TCAM Resource Issue Workarounds Explained

Contents

Introduction

This document describes TCAM resource issue workarounds.

Common TCAM Errors

%ACLQOS-SLOT3-4-ACLQOS_OVER_THRESHOLD Tcam 0 Bank 0's usage has reached its threshold

%ACLMGR-3-ACLMGR_VERIFY_FAIL Verify failed: client 8200016E, Sufficient free entries are not available in TCAM bank

"ERROR: Insertion of TCAM entry failed due to Spanslogic TCAM constraints" -- on XL modules only

For more spanslogic TCAM constraints, please refer to .

Hardware ACL Resource Utilization

Command:
show hardware access-list resource utilization module <mod>

```
SITE1-AGG1# show hardware access-list resource utilization mod 3


INSTANCE 0x0
-------------
```

```
        ACL Hardware Resource Utilization (Mod 3)
        --------------------------------------------
                         Used    Free    Percent
                                         Utilization
-----------------------------------------------------
Tcam 0, Bank 0           9       16375   0.05
Tcam 0, Bank 1           2       16382   0.01
Tcam 1, Bank 0           7       16377   0.04
Tcam 1, Bank 1           246     16138   1.50

LOU                       3       101     2.88
Both LOU Operands         2
Single LOU Operands       1
LOU L4 src port:          0
LOU L4 dst port:          1
LOU L3 packet len:        0
LOU IP tos:               0
LOU IP dscp:              0
LOU ip precedence:        0
LOU ip TTL:               0
TCP Flags                 0       16      0.00

Protocol CAM              4       3       57.14
Mac Etype/Proto CAM       9       5       64.28

Non L4op labels, Tcam 0   2       6141    0.03
Non L4op labels, Tcam 1   3       6140    0.04
L4 op labels, Tcam 0      0       2047    0.00
L4 op labels, Tcam 1      1       2046    0.04

Ingress Dest info table   131072  510     0.39
Egress Dest info table    65536   511     0.19
SITE1-AGG1#
```

Options

The listed below are the few options when the TCAM usage are high.

- Atomic update
  Command: **no hardware access-list update atomic**

- Disable statistics per entry in all the ACLs
  Command: **no statistics per-entry**

- Fragments handling
  Command: **fragments deny-all/permit-all**

- ACE expansion threshold
  Command: **hardware access-list lou resource threshold**

- Resource pooling (not service impacting since the existing entries are not moved)
  Command: **hardware access-list resource pooling mod <x>**

Atomic Update

By default, N7K performs an atomic Access Control List (ACL) update to a module when there's ACL change. An atomic update does not disrupt traffic that the updated ACL applies to. However, an atomic update requires that an I/O module that receives an ACL update has enough available resources to store each updated ACL entry in addition to all pre-existing entries in the affected ACL. After the update occurs, the additional resources used for the update are freed. If the I/O module lacks the required resources, the device generates an error message and the ACL update to the I/O module fails.

If an I/O module lacks the resources required for an atomic update, you can disable atomic updates by using

```
no hardware access-list update atomic
```

However, during the brief time required for the device to remove the pre-existing ACL and implement the updated ACL, traffic that the ACL applies to is dropped by default. If you want to permit all traffic that an ACL applies to, while it receives a nonatomic updat. Use the **hardware access-list update default-result permit** command.

> **Note**: If atomic and non-atomic update are both possible (say the TCAM has enough free space), atomic is preferable. If there is not enough free space to do atomic update, then non-atomic is tried. Therefore the current implementation is always try atomic first, even when atomic update is disabled. However, currently upon failure due to spanslogic constrains, it is not switched to non atomic, and CSCud36802 is filed to address this (to be fixed in Freetown as of today).

> **Note**: When trying to remove ACE while TCAM usage is high, since atomic update is always tried first as mentioned above, the spanslogic contrains could still be hit and CSCua24513 was filed to address this issue (fixed in 5.2.7).

Statistics per Entry

By default N7K would try to merge features when programming the TCAM, which helps saving the TCAM resource. When **statistics per entry** is configured, the entries are not merged to maintain per-Access Control Entries (ACE) stats, in which case it could take more resources.

This command does not have any preformance impact since ACL processing is always in the hardware.

There are two options to display the stats:

```
show ip access-list <acl>
```

> **Note**: Displays counters for only those hardware entries hit that are programmed of policy type PACL/RACL (e.g. acl applied on interfaces)

```
show hardware internal access-list input entries detail module <x>
```

> **Note**: ACL used inside copp policy is used for classification of packets. Decision to whether allow/deny/rate-limit the packet is done by the control-plane qos policy/class-map config. Permit/deny actions specified in acl is not effective when used inside copy policy.

If you enable stats on the copp acl and even if you use the same acl inside copp class-map, **show ip access <acl>** wouldn't reflect this due to the reason above. Essentially an acl used inside a copp qos policy is programmed as policy type – QoS. If you want to see the packets hitting the copp control-plane qos policy, this command can be used:

```
show system internal access-list input entries detail module <x> | b CoPP
```

Fragments Handling

Default programming model creates parallel non-first fragment entry in hardware for each ACE. This entry matches same source/destination IP addresses and protocol as original ACE, but with no L4 port information and matching on non-initial fragments.

> **Note**: Fragment entries for L3 ACEs not programmed on non-XL forwarding engines.

Default fragment handling results in 2X CL TCAM utilization. Configuration knob provided to permit or deny ALL non-initial fragments:

```
fragments {permit-all | deny-all}
```

Optimizes CL TCAM utilization – consumes a single CL TCAM entry for entire ACL (versus one entry per L4 ACE)

ACE Expansion Threshold

ACEs using L4 operators - range, gt, lt, neq. There are two ways for software to handle L4 operators:

- Allocate L4op (hardware resource) and program LOU register (another hardware resource)
- Expand the ACE into multiple eq entries (i.e., CL TCAM entries)

Global command **hardware access-list lou resource threshold** controls when option 1 vs option 2 occurs for an ACE. The expansion threshold controls when expansion occurs, the default threshold is 5. If an ACE can be expanded into <=5 CL TCAM entries, no L4op allocated.

Pros/cons:

- Expansion results in more TCAM entry consumption
- L4op/LOU usage limited by L4ops per label (10) and LOU registers (208)

Resource Pool

A.K.A. Bank Chain. Explained in detail in

Related information

Cisco BUG ID CSCtd24377 AD-XL: Spanslogic algorithm constraints

Cisco BUG ID CSCuc98853 ACLQOS is not honoring fragment deny-all/permit-all for route-map for XL

Tim Stevens's classification slides