Multi-Queue hotfix

12 July 2015
Troubleshooting

1. After reboot, the wrong interfaces are configured for Multi-Queue...

2. When changing the status of interfaces, all the interface IRQs are assigned to CPU 0 or to all of the CPU cores...

3. In VSX mode (R75.40VS and above): An 'fwk' process runs on the same CPU as some of the interface queues...

4. In Security Gateway mode: After changing the number of CoreXL Firewall instances, Multi-Queue is disabled on all interfaces...

5. Changing the affinity of CoreXL Firewall instances...

6. Make the decision to use the Multi-Queue hotfix...

7. Make sure that the network interface cards support Multi-Queue...

8. Decide whether more CPU cores can be allocated to the SND...

9. Examine the CPU utilization...

10. Examine the CPU roles allocation...

11. Make sure that SecureXL is enabled...

12. Decide whether more CPU cores can be allocated to the SND...

13. Changing the status of an interface with Multi-Queue...

14. Changing the affinity of CoreXL Firewall instances...

15. In VSX mode (R75.40VS and above): Changing the number of CPU cores, to which the fwk processes are assigned...

16. Checking the Multi-Queue CPU utilization...

17. Controlling the number of traffic queues...

18. In Security Gateway mode: Changing the number of CoreXL Firewall instances when Multi-Queue is enabled on some, or on all interfaces...

19. Adding a network interface card...

20. Make the decision to use the Multi-Queue hotfix...

21. Examine the CPU utilization...
Getting Started Guide

Introduction

Today, each network interface card has one traffic queue that is handled by one CPU at a time.

Since the Secure Network Distributor (SND) - SecureXL and CoreXL dispatcher is running on the CPU cores that handle the traffic queues, user cannot use more CPU cores for acceleration than the number of network interface cards passing the traffic.

Multi-Queue lets the user configure more than one traffic queue for each network interface card, which allows using more CPU cores for acceleration. Multi-Queue is supported on network interface cards that use  

Notes

1. Currently, the Multi-Queue feature is not integrated into Check Point operating systems. Therefore, a Multi-Queue hotfix (package) is provided in order to add the Multi-Queue feature.
2. The Multi-Queue hotfix does not depend on any other Check Point hotfix, and other Check Point hotfixes do not depend on the Multi-Queue hotfix.
3. If an older version of Multi-Queue hotfix is installed on the Security Gateway, then the hotfix and its configuration will be lost during the upgrade from one Check Point version to another (e.g., from R70.50 to R75.40VS). This hotfix must be installed after upgrading in order to add the Multi-Queue functionality. Once the Multi-Queue hotfix is installed, user must configure the Multi-Queue again.
How to install the Multi-Queue hotfix

Refer to the "Deciding if the Multi-Queue hotfix is needed" paragraph for deciding whether the given Security Gateway can benefit from the Multi-Queue hotfix.

**Note:** Links to the files and step-by-step instructions are provided in the sk80940 (Multi-Queue hotfix for Security Gateway).

**General action plan:**

1. Download the FireWall hotfix package (fw1_<HOTFIX_NAME>.tgz) to Security Gateway (for example, into "/path_to_hotfix/fw/").
2. Download the SecureXL hotfix package (sim_<HOTFIX_NAME>.tgz) to Security Gateway (for example, into "/path_to_hotfix/sim/").
   **Note:** R75.40 and R75.40VS do not require this package.
3. Download the relevant Linux kernel RPMs with necessary modified NIC drivers igb v2.4.13 and ixgbe v3.1.17 to Security Gateway (for example, into "/path_to_hotfix/kernel/"):  
   A) 32-bit kernel for CPU with PAE support  
   B) 32-bit kernel for CPU without PAE support  
   C) 64-bit kernel
   • To check whether the CPU supports PAE, run this long command:
   
   `[Expert@HostName]# (cat /proc/cpuinfo | grep 'flags' | awk 'NR==1' | grep -qi 'pae' && echo PAE) || (echo No-PAE)`
   
   • To check whether the machine runs with 32-bit or 64-bit kernel, run:

   `
   [Expert@HostName]# uname -r
   '2.6.18-92cp' means 32-bit/'2.6.18-92cpx86_64' means 64-bit`
4. Unpack the FireWall hotfix package:

