This article describes how to troubleshoot memory issues that may occur when configuring and using Cisco NX-OS.

Guide Contents
Troubleshooting Overview
Troubleshooting Installs, Upgrades, and Reboots
Troubleshooting Licensing
Troubleshooting VDCs
Troubleshooting CFS
Troubleshooting Ports
Troubleshooting vPCs
Troubleshooting VLANs
Troubleshooting STP
Troubleshooting Routing
Troubleshooting Unicast Traffic
Troubleshooting WCCP
Troubleshooting Memory (this section)
Troubleshooting Packet Flow Issues
Troubleshooting FCoE
Before Contacting Technical Support
Troubleshooting Tools and Methodology

Contents

Overview

Dynamic random access memory (DRAM) is a limited resource on all platforms and must be controlled/monitored to ensure utilization is kept in check.

Cisco NX-OS uses memory in the following three ways:

Page cache

When you access files from persistent storage (CompactFlash), the kernel reads the data into the page cache, which means that when you access the data in the future, you can avoid the slow access times that are associated with disk storage. Cached pages can be released by the kernel if the memory is needed by other processes.

Some file systems (tmpfs) exist purely in the page cache (for example, /dev/sh, /var/sysmgr, /var/tmp), which means that there is no persistent storage of this data and that when the data is

removed from the page cache, it cannot be recovered. tmpfs-cached files release page-cached pages only when they are deleted.

Kernel

> The kernel needs memory to store its own text, data, and Kernel Loadable Modules (KLMs). KLMs are pieces of code that are loaded into the kernel (as opposed to being a separate user process). An example of kernel memory usage is when an inband port driver allocates memory to receive packets.

User processes

> This memory is used by Cisco NX-OS/Linux processes that are not integrated in the kernel (such as text, stack, heap, and so on).

When you are troubleshooting high memory utilization, you must first determine what type of utilization is high (process, page cache, or kernel). Once you have identified the type of utilization, you can use additional troubleshooting commands to help you figure out which component is causing this behavior.

General/High Level Assessment of Platform Memory Utilization

You can assess the overall level of memory utilization on the platform by using two basic CLI commands: **show system resources** and **show processes memory**.

> 🖊**Note**: From these command outputs, you might be able to tell that platform utilization is higher than normal/expected, but you will not be able to tell what type of memory usage is high.

The show system resources command displays platform memory statistics (not per VDC).

```
N7K# show system resources
Load average: 1 minute: 0.43 5 minutes: 0.30 15 minutes: 0.28
Processes : 884 total, 1 running
CPU states : 2.0% user, 1.5% kernel, 96.5% idle
Memory usage: 4135780K total, 3423272K used, 712508K free
0K buffers, 1739356K cache
```

> 🖊**Note**: This output is derived from the Linux memory statistics in /proc/meminfo.

> **total** - The amount of physical RAM on the platform

> **free** - The amount of unused or available memory

> **used** - The amount of allocated (permanent) and cached (temporary) memory

> The cache and buffers are not relevant to customer monitoring.

This information provides a general representation of the platform utilization only. You need more information to troubleshoot why memory utilization is high.

The **show process memory** command displays the memory allocation per process for the current VDC (the output will contain non-VDC global processes also).

```
N7K# show processes memory
PID MemAlloc MemLimit MemUsed StackBase/Ptr Process
----- -------- ---------- ---------- ----------------- ----------------
4662 52756480 562929945 150167552 bfffdf00/bfffd970 netstack
```

While this output is more detailed, it is only useful for verifying process-level memory allocation within a specific VDC.

