opFlow 3 Scalability Features

Overview

Dealing with NetFlow traffic analysis commonly involves large to huge amounts of data, which poses quite a few scalability challenges. Meeting these requires certain trade-offs to be made by any NetFlow analyzer, not just opFlow. This page documents the configuration options and behaviours related to making opFlow scale well.

opFlow Daemon

Database Size Capping

opFlow has supported database size capping for a long time, and it's proven a vital feature to ensure that you don't exhaust your available disk space.

In opFlow 3, the installer guides you though the initial database configuration steps and allows the database size to be capped. We **highly recommend** that you perform this step!

The default cap sizes are set in the configuration file (`opCommon.nmis`), entries `opflow_db_flows_collection_size` (for the raw flows, default 5GB) and `opflow_db_conversations_collection_size` (for conversations, default 15GB). Once the database reaches those limits, old records are automatically removed.

If these defaults are not suitable for your environment, then you can either choose to use a disk-percentage based formula during the installation procedure, or you can leave the database uncapped, adjust the parameters and manually perform the capping using `opflow-cli.pl act=setup-db`.

If necessary, the capping can be repeated with different parameters later (but a re-capping operation may take quite a bit of time), again using `opflow-cli.pl act=setup-db`.

High-Volume Mode

By default, opFlow 3 comes with the config option `opflow_high_volume` enabled.

In this mode, `opflowd` reads pre-aggregated conversation data from the collector application `nfdump`, every `opflow_summarisation_interval` seconds (default: 2 minutes).

Important aspects of High-Volume Mode

- `nfdump` must be configured with a rotation interval that is the same as `opflow_summarisation_interval`.
  This is done by setting `DATA_ROTATE_INTERVAL=120` in the config file /etc/sysconfig/nfdump (CentOS/RedHat) or /etc/default/nfdump (Debian /Ubuntu).
- In this mode, no raw flows are stored in MongoDB, and the capping size of the `flows` collection is unimportant. Only the capping on the `conversations` collection is relevant.
- The pre-aggregation combines all flows within the respective interval and groups them by the involved endpoints, the communication protocol and application in question.
  Some granularity present in the raw flow records is sacrificed for scalability: a conversation includes a list of the port numbers involved, and *cumulative* counters for packets and bytes for the *whole* summarisation interval. Any number of network interactions between the same endpoints, using the same application and which fall into one summarisation period are lumped up into one conversation record.
- If you use a short summarisation interval, the pre-aggregation will be less efficient at combining multiple flows into conversations.
  You will also likely experience higher database loads and may hit insertion speed limits at a lower volume of incoming netflow records.
- If you use a long summarisation interval then the summarisation will be maximally efficient, but the opFlow GUI will exhibit time lag and show somewhat more outdated data.

Low-Volume Mode

If you set `opflow_high_volume` to 0 or "false", `opflowd` runs in low-volume mode. In this mode, individual "raw" flow records are initially stored in the `flows` collection. Periodically, these raw individual flow records are then summarised into conversations in the same manner as in high-volume mode.

Important aspects of Low-Volume Mode

- The summarisation interval *should* be a multiple of the `nfdump` rotation interval for optimal performance, but those parameters are not as closely tied in this mode.
- Both `flows` and `conversations` collections will be used. The size capping on `flows` must be sized to retain records for at least the most recent summarisation interval.
- Whenever `nfdump` rotates its flow collection file, opflowd picks that up and starts collection and insertion of the raw flows contained therein.
- Inserting lots of raw flows requires more database performance (and possibly fine-tuning of the `opflow_batch_insert_size` config parameter) and you will hit database limitations much earlier than in high-volume mode.
- As long as you don't run into the size capping limits on the raw `flows` collection, full data of the utmost precision remains available.
- As of version 3.0.2, the opFlow GUI does not expose the raw flows to the user.

Parallel Processing

In both low- and high-volume modes, opflowd can make use of parallel processing to cope with high flow volumes: the config item `opflowd_max_processes` (default: 4) sets the maximum number of opflowd worker processes that can be run in parallel.

opflowd will start up to that many flow consumer and summariser processes. With the default settings your opFlow installation would thus keep up with inbound flow volumes until the processing of each `nfdump` flow file takes four times the file's time period.