   `[Expert@HostName]# cd /path_to_hotfix/fw/`
   `[Expert@HostName]# tar -zxvf fw1_<HOTFIX_NAME>.tgz`
5. Install the FireWall hotfix package:

   `[Expert@HostName]# ./fw1_<HOTFIX_NAME>
   All Check Point services will be stopped ('cpstop').`
The installation will prompt for a reboot once it finishes. **Do not reboot yet.**

6. Unpack the SecureXL hotfix package:
   
   ```
   [Expert@HostName]# cd /path_to_hotfix/sim/
   [Expert@HostName]# tar -zxvf sim_<HOTFIX_NAME>.tgz
   ```

7. Install the SecureXL hotfix package:
   
   ```
   [Expert@HostName]# ./sim_<HOTFIX_NAME>
   ```
   All Check Point services will be stopped ('cpstop'). The installation will prompt for a reboot once it finishes. **Do not reboot yet.**

8. Install the Linux kernel(s) with modified NIC drivers:
   
   ```
   [Expert@HostName]# cd /path_to_hotfix/kernel/
   [Expert@HostName]# rpm -Uhv –force kernel*.rpm
   ```

9. Reboot the Security Gateway:
   
   ```
   [Expert@HostName]# reboot
   ```

10. Configure the Multi-Queue hotfix as described in “Multi-Queue Basic Configuration”.

---

**How to uninstall the Multi-Queue hotfix**

**General action plan:**

1. Disable the Multi-Queue completely on each interface:
   
   ```
   [Expert@HostName]# cpmq set
   ```

2. Run the uninstall script for FireWall hotfix:
   
   ```
   [Expert@HostName]# sh /opt/CPsuite-R7<x>/uninstall_fw1_<HOTFIX_NAME>.sh
   ```

3. Run the uninstall script for SecureXL hotfix:
   
   ```
   [Expert@HostName]# sh /opt/CPsuite-R7<x>/uninstall_sim_<HOTFIX_NAME>.sh
   ```

4. Install the native (default) Linux kernel RPM for the current version (to replace the NIC drivers). **Contact Check Point Support** to get native (default) Linux kernel RPM, or just the native (default) NIC driver files.

5. Reboot the Security Gateway.
Multi-Queue Basic Configuration

Use the `cpmq` utility to see and to change the current Multi-Queue configuration.

1. Configuring the Multi-Queue
   The `cpmq set` command allows the user to configure Multi-Queue on the supported network interface cards.

   To configure Multi-Queue:
   Run `cpmq set`

   The command shows all supported interfaces that are active and allows changing the Multi-Queue configuration for each network interface cards. Network interface cards that are administratively down, do not appear in the output.

   **Note:** Changes in the Multi-Queue configuration require reboot.

2. Querying the Multi-Queue configuration
   The `cpmq get` command shows the current Multi-Queue status on the supported network interface cards.

   Run `cpmq get [-a]`

   The `-a` flag shows the Multi-Queue configuration for all supported network interface cards (both active and inactive). Without this flag, the command shows only the active network interface cards.

   **Output example:**
   ```
   [Expert@FW]# cpmq get -a
   Active igb interfaces:
   eth1-05 [On]
   eth1-06 [Off]
   eth1-01 [Off]
   eth1-03 [Off]
   eth1-04 [On]
   Non active igb interfaces:
   eth1-02 [Off]
   [Expert@FW]#
   ```
In this example, there are:

- 2 interfaces that are up with Multi-Queue enabled (eth1-05, eth1-04)
- 3 interfaces that are up with Multi-Queue disabled (eth1-06, eth1-01, eth1-03)
- 1 interface that supports Multi-Queue, but is currently down (eth1-02).

Status explanation:

- **On** - Multi-Queue is enabled on the interface.
- **Off** - Multi-Queue is disabled on the interface.
- **Pending On** - Multi-Queue will be enabled on this interface after reboot (currently disabled).
- **Pending Off** - Multi-Queue will be disabled on this interface after reboot (currently enabled).

### Multi-Queue Limitations

- Multi-Queue is supported only on **igb** (1 GbE) and **ixgbe** (10 GbE) drivers. Only network interface cards that use these drives can benefit from the Multi-Queue hotfix. See “Make sure the network interface cards support Multi-Queue” for further details.
- When updating these NIC drivers to a newer version, make sure that Multi-Queue is supported by the new version of these drivers.
- Multi-Queue is not supported on single-CPU machines.
- Multi-Queue can be configured only on **two** network interface cards due to IRQ limitations.