Detailed Assessment of Platform Memory Utilization

Use the **show system internal kernel** command or the **show system internal memory-alerts-log** command for a more detailed representation of memory utilization in Cisco NX-OS.

```
N7K# show system internal kernel meminfo
MemTotal: 4135780 kB
MemFree: 578032 kB
Buffers: 5312 kB
Cached: 1926296 kB
RAMCached: 1803020 kB
Allowed: 1033945 Pages
Free: 144508 Pages
Available: 177993 Pages
SwapCached: 0 kB
Active: 1739400 kB
Inactive: 1637756 kB
HighTotal: 3287760 kB
HighFree: 640 kB
LowTotal: 848020 kB
LowFree: 577392 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
Mapped: 1903768 kB
Slab: 85392 kB
CommitLimit: 2067888 kB
Committed_AS: 3479912 kB
PageTables: 20860 kB
VmallocTotal: 131064 kB
VmallocUsed: 128216 kB
VmallocChunk: 2772 kB
```

In the output above, the most important fields are as follows:

> MemTotal (kB)- Total amount of memory in the system (4 GB in the Cisco Nexus 7000 Series Sup1)
> Cached (kB) - Amount of memory used by the page cache (includes files in tmpfs mounts and data cached from persistent storage /bootflash)
> RamCached (kB) - Amount of memory used by the page cache that cannot be released (data not backed by persistent storage)
> Available (Pages) - Amount of free memory in pages (includes the space that could be made available in the page cache and free lists)
> Mapped (Pages) - Memory mapped into page tables (data being used by nonkernel processes)
> Slab (Pages) - Rough indication of kernel memory consumption

Note: One page of memory is equivalent to 4 kB of memory.

The **show system internal kernel memory global** command displays the memory usage for the page cache and kernel/process memory.

```
N7K# show system internal kernel memory global
Total memory in system : 4129600KB
Total Free memory : 1345232KB
Total memory in use : 2784368KB
Kernel/App memory : 1759856KB
```

```
RAM FS memory : 1018616KB
```

📝**Note**: In Cisco NX-OS, the Linux kernel monitors the percentage of memory that is used (relative to the
total RAM present) and platform manager generates alerts as utilization passes default or
configured thresholds. If an alert has occurred, it is useful to review the logs captured by the
platform manager against the current utilization. Additional information about this monitoring is
included later in this article.

By reviewing the output of these commands, you can determine if the utilization is high as a result of the
page cache, processes holding memory, or kernel.

For more detailed information, see the following topics:

> Page Cache

> Kernel

> User Processes

Page Cache

If Cached or RAMCached is high, you should check the file system utilization and determine what kind of
files are filling the page cache.

> The **show system internal flash** command displays the file system utilization (the output is similar to df
> -hT included in the memory alerts log).

```
N7K# show system internal flash
Mount-on                 1K-blocks     Used   Available    Use%  Filesystem
/                           409600     43008      367616     11  /dev/root
/proc                            0         0           0      0  proc
/sys                             0         0           0      0  none
/isan                       409600    269312      140288     66  none
/var/tmp                    307200       876      306324      1  none
/var/sysmgr                1048576    999424       49152     96   none
/var/sysmgr/ftp             307200     24576      282624      8  none
/dev/shm                   1048576    412672      635904     40  none
/volatile                   204800         0      204800      0  none
/debug                        2048        16        2032      1  none
/dev/mqueue                      0         0           0      0  none
/mnt/cfg/0                   76099      5674       66496      8  /dev/hda5
/mnt/cfg/1                   75605      5674       66027      8  /dev/hda6
/bootflash                 1796768    629784     1075712     37  /dev/hda3
/var/sysmgr/startup-cfg     409600     27536      382064      7  none
/mnt/plog                    56192      3064       53128      6  /dev/mtdblock2
/dev/pts                         0         0           0      0  devpts
/mnt/pss                     38554      6682       29882     19  /dev/hda4
/slot0                     2026608         4     2026604      1  /dev/hdc1
/logflash                  7997912    219408     7372232      3  /dev/hde1
/bootflash_sup-remote      1767480   1121784      555912     67  127.1.1.6:/mnt/bootflash/
/logflash_sup-remote       7953616    554976     6994608      8  127.1.1.6:/mnt/logflash/
```

Note: When reviewing this output, the value of none in the Filesystem column means that it is a tmpfs type.

