

RED HAT CUSTOMER PORTAL

Memory management and tuning options in Red Hat Enterprise Linux



Updated 08 May 2012, 7:15 AM GMT

Issue

- How can I tune and monitor performance of memory.

Environment

- Red Hat Enterprise Linux 3
- Red Hat Enterprise Linux 4
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6

Resolution

LowMem Starvation

- Memory usage on 32-bit system can become problematic under some workloads, especially for I/O intensive applications such as
 - Oracle Database or Application Server
 - Java
 - With the x86 architecture the first 16MB-896MB of physical memory is known as "low memory" (ZONE_NORMAL) which is permanently mapped into kernel space. Many kernel resources must live in the low memory zone. In fact, many kernel operations can only take place in this zone. This means that the low memory area is the most performance critical zone. For example, if you run many resources intensive applications/programs and/or use large physical memory, then "low memory" can become low since more kernel structures must be allocated in this area. Under heavy I/O workloads the kernel may become starved for LowMem even though there is an abundance of available HighMem. As the kernel tries to keep as much data in cache as possible this can lead to oom-killers or complete system hangs.
 - In 64-bit systems all the memory is allocated in ZONE_NORMAL. So lowmem starvation will not affect 64-bit systems. Moving to 64-bit would be a permanent fix for lowmem starvation.

Diagnosing

- The amount of **LowMem** can be checked in `/proc/meminfo`. If the LowFree falls below 50Mb it may be cause for concern. However this does not always indicate a problem as the kernel will try to use the entire LowMem zone and it may be able to reclaim some of the cache.

```
MemTotal:      502784 kB
MemFree:       29128 kB
HighTotal:     162088 kB
HighFree:      22860 kB
LowTotal:      340696 kB
LowFree:       6268 kB
```

- **OOM-KILLER**: the kernel should print `sysrq-M` information to messages and the console. You may see the Normal zone reporting `all_unreclaimable? yes`, meaning the kernel could not reclaim any memory in this zone.

```
Aug 25 15:44:24 dompar158 kernel: DMA free:12544kB min:16kB low:32kB hig
Aug 25 15:44:24 dompar158 kernel: Normal free:888kB min:928kB low:1856kB
Aug 25 15:44:24 dompar158 kernel: HighMem free:8731264kB min:512kB low:1
```

- In this case we can also see that the largest contiguous block of memory in the LowMem range is 32kB, so if the kernel requires anything larger than that the allocation may fail.

```
Aug 25 15:44:24 dompar158 kernel: DMA: 4*4kB 4*8kB 3*16kB 3*32kB 3*64kB
Aug 25 15:44:24 dompar158 kernel: Normal: 80*4kB 18*kB 29*16kB 3*32kB 0*
Aug 25 15:44:24 dompar158 kernel: HighMem: 660*4kB 160*8kB 61*16kB 55*32
```

- **SYSTEM HANGS:** A core file captured at the time of the hang can often provide evidence for LowMem starvation:

```
crash> kmem -i
```

	PAGES	TOTAL	PERCENTAGE
TOTAL MEM	1021613	3.9 GB	
FREE	159502	623.1 MB	15% of TOTAL MEM
USED	862111	3.3 GB	84% of TOTAL MEM
SHARED	198019	773.5 MB	19% of TOTAL MEM
BUFFERS	43212	168.8 MB	4% of TOTAL MEM
CACHED	103623	404.8 MB	10% of TOTAL MEM
SLAB	63170	246.8 MB	6% of TOTAL MEM
TOTAL HIGH	802802	3.1 GB	78% of TOTAL MEM
FREE HIGH	155824	608.7 MB	19% of TOTAL HIGH
TOTAL LOW	218811	854.7 MB	21% of TOTAL MEM
FREE LOW	3678	14.4 MB	1% of TOTAL LOW
TOTAL SWAP	1048554	4 GB	
SWAP USED	52	208 KB	0% of TOTAL SWAP
SWAP FREE	1048502	4 GB	99% of TOTAL SWAP

- Installing hangwatch can also be useful if the M flag is enabled for sysrq.

