linux 中 free 命令与 top 命令显示虚拟机内存已使用状态不一致问题

一台长期空闲的 Vmware 上的虚拟机，发现 free 命令返回已使用内存远远大于 top 命令中所有进程使用内存之和，现象如下图所示：

```
[root@cdp1 ~]# vmware-toolbox-cmd stat balloon
20316 MB
[root@cdp1 ~]# free -h
              total        used        free      shared  buff/cache   available
Mem:            30G         20G        6.4G         32M        3.9G        9.9G
Swap:           15G          0B         15G
[root@cdp1 ~]# top
top - 01:47:20 up 90 days,  3:06,  1 user,  load average: 0.05, 0.03, 0.05
Tasks: 128 total,   2 running, 126 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.4 us,  0.5 sy,  0.0 ni, 99.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 32006264 total,  6719808 free, 21240236 used,  4046220 buff/cache
KiB Swap: 16121852 total, 16121852 free,        0 used. 10349668 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
  966 root      20   0  574288  17464   6136 S   0.0  0.1  22:39.90 tuned
  734 polkitd   20   0  612240  10064   4656 S   0.0  0.0   0:09.74 polkitd
  535 root      20   0   39060   9228   8904 S   0.0  0.0   0:45.63 systemd-journal
  745 root      20   0  474324   8640   6648 S   0.0  0.0 165:42.75 NetworkManager
  965 root      20   0  240976   7072   5608 S   0.0  0.0  11:31.61 rsyslogd
 9021 root      20   0  162120   6756   5188 S   0.0  0.0   0:03.56 sshd
 9023 root      20   0  161804   6444   5056 S   0.0  0.0   0:00.35 sshd
  735 root      20   0  168148   5048   3636 S   0.0  0.0   0:01.59 VGAuthService
  737 root      20   0  275072   4944   3672 S   0.5  0.0 229:07.79 vmtoolsd
  961 root      20   0  113000   4364   3332 S   0.0  0.0   0:01.27 sshd
    1 root      20   0  191000   4012   2588 S   0.0  0.0   1:44.22 systemd
 9043 root      20   0   72348   2864   2104 S   0.0  0.0   0:00.16 sftp-server
  572 root      20   0   46112   2516   1332 S   0.0  0.0   0:01.85 systemd-udevd
  739 dbus      20   0   58224   2472   1820 S   0.0  0.0   0:24.92 dbus-daemon
14808 root      20   0  162108   2328   1592 R   0.0  0.0   0:00.05 top
 9025 root      20   0  115680   2212   1668 S   0.0  0.0   0:00.49 bash
  738 root      20   0   26384   1784   1480 S   0.0  0.0   0:40.13 systemd-logind
 9050 root      20   0  113420   1720   1336 S   3.6  0.0   0:04.51 bash
  746 root      20   0  126388   1592    968 S   0.0  0.0   0:18.93 crond
  573 root      20   0  272304   1440    976 S   0.0  0.0   0:00.01 lvmetad
  732 root      20   0   21688   1336   1012 S   0.5  0.0  10:28.49 irqbalance
  970 root      20   0  110208    864    732 S   0.0  0.0   0:00.03 agetty
  708 root      16  -4   55532    860    456 S   0.0  0.0   0:18.01 auditd
14826 root      20   0  108056    360    284 S   0.0  0.0   0:00.00 sleep
    2 root      20   0       0      0      0 S   0.0  0.0   0:02.67 kthreadd
    4 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S   0.0  0.0   0:04.50 ksoftirqd/0
    7 root      rt   0       0      0      0 S   0.0  0.0   0:08.95 migration/0
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
```

调查发现，原来是 VMware 的 virtual memory ballooning 技术导致宿主机偷了虚拟机内存导致的。可以使用 VMware 工具命令查看具体占用情况，命令为：vmware-toolbox-cmd stat balloon

High memory usage but no process is using it

Asked 7 years, 6 months ago Modified 7 years, 6 months ago Viewed 63k times

I run `free -m` on a debian VM running on Hyper-V:

```
             total       used       free     shared    buffers     cached
Mem:         10017       9475        541        147         34        909
-/+ buffers/cache:       8531       1485
Swap:         1905          0       1905
```

So out of my 10GB of memory, 8.5GB is in use and only 1500MB is free (excluding cache).

But I struggle to find what is using the memory. The output of `ps aux | awk '{sum+=$6} END {print sum / 1024}'`, which is supposed to add up the RSS utilisation is:

```
1005.2
```

In other words, my processes only use 1GB of memory but the system as a whole (excluding cache) uses 8.5GB.

What could be using the other 7.5GB?

ps: I have another server with a similar configuration that shows used mem of 1200 (free mem = 8.8GB) and the sum of RSS usage in ps is 900 which is closer to what I would expect...

