

排除CVIM中计算节点上的高内存使用率故障

目录

[简介](#)

[先决条件](#)

[要求](#)

[什么是HugePages?](#)

[问题说明](#)

[分析](#)

[故障排除](#)

简介

本文档介绍在思科虚拟化基础设施管理器(CVIM)计算节点中分析高内存使用率相关问题的过程。

先决条件

要求

思科建议您了解内存管理和Linux中的HugePages。

什么是HugePages?

启用HugePages允许操作系统支持大于默认值 (通常为4 KB) 的内存页。使用非常大的页面大小可以通过减少访问页面表条目所需的系统资源来增强系统性能。因此，通常使用HugePages来缓解内存延迟。

问题说明

CVIM计算节点上的高内存使用率警报，而CVIM未触发任何警报。与内存利用率相关的警报可通过第三方监控工具或监控控制面板发出。

分析

据观察，OS中的高内存利用率 `free` 和 `sar` 命令输出。

```
[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	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	

请使用 `ps` 命令来识别内存使用率最高的进程。

```
[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
```

通过使用检查统计信息，验证容器内存使用情况 `podman` 或 `docker` 命令。

```
[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
```

根据提供的输出，似乎没有进程显示高内存使用率。此外，这些容器似乎正在利用少量的内存。

此 `free` 命令仍显示高内存使用率。

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

故障排除

要理解这种内存利用率，必须了解HugePage内存。

如果Pod是通过HugePages启用的，必须注意使用正确的风格，以确保系统内存不用于启动VM。VM使用系统内存可能会导致CVIM不稳定，因为工作负载和基础设施都在争夺为基础设施预留的资源。

检查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` 是HugePages的总数。

$(90001 + 90000) \times 2M = 360GB$ 预留给HugePage。

此外，请注意，总物理内存的5%保留给普通内存页(4KB)以供操作系统使用，即使已配置100%的HugePage也是如此。

$385GB$ (免费总计) — $360GB$ (为HugePage保留) = $25GB$ 为普通页保留。

因此，正如在中观察到的那样，`sar` 和 `free` 命令。

使用上述命令检查实际内存使用情况。

<#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
```

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": "cvim-computey",
"instance": "10.x.x.x:9273",
"job": "telegraf",
"node_type": "compute"
},
"value": [
1693479719.383,
"76.63431887455388"

```

仅当可用内存小于10%时，CVIM才会触发警报。

警报名称 — mem_available_percent

可用的系统内存不足10%。系统和OpenStack基础设施服务都使用常规的4K页内存，不包含超大页。此警报可能表示RAM不足或系统或基础设施使用内存异常。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。