Troubleshooting Memory

About Troubleshooting Memory

Dynamic random access memory (DRAM) is a limited resource on all platforms and must be controlled or 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 or 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.

Note If the *show system resources* command output shows a decline in the free memory, it may be because of Linux kernel caching. Whenever the system requires more memory, Linux kernel will release cached memory. The *show system internal kernel meminfo* command displays cached memory in the system.

The **show system resources** command displays platform memory statistics.

```
switch# show system resources
Load average:   1 minute: 0.70   5 minutes: 0.89   15 minutes: 0.88
Processes   :   805 total, 1 running
CPU states  :   7.06% user,   5.49% kernel,   87.43% idle
                  CPU0 states  :   9.67% user,   6.45% kernel,   83.87% idle
                  CPU1 states  :   10.41% user,   7.29% kernel,   82.29% idle
                  CPU2 states  :   5.20% user,   4.16% kernel,   90.62% idle
                  CPU3 states  :   5.15% user,   2.06% kernel,   92.78% idle
Memory usage:   16399900K total,   6557936K used,   9841964K free
Kernel vmalloc:   36168240K total,   18446744039385981489K free     >>>>>>>>>>>>
Kernel buffers:   10860132K Used>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>.>>>>>>>
Kernel cached :   120072K Used >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> seeing these extra
logs
Current memory status: OK
```

```
switch# 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 processes memory** command displays the memory allocation per process.

```
switch# show processes memory
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
PID  MemAlloc MemLimit  MemUsed   StackBase/Ptr     Process
---- -------- --------- --------- ----------------- ----------------
```

```
4662 52756480 562929945 150167552 bfffdf00/bfffd970 netstack
```

Detailed Assessment of Platform Memory Utilization

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

```
switch# 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.
- **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.

```
switch# 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.

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.

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).

```
switch# 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.

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).

Kernel

Kernel issues are less common, but you can determine the problem by reviewing the slab utilization in the **show system internal kernel 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.

```
switch# 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).

```
switch# 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 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 become an issue if a crash occurs.

Determining 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.

```
switch# 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 limit.

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

To determine how much memory the processes are really using, 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 **show system internal processes memory** command.

```
switch# show system internal processes memory
 PID TTY STAT TIME MAJFLT TRS RSS VSZ %MEM COMMAND
 4811 ?        Ssl  00:00:16      0    0 49772 361588  0.3 /isan/bin/routing-sw/clis
-cli /isan/etc/routing-sw/cli
 4928 ?       Ssl  00:18:41      0    0 44576 769512  0.2 /isan/bin/routing-sw/netstack
 /isan/etc/routing-sw/pm.cfg
 4897 ?        Ssl  00:00:18      0    0 42604 602216  0.2 /isan/bin/routing-sw/arp
 4791 ?        Ss   00:00:00      0    0 34384 318856  0.2 /isan/bin/pixm_vl
 4957 ?       Ssl  00:00:26      0    0 30440 592348  0.1 /isan/bin/snmpd -f -s udp:161
 udp6:161 tcp:161 tcp6:161
 5097 ?        Ssl  00:06:53      0    0 28052 941880  0.1 /isan/bin/routing-sw/pim -t

 5062 ?        Ss   00:01:00      0    0 27300 310596  0.1 /isan/bin/diag_port_lb
 5087 ?        Ssl  00:03:53      0    0 24988 992756  0.1 /isan/bin/routing-sw/bgp -t
 65001
 4792 ?        Ss   00:00:00      0    0 24080 309024  0.1 /isan/bin/pixm_gl
 5063 ?        Ss   00:00:01      0    0 21940 317440  0.1 /isan/bin/ethpm
 5044 ?        Ss   00:00:00      0    0 21700 304032  0.1 /isan/bin/eltm
 5049 ?        Ss   00:00:14      0    0 20592 306156  0.1 /isan/bin/ipqosmgr
 5042 ?        Ssl  00:00:05      0    0 20580 672640  0.1 /isan/bin/routing-sw/igmp
 5082 ?        Ssl  00:00:25      0    0 19948 914088  0.1 /isan/bin/routing-sw/mrib
-m 4
 5091 ?        Ssl  00:01:58      0    0 19192 729500  0.1 /isan/bin/routing-sw/ospfv3
 -t 8893
 5092 ?        Ssl  00:01:55      0    0 18988 861556  0.1 /isan/bin/routing-sw/ospf
-t 6464
 5083 ?        Ss   00:00:06      0    0 18876 309516  0.1 /isan/bin/mfdm
 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.