If opFlow detects resource exhaustion of this kind, an Operational Status record (and suitable log messages) is created to notify you of the problem. Additionally, opflowd generates statistics for every processing run which can be viewed on the Operational Status page.

Long-Term Summarisation Stages

Database capping is likely to interfere and limit long-term data availability. To address this point, opFlow 3 also supports an arbitrary number of optional longer-term summarisation stages. These reside in separate database collections and can be capped independently. This functionality is used for (re-) creating traffic overview reports retrospectively.

By default a one-hour summary stage is enabled, which furthermore collapses and combines all conversations that produced less than 1024 bytes or less than 5 packets during the respective hour. In our tests these settings have proven to provide a very high degree of compression efficiency without much loss of detail.

Both the opFlow GUI and the report generation code look for the 'best available' source of data and fall back to using summarisation stage data where required. This means that even though your main conversations may have been purged due to high incoming flow volume and size capping after just a few hours, you would still be able to access historic data reaching back to the oldest summarisation stage result (but you may have to select a longer Summarise Interval in the Advanced menu).

You can define summary stages in the configuration file, under `opflow_summary_stages`; a stage definition requires a name (allowed characters A-Z, a-z, 0-9, _ and -), and a `period` (in seconds). The summarised data will be stored in a collection named `summary_`<stagename>. You can optionally set up database capping for this collection (with the `collection_size` property, in bytes), and collapsing of unimportant conversations (with the `collapse_min_bytes` and/or `collapse_min_pkts` settings - zero or not set disables collapsing, and collapsing happens if either of the two criteria is met).

Extended Summarisation

Since opFlow 3.0 and all opFlowSP versions, you are able to configure further summarised flows in order to be able to keep more historical information. In opFlow you will notice that the installer comes with the following extra summary options:

```
        'opflow_summary_stages' => {
                'daily' => {
                'collapse_min_bytes' => 102400,
                'collapse_min_pkts' => 128,
                'collection_size' => 1073741824,
                'period' => 3600
              },
        'hourly' => {
            collapse_min_bytes => 1024,
            collapse_min_pkts => 5,
                        collection_size => 1073741824, # 1gb
                        period => 3600,
        },
                'quarterhr' => {
                        'collapse_min_bytes' => 1024,
                        'collapse_min_pkts' => 5,
                        'collection_size' => 1073741824,
                        'period' => 900
                }
        },
```

Currently opFlowSP has the a single summary option, but you can easily add more:

```
        'opflow_summary_stages' => {
                'quarterhr' => {
                        'collapse_min_bytes' => 1024,
                        'collapse_min_pkts' => 5,
                        'collection_size' => 1073741824,
                        'period' => 900
                }
        },
```

configuration	Unit	Description
collapse_min_bytes	bytes	Collapse all conversations with less than X bytes of traffic in the whole period into one; set to zero/undef to disable
collapse_min_pkts	packets	or window by minimum number of packets. a match of either criterion will cause collapsing
collection_size	bytes	Collection capped size
period	seconds	Summary period

Changing Summarisations

You can remove an existing summarisation from the configuration and restart the opflow daemon, you will need to drop the database collection manually once you are sure.

When adding additional summarisation intervals you will need to run the DB setup command as it will setup collection capping. You will want to stop the opflow daemon as well as omkd then restart them once you have made the changes you wanted.

```
service opflowd stop
opflow-cli.exe act=setup-db
service opflowd start
```

Also, and this is worth noting, the daemon will create all the summaries when it starts, so depending on how much data is already present, this may increase the load on the server.

opFlow GUI Modes

The opFlow GUI in version 3 includes two different dashboard pages, one optimised for high-volume and one optimised for high precision.

By default, the high-volume mode is active and the dashboard page shows *one* traffic summarised in one way (default is Top Applications, sorted by traffic volume in bytes). You can change the summary displayed using the Advanced menu (Summary Type and Summary Field). Changing Summary Type selects a different summary section, and affects the Flows over Time chart (i.e. the charted data is grouped according to your selection).

If you set `opflow_gui_high_flow_volume` to 0 or "false", the opFlow GUI switches to low-volume mode.

In this mode the dashboard shows the data summarised in *three* different ways, Top Talkers, Top Applications and Top Applications plus Sources, again sorted by traffic volume in bytes. Again, the Advanced menu lets you select the sort field (Summary Field), changing Summary Type changes *only* the Flows over Time chart in this mode..