**1.说明**

vmstat命令是最常见的Linux/Unix监控工具，属于sysstat包。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)。

**2.安装**

yum install -y sysstat

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:

1. root@local:~# vmstat 2 1
2. procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
3. r b swpd free buff cache si so bi bo in cs us sy id wa
4. 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:

1. root@local:~# vmstat 2
2. procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
3. r b swpd free buff cache si so bi bo in cs us sy id wa
4. 1 0 0 3499840 315836 3819660 0 0 0 1 2 0 0 0 100 0
5. 0 0 0 3499584 315836 3819660 0 0 0 0 88 158 0 0 100 0
6. 0 0 0 3499708 315836 3819660 0 0 0 2 86 162 0 0 100 0
7. 0 0 0 3499708 315836 3819660 0 0 0 10 81 151 0 0 100 0
8. 1 0 0 3499732 315836 3819660 0 0 0 2 83 154 0 0 100 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。

**3.字段含义说明：**

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **项目** | **含义** | **说明** |
| **Procs（进程）** | r | **等待执行的任务数** | **展示了正在执行和等待cpu资源的任务个数。当这个值超过了cpu个数，就会出现cpu瓶颈。** |
| B | 等待IO的进程数量 |  |
| **Memory(内存)** | swpd | 正在使用虚拟的内存大小，单位k |  |
| free | 空闲内存大小 |  |
| buff | 已用的buff大小，对块设备的读写进行缓冲 |  |
| cache | 已用的cache大小，文件系统的cache |  |
| inact | 非活跃内存大小，即被标明可回收的内存，区别于free和active | 具体含义见：概念补充（当使用-a选项时显示） |
| active | 活跃的内存大小 | 具体含义见：概念补充（当使用-a选项时显示） |
| **Swap** | si | 每秒从交换区写入内存的大小（单位：kb/s） |  |
| so | 每秒从内存写到交换区的大小 |  |
| **IO** | bi | 每秒读取的块数（读磁盘） | 块设备每秒接收的块数量，单位是block，这里的块设备是指系统上所有的磁盘和其他块设备，现在的Linux版本块的大小为1024bytes |
| bo | 每秒写入的块数（写磁盘） | 块设备每秒发送的块数量，单位是block |
| **system** | in | **每秒中断数，包括时钟中断** | **这两个值越大，会看到由内核消耗的cpu时间sy会越多**    **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** |
| cs | **每秒上下文切换数** |
| **CPU（以百分比表示）** | us | 用户进程执行消耗cpu时间(user time) | us的值比较高时，说明用户进程消耗的cpu时间多，**但是如果长期超过50%的使用，那么我们就该考虑优化程序算法或其他措施了** |
| sy | 系统进程消耗cpu时间(system time) | sys的值过高时，说明系统内核消耗的cpu资源多，这个不是良性的表现，我们应该检查原因。**这里us + sy的参考值为80%，如果us+sy 大于 80%说明可能存在CPU不足** |
| Id | 空闲时间(包括IO等待时间) | 一般来说 us+sy+id=100 |
| wa | 等待IO时间 | wa过高时，说明io等待比较严重，这可能是由于磁盘大量随机访问造成的，也有可能是磁盘的带宽出现瓶颈。 |

**4、常见问题处理**

**4.1 常见问题及解决方法**

如果r经常大于4，且id经常少于40，表示cpu的负荷很重。

如果pi，po长期不等于0，表示内存不足。

如果disk经常不等于0，且在b中的队列大于3，表示io性能不好。

1.)如果在processes中运行的序列(process r)是连续的大于在系统中的CPU的个数表示系统现在运行比较慢,有多数的进程等待CPU。

2.)如果r的输出数大于系统中可用CPU个数的4倍的话,则系统面临着CPU短缺的问题,或者是CPU的速率过低,系统中有多数的进程在等待CPU,造成系统中进程运行过慢。

3.)如果空闲时间(cpu id)持续为0并且系统时间(cpu sy)是用户时间的两倍(cpu us)系统则面临着CPU资源的短缺。

当发生以上问题的时候请先调整应用程序对CPU的占用情况.使得应用程序能够更有效的使用CPU.同时可以考虑增加更多的CPU.  关于CPU的使用情况还可以结合mpstat,  ps aux top  prstat –a等等一些相应的命令来综合考虑关于具体的CPU的使用情况,和那些进程在占用大量的CPU时间.一般情况下，应用程序的问题会比较大一些.比如一些sql语句不合理等等都会造成这样的现象.

内存问题现象:

内存的瓶颈是由scan rate (sr)来决定的.scan rate是通过每秒的始终算法来进行页扫描的.如果scan rate(sr)连续的大于每秒200页则表示可能存在内存缺陷.同样的如果page项中的pi和po这两栏表示每秒页面的调入的页数和每秒调出的页数.如果该值经常为非零值,也有可能存在内存的瓶颈,当然,如果个别的时候不为0的话,属于正常的页面调度这个是虚拟内存的主要原理.

解决办法:   
1.调节applications & servers使得对内存和cache的使用更加有效.

2.增加系统的内存.

3. Implement priority paging in s in pre solaris 8 versions by adding line "set priority paging=1" in /etc/system. Remove this line if upgrading from Solaris 7 to 8 & retaining old /etc/system file.

关于内存的使用情况还可以结ps aux top  prstat –a等等一些相应的命令来综合考虑关于具体的内存的使用情况,和那些进程在占用大量的内存.一般情况下，如果内存的占用率比较高,但是,CPU的占用很低的时候,可以考虑是有很多的应用程序占用了内存没有释放,但是,并没有占用CPU时间,可以考虑应用程序,对于未占用CPU时间和一些后台的程序,释放内存的占用。

**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。

**5.常见性能问题分析**

**IO/CPU/men连锁反应**  
    1.free急剧下降  
    2.buff和cache被回收下降，但也无济于事  
    3.依旧需要使用大量swap交换分区swpd  
    4.等待进程数，b增多  
    5.读写IO，bi bo增多  
    6.si so大于0开始从硬盘中读取  
    7.cpu等待时间用于 IO等待，wa增加  
**内存不足**  
    1.开始使用swpd，swpd不为0  
    2.si so大于0开始从硬盘中读取  
**io瓶颈**  
    1.读写IO，bi bo增多超过2000  
    2.cpu等待时间用于 IO等待，wa增加 超过20  
    3.sy 系统调用时间长，IO操作频繁会导致增加 >30%  
    4.wa io等待时间长  
        iowait% <20%            良好  
        iowait% <35%            一般  
        iowait% >50%  
    5.进一步使用iostat观察  
**CPU瓶颈：load,vmstat中r列**  
    1.反应为CPU队列长度  
    2.一段时间内，CPU正在处理和等待CPU处理的进程数之和，直接反应了CPU的使用和申请情况。  
    3.理想的load average：核数\*CPU数\*0.7  
        CPU个数：grep 'physical id' /proc/cpuinfo | sort -u  
        核数：grep 'core id' /proc/cpuinfo | sort -u | wc -l  
    4.超过这个值就说明已经是CPU瓶颈了  
**CPU瓶颈**  
    1.us 用户CPU时间高超过90%  
涉及到web服务器，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。  
    1.cs可以对apache和nginx线程和进程数限制起到一定的参考作用  
    2.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  
较好的趋势：主要是 swap使用少，swpd数值低。si so分页读取写入数值趋近于零

**6.其他说明：**

**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。