Linux压测环境内存监控手段及方法

**影响内存性能的2个因素**

* 系统swap频繁切换
* JVM的GC过于频繁

**如何判断系统内存是否还有空余？**

* 从OS的角度来看

内核（OS）使用+Application(X, oracle,etc)使用的+buffers+cached.

* 从应用程序的角度来说

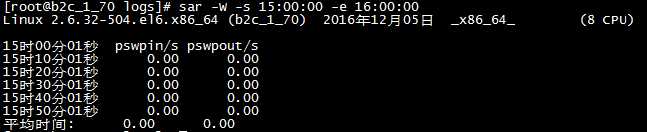
可用内存=系统free（memory+buffers+cached.）

因此，**只需要确认nmon输出仍然有free的内存，即使swap已部分占用也可以忽略**，这是由于SWAP的第一和第二个特性决定的，如下。

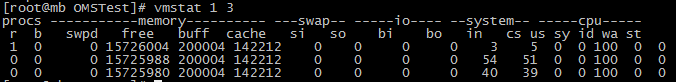
* Linux系统会不时的进行页面交换操作
* Linux进行页面交换是有条件的，不是所有页面在不用时都交换到虚拟内存
* 交换空间的页面在频繁使用后会导致Linux出现假死机、服务异常等问题

通常在压测过程中，在运行多个服务的服务器上可用内存时常几乎为0，此时可以观察swap in/swap out和swapd的值来判断交换频率。

* sar –w –s hh:mm:ss –e hh:mm:ss



* vmstat 1 3 （1 为监控频率、3为监控次数）

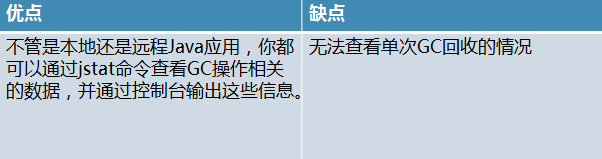


swpd列表示切换到内存交换区的内存数量(以k为单位)。如果swpd的值不为0，或者比较大，只要si、so的值长期为0，这种情况下一般不用担心，不会影响系统性能。反之则表示内存不足，需要增加系统内存。

**如何查看GC回收频率以及频次标准**

jstat: Java Virtual Machine Statistics Monitoring Tool

在使用**-gcutil**选项时，会输出如下字段的信息。在做GC调优时，尤其要关注**YGC**, **YGCT**, **FGC**, **FGCT**和**GCT**的数据变化。



命令行：

|  |
| --- |
| jstat -gcutil <pid> <interval> <count>  例：三分钟间隔输出2次  jstat -gcutil 6878 120000 2  S0 S1 E O P YGC YGCT FGC FGCT GCT  0.00 6.14 64.64 81.46 43.58 4413 24.675 2 0.171 24.846  59.25 0.00 75.54 26.82 44.62 4576 25.611 3 0.237 25.849 |

YGC：从应用程序启动到采样时年轻代中gc次数   
YGCT：从应用程序启动到采样时年轻代中gc所用时间(s)   
FGC：从应用程序启动到采样时old代(全gc)gc次数   
FGCT：从应用程序启动到采样时old代(全gc)gc所用时间(s)   
GCT：从应用程序启动到采样时gc用的总时间(s)

说明：

在监控的三分钟里FullGC发生1次(FGC从2->3)，FullGC所用时间为0.066ms(FGCT：0.237-0.171)

**当Full GC < 半小时/次且Fulll GC time < 1s，可认为GC工作良好。**

* **Minor GC执行迅速(50毫秒以内)**
* **Minor GC执行不频繁(间隔10秒左右一次)**
* **Full GC执行迅速(1秒以内)**
* **Full GC执行不频繁(间隔10分钟左右一次)**

如不满足，则说明GC有优化的空间，但不是必须，需结合实际业务处理速度来看。

如GC进行优化，则需进行长时间的稳定性测试，同时监控GC的指标来判断GC优化的有效性。

**如何使用监控脚本**

Gclogger.sh: 调用jstat命令定时输出jvm参数至gc\_xxx.log中。

GCResult.py:分析/logs下所有log文件中的Young GC和Full GC情况。