In this example, utilization is high because the /var/sysmgr (or subfolders) is using a lot of space. /var/sysmgr is a tmpfs mount, which means that the files exist in RAM only. You need to determine what type of files are filling the partition and where they came from (cores/debugs/etc). Deleting the files will reduce utilization, but you should try to determine what type of files are taking up the space and what process left them in tmpfs.

In Cisco NX-OS release 4.2(4) and later releases, use the following commands to display and delete the problem files from the CLI:

The **show system internal dir** *full directory path* command lists all the files and sizes for the specified path (hidden command).

The **filesys delete** *full file path* command deletes a specific file (hidden command).
Note: Use caution when using this command. You cannot recover a deleted file.

Note: If you are running a Cisco NX-OS release prior to Cisco NX-OS release 4.2(4), you should contact your customer support representative.

You can also use the **show hardware internal proc-info pcacheinfo** command to determine how much space each file system is using in the page cache (Cached). The command output may help you determine which persistent file systems are using the page cache and how much memory they are using.

Kernel

Kernel issues are less common, but you can determine the problem by reviewing the slab utilization in the **show system internal meminfo** command output. Generally, kernel troubleshooting requires Cisco customer support assistance to isolate why the utilization is increasing.

If slab memory usage grows over time, use the following commands to gather more information:

The **show system internal kernel malloc-stats** command displays all the currently loaded KLMs, malloc, and free counts.

```
N7K# show system internal kernel malloc-stats
Kernel Module Memory Tracking
-------------------------------------------------------------
Module kmalloc kcalloc kfree diff
klm_usd 00318846 00000000 00318825 00000021
klm_eobcmon 08366981 00000000 08366981 00000000
klm_utaker 00001306 00000000 00001306 00000000
klm_sysmgr-hb 00000054 00000000 00000049 00000005
klm_idehs 00000001 00000000 00000000 00000001
klm_sup_ctrl_mc 00209580 00000000 00209580 00000000
klm_sup_config 00000003 00000000 00000000 00000003
klm_mts 03357731 00000000 03344979 00012752
klm_kadb 00000368 00000000 00000099 00000269
klm_aipc 00850300 00000000 00850272 00000028
klm_pss 04091048 00000000 04041260 00049788
klm_rwsem 00000001 00000000 00000000 00000001
klm_vdc 00000126 00000000 00000000 00000126
klm_modlock 00000016 00000000 00000016 00000000
klm_e1000 00000024 00000000 00000006 00000018
```

```
klm_dc_sprom 00000123 00000000 00000123 00000000
klm_sdwrap 00000024 00000000 00000000 00000024
klm_obfl 00000050 00000000 00000047 00000003
```

By comparing several iterations of this command, you can determine if some KLMs are allocating a lot of memory but are not freeing/returning the memory back (the differential value will be very large compared to normal).

The **show system internal kernel skb-stats** command displays the consumption of SKBs (buffers used by KLMs to send and receive packets).

```
N7K# show system internal kernel skb-stats
Kernel Module skbuff Tracking
-----------------------------------------------------------
Module alloc free diff
klm_shreth 00028632 00028625 00000007
klm_eobcmon 02798915 02798829 00000086
klm_mts 00420053 00420047 00000006
klm_aipc 00373467 00373450 00000017
klm_e1000 16055660 16051210 00004450
```

Compare the output of several iterations of this command to see if the differential value is growing or very high.

The **show hardware internal proc-info slabinfo** command dumps all of the slab information (memory structure used for kernel management). The output can be large.

User Processes

If page cache and kernel issues have been ruled out, utilization might be high as a result of some user processes taking up too much memory or a high number of running processes (due to the number of VDCs/features enabled).

Note: Cisco NX-OS defines memory limits for most processes (rlimit). If this rlimit is exceeded, sysmgr will crash the process and a core file is usually generated. Processes close to their rlimit may not have a large impact on platform utilization but could still become an issue if a crash occurs.

