Linux服务器信息

# 查看cpu详细信息命令

cat /proc/cpuinfo

输出项的含义：

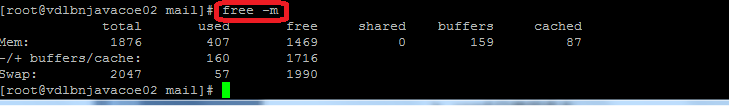
|  |
| --- |
| processor　：系统中逻辑处理核的编号。对于单核处理器，则课认为是其CPU编号，对于多核处理器则可以是物理核、或者使用超线程技术虚拟的逻辑核  vendor\_id：CPU制造商  cpu family：CPU产品系列代号  model：CPU属于其系列中的哪一代的代号  model name：CPU属于的名字及其编号、标称主频  stepping　：CPU属于制作更新版本  cpu MHz：CPU的实际使用主频  cache size：CPU二级缓存大小  physical id：单个CPU的标号  siblings ：单个CPU逻辑物理核数  core id：当前物理核在其所处CPU中的编号，这个编号不一定连续  cpu cores：该逻辑核所处CPU的物理核数  apicid：用来区分不同逻辑核的编号，系统中每个逻辑核的此编号必然不同，此编号不一定连续  fpu：是否具有浮点运算单元（Floating Point Unit）  fpu\_exception：是否支持浮点计算异常  cpuid level ：执行cpuid指令前，eax寄存器中的值，根据不同的值cpuid指令会返回不同的内容  wp ：表明当前CPU是否在内核态支持对用户空间的写保护（Write Protection）  flags：当前CPU支持的功能  bogomips：在系统内核启动时粗略测算的CPU速度（Million Instructions Per Second）  clflush size ：每次刷新缓存的大小单位  cache\_alignment：缓存地址对齐单位  address sizes：可访问地址空间位数  power management：对能源管理的支持 |

## 常用命令

|  |
| --- |
| #显示物理CPU个数  cat /proc/cpuinfo | grep "physical id" | sort | uniq | wc –l  #显示每个物理CPU中core的个数  cat /proc/cpuinfo | grep "cpu cores" | uniq  #显示逻辑CPU的个数  cat /proc/cpuinfo | grep "processor" | wc –l  注意：物理CPU个数×核数=逻辑CPU的个数 |

# 查看服务器内存的使用情况

## free命令

第二行：

* total:总计物理内存的大小。
* 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/cached:磁盘缓存的大小。

第三行(-/+ buffers/cached):

* -buffers/cache：（已用）的内存数，即used-buffers-cached。
* +buffers/cache：（可用）的内存数，即free + buffers + cached。
* used:已使用多大。
* free:可用有多少。

由此得出结论，可用内存的计算公式为

可用内存=free+buffers+cached

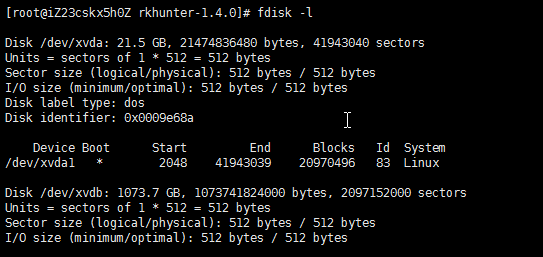
注意：free：物理内存有一部分充当缓冲和缓存（buffers，cached）

缓冲和缓存的区别：缓存cache是用来加速数据从硬盘中读取（将从硬盘读取的数据放入缓存中）；缓冲buffer是用来加速数据“写入”硬盘（数据暂时写入数据到缓冲中）。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能有效地缩短I/O系统调用（比如read、write、getdents）的时间。

# 查看linux服务器的硬盘使用情况

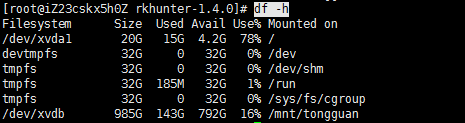
## 查看硬盘和分区情况

fdisk -l



## 检查文件系统的磁盘空间占用情况

df –h



## 查看硬盘的I/O性能

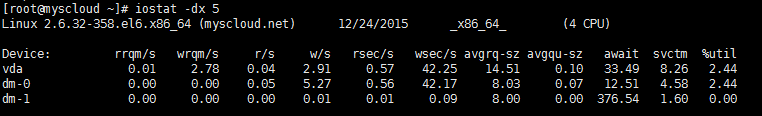
iostat命令被用于监视系统输入输出设备和CPU的使用情况。它的特点是汇报磁盘活动统计情况，同时也会汇报出CPU使用情况。同vmstat一样，iostat也有一个弱点，就是它不能对某个进程进行深入分析，仅对系统的整体情况进行分析。

