细心的朋友会注意到,当你在linux下频繁存取文件后,物理内存会很快被用光,当程序结束后,内存不会被正常释放,而是一直作为caching.这个问题,貌似有不少人在问,不过都没有看到有什么很好解决的办法.那么我来谈谈这个问题.

先来说说free命令

[root@server ~]# free -m  
                    total       used       free     shared    buffers     cached  
Mem:           249        163         86          0         10             94  
-/+ buffers/cache:         58        191  
Swap:          511          0          511

其中:

total 内存总数

used 已经使用的内存数

free 空闲的内存数

shared 多个进程共享的内存总额

buffers Buffer Cache和cached Page Cache 磁盘缓存的大小

-buffers/cache 的内存数:used - buffers - cached

+buffers/cache 的内存数:free + buffers + cached

可用的memory=free memory+buffers+cached

有了这个基础后,可以得知,我现在used为163MB,free为86,buffer和cached分别为10,94

那么我们来看看,如果我执行复制文件,内存会发生什么变化.

[root@server ~]# cp -r /etc ~/test/  
[root@server ~]# free -m  
                    total       used       free     shared    buffers     cached  
Mem:           249        244          4          0          8             174  
-/+ buffers/cache:         62        187  
Swap:          511          0          511

在我命令执行结束后,used为244MB,free为4MB,buffers为8MB,cached为174MB,天呐都被cached吃掉了.别紧张,这是为了提高文件读取效率的做法.

为了提高磁盘存取效率, Linux做了一些精心的设计, 除了对dentry进行缓存(用于VFS,加速文件路径名到inode的转换), 还采取了两种主要Cache方式：Buffer Cache和Page Cache。前者针对磁盘块的读写，后者针对文件inode的读写。这些Cache有效缩短了 I/O系统调用(比如read,write,getdents)的时间。"

那么有人说过段时间,linux会自动释放掉所用的内存,我们使用free再来试试,看看是否有释放>?

[root@server test]# free -m  
                    total       used       free     shared    buffers     cached  
Mem:           249        244          5          0          8            174  
-/+ buffers/cache:         61        188  
Swap:          511          0          511

MS没有任何变化,那么我能否手动释放掉这些内存呢???回答是可以的!

/proc是一个虚拟文件系统,我们可以通过对它的读写操作做为与kernel实体间进行通信的一种手段.也就是说可以通过修改/proc中的文件,来对当前kernel的行为做出调整.那么我们可以通过调整/proc/sys/vm/drop\_caches来释放内存.操作如下:

[root@server test]# cat /proc/sys/vm/drop\_caches  
0  
首先,/proc/sys/vm/drop\_caches的值,默认为0

[root@server test]# sync

手动执行sync命令(描述:sync 命令运行 sync 子例程。如果必须停止系统，则运行 sync 命令以确保文件系统的完整性。sync 命令将所有未写的系统缓冲区写到磁盘中，包含已修改的 i-node、已延迟的块 I/O 和读写映射文件)

[root@server test]# echo 3 > /proc/sys/vm/drop\_caches  
[root@server test]# cat /proc/sys/vm/drop\_caches  
3

将/proc/sys/vm/drop\_caches值设为3

[root@server test]# free -m  
                   total       used       free     shared    buffers     cached  
Mem:           249         66        182          0          0           11  
-/+ buffers/cache:         55        194  
Swap:          511          0          511

再来运行free命令,发现现在的used为66MB,free为182MB,buffers为0MB,cached为11MB.那么有效的释放了buffer和cache.

有关/proc/sys/vm/drop\_caches的用法在下面进行了说明

/proc/sys/vm/drop\_caches (since Linux 2.6.16)  
              Writing  to  this  file  causes the kernel to drop clean caches,  
              dentries and inodes from memory, causing that memory  to  become  
              free.

              To  free  pagecache,  use  echo 1 > /proc/sys/vm/drop\_caches; to  
              free dentries and inodes, use echo 2 > /proc/sys/vm/drop\_caches;  
              to   free   pagecache,   dentries  and  inodes,  use  echo  3  >  
              /proc/sys/vm/drop\_caches.

              Because this is a non-destructive operation  and  dirty  objects  
              are not freeable, the user should run sync(8) first.

==================================================

buffer 与cache 的区别

　　A buffer is something that has yet to be "written" to disk. A cache is something that has been "read" from the disk and stored for later use.

　　更详细的解释参考：Difference Between Buffer and Cache

**对于共享内存（Shared memory），主要用于在UNIX 环境下不同进程之间共享数据，是进程间通信的一种方法，一般的应用程序不会申请使用共享内存**，笔者也没有去验证共享内存对上面等式的影响。如果你有兴趣，请参考：What is Shared Memory?

　　cache 和 buffer的区别：

**Cache：高速缓存，是位于CPU与主内存间的一种容量较小但速度很高的存储器。**由于CPU的速度远高于主内存，CPU直接从内存中存取数据要等待一定时间周期，Cache中保存着CPU刚用过或循环使用的一部分数据，当CPU再次使用该部分数据时可从Cache中直接调用,这样就减少了CPU的等待时间,提高了系统的效率。Cache又分为一级Cache(L1 Cache)和二级Cache(L2 Cache)，L1 Cache集成在CPU内部，L2 Cache早期一般是焊在主板上,现在也都集成在CPU内部，常见的容量有256KB或512KB L2 Cache。

**Buffer：缓冲区，一个用于存储速度不同步的设备或优先级不同的设备之间传输数据的区域。通过缓冲区，可以使进程之间的相互等待变少，从而使从速度慢的设备读入数据时，速度快的设备的操作进程不发生间断。**

===========================================

# sync  
# echo 1 > /proc/sys/vm/drop\_caches  
   echo 2 > /proc/sys/vm/drop\_caches  
   echo 3 > /proc/sys/vm/drop\_caches

cache释放：  
To free pagecache:  
echo 1 > /proc/sys/vm/drop\_caches  
To free dentries and inodes:  
echo 2 > /proc/sys/vm/drop\_caches  
To free pagecache, dentries and inodes:  
echo 3 > /proc/sys/vm/drop\_caches

说明，释放前最好sync一下，防止丢数据。

因为LINUX的内核机制，一般情况下不需要特意去释放已经使用的cache。这些cache起来的内容可以增加文件以及的读写速度。

===============================================