Figuring Out Which Process is Using a Lot of Memory

The following commands can help you identify if a specific process is using a lot of memory:

The **show process memory** command displays the memory allocation per process for the current VDC (the output will contain non-VDC global processes also).

```
N7K# show processes memory
PID MemAlloc MemLimit MemUsed StackBase/Ptr Process
----- -------- ---------- ---------- ---------------- ----------------
4662 52756480 562929945 150167552 bfffdf00/bfffd970 netstack
```

Note: The output of the **show process memory** command might not provide a completely accurate picture of the current utilization (allocated does not mean in use). This command is useful for determining if a process is approaching its rlimit.

To determine how much memory the processes are really using, you should check the Resident Set Size (RSS). This value will give you a rough indication of the amount of memory (in KB) that is being consumed by the processes. You can gather this information by using the following command:

The **show system internal processes memory** command displays the process information in the memory alerts log (if the event occurred).

```
N7K# show system internal processes memory
PID TTY STAT TIME MAJFLT TRS RSS VSZ %MEM COMMAND
4727 ? Ss 00:00:00 0 1549 123248 132832 2.9 /isan/bin/pixm
4728 ? Ssl 00:00:00 0 408 78388 143104 1.8 /isan/bin/routing-sw/mrib -m 4
6662 ? Ssl 00:00:05 0 2762 64024 144396 1.5 /isan/bin/routing-sw/netstack /isan/etc/routing-sw/pm
4538 ? Ssl 00:00:00 0 2762 60448 211664 1.4 /isan/bin/routing-sw/netstack /isan/etc/routing-sw/pm
5865 ? Ssl 00:00:01 0 2762 60416 113320 1.4 /isan/bin/routing-sw/netstack /isan/etc/routing-sw/pm
6395 ? Ssl 00:00:00 0 2762 52008 105552 1.2 /isan/bin/routing-sw/netstack /isan/etc/routing-sw/pm
4271 ? Ssl 00:00:00 0 609 49812 61420 1.2 /isan/bin/routing-sw/urib
7879 ? Ssl 00:00:00 0 1909 44800 90508 1.0 /isan/bin/routing-sw/bgp -t 64000
5696 ? Ssl 00:00:17 0 337 44696 55252 1.0 /isan/bin/routing-sw/clis -cli /isan/etc/routing-sw/cli
5333 ? Ssl 00:00:14 0 337 44652 55208 1.0 /isan/bin/routing-sw/clis -cli /isan/etc/routing-sw/cli
4182 ? Ssl 00:00:15 0 337 44648 55204 1.0 /isan/bin/routing-sw/clis -cli /isan/etc/routing-sw/cli
6076 ? Ssl 00:00:14 0 337 44624 55284 1.0 /isan/bin/routing-sw/clis -cli /isan/etc/routing-sw/cli
6825 ? Ssl 00:00:00 0 1402 44576 84020 1.0 /isan/bin/routing-sw/pim -t
4268 ? Ssl 00:00:00 0 363 27132 38896 0.6 /isan/bin/routing-sw/u6rib
4732 ? Ssl 00:00:00 0 404 25220 65360 0.6 /isan/bin/routing-sw/m6rib
4726 ? S<s 00:00:00 0 144 25208 30188 0.6 /isan/bin/pixmc
remaining output omitted
```

If you see an increase in the utilization for a specific process over time, you should gather additional information about the process utilization.

Figuring Out How a Specific Process is Using Memory

If you have determined that a process is using more memory than expected, it is helpful to investigate how the memory is being used by the process.

The **show system internal sysmgr service pid** *<PID in decimal>* command dumps the service information running the specified PID.

```
N7K# show system internal sysmgr service pid 4727
Service "pixm" ("pixm", 109):
UUID = 0x133, PID = 4727, SAP = 176
State: SRV_STATE_HANDSHAKED (entered at time Fri Nov 12 01:42:01 2010).
Restart count: 1
Time of last restart: Fri Nov 12 01:41:11 2010.
The service never crashed since the last reboot.
Tag = N/A
Plugin ID: 1
```

Convert the UUID from the above output to decimal and use in the next command.