语法：iostat(选项)(参数)

选项：

* -c：仅显示CPU使用情况；
* -d：仅显示设备利用率；
* -k：显示状态以千字节每秒为单位，而不使用块每秒；
* -m：显示状态以兆字节每秒为单位；
* -p：仅显示块设备和所有被使用的其他分区的状态；
* -t：显示每个报告产生时的时间；
* -V：显示版号并退出；
* -x：显示扩展状态。

参数：间隔时间：每次报告的间隔时间（秒）； 次数：显示报告的次数。



* Device 监测设备名称
* rrqm/s 每秒需要读取需求的数量
* wrqm/s 每秒需要写入需求的数量
* r/s 每秒实际读取需求的数量
* w/s 每秒实际写入需求的数量
* rsec/s 每秒读取区段的数量
* wsec/s 每秒写入区段的数量
* rkB/s 每秒实际读取的大小，单位为KB
* wkB/s 每秒实际写入的大小，单位为KB
* avgrq-sz 需求的平均大小区段
* avgqu-sz 需求的平均队列长度
* await 等待I/O平均的时间（milliseconds）
* svctm I/O需求完成的平均时间(毫秒)
* %util 被I/O需求消耗的CPU百分比

系统分析

CPU密集型的机器：vmstat的us会很高，sy>20%,r很高

I/O密集型机器：vm的b,wa值很高

内存交换：vmstat的si,so很高

## 查看Linux系统中某目录的大小

du -sh 目录名



# 查看linux系统的平均负载

## uptime命令

[root@vdlbnjavacoe01 test]# uptime

11:25:19 up 5 days, 22:04, 1 user, load average: 0.00, 0.01, 0.05

信息分析：

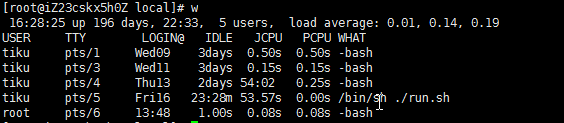
* 当前时间11:25:19
* 系统已运行的时间5 days, 22:04,
* 前在线用户 1 user
* 平均负载：0.00, 0.01, 0.05，最近1分钟、5分钟、15分钟系统的负载

一般来说，每个CPU内核当前活动进程数不大于3，则系统运行表现良好！当然这里说的是每个cpu内核，也就是如果你的主机是四核cpu的话，那么只要uptime最后输出的一串字符数值小于12即表示系统负载不是很严重.当然如果达到20，那就表示当前系统负载非常严重，估计打开执行web脚本非常缓慢.

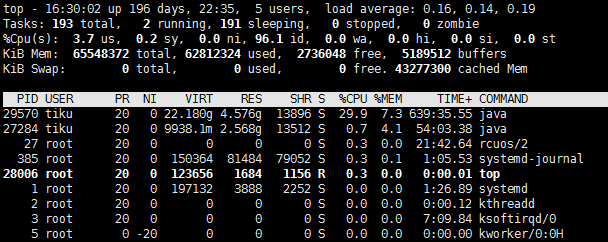
注意：cat /proc/cpuinfo #查看cpu信息

## 查看系统当前用户

W



## top动态显示系统平均负载

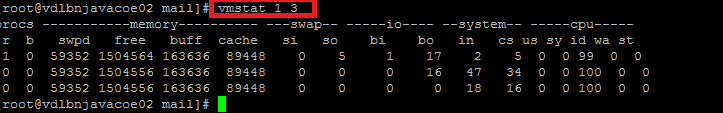


# 查看Linux系统的其他参数

## vmstat命令

#监控系统资源

vmstat刷新时延 次数



字段说明：

Procs（进程）

* r: 运行队列中进程数量，这个值也可以判断是否需要增加CPU。（长期大于1）
* b: 等待IO的进程数量。

Memory（内存）

* swpd: 使用虚拟内存大小，如果swpd的值不为0，但是SI，SO的值长期为0，这种情况不会影响系统性能。
* free: 空闲物理内存大小。
* buff: 用作缓冲的内存大小。
* cache: 用作缓存的内存大小，如果cache的值大的时候，说明cache处的文件数多，如果频繁访问到的文件都能被cache处，那么磁盘的读IO bi会非常小。

Swap

* si: 每秒从交换区写到内存的大小，由磁盘调入内存。
* so: 每秒写入交换区的内存大小，由内存调入磁盘。

注意：内存够用的时候，这2个值都是0，如果这2个值长期大于0时，系统性能会受到影响，磁盘IO和CPU资源都会被消耗。

有些朋友看到空闲内存（free）很少的或接近于0时，就认为内存不够用了，不能光看这一点，还要结合si和so，如果free很少，但是si和so也很少（大多时候是0），那么不用担心，系统性能这时不会受到影响的。