EDIT

`cat /proc/meminfo` on machine 1 (low memory):

```
MemTotal:       10257656 kB
MemFree:          395840 kB
MemAvailable:    1428508 kB
```

```
Buffers:          162640 kB
Cached:          1173040 kB
SwapCached:          176 kB
Active:          1810200 kB
Inactive:         476668 kB
Active(anon):     942816 kB
Inactive(anon):   176184 kB
Active(file):     867384 kB
Inactive(file):   300484 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:       1951740 kB
SwapFree:        1951528 kB
Dirty:                16 kB
Writeback:             0 kB
AnonPages:        951016 kB
Mapped:           224388 kB
Shmem:            167820 kB
Slab:              86464 kB
SReclaimable:      67488 kB
SUnreclaim:        18976 kB
KernelStack:        6736 kB
PageTables:        13728 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     7080568 kB
Committed_AS:    1893156 kB
VmallocTotal:   34359738367 kB
VmallocUsed:       62284 kB
VmallocChunk:   34359672552 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
HugePages_Total:       0
```

`cat /proc/meminfo` on machine 2 (normal memory usage):

```
MemTotal:       12326128 kB
MemFree:         8895188 kB
MemAvailable:   10947592 kB
Buffers:          191548 kB
Cached:          2188088 kB
SwapCached:            0 kB
Active:          2890128 kB
Inactive:         350360 kB
Active(anon):    1018116 kB
Inactive(anon):    33320 kB
Active(file):    1872012 kB
Inactive(file):   317040 kB
```

```
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:       3442684 kB
SwapFree:        3442684 kB
Dirty:                44 kB
Writeback:             0 kB
AnonPages:        860880 kB
Mapped:           204680 kB
Shmem:            190588 kB
Slab:              86812 kB
SReclaimable:      64556 kB
SUnreclaim:        22256 kB
KernelStack:       10576 kB
PageTables:        11924 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     9605748 kB
Committed_AS:    1753476 kB
VmallocTotal:   34359738367 kB
VmallocUsed:       62708 kB
VmallocChunk:   34359671804 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
HugePages_Total:       0
HugePages_Free:        0
```

linux debian memory

Share Improve this question Follow

edited Jun 11, 2020 at 12:04
Community Bot
1

asked Feb 3, 2016 at 18:49

assylias
718 1 8 18

2 Answers

Sorted by: Highest score (default)

23

I understand you're using Hyper-V, but the concepts are similar. Maybe this will set you on the right track.

Your issue is likely due to virtual memory ballooning, a technique the hypervisor uses to optimize memory. See this link for a description

I observed your exact same symptoms with my VMs in vSphere. A 4G machine with nothing running on it would report 30M used by cache, but over 3G "used" in the "-/+ buffers" line.

Here's sample output from VMWare's statistics command. This shows how close to 3G is being tacked on to my "used" amount:

```
vmware-toolbox-cmd stat balloon
3264 MB
```

In my case, somewhat obviously, my balloon driver was using ~3G

I'm not sure what the similar command in Hyper-V is to get your balloon stats, but I'm sure you'll get similar results

Share Improve this answer Follow

edited Feb 11, 2016 at 1:30

answered Feb 10, 2016 at 18:07

Matt
396 1 4

Thanks - you are definitely onto something. lsmod | grep hv_ shows hv_balloon on the low memory machine but not on the other - so the balloon module is loaded on one and not the other. And the behaviour looks very much like this description. – assylias Feb 11, 2016 at 5:06

Not sure what the equivalent to `vmware-toolbox-cmd` is on Hyper V though. – assylias Feb 11, 2016 at 5:06

@assylias I know, sorry. I looked myself while writing this answer and came up empty. However, if you write a program that quickly allocates a lot of memory, that may convince the hypervisor that your VM needs the resources. Similar to disk cache eviction test case, but different root cause. – Matt Feb 11, 2016 at 13:20

You can unset the dynamic ram feature in Hyper-V to solve this issue. – Ashish Negi Aug 23, 2017 at 11:06

3 I don't really see the solution here I'm afraid. – Jamie Hutber Nov 2, 2018 at 11:51

DEFINITION

virtual memory ballooning

Ryan Lanigan

Virtual memory ballooning is a computer memory reclamation technique used by a hypervisor to allow the physical host system to retrieve unused memory from certain guest virtual machines (VMs) and share it with others. Memory ballooning allows the total amount of RAM required by guest VMs to exceed the amount of physical RAM available on the host. When the host system runs low on physical RAM resources, memory ballooning allocates it selectively to VMs.

If a VM only uses a portion of the memory that it was allocated, the ballooning technique makes it available for the host to use. For example, if all the VMs on a host are allocated 8 GB of memory, some of the VMs will only use half the allotted share. Meanwhile, one VM might need 12 GB of memory for an intensive process. Memory ballooning allows the host to borrow that unused memory and allocate it to the VMs with higher memory demand.

The guest operating system runs inside the VM, which is allocated a portion of memory. Therefore, the guest OS is unaware of the total memory available. Memory ballooning makes the guest operating system aware of the host's memory shortage.

Virtualization providers such as VMware enable memory ballooning. VMware memory ballooning, Microsoft Hyper-V dynamic memory, and the open source KVM balloon process are similar in concept. The host uses balloon drivers running on the VMs to determine how much memory it can take back from an under-utilizing VM. Balloon drivers must be installed on any VM that participates in the memory ballooning technique.

Balloon drivers get the target balloon size from the hypervisor and then inflate by allocating the proper number of guest physical pages within the VM. This process is known as inflating the balloon; the process of releasing the available pages is known as deflating the balloon.

VM memory ballooning can create performance problems. When a balloon driver inflates to the point where the VM no longer has enough memory to run its processes within itself, it starts using another VM memory technique known as memory swapping. This will slow down the VM, depending upon the amount of memory to recoup and/or the quality of the storage IOPS delivered to it.

Other virtual memory management techniques include memory overcommit, memory paging, memory mirroring and transparent page sharing.

This was last updated in October 2014