Determining 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.

```
switch# 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 May 10 01:42:01 2013).
Restart count: 1
Time of last restart: Fri May 10 01:41:11 2013.
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 you are troubleshooting in a lab, you can use Cisco NX-OS hexadecimal/decimal conversion using the following hidden commands:

- **hex**<decimal to convert>
- **dec**<hexadecimal 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).

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

 ----                         ---   ----    ----  ---    ----   --  ---   --- --- ----

 /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/mts                         0     0       0    0       1    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.

```
switch# 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                       35     35     132132     149940
   3 MT_MEM_mtrack_info                     598    866       9568      13856
   4 MT_MEM_mtrack_lib_name                 598    866      15860      22970
-----------------------------------------------------------------------------
Total bytes: 157560 (153k)
```

```
-----------------------------------------------------------------------------

Private Mem stats for UUID : Non mtrack users(0) Max types: 157
-----------------------------------------------------------------------------
TYPE NAME                                       ALLOCS                 BYTES
                                           CURR    MAX       CURR        MAX
   1 [0x41000000]ld-2.15.so                 283    283      48255      48256
   2 [0x41024000]libc-2.15.so               142    144       4979       5587
   8 [0x41241000]libglib-2.0.so.0.3200.3    500    771      10108      15588
  39 [0xf68af000]libindxobj.so                7      7        596        596
  45 [0xf68ca000]libavl.so                   73     73       1440       1440
  67 [0xf71b3000]libsdb.so                   56     58       3670      73278
  75 [0xf7313000]libmpmts.so                 35     37        280        380
  86 [0xf7441000]libutils.so                 23     28       3283       5766
  89 [0xf74bf000]libpss.so                   59     60       8564     483642
  90 [0xf750b000]libmts.so                    7      8        816        828
  92 [0xf754c000]libacfg.so                   0      4          0      51337
-----------------------------------------------------------------------------
Total bytes: 82817 (80k)
--------------------------------------------------------------------------------
 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 the Border Gateway Protocol's (BGP's) graceful low memory handling, that allows processes to adjust their behavior to keep memory utilization under control.

Memory Thresholds

When many features are deployed, baseline memory requires the following thresholds:

- MINOR
- SEVERE
- CRITICAL

Because the default thresholds are calculated on boot up depending on the DRAM size, its value varies depending on the DRAM size that is used on the platform. The thresholds are configurable using the **system memory-thresholds minor** *percentage* **severe** *percentage* **critical** *percentage* command.

Beginning with Cisco NX-OS Release 10.2(4)M, the default system memory thresholds are as follows:

Beginning with Cisco NX-OS Release 10.3(1)F, the default system memory thresholds are as follows:

- Critical: 91

- Severe: 89
- Minor: 88

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

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

Switches running scaled deployment, including scaled BGP EVPN VxLAN VNI (please see *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* for supported scale), the memory alert may be seen during Non-Disruptive ISSU as the default system memory threshold has been lowered beginning with Cisco NX-OS Release 10.3(3)F release. To avoid system reacting to critical memory alert, before upgrade configure higher value for system memory thresholds. For example: Set system memory thresholds as 90 for minor, 94 for severe, and 95 for critical.

Memory Alerts

When a memory threshold has been passed (OK -> MINOR, MINOR -> SEVERE, SEVERE -> CRITICAL), the Cisco NX-OS platform manager captures a snapshot of memory utilization and logs an alert to syslog. 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 time stamps 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.