

Troubleshooting Memory

- About Troubleshooting Memory, on page 1
- General/High Level Assessment of Platform Memory Utilization, on page 1
- User Processes, on page 3
- Built-in Platform Memory Monitoring, on page 3

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
```

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
```

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.

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 10.3(1)F, the default system memory thresholds are as follows:

- Critical: 91
- Severe: 89
- Minor: 88

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.