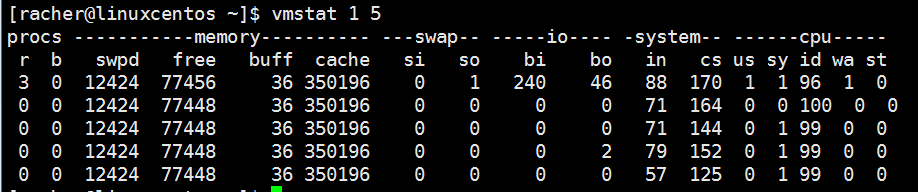
## Vmstat 命令

Vmstat 1 5 这表示vmstat 每1秒采集数据，一直采集5次结束。



**r** （run queue）表示运行队列(就是说多少个进程真的分配到CPU)，我测试的服务器目前CPU比较空闲，没什么程序在跑，当这个值超过了CPU数目，就会出现CPU瓶颈 了。这个也和top的负载有关系，一般负载超过了3就比较高，超过了5就高，超过了10就不正常了，服务器的状态很危险。top的负载类似每秒的运行队 列。如果运行队列过大，表示你的CPU很繁忙，一般会造成CPU使用率很高。

**b** （blocked）表示阻塞的进程,这个不多说，进程阻塞，大家懂的。

**swpd** 虚拟内存已使用的大小，如果大于0，表示你的机器物理内存不足了，如果不是程序内存泄露的原因，那么你该升级内存了或者把耗内存的任务迁移到其他机器。

**free**   空闲的物理内存的大小，我的机器内存总共8G，剩余3415M。

**buff**   Linux/Unix系统是用来存储，目录里面有什么内容，权限等的缓存，我本机大概占用300多M

**cache** cache直接用来记忆我们打开的文件,给文件做缓冲，我本机大概占用300多M(这里是Linux/Unix的聪明之处，把空闲的物理内存的一部分拿来做文件和目录的缓存，是为了提高 程序执行的性能，当程序使用内存时，buffer/cached会很快地被使用。)

**si**  每秒从磁盘读入虚拟内存的大小，如果这个值大于0，表示物理内存不够用或者内存泄露了，要查找耗内存进程解决掉。我的机器内存充裕，一切正常。

**so**  每秒虚拟内存写入磁盘的大小，如果这个值大于0，同上。

**bi**  块设备每秒接收的块数量，这里的块设备是指系统上所有的磁盘和其他块设备，默认块大小是1024byte，我本机上没什么IO操作，所以一直是0，但是我曾在处理拷贝大量数据(2-3T)的机器上看过可以达到140000/s，磁盘写入速度差不多140M每秒

**bo** 块设备每秒发送的块数量，例如我们读取文件，bo就要大于0。bi和bo一般都要接近0，不然就是IO过于频繁，需要调整。

**In(interrupts)** 每秒CPU的中断次数，包括时间中断

**cs** （context switches）每秒上下文切换次数，例如我们调用系统函数，就要进行上下文切换，线程的切换，也要进程上下文切换，这个值要越小越好，太大了，要考虑调低线程或者进程的 数目,例如在apache和nginx这种web服务器中，我们一般做性能测试时会进行几千并发甚至几万并发的测试，选择web服务器的进程可以由进程或 者线程的峰值一直下调，压测，直到cs到一个比较小的值，这个进程和线程数就是比较合适的值了。系统调用也是，每次调用系统函数，我们的代码就会进入内核 空间，导致上下文切换，这个是很耗资源，也要尽量避免频繁调用系统函数。上下文切换次数过多表示你的CPU大部分浪费在上下文切换，导致CPU干正经事的 时间少了，CPU没有充分利用，是不可取的。