**To override this limitation:**

Run `cpmq set -f`

- Maximum CPU allocations:
The number of traffic queues is limited by the number of CPU cores and by these additional limitations:
  - **igb** - a maximum of 4 traffic queues is allowed.
  - **ixgbe** - a maximum of 16 traffic queues is allowed.
To override this limitation:

Run `cpmq rx_num <igb/ixgbe> number_of_active_RX_queues -f`

- When installing a new interface that uses the `igb` or `ixgbe` driver, run `cpmq reconfigure`, and reboot the machine.

Multi-Queue Administration Guide

General information about CPU roles

There are two main roles for CPU cores applicable to SecureXL and CoreXL:

1. **SecureXL and CoreXL dispatcher CPU** (a.k.a the SND - Secure Network Distributor). This role can be configured manually with the `sim affinity -s` command.

2. **CoreXL Firewall instance CPU**. This role can be configured manually with the `fw ctl affinity` command.

For best performance, the same CPU core should not work in both roles. During the installation, a default CPU role configuration is set. For example, on a machine with 8 CPU cores: the 2 CPU cores with the lowest CPU ID are set as SNDs, and the 6 CPU cores with the highest CPU IDs are set as CoreXL Firewall instances.

Without this Multi-Queue hotfix, the number of CPU cores allocated to the SND role is limited by the number of network interface cards passing the traffic. Since each network interface card has one traffic queue, each queue can be handled by only one CPU core at a time. This means that the SND can use only one CPU core at a time per network interface card.

When most traffic is accelerated, the CPU load for SND can be very high while the CPU load for CoreXL Firewall instances can be very low. This is inefficient utilization of CPU capacity of the machine.

The Multi-Queue hotfix allows the user to configure more than one traffic queue for each supported network interface card, so that more than one SND CPU core can handle the
traffic of a single network interface card at a time. This balances the load more efficiently between SND CPU cores and CoreXL Firewall instances CPU cores.

**Terminology**

- **SND** - Secure Network Distributor. A CPU core that runs SecureXL and CoreXL dispatcher.
- **rx queue** - receive packet queue
- **tx queue** - transmit packet queue
- **IRQ affinity** - binding an IRQ to specific CPU core(s)

**Deciding if the Multi-Queue hotfix is needed**

This section is intended to help the user to decide if the given Security Gateway can benefit from the Multi-Queue hotfix.

The given Security Gateway can benefit from the Multi-Queue hotfix when the CPU load for SND is high while the CPU load for CoreXL Firewall instances is low, and, when it is not possible to allocate more CPU cores for SND by changing the IRQ affinity of the network interface cards.

It is strongly recommended performing the following steps *before* installing the Multi-Queue hotfix.

1. **Make sure that SecureXL is enabled**

   The Multi-Queue hotfix is relevant only if SecureXL is enabled.

   Run `fwaccel stat`.

   Look at the “Accelerator Status” field.
Output example:

[Expert@FW]# fwaccel stat
Accelerator Status : on
Accept Templates : enabled
Drop Templates : disabled
............................
[Expert@FW]#

2. Examine the CPU roles allocation

To see the CPU roles allocation, run `fw ctl affinity -l`.
This command shows the CPU affinity of the interfaces and the CPU affinity of
CoreXL Firewall instances.

Output example from R75.40VS in Security Gateway mode:

[Expert@FW]# fw ctl affinity -l
Mgmt: CPU 0
eth1-05: CPU 0
eth1-06: CPU 1
fw_0: CPU 5
fw_1: CPU 4
fw_2: CPU 3
fw_3: CPU 2
[Expert@FW]#

In this example, the SND is running on CPU0 and CPU1, and CoreXL Firewall
instances are running on CPU cores 2-5.

Output example from R75.40VS in VSX mode:

[Expert@FW:0]# fw ctl affinity -l
Mgmt: CPU 0
eth1-05: CPU 0
eth1-06: CPU 1
VS_0 fwk: CPU 2 3 4 5
VS_1 fwk: CPU 2 3 4 5
[Expert@FW:0]#

In this example the SND is running on CPU0 and CPU1, and the CoreXL Firewall
instances (part of the `fwk` processes) of all the Virtual Systems are running on CPU
cores 2-5.
3. Examine the CPU utilization

To check the CPU utilization, run `top`.

