

Troubleshoot High Memory Usage on Compute Nodes in CVIM

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[What are HugePages?](#)

[Problem Description](#)

[Analysis](#)

[Troubleshoot](#)

Introduction

This document describes the procedure to analyze the issue related to high memory usage in Cisco Virtualized Infrastructure Manager (CVIM) compute nodes.

Prerequisites

Requirements

Cisco recommends you have knowledge of memory management and HugePages in Linux.

What are HugePages?

Enabling HugePages allows the operating system to support memory pages larger than the default (usually 4 KB). The use of very large page sizes can enhance system performance by reducing the system resources needed to access page table entries. Consequently, HugePages are typically employed to mitigate memory latency.

Problem Description

High memory usage alerts on CVIM compute nodes while CVIM has not triggered any alerts. Alerts related to memory utilization could be via a third-party monitoring tool or monitoring dashboard.

Analysis

It is observed that high memory utilization in the OS as per the `free` and `sar` command output in Linux.

```
[root@cvim-computex ~]# free -m
              total    used    free   shared  buff/cache  available
Mem:          385410  365882   7602     3621     11925     8411
Swap:           2047         0    2047
```

```
[root@cvim-computex ~]# sar -r
Linux 4.18.0-193.81.1.el8_2.x86_64 (pod1-compute4.mx2) 08/24/2023 _x86_64_ (112 CPU)
```

Time	kbmemfree	kbavail	kbmemused	%memused	kbbuffers	kbcached	kbcommit	%commit	kbactive	kbinact
12:00:46 AM	7493576	7871200	387166528	98.10	4240	9334356	12893752	3.25	4891940	6325076
12:10:34 AM	7503208	7883396	387156896	98.10	4240	9337364	12872708	3.24	4885008	6328096
12:20:11 AM	7485648	7869540	387174456	98.10	4240	9340556	12902748	3.25	4892948	6331276
12:30:34 AM	7494396	7880940	387165708	98.10	4240	9343636	12866964	3.24	4886908	6334364
12:40:46 AM	7479616	7869772	387180488	98.10	4240	9346720	12905156	3.25	4892408	6337444
01:00:46 AM	7490304	7883016	387169800	98.10	4240	9349832	12860152	3.24	4885308	6340500
01:10:34 AM	7472248	7868672	387187856	98.11	4240	9352836	12896932	3.25	4892604	6343556
01:20:46 AM	7484308	7883276	387175796	98.10	4240	9355948	12867972	3.24	4885172	6346676
01:30:34 AM	7475092	7869596	387185012	98.11	4240	9350840	12904328	3.25	4892448	6341556
01:40:46 AM	7485436	7882508	387174668	98.10	4240	9353932	12864252	3.24	4885148	6344660
01:50:34 AM	7468840	7869520	387191264	98.11	4240	9357036	12907464	3.25	4893552	6347752
02:00:46 AM	7479076	7882428	387181028	98.10	4240	9360124	12861892	3.24	4886044	6350844

Use the `ps` command to identify the processes with the highest memory usage.

```
[root@cvim-computex ~]# ps -aux --sort -rss
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      328199 1207  0.2 541893584 ?    RLl  Mar12 2948779:31 /usr/bin/vpp -c /etc/vpp/vpp.conf
root      1829  0.0  0.0 379024 227692 ?    Ss   Mar12 14:21 /usr/lib/systemd/systemd-journald
```

Verify the container memory usage by checking the statistics using the `podman` or `docker` commands.

```
[root@cvim-computex ~]# podman stats
ID          NAME                CPU % MEM USAGE / LIMIT MEM % NET IO  BLOCK IO          PIDS
2f8fdc4b63a4 fluentd_31902      --    301.2MB / 404.1GB 0.07% -- / -- 9.265MB / 89.68GB 75
34d806a30733 novalibvirt_31902 --    42.16MB / 404.1GB 0.01% -- / -- 589.8kB / 22.13MB 44
48292d2fa956 novassh_31902     --    5.882MB / 404.1GB 0.00% -- / -- 475.1kB / 167.3MB 2
7b2ce84e86b3 novacompute_31902 --    231.8MB / 404.1GB 0.06% -- / -- 761.9kB / 2.43GB 49
89c01c14ef3f neutron_vpp_31902 --    1.209GB / 404.1GB 0.30% -- / -- 0B / 7.66MB 35
```

Based on the provided output, it appears that no processes are exhibiting high memory usage. Additionally, the containers seem to be utilizing a low amount of memory.

The `free` command still shows high memory usage.

```
root@cvim-computex ~]# free -m
              total        used         free       shared    buff/cache   available
Mem:           385410      366751         7310         3496         11348      7696
Swap:            2047           5         2042
[root@cvim-computex ~]#
```

Troubleshoot

To comprehend this memory utilization, knowledge of HugePage memory is essential.

If the pod is enabled with HugePages, care must be taken to use the right flavor, to ensure that the system memory is not used to launch the VMs. The usage of system memory for VMs can lead to CVIM instability, as both the workload and the infrastructure are competing for the resources reserved for the infrastructure.

Check the HugePages:

```
[root@cvim-computex ~]# tail /sys/devices/system/node/node0/hugepages/hugepages-2048kB/nr_hugepages
90001
[root@cvim-computex ~]# tail /sys/devices/system/node/node0/hugepages/hugepages-1048576kB/nr_hugepages
0
[root@cvim-computex ~]# tail /sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr_hugepages
90000
[root@cvim-computex ~]# tail /sys/devices/system/node/node1/hugepages/hugepages-1048576kB/nr_hugepages
0
[root@cvim-computex ~]#
```

`nr_hugepages` is the total number of HugePages.

$(90001 + 90000) \times 2M = 360GB$ is reserved for HugePage.

Also, note that 5% memory of total physical memory is reserved for normal memory pages (4KB) for OS usage even if 100% HugePage is configured.

$385GB$ (free total) - $360GB$ (reserved for HugePage) = $25GB$ is reserved for Normal Pages.

So, high memory utilization, as observed in the `sar` and `free` commands is expected.

Use the mentioned command to check the actual memory usage.

<#root>

```
[root@mgmt-node ~]# ip -br -4 a s br_api
br_api UP 10.x.x.x/24
```

```
[root@mgmt-node ~]# curl -sS -g -u admin:password --cacert /var/www/mercury/mercury-ca.crt https://10.x.x.x
```

sample output:

```
{
"status": "
success
",
"data": {
"resultType": "vector",
"result": [
{
"metric": {
"host": "cvim-computex",
"instance": "10.x.x.x:9273",
```

```
"job": "telegraf",
"node_type": "compute"
    },
"value": [
1693479719.383,
"
76.16486394450624
" --> Actual available memory percentage.
    ]
    },
    {
"metric": {
"host": "cvm-compute",
"instance": "10.x.x.x:9273",
"job": "telegraf",
"node_type": "compute"
    },
"value": [
1693479719.383,
"76.63431887455388"
```

CVIM triggers an alert only when the available memory is less than 10%.

Alert Name - mem_available_percent

There is less than 10% of available system memory. Regular 4K pages memory is used by both the system and OpenStack infrastructure services and does not include huge pages. This alert can indicate either an insufficient amount of RAM or abnormal memory usage by the system or infrastructure.