**us** 用户CPU时间，我曾经在一个做加密解密很频繁的服务器上，可以看到us接近100,r运行队列达到80(机器在做压力测试，性能表现不佳)。

**sy** 系统CPU时间，如果太高，表示系统调用时间长，例如是IO操作频繁。

**id**  空闲 CPU时间，一般来说，id + us + sy = 100,一般我认为id是空闲CPU使用率，us是用户CPU使用率，sy是系统CPU使用率。

**wt** 等待IO CPU时间。

## 瓶颈分析：



分析哪些方面？

1. 操作系统OS（硬件方面） 包括：CPU、Memory、IO、Nerwork 等
2. 软件方面 攻城狮们的代码是否高效？

第一步 确定我们的系统跑的应用类型

应用类型大致分2中

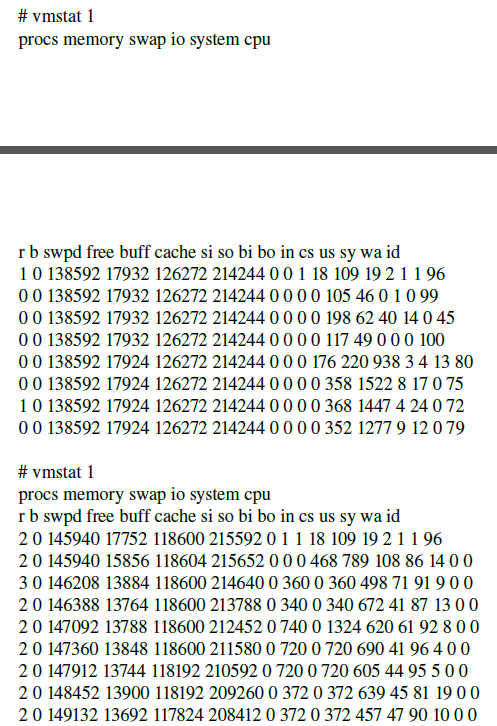
分别为 IO 范畴的 此类范畴的应用，一般都是高负荷的内存使用以及存储系统。通常数据库软件（mysal，oracle）

CPU 范畴的 这个范畴的应用，一般都是高负荷的cpu占用。CPU范畴的应用，就是一个批量处理cpu请求以及数学计算的工程。通常 web server ，mail servce

第二步 确定基准线

所谓基准线，就是我们的系统希望达到一个怎样的效果？以及那些方面需要优化，一个参考值，因此建立这个基准线，这个数据必须是系统可用性能状态值，用来比较不可用性能状态值。（一个标尺，参考物）

Eg：系统基准线快照



从上面基准线快照可以看到，最后一id列（表示的是空闲时间），可以看到基准线统计是cpu空闲时间在72% ~100%。第二个的出来的结果系统没有空闲时间，占用率100%，这样就可以确定是否cpu使用率应该被优化。

### \*CUP 介绍

**cpu使用率**其实就是运行的程序占用得cpu资源，反应机器在某个时间段运行程序的情况。使用率越高，说明机器在这个时间上运行了很多程序。

**上下文切换**（类似你插队了，前后挪动了下位置，这样对整体而言反而浪费时间）一个标准的Linux内核可以支持运行50～50000个进程运行，对于普通的CPU，内核会调度和执行这些进程。每个进程都会分到CPU的时间片来运行，当一个进程用完时间片或者被更高优先级的进程抢占后，它会备份到CPU的运行队列中，同时其他进程在CPU上运行。这个进程切换的过程被称作上下文切换。过多的上下文切换会造成系统很大的开销。

**运行队列** 每个CPU都会维持一个运行队列，理想情况下，调度器会不断让队列中的进程运行。进程不是处在sleep状态就是run able状态。如果CPU过载，就会出现调度器跟不上系统的要求，导致可运行的进程会填满队列。队列愈大，程序执行时间就愈长。

**性能判断标准**

1、Run-Queues :每个运行队列不超过1-3个线程，一个双核处理器应该运行队列不超过6个线程。

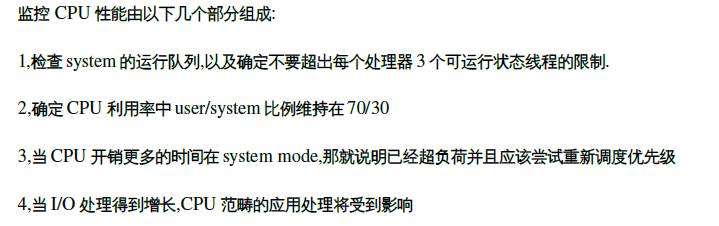
@2、CPU Utiliation 均衡比例

65% - 70% User Time

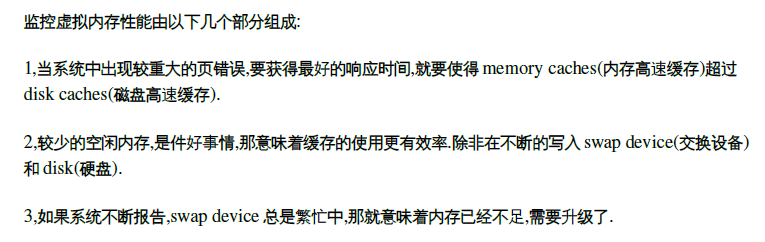
30% - 35% System Time

0% - 5% Idle Time

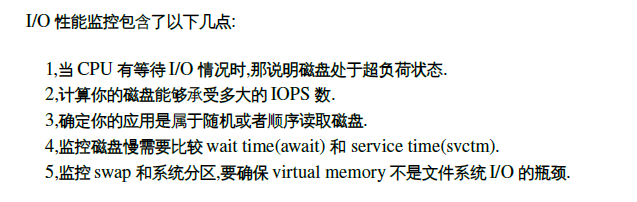
3、Context Switches ：上下文切换的数目直接关系到cpu的使用率，如果cpu利用率保持在上述均衡状态时，大量的上下文切换时正常的。



### \*Menory



### \*I/O



### \*Network

1、网卡配置设置（100baseT 100Mbit/s 基带传输）

2、网络吞吐量

3、本地吞吐量 iptraf

4、netperf 终端吞吐量

5、iperf 评估网络效率