Press digit '1' to display the utilization for each CPU core.

Output example:

```
top - 18:02:33 up 28 days,  1:18,  1 user,  load average: 1.22, 1.38, 1.48
Tasks: 137 total,  3 running, 134 sleeping,  0 stopped,  0 zombie
Cpu0 : 2.0%us, 0.0%sy, 0.0%ni, 32.7%id,  5.9%wa, 0.0%hi, 59.4%si,  0.0%st
Cpu1 : 0.0%us, 1.0%sy, 0.0%ni, 35.2%id,  0.0%wa, 0.0%hi, 63.8%si,  0.0%st
Cpu2 : 2.0%us, 2.0%sy, 0.0%ni, 69.5%id,  0.0%wa, 4.0%hi, 22.5%si,  0.0%st
Cpu3 : 1.0%us, 2.0%sy, 0.0%ni, 74.5%id,  0.0%wa, 0.0%hi, 22.5%si,  0.0%st
Cpu4 : 5.0%us, 1.0%sy, 0.0%ni, 72.5%id,  0.0%wa, 0.0%hi, 21.5%si,  0.0%st
Mem: 12224020k total, 7005820k used, 5218200k free, 273536k buffers
Swap:14707496k total, 0k used, 14707496k free, 484340k cached
```

In this example we can see that SND CPU cores (CPU0 and CPU1) are about 30% idle, and the CoreXL Firewall instances CPU cores (CPU2, CPU3 and CPU4) are about 70% idle.

4. Decide whether more CPU cores can be allocated to the SND

If the number of the network interfaces handling traffic is bigger than the number of the CPU cores assigned to SND (see output of `fw ctl affinity -l` in Step 2 above), it is possible to allocate more CPU cores for SND.

Note: for configuration instructions, refer to any R7x Firewall Administration Guide - chapter “CoreXL Administration”.

In the following example, it is possible to change the IRQ affinity of the interfaces to use more CPU cores for the SNDs:

- `eth1-01` - connected to an internal network
- `eth1-02` - connected to an internal network
- `eth2-01` - connected to the DMZ
- `eth3-01` - connected to the external network
- `fw ctl affinity` -1 shows the following IRQ affinity:
5. Make sure the network interface cards support Multi-Queue

Note: Multi-Queue is supported only on machines that run SecurePlatform or Gaia operating system, and only for network interface cards that use **igb** (1 GbE) and **ixgbe** (10 GbE) drivers.

For most up-to-date list of supported interface network cards, refer to [sk80940 (Multi-Queue hotfix for Security Gateway)].

A) Security Appliances running SecurePlatform / Gaia OS

<table>
<thead>
<tr>
<th>1 GbE cards</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Ethernet Controller</th>
<th>CPAC-ACC-2-1F</th>
<th>CPAC-ACC-4-1F</th>
<th>CPAC-ACC-4-1C</th>
<th>CPAC-ACC-4-1C-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Controller Speed</td>
<td>1 GbE</td>
<td>1 GbE</td>
<td>1 GbE</td>
<td>1 GbE</td>
</tr>
<tr>
<td>Driver Type</td>
<td>igb</td>
<td>igb</td>
<td>igb</td>
<td>igb</td>
</tr>
<tr>
<td>Supported Platforms</td>
<td>4200</td>
<td>4200</td>
<td>4200</td>
<td>13500</td>
</tr>
<tr>
<td></td>
<td>4400</td>
<td>4400</td>
<td>4400</td>
<td>13800</td>
</tr>
<tr>
<td></td>
<td>4600</td>
<td>4600</td>
<td>4600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4800</td>
<td>4800</td>
<td>4800</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12200</td>
<td>12200</td>
<td>12200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12400</td>
<td>12400</td>
<td>12400</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12600</td>
<td>12600</td>
<td>12600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13500</td>
<td>13500</td>
<td>13500</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13800</td>
<td>13800</td>
<td>13800</td>
<td></td>
</tr>
</tbody>
</table>
### 1 GbE cards (continue)

