[Linux vmstat命令实战详解](http://www.cnblogs.com/ggjucheng/archive/2012/01/05/2312625.html)

vmstat命令是最常见的Linux/Unix监控工具，可以展现给定时间间隔的服务器的状态值,包括服务器的CPU使用率，内存使用，虚拟内存交换情况,IO读写情况。这个命令是我查看Linux/Unix最喜爱的命令，一个是Linux/Unix都支持，二是相比top，我可以看到整个机器的CPU,内存,IO的使用情况，而不是单单看到各个进程的CPU使用率和内存使用率(使用场景不一样)。

一般vmstat工具的使用是通过两个数字参数来完成的，第一个参数是采样的时间间隔数，单位是秒，第二个参数是采样的次数，如:

root@ubuntu:~# vmstat 2 1  
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----  
 r b swpd free buff cache si so bi bo in cs us sy id wa  
 1 0 0 3498472 315836 3819540 0 0 0 1 2 0 0 0 100 0

2表示每个两秒采集一次服务器状态，1表示只采集一次。

实际上，在应用过程中，我们会在一段时间内一直监控，不想监控直接结束vmstat就行了,例如:

[复制代码](javascript:void(0);)

root@ubuntu:~# vmstat 2   
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----  
 r b swpd free buff cache si so bi bo in cs us sy id wa  
 1 0 0 3499840 315836 3819660 0 0 0 1 2 0 0 0 100 0  
 0 0 0 3499584 315836 3819660 0 0 0 0 88 158 0 0 100 0  
 0 0 0 3499708 315836 3819660 0 0 0 2 86 162 0 0 100 0  
 0 0 0 3499708 315836 3819660 0 0 0 10 81 151 0 0 100 0  
 1 0 0 3499732 315836 3819660 0 0 0 2 83 154 0 0 100 0

[复制代码](javascript:void(0);)

这表示vmstat每2秒采集数据，一直采集，直到我结束程序，这里采集了5次数据我就结束了程序。

好了，命令介绍完毕，现在开始实战讲解每个参数的意思。

**r** 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** 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** 每秒CPU的中断次数，包括时间中断

**cs** 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。