IO（现在的Linux版本块的大小为1kb）

* bi: 每秒读取的块数
* bo: 每秒写入的块数

注意：随机磁盘读写的时候，这2个值越大（如超出1024k)，能看到CPU在IO等待的值也会越大。

system（系统）

* in: 每秒中断数，包括时钟中断。
* cs: 每秒上下文切换数。

注意：上面2个值越大，会看到由内核消耗的CPU时间会越大。

CPU（以百分比表示）

* us: 用户进程执行时间百分比(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50%的使用，那么我们就该考虑优化程序算法或者进行加速。
* sy: 内核系统进程执行时间百分比(system time) sy的值高时，说明系统内核消耗的CPU资源多，这并不是良性表现，我们应该检查原因。
* wa: IO等待时间百分比 wa的值高时，说明IO等待比较严重，这可能由于磁盘大量作随机访问造成，也有可能磁盘出现瓶颈（块操作）。
* id: 空闲时间百分比

## dmesg命令

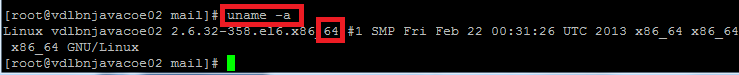
#开机时内核检测信息

dmesg | grep CPU

## uname命令

uname [选项]

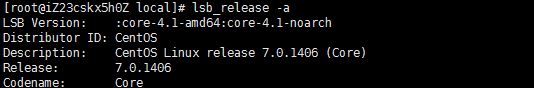
* -a:查看系统所有相关信息
* -r:查看内核信息
* -s:查看内核名称



## lsb\_release命令

#查看发行版本

lsb\_release -a

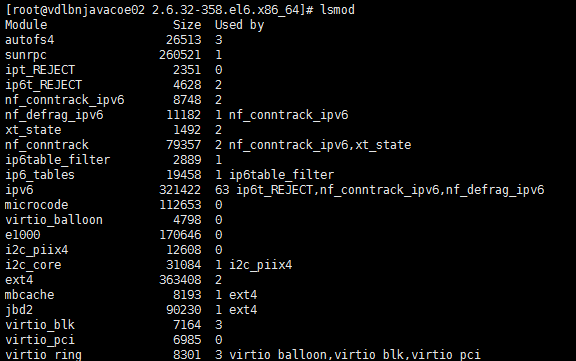


## linux系统内核参数

linux内核模块存储在/lib/modules/$(uname -r)

### 查看已加载内核模块

lsmod命令用来显示当前linux内核状态，不用任何参数说明会显示当前应经加载的所有内核模块。输出的三列分别表示模块名称，占用内存大小，是否被使用。如果第三列为0，则该模块可以随时卸载，非0则无法执行modprode删除模块。

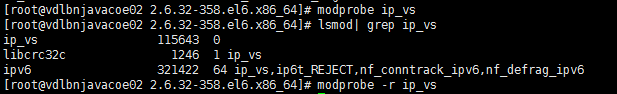


### 加载和卸载内核模块

modprode ip\_vs #动态加载ip\_vs模块

lsmod| grep ip\_vs #查看加载模块是否成功

modprobe -r ip\_vs #动态卸载ip\_vs模块



查看内核模块信息：modinfo ip\_vs

自动挂载内核的方法：

echo "modprobe ip\_vs" >> /etc/rc.sysinit

### 修改内核参数

Linux内核参数随着系统的启动会被写入内存，我们可以直接修改/proc目录下的文件来调整内核参数，并且这种调整是立即生效的。

临时调整内核参数:

echo "1" > /proc/sys/net/ipv4/ip\_forward

echo "1" > /proc/sys/net/ipv4/icmp\_echo\_ignore\_all

echo "108248">/proc/sys/fs/file-max

永久调整参数：vi /etc/sysctl.conf

|  |
| --- |
| net.ipv4.ip\_forward=1  net.ipv4.icmp\_echo\_ignore\_all=1  fs.file-max=108248  -- sysctl –p #立即生效 |

## Linux中查找PCI设置

有时需要在Linux下查找PCI设置，可以用lspci命令，它能列出机器中的PCI设备信息，比如声卡、显卡、Modem、网卡等的信息，也能列出主板集成设备的信息。lspci读取的是hwdata数据库。可能有的读者和我一样，最关心的还是网卡型号：lspci | grep Ether