<table>
<thead>
<tr>
<th>Expansion Card Model</th>
<th>CPAC-ACC-8-1C</th>
<th>CPAC-ACC-8-1C-L</th>
<th>CPAC-12-1C-21000</th>
<th>CPAC-12-1F-21000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Controller</td>
<td>Intel 82580EB</td>
<td>Intel 82580EB</td>
<td>Intel 82576EB</td>
<td>Intel 82576EB</td>
</tr>
<tr>
<td>Ethernet Controller</td>
<td>1 GbE</td>
<td>1 GbE</td>
<td>1 GbE</td>
<td>1 GbE</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Driver Type</td>
<td>igb</td>
<td>igb</td>
<td>igb</td>
<td>igb</td>
</tr>
<tr>
<td>Supported Platforms</td>
<td>4800</td>
<td>12200</td>
<td>21400</td>
<td>21400</td>
</tr>
<tr>
<td></td>
<td>12200</td>
<td>13500</td>
<td>21600</td>
<td>21600</td>
</tr>
<tr>
<td></td>
<td>12400</td>
<td>13800</td>
<td>21700</td>
<td>21700</td>
</tr>
<tr>
<td></td>
<td>12600</td>
<td></td>
<td>21800</td>
<td>21800</td>
</tr>
</tbody>
</table>

### 10 GbE cards

<table>
<thead>
<tr>
<th>Expansion Card Model</th>
<th>CPPWR-ACC-2-10SRF</th>
<th>CPPWR-ACC-2-10LRF</th>
<th>CPAC-ACC-2-10F</th>
<th>CPAC-ACC-4-10F</th>
<th>CPAC-4-10F-21000</th>
<th>CPAC-ACCL-4-10F-21000 *</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Controller</td>
<td>Intel 82598EB</td>
<td>Intel 82598EB</td>
<td>Intel 82598</td>
<td>Intel 82599EB</td>
<td>Intel 82599EB</td>
<td>Intel 82599EB</td>
</tr>
<tr>
<td>Ethernet Controller</td>
<td>10 GbE</td>
<td>10 GbE</td>
<td>10 GbE</td>
<td>10 GbE</td>
<td>10 GbE</td>
<td>10 GbE</td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Driver Type</td>
<td>ixgbe</td>
<td>ixgbe</td>
<td>ixgbe</td>
<td>ixgbe</td>
<td>ixgbe</td>
<td>ixgbe</td>
</tr>
<tr>
<td>Supported Platforms</td>
<td>Power-1 50xx</td>
<td>Power-1 50xx</td>
<td>4800</td>
<td>12200</td>
<td>13800</td>
<td>21400</td>
</tr>
<tr>
<td></td>
<td>Power-1 90xx</td>
<td>Power-1 90xx</td>
<td>12400</td>
<td>12400</td>
<td>21400</td>
<td>21600</td>
</tr>
<tr>
<td></td>
<td>Power-1 11xxx</td>
<td>Power-1 11xxx</td>
<td>12600</td>
<td>13500</td>
<td>21700</td>
<td>21700</td>
</tr>
<tr>
<td></td>
<td>13500</td>
<td>13800</td>
<td>21800</td>
<td>21800</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: 'CPAC-ACCL-4-10F-21000' is a Security Acceleration Module (SAM108).
**Fail Open cards (also known as Bypass cards)**