Tuning Sysctl

- RHEL 4
 - Attempt to protect 100Mb of LowMem from userspace allocations (defaults to 0)

```
vm.lower_zone_protection=100
```

- RHEL 5 and 6
 - Attempt to protect approximately 1/9 (98Mb) of LowMem from userspace allocations (defaults to 1/32, or 27.5 Mb)

```
vm.lowmem_reserve_ratio=256 256 9
```

- RHEL 4, 5 and 6
 - Have a higher tendency to swap out to disk. This value can go from 0 to 100 (default 60). Setting below 10 is not recommended

```
vm.swappiness=80
```

- Try to keep at least 19Mb of memory free (default varies). Adjust this to something higher than what is currently in use

```
vm.min_free_kbytes=19000
```

- Decrease the amount of time for a page to be considered old enough for flushing to disk via the pdflush daemon (default 2999). Expressed in 100'ths of a second

```
vm.dirty_expire_centisecs=2000
```

- Shorten the interval at which the pdflush daemon wakes up to write dirty data to disk (default 499). Expressed in 100'ths of a second

```
vm.dirty_writeback_centisecs=400
```

- Decrease the tendency of the kernel to reclaim the memory which is used for caching of directory and inode objects (default 100, do not increase this beyond 100 as it can cause excessive reclaim).

```
vm.vfs_cache_pressure=50
```

- RHEL 5
 - The pagecache value represents a percentage of physical RAM. When the size of the filesystem cache exceeds this size then cache pages are added only to the inactive list so under memory reclaim conditions the kernel is more likely to reclaim pages from the cache instead of swapping anonymous pages.

```
vm.pagecache=100
```

- RHEL 6
 - Will take into account highmem along with lowmem when calculating dirty_ratio and dirty_background_ratio. This will will make page reclaiming faster.

```
vm.highmem_is_dirtyable=1
```

Overcommit Memory

- Overcommitting memory allows the kernel to potentially allocate more memory than the system actually has. This is perfectly safe, and in fact default behavior, as the Linux VM will handle the management of memory. However, to tune it, consider the following information:

```
/proc/sys/vm/overcommit_memory
```

- 0 - Heuristic overcommit handling. Obvious overcommits of address space are refused. Used for a typical system. It ensures a seriously wild allocation fails while allowing overcommit to reduce swap usage. root is allowed to allocate slightly more memory in this mode. This is the default.
- 1 - Always overcommit. Appropriate for some scientific applications.
- 2 - Don't overcommit. The total address space commit for the system is not permitted to exceed swap plus a configurable percentage (default is 50) of physical RAM. Depending on the percentage you use, in most situations this means a process will not be killed while attempting to use already-allocated memory but will receive errors on memory allocation as appropriate.

HugePages

- Enabling an application to use HugePages provides many benefits for the VM as it allows that application to lock data into memory and prevent it from swapping. Some advantages of such a configuration:
 - Increased performance by through increased TLB hits
 - Pages are locked in memory and are never swapped out which guarantees that shared memory like SGA remains in RAM
 - Contiguous pages are preallocated and cannot be used for anything else but for System V shared memory (e.g. SGA)
 - Less bookkeeping work for the kernel for that part of virtual memory due to larger page sizes
- HugePages are only useful for applications that are aware of them (i.e., don't recommend them as a way to solve all memory issues). They are only used for shared memory allocations so be sure not to allocate too many pages. By default the size of one HugePage is 2Mb. For Oracle systems allocate enough HugePages to hold the entire SGA in memory.

- To enable HugePages use a sysctl setting to define how many pages should be allocated:

- RHEL 3

```
vm.hugetlb_pool=1024
```

- RHEL 4 onwards

```
vm.nr_hugepages=1024
```

- The application user must also have its memlock limit increased in `/etc/security/limits.conf` so they can lock that many pages into memory:
 - oracle - memlock 2097152
- On RHEL 4, 5 or 6 this user must be logged out and back in (i.e., the application restarted) before the settings will be applied. On RHEL 3 the system must be rebooted.

Resources

- [Tuning and Optimizing RHEL for Oracle 9i and 10g](http://www.redhat.com/t/pdf/rhel/Oracle-10-g-recommendations-v1_2.pdf) (http://www.redhat.com/t/pdf/rhel/Oracle-10-g-recommendations-v1_2.pdf)
- [Puschitz Guide to Tuning Linux for Oracle](http://www.puschitz.com/TuningLinuxForOracle.shtml) (<http://www.puschitz.com/TuningLinuxForOracle.shtml>)
- [Understanding Virtual Memory in Red Hat Enterprise Linux 3](http://people.redhat.com/nhorman/papers/rhel3_vm.pdf) (http://people.redhat.com/nhorman/papers/rhel3_vm.pdf)

Product(s) [Red Hat Enterprise Linux](/knowledge/products/red-hat-enterprise-linux) (/knowledge/products/red-hat-enterprise-linux)

Tags [memory](/knowledge/tags/memory) (/knowledge/tags/memory) , [performance](/knowledge/tags/performance) (/knowledge/tags/performance) , [rhel3](/knowledge/tags/rhel3) (/knowledge/tags/rhel3) , [rhel4](/knowledge/tags/rhel4) (/knowledge/tags/rhel4) , [rhel5](/knowledge/tags/rhel5) (/knowledge/tags/rhel5)



Copyright © 2013 Red Hat, Inc.