🖉**Note**: If troubleshooting in lab, you can use NX-OS hex/dec conversion using following hidden commands :

> **hex** *<dec to convert>*
> **dec** *<hex to convert>*

The **show system internal kernel memory uuid** *<UUID in decimal>* command displays the detailed process memory usage including its libraries for a specific UUID in the system (convert UUID from the sysmgr service output).

```
N7K# show system internal kernel memory uuid 307
Note: output values in KiloBytes
Name rss shrd drt map heap ro dat bss stk misc
```

Figuring Out Which Process is Using a Lot of Memory   7

```
---- --- ---- --- --- ---- -- --- --- --- ----
/isan/bin/pixm 7816 5052 2764 1 0 0 0 0 52 0
/isan/plugin/1/isan/bin/pixm 115472 0 115472 0 109176 752 28 6268 0 24
/lib/ld-2.3.3.so 84 76 8 2 0 76 0 0 0 8
/usr/lib/libz.so.1.2.1.1 16 12 4 1 0 12 4 0 0 0
/usr/lib/libstdc++.so.6.0.3 296 272 24 1 0 272 20 4 0 0
/lib/libgcc_s.so.1 1824 12 1812 1 1808 12 4 0 0 0
/isan/plugin/1/isan/lib/libtmifdb.so.0 12 8 4 1 0 8 4 0 0 0
/isan/plugin/0/isan/lib/libtmifdb_stub 12 8 4 1 0 8 4 0 0 0
/dev/mts0 0 0 0 1 0 0 0 0 0 0
/isan/plugin/1/isan/lib/libpcm_sdb.so. 16 12 4 1 0 12 4 0 0 0
/isan/plugin/1/isan/lib/libethpm.so.0. 76 60 16 1 0 60 16 0 0 0
/isan/plugin/1/isan/lib/libsviifdb.so. 20 4 16 1 12 4 4 0 0 0
/usr/lib/libcrypto.so.0.9.7 272 192 80 1 0 192 76 4 0 0
/isan/plugin/0/isan/lib/libeureka_hash 8 4 4 1 0 4 4 0 0 0
remaining output omitted
```

This output helps you to determine if a process is holding memory in a specific library and can assist with
memory leak identification.

> The **show system internal** *<service>* **mem-stats detail** command displays the detailed memory
> utilization including the libraries for a specific service.

```
N7K# show system internal pixm mem-stats detail
Private Mem stats for UUID : Malloc track Library(103) Max types: 5
------------------------------------------------------------------------------
TYPE NAME ALLOCS BYTES
CURR MAX CURR MAX
2 MT_MEM_mtrack_hdl 32 33 16448 16596
3 MT_MEM_mtrack_info 424 531 6784 8496
4 MT_MEM_mtrack_lib_name 636 743 30054 35112
------------------------------------------------------------------------------
Total bytes: 53286 (52k)
------------------------------------------------------------------------------
Private Mem stats for UUID : Non mtrack users(0) Max types: 105
------------------------------------------------------------------------------
TYPE NAME ALLOCS BYTES
CURR MAX CURR MAX
4 [r-xp]/isan/plugin/0/isan/lib/libacfg.s 0 4 0 51337
9 [r-xp]/isan/plugin/0/isan/lib/libavl.so 79 81 1568 1608
25 [r-xp]/isan/plugin/0/isan/lib/libfsrv.s 6 6 34 34
32 [r-xp]/isan/plugin/0/isan/lib/libindxob 6 6 456 456
46 [r-xp]/isan/plugin/0/isan/lib/libmpmts. 0 2 0 100
48 [r-xp]/isan/plugin/0/isan/lib/libmts.so 7 10 816 972
51 [r-xp]/isan/plugin/0/isan/lib/libpfm_in 0 1 0 3490
53 [r-xp]/isan/plugin/0/isan/lib/libpss.so 169 196 27316 114880
57 [r-xp]/isan/plugin/0/isan/lib/libsdb.so 140 140 5632 5632
62 [r-xp]/isan/plugin/0/isan/lib/libsrg.so 0 1 0 3480
68 [r-xp]/isan/plugin/0/isan/lib/libsysmgr 3 3 2094 2094
79 [r-xp]/isan/plugin/0/isan/lib/libutils. 61 69 512 55389
84 [r-xp]/isan/plugin/1/isan/bin/pixm 238 240 532920 533440
88 [r-xp]/isan/plugin/1/isan/lib/libpixm.s 0 1 0 48
92 [r-xp]/lib/ld-2.3.3.so 21 26 3483 4233
94 [r-xp]/lib/tls/libc-2.3.3.so 286 287 8163 8490
100 [r-xp]/usr/lib/libglib-2.0.so.0.600.1 12 19 6328 6800
------------------------------------------------------------------------------
Total bytes: 589322 (575k)
remaining output omitted
```