<table>
<thead>
<tr>
<th>Expansion Card Model</th>
<th>CPAC-4-1C-BP</th>
<th>CPAC-4-1FSR-BP</th>
<th>CPAC-4-1FLR-BP</th>
<th>CPAC-2-10FSR-BP</th>
<th>CPAC-2-10FLR-BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Controller</td>
<td>Intel 82580EB</td>
<td>Intel 82580EB</td>
<td>Intel 82580EB</td>
<td>Intel 82599ES</td>
<td>Intel 82599ES</td>
</tr>
<tr>
<td>Ethernet Controller Speed</td>
<td>1 GbE</td>
<td>1 GbE</td>
<td>1 GbE</td>
<td>10 GbE</td>
<td>10 GbE</td>
</tr>
<tr>
<td>Driver Type</td>
<td>igb</td>
<td>igb</td>
<td>igb</td>
<td>ixgbe</td>
<td>ixgbe</td>
</tr>
<tr>
<td>Supported Platforms</td>
<td>4200</td>
<td>4200</td>
<td>4200</td>
<td>4800</td>
<td>4800</td>
</tr>
<tr>
<td></td>
<td>4400</td>
<td>4400</td>
<td>4400</td>
<td>12200</td>
<td>12200</td>
</tr>
<tr>
<td></td>
<td>4600</td>
<td>4600</td>
<td>4600</td>
<td>12400</td>
<td>12400</td>
</tr>
<tr>
<td></td>
<td>4800</td>
<td>4800</td>
<td>4800</td>
<td>12600</td>
<td>12600</td>
</tr>
<tr>
<td></td>
<td>12200</td>
<td>12200</td>
<td>12200</td>
<td>13500</td>
<td>13500</td>
</tr>
<tr>
<td></td>
<td>12400</td>
<td>12400</td>
<td>12400</td>
<td>13800</td>
<td>13800</td>
</tr>
<tr>
<td></td>
<td>12600</td>
<td>12600</td>
<td>12600</td>
<td>Smart-1 225</td>
<td>Smart-1 225</td>
</tr>
<tr>
<td></td>
<td>13500</td>
<td>13500</td>
<td>13500</td>
<td>Smart-1 3050</td>
<td>Smart-1 3050</td>
</tr>
<tr>
<td>Smart-1 225</td>
<td>13800</td>
<td>13800</td>
<td>13800</td>
<td>Smart-1 3150</td>
<td>Smart-1 3150</td>
</tr>
<tr>
<td>Smart-1 3050</td>
<td>Smart-1</td>
<td>Smart-1</td>
<td>Smart-1</td>
<td>Smart-1</td>
<td>Smart-1</td>
</tr>
<tr>
<td>Smart-1 3150</td>
<td>Smart-1</td>
<td>Smart-1</td>
<td>Smart-1</td>
<td>Smart-1</td>
<td>Smart-1</td>
</tr>
</tbody>
</table>

**B) IP Appliances running Gaia OS**

Multi-Queue is supported on IP1280, IP2450 machines for the XMC 1 GbE network interface cards.

**C) Open Servers running SecurePlatform / Gaia OS**

To check, which driver is used by an interface, run

```
[Expert@FW]# ethtool -i <interface_name>
```
6. Make the decision to use the Multi-Queue hotfix

We recommend installing and configuring the Multi-Queue hotfix:

1. If the CPU load for SND is high (idle is less than 20%), and the CPU load for CoreXL Firewall instances is low (idle is greater than 50%).
2. If it is not possible to assign more CPU cores to the SND by changing the IRQ affinity of interfaces.

Advanced Multi-Queue settings

1. Controlling the number of traffic queues

   - ixgbe driver:
     When configuring Multi-Queue for an ixgbe-interface, an RxTx queue is created per CPU. It is not possible to control the number of active rx queues (all tx queues are active).

   - igb driver:
     When configuring Multi-Queue for an igb-interface, the number of tx queues and rx queues is calculated by the number of active rx queues.

   - On the Security Gateway, and on R75.40VS in Security Gateway mode:
     By default, the number of active rx queues is calculated by:
     \[
     \text{active rx queues} = [\text{number of CPU cores} - \text{number of CoreXL Firewall instances}]
     \]

   - On R75.40VS in VSX mode:
     By default, the number of active rx queues is calculated by:
     \[
     \text{active rx queues} = \text{the lowest CPU ID to which an fwk process is assigned}
     \]

   - The number of active rx queues can be controlled by running:
     \[
     \text{cpmq set rx\_num <igb/ixgbe> number\_of\_active\_RX\_queues}
     \]
     This command overrides the default value.

   - The number of active rx queues can be checked by running:
     \[
     \text{cpmq get rx\_num <igb/ixgbe>}
     \]

   - On Security Gateway, the number of active queues changes automatically when changing the number of CoreXL Firewall instances (via cpconfig menu).
This automatic change is not applicable, if the number of rx queues was configured manually.

- Manual configuration of active rx queues can be deleted by running:
  
cpmq set rx_num <igb/ixgbe> default

2. IRQ affinity for Multi-Queue

- The IRQ affinity of the traffic queues is set automatically in the following way during the operating system boot:
  
  rxtx-0 -> CPU 0  
rxtx-1 -> CPU 1  
rxtx-2 -> CPU 2  
  and so on.

