updates are performed infrequently, usually just once a day. The collect operation on the other hand is the 'work horse' of NMIS where the main SNMP and other protocol polling activity happens. Collects are performed every 5 minutes by default.

Depending on your configuration, the thresholding and summary processes will run with the main collect operation or can be run separately; details about that separation can be found in the article Scaling NMIS Polling.

Related Features in NMIS

To prevent individual nodes from holding up NMIS, a system to manage long running processes was added. This feature was summarised in the release notes for 8.5.4G as follows:

NMIS now handles critical sections and long-running NMIS processes much better than before:

- New config option `max_child_runtime` can be used to limit collect/update job runtime.
- If NMIS jobs runs over-time, a warning event is generated unless config option `disable_nmis_process_events` is set to false.
- Collect jobs can now optionally run longer than a single collect cycle, if the command line option `ignore_running=true` is given - in which case NMIS will just warn about old NMIS processes that are still running, not kill them as it does by default.

max_child_runtime configuration option

By default this is set to undef, and the feature is disabled if the value is undef or 0 (zero).

When enabled (by setting the value to the maximum allowed runtime in seconds), this feature sets up a process timeout (alarm) for each NMIS child process (in the documentation also often called a "thread"), which terminates the process if it remains stuck or running for too long. This serves as a safeguard for nodes that are totally unresponsive or much too slow in responding, and which would otherwise keep the collect operation from completing.

NMIS Event "NMIS runtime exceeded"

By default, NMIS 8.5.4 and greater will monitor all the processes on the NMIS server and when starting a new collect cycle (polling cycle), NMIS will check if there are still processes running from the last poll cycle. If it finds any "old" processes it will politely ask them to stop (kill "TERM"), the child processes will receive this request and complete what they are doing fairly quickly and die peacefully. The event "NMIS runtime exceeded" will be generated for the NMIS server node and added to the NMIS Event Log, so as to inform you of the unexpectedly slow or delayed process.

If you do not want NMIS to terminate old processes at all, then you can add `ignore_running=true` to your command line in the cron setup, e.g. "`nmis.pl type=collect ignore_running=true...`"

If you like the feature but do not what the events to be generated, you can disable the events with the configuration option `disable_nmis_process_events=false`, which is found in Config.nmis or can be modified using the NMIS Configuration GUI in the "System -> System Configuration" menu.

Please note that the operation of the `max_child_runtime` feature is independent of `ignore_running`; that is, you can use neither, one, or both of these features at the same time.