Linux 中的“大内存页”（hugepage）是个什么？

Linux爱好者 3月24日

> 英文：Shrikant Lavhate，翻译：Linux中国/DarkSun
> linux.cn/article-9450-1.html

> 学习 Linux 中的大内存页hugepage。理解什么是“大内存页”，如何进行配置，如何查看当前状态以及如何禁用它。

本文中我们会详细介绍大内存页huge page，让你能够回答：Linux 中的“大内存页”是什么？在 RHEL6、RHEL7、Ubuntu 等 Linux 中，如何启用/禁用“大内存页”？如何查看“大内存页”的当前值？

首先让我们从“大内存页”的基础知识开始讲起。

Linux 中的“大内存页”是个什么玩意？

“大内存页”有助于 Linux 系统进行虚拟内存管理。顾名思义，除了标准的 4KB 大小的页面外，它们还能帮助管理内存中的巨大的页面。使用“大内存页”，你最大可以定义 1GB 的页面大小。

在系统启动期间，你能用“大内存页”为应用程序预留一部分内存。这部分内存，即被“大内存页”占用的这些存储器永远不会被交换出内存。它会一直保留其中，除非你修改了配置。这会极大地提高像 Oracle 数据库这样的需要海量内存的应用程序的性能。

为什么使用“大内存页”？

在虚拟内存管理中，内核维护一个将虚拟内存地址映射到物理地址的表，对于每个页面操作，内核都需要加载相关的映射。如果你的内存页很小，那么你需要加载的页就会很多，导致内核会加载更多的映射表。而这会降低性能。

使用“大内存页”，意味着所需要的页变少了。从而大大减少由内核加载的映射表的数量。这提高了内核级别的性能最终有利于应用程序的性能。

简而言之，通过启用“大内存页”，系统具只需要处理较少的页面映射表，从而减少访问/维护它们的开销！

如何配置“大内存页”？

运行下面命令来查看当前“大内存页”的详细内容。

```
root@kerneltalks # grep Huge /proc/meminfo
AnonHugePages:         0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
```

从上面输出可以看到，每个页的大小为 2MB（Hugepagesize），并且系统中目前有 0 个“大内存页”（HugePages_Total）。这里“大内存页”的大小可以从 2MB 增加到 1GB。

运行下面的脚本可以知道系统当前需要多少个巨大页。该脚本取之于 Oracle。

```
#!/bin/bash
#
# hugepages_settings.sh
#
# Linux bash script to compute values for the
# recommended HugePages/HugeTLB configuration
#
# Note: This script does calculation for all shared memory
# segments available when the script is run, no matter it
# is an Oracle RDBMS shared memory segment or not.
# Check for the kernel version
```

```
KERN=`uname -r | awk -F. '{ printf("%d.%d\n",$1,$2); }'`
# Find out the HugePage size
HPG_SZ=`grep Hugepagesize /proc/meminfo | awk {'print $2'}`
# Start from 1 pages to be on the safe side and guarantee 1 free HugePage
NUM_PG=1
# Cumulative number of pages required to handle the running shared memory segments
for SEG_BYTES in `ipcs -m | awk {'print $5'} | grep "[0-9][0-9]*"`
do
  MIN_PG=`echo "$SEG_BYTES/($HPG_SZ*1024)" | bc -q`
  if [ $MIN_PG -gt 0 ]; then
    NUM_PG=`echo "$NUM_PG+$MIN_PG+1" | bc -q`
  fi
done
# Finish with results
case $KERN in
  '2.4') HUGETLB_POOL=`echo "$NUM_PG*$HPG_SZ/1024" | bc -q`;
        echo "Recommended setting: vm.hugetlb_pool = $HUGETLB_POOL" ;;
  '2.6' | '3.8' | '3.10' | '4.1' ) echo "Recommended setting: vm.nr_hugepages = $NUM_PG" ;;
   *) echo "Unrecognized kernel version $KERN. Exiting." ;;
esac
# End
```

将它以 hugepages_settings.sh 为名保存到 /tmp 中，然后运行之：

```
root@kerneltalks # sh /tmp/hugepages_settings.sh
Recommended setting: vm.nr_hugepages = 124
```

你的输出类似如上结果，只是数字会有一些出入。

这意味着，你系统需要 124 个每个 2MB 的“大内存页”！若你设置页面大小为 4MB，则结果就变成了 62。你明白了吧？

配置内核中的“大内存页”

本文最后一部分内容是配置上面提到的 内核参数 ，然后重新加载。将下面内容添加到 /etc/sysctl.conf 中，然后输入 sysctl -p 命令重新加载配置。

```
vm.nr_hugepages=126
```

注意我们这里多加了两个额外的页，因为我们希望在实际需要的页面数量之外多一些额外的空闲页。

现在，内核已经配置好了，但是要让应用能够使用这些“大内存页”还需要提高内存的使用阀值。新的内存阀值应该为 126 个页 x 每个页 2 MB = 252 MB，也就是 258048 KB。

你需要编辑 /etc/security/limits.conf 中的如下配置：

```
soft memlock 258048
hard memlock 258048
```

某些情况下，这些设置是在指定应用的文件中配置的，比如 Oracle DB 就是在 /etc/security/limits.d/99-grid-oracle-limits.conf 中配置的。

这就完成了！你可能还需要重启应用来让应用来使用这些新的巨大页。

（LCTT 译注：此外原文有误，“透明大内存页”和“大内存页”不同，而且，在 Redhat 系统中，“大内存页” 不是默认启用的，而“透明大内存页”是启用的。因此这个段落删除了。）