Tracking Flow Volume and Import Performance

For Information about performance management in the latest version please refer to:

[opFlow 3 Scalability Features]

> Please note a lot of the below still applies in V3 and is still relevant to analysing the flow volumes you are or will be expecting to receive.

- For Information about performance management in the latest version please refer to:
- opFlow 3 Scalability Features
- Flow Volume - sent from Cisco Device
- Flow Volume - received by opFlow
- Import Performance

Flow Volume - sent from Cisco Device

Cisco devices will tell you the number of flows currently exported, **no matter if they were successfully received by a server or not**. This can be very useful to help calculate the amount of flow data that will come from a device before sending the flows to an opFlow server.

1. Get the current time
2. Get the number of flows
3. Wait X minutes, for example 2 or 60
4. Get the current time
5. Get the number of flows
6. Do some math, subtract second flow number from first flow number, that is the number of flows for X minutes, divide by X * 60 (so the number of seconds) and you have the number per second. Doing this over more time will give you a better idea of the overall number, it is suggested that you wait much more than 2 minutes

```
router> show clock
11:57:51.155 AEST Wed Apr 2 2014
```

```
router>sh ip flow export
Flow export v9 is enabled for main cache
  Export source and destination details :
  VRF ID : Default
    Destination(1)  192.168.1.7 (12345)
    Destination(2)  192.168.1.42 (12345)
  Version 9 flow records
  25716317 flows exported in 890127 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
  0 export packets were dropped due to no fib
  1 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
```

So 25716317 have been sent. Now wait 2 minutes, then get the time again

```
router>show clock
11:59:53.155 AEST Wed Apr 2 2014
```

Now get the number of flows again:

```
meatball>sh ip flow export
Flow export v9 is enabled for main cache
  Export source and destination details :
  VRF ID : Default
    Destination(1)  192.168.1.7 (12345)
    Destination(2)  192.168.1.42 (12345)
  Version 9 flow records
  25717645 flows exported in 890173 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
  0 export packets were dropped due to no fib
  1 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
```

Our clock difference was almost exactly 2 minutes, so we will use that number in the example for our calculations.

To figure out flows / 2 min or flows / second:

```
25717645 flows - 25716317 flows = 1328 flows / 2 min
```

```
1328 / 120 = 11 flows / second
```

The numbers you see will likely be higher, this is just an example. Please contact support@opmantek.com if you would like help correctly sizing your server.

Flow Volume - received by opFlow

opflow_stats.pl calculates how many flows per second are being received by the server. To view this info just run it

```
/usr/local/opmantek/bin/opflow_stats.pl
```

Output:

```
.........................................
Average Flows Per Second: 762
Average Flows per Cycle: 34
Flows Inserted per Cycle: 33
Conversations per Cycle: 23
Conversations Compression: 69%
Average LoadFlow: 0.0237999999999999
Average FlowReader: 0.0104999999999999
Average MvKill: 0.0101249999999999
```

In this example we are receiving 762 flows a second.

Average flows per cycle is how many flows are received in a 2 minute cycle. In this case it's saying 34, you may immediately think "this makes no sense" and you are correct. The system this info was taken from had been saving up flows (opflowd was not running) and then it inserted them all at once followed by very few flows being received after. This is a good representation of inconsistent flow records making the numbers look incorrect.

Import Performance

In addition to saving the flows to disk, opFlow also needs to parse and import the flows into the database. To get an idea of how many flows per second (fps) opflowd is importing into mongo run this command:

```
tail -500 /usr/local/opmantek/logs/opflow.log | grep "STATS Flows"
```

Output will have lines like this:

```
10-Mar-2013 17:20:06,opflowd.pl::runFlows#338<br>opFlowd: STATS Flows: FPS=1964 Flows=10682 FlowInserts=10681
ConvInserts=7300 LoadFlow=5.44 FlowReader=0.25 MvKill=0.00
10-Mar-2013 17:24:00,opflowd.pl::runFlows#338<br>opFlowd: STATS Flows: FPS=1050 Flows=21 FlowInserts=20
ConvInserts=16 LoadFlow=0.02 FlowReader=0.01 MvKill=0.01
```

As you can see the system this info was taken from is very erratic (as mentioned above).

By default opFlowd runs every 2 minutes, the output tells us how many FPS (flows per second) it is able to import, how many inserts were made as well has how many Converstation Inserts were made.

Conversations are a group of flows between two endpoints that have been summarised over a given period defined in the opFlow.nmis config file, this is done to reduce the amount of data that is saved and therefor use less space. This will only happen if they are enabled, and you will see this data displayed (instead of raw flows) if display is enabled:

```
  'opflow_summarisation_interval' => '60',
  'opflow_summarisation_enabled' => 'true',
  'opflow_summarisation_display' => 'true',
```

If you wish to only use summarised data (and not raw data, most likely to save space) you will want to turn off raw flow data:

```
  'opflow_keep_raw_flows' => 'true',
```