These outputs are usually requested by the Cisco customer support representative when investigating a potential memory leak in a process or its libraries.

Built-in Platform Memory Monitoring

Cisco NX-OS has built-in kernel monitoring of memory usage to help avoid system hangs, process crashes, and other undesirable behavior. The platform manager periodically checks the memory utilization (relative to the total RAM present) and automatically generates an alert event if the utilization passes the configured threshold values. When an alert level is reached, the kernel attempts to free memory by releasing pages that are no longer needed (for example, the page cache of persistent files that are no longer being accessed), or if critical levels are reached, the kernel will kill the highest utilization process. Other Cisco NX-OS components have introduced memory alert handling, such as BGP's graceful low memory handling, that allow processes to adjust their behavior to keep memory utilization under control.

Note: While Cisco NX-OS implements VDCs, it is important to remember that a specific VDC's memory utilization is not limited. Platform memory issues will impact all configured VDCs.

Memory Thresholds

Prior to Release 4.2(4), the default memory alert thresholds were as follows:

- 70% MINOR
- 80% SEVERE
- 90% CRITICAL

From Release 4.2(4) and later releases, the memory alert thresholds were changed to the following:

- 85% MINOR
- 90% SEVERE
- 95% CRITICAL

This change was introduced in part due to baseline memory requirements when many features/VDCs are deployed.

The thresholds are configurable, using the following command:

system memory-thresholds minor *percentage* **severe** *percentage* **critical** *percentage*

The **show system internal memory-status** command allows you to check the current memory alert status.

```
N7K# show system internal memory-status
MemStatus: OK
```

Memory Alerts

If a memory threshold has been passed (OK -> MINOR, MINOR -> SEVERE, SEVERE -> CRITICAL), the Cisco NX-OS platform manager will capture a snapshot of memory utilization and log an alert to SYSLOG (as of Release 4.2(4), default VDC only). This snapshot is useful in determining why memory utilization is

high (process, page cache, or kernel). The log is generated in the Linux root path (/) and copy is moved to OBFL (/mnt/plog) if possible. This log is very useful for determining if memory utilization is high due to the memory that was consumed by the page cache, kernel, or Cisco NX-OS user processes.

The **show system internal memory-alerts-log** command displays the memory alerts log.

The memory alerts log consists of the following outputs:

Command	Description
cat /proc/memory_events	Provides a log of timestamps when memory alerts occurred.
cat /proc/meminfo	Shows the overall memory statistics including the total RAM, memory consumed by the page cache, slabs (kernel heap), mapped memory, available free memory, and so on.
cat /proc/memtrack	Displays the allocation/deallocation counts of the KLMs (Cisco NX-OS processes running in kernel memory).
df -hT	Displays file system utilization information (with type).
du --si -La /tmp	Displays file information for everything located in /tmp (symbolic link to /var/tmp).
cat /proc/memory_events	Dumped a second time to help determine if utilization changed during data gathering.
cat /proc/meminfo	Dumped a second time to help determine if utilization changed during data gathering.