  This is also true in cases where rx and tx queues are assigned with a separated IRQ:
  
rx-0 -> CPU 0  
  tx-0 -> CPU 0  
rx-1 -> CPU 1  
  tx-1 -> CPU 1  
  and so on.

- The `sim affinity` or the `fw ctl affinity` commands cannot be used to change and query the IRQ affinity for interfaces with enabled Multi-Queue.

- The affinity of Multi-Queue IRQs can be reset by running `cpmq set affinity`.

- The affinity of Multi-Queue IRQs can be checked by running `cpmq get -v`.

- Do not change the IRQ affinity of queues manually. Changing the IRQ affinity of the traffic queues manually can affect performance.
3. Checking the Multi-Queue CPU utilization

1. Find the CPU cores assigned to Multi-Queue IRQs by running `cpmq get -v`.

   Output example:
   
   ```
   [Expert@FW]# cpmq get -v
   
   Active igb interfaces:
   eth1-05 [On]
   eth1-06 [Off]
   eth1-01 [Off]
   eth1-03 [Off]
   eth1-04 [On]
   
   multi-queue affinity for igb interfaces:
   
   eth1-05:
   
<table>
<thead>
<tr>
<th>irq</th>
<th>cpu</th>
<th>queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>178</td>
<td>0</td>
<td>TxRx-0</td>
</tr>
<tr>
<td>186</td>
<td>1</td>
<td>TxRx-1</td>
</tr>
</tbody>
</table>
   
   eth1-04:
   
<table>
<thead>
<tr>
<th>irq</th>
<th>cpu</th>
<th>queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>0</td>
<td>TxRx-0</td>
</tr>
<tr>
<td>131</td>
<td>1</td>
<td>TxRx-1</td>
</tr>
</tbody>
</table>
   
   [Expert@FW]#
   ```

   In this example, Multi-Queue is enabled on two `igb`-interfaces (`eth1-05` and `eth1-04`). The number of active rx queues is configured to 2 (for `igb`, the number of queues is calculated by the number of active rx queues). The IRQs for both interfaces are assigned to CPU 0 and CPU 1.

2. Check the CPU utilization by running `top`.

   Press digit '1' to display the utilization for each CPU core.

   Output example:
   
   ```
   top - 18:02:33 up 28 days, 1:18, 1 user, load average: 1.22, 1.38, 1.48
   Tasks: 137 total, 3 running, 134 sleeping, 0 stopped, 0 zombie
   Cpu0 : 2.0%us, 0.0%sy, 0.0%ni, 42.7%id, 5.9%wa, 0.0%hi, 49.4%si, 0.0%st
   Cpu1 : 0.0%us, 1.0%sy, 0.0%ni, 55.2%id, 0.0%wa, 0.0%hi, 43.8%si, 0.0%st
   Cpu2 : 2.0%us, 2.0%sy, 0.0%ni, 45.5%id, 0.0%wa, 4.0%hi, 46.5%si, 0.0%st
   Cpu3 : 1.0%us, 2.0%sy, 0.0%ni, 74.5%id, 0.0%wa, 0.0%hi, 22.5%si, 0.0%st
   Cpu4 : 5.0%us, 1.0%sy, 0.0%ni, 42.6%id, 0.0%wa, 0.0%hi, 51.5%si, 0.0%st
   
   Mem: 12224020k total, 7005820k used, 5218200k free, 273536k buffers
   Swap: 14707496k total, 0k used, 14707496k free, 484340k cached
   
   PID USER FR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
   3301 root 15 0 0 0 0 R 17 0.0 2747:04 [fw_worker_0]
   3246 root 15 0 0 0 0 R 15 0.0 2683:27 [fw_worker_1]
   3226 root 15 0 0 0 0 R 16 0.0 2593:35 [fw_worker_2]
   ```
Special Scenarios and Configuration

1. In Security Gateway mode: Changing the number of CoreXL Firewall instances when Multi-Queue is enabled on some, or on all interfaces

For best performance, the default number of active rx queues is calculated by:

\[
\text{active rx queues} = \left[ \text{number of CPU cores} - \text{number of CoreXL Firewall instances} \right]
\]

This configuration is set automatically when configuring Multi-Queue. When changing the number of CoreXL Firewall instances, the number of active rx queues will change automatically, if it was not set manually.

2. In VSX mode (R75.40VS and above): Changing the number of CPU cores, to which the fwk processes are assigned

The default number of active rx queues is calculated by:

\[
\text{active rx queues} = \text{the lowest CPU ID to which an fwk process is assigned}
\]

Output example:

```
[Expert@FW:0]# fw ctl affinity -l
Mgmt: CPU 0
et1-05: CPU 0
et1-06: CPU 1
VS_0 fwk: CPU 2 3 4 5
VS_1 fwk: CPU 2 3 4 5
[Expert@FW:0]#
```

In this example, the number of active rx queues is set to 2. This configuration is set automatically when configuring Multi-Queue. It will not automatically update when changing the affinity of the Virtual Systems. When changing the affinity of the Virtual Systems, make sure to follow the instructions in “Controlling the number of traffic queues”.
3. Changing the status of an interface with Multi-Queue

1. Changing the status to “down”
   Multi-Queue configuration will be saved when changing the status of an interface to “down”. Since the number of interfaces with Multi-Queue enabled is limited to two, user may need to disable Multi-Queue on an interface after changing its status to “down” in order to enable Multi-Queue on another interface. In this case, the `cpmq set` command will suggest and allow the disabling of Multi-Queue on the non-active interfaces.

2. Changing the status to “up”
   Set the IRQ affinity for the Multi-Queue interfaces, when:
   - Multi-Queue on an interface is enabled
   - the interface status is changed to “down”
   - the machine was rebooted
   - the interface status is changed back to “up”
   To set the IRQ affinity of Multi-Queue interfaces, run `cpmq set affinity`

4. Adding a network interface card

When adding a network interface card to the machine that uses `igb` or `ixgbe` drivers, the Multi-Queue configuration can change due to interface indexing. If a network interface card that uses `igb` or `ixgbe` is added to the machine, make sure to re-configure the Multi-Queue by running `cpmq reconfigure`. If a reconfiguration change is required, user will be prompted to reboot the machine.

5. Changing the affinity of CoreXL Firewall instances

For best performance, we recommend not to assign both SND and a CoreXL Firewall instance to the same CPU core.
When changing the affinity of the CoreXL Firewall instances to a CPU assigned with one of the Multi-Queue queues (as shown by `cpmq get -v`), we recommend to reconfigure the number of active rx queues following this rule:

\[ \text{active rx queues} = \text{the lowest CPU number to which a CoreXL Firewall instance is assigned.} \]

The number of active rx queues can be configured by running:

\[ \text{cpmq set rx\_num <igb/ixgbe> <value/default>} \]

As explained in “Controlling the number of traffic queues”.
Troubleshooting

1. After reboot, the wrong interfaces are configured for Multi-Queue

This can happen after changing the physical interfaces on the computer. To solve this issue, run `cpmq reconfigure`, and reboot the machine. Alternatively, configure Multi-Queue from the scratch.

2. When changing the status of interfaces, all the interface IRQs are assigned to CPU 0 or to all of the CPU cores

This can happen when an interface status is changed to “up” after the automatic affinity procedure runs (the affinity procedure runs automatically during boot). To solve this issue, run `cpmq set affinity`.

3. In VSX mode (R75.40VS and above): An ‘fwk’ process runs on the same CPU as some of the interface queues

This can happen when the affinity of Virtual System was manually changed, but Multi-Queue was not reconfigured accordingly. To solve this issue, configure the number of active rx queues manually, or run `cpmq reconfigure`, and reboot the machine.

4. In Security Gateway mode: After changing the number of CoreXL Firewall instances, Multi-Queue is disabled on all interfaces

When changing the number of CoreXL Firewall instances, the number of active rx queues automatically changes according to this rule (if not configured manually):

\[
\text{active rx queues} = [\text{number of CPUs} - \text{number of CoreXL Firewall instances}].
\]
If the number of CoreXL Firewall instances is equal to the number of CPU cores, or if the
difference between the number of CPU cores and the number of CoreXL Firewall
instances is 1, then the Multi-Queue will be disabled.
To solve this issue, configure the number of active rx queues manually by running:
cpmq set rx_num <igb/ixgbe> <value>