常用命令

linux中一些常用的命令，如：查看内存、CPU、I/O、磁盘、带宽。 sql导入/导出 vi和vim中的常用命令。

Linux常用基本操作命令

查看内存

Linux查看内存我们一般使用free命令：

面是对Linux查看内存命令中这些数值的解释：

total:总计物理内存的大小。

used:已使用多大。

free:可用有多少。

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):

used:已使用多大。

free:可用有多少。

获得cpu的详细信息：

Linux命令：cat /proc/cpuinfo

重要参数:

processor 逻辑处理器的id。

physical id 物理封装的处理器的id。

core id 每个核心的id。

cpu cores 位于相同物理封装的处理器中的内核数量。

siblings 位于相同物理封装的处理器中的逻辑处理器的数量

查看磁盘使用情况

df -lh 命令

Filesystem(文件系统)	Size(容量)	Used(已用)	Avail(可用)	Use%(已用%)	Mounted on(挂载点)
/dev/xvda1	40G	3.9G	34G	11%	/
tmpfs	498M	0	498M	0%	/dev/shm

查看系统I/O使用情况

使用vmstat命令,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,比如

2表示每个两秒采集一次服务器状态，1表示只采集一次。

重要信息:

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。
查看带宽
可以使用iptraf命令
Iptraf命令需要安装iptraf才可以使用,安装方法很简单:
centos系统：sudo yum install iptraf
ubuntu系统：sudo apt-get install iptraf
安装成功后，在命令行执行“iptart”命令启动软件。

然后按键盘的上下键选择菜单,选择第三项“Detailed interface statistics”回车

选择网卡，有些服务器有内网网卡和外网网卡，具体哪个是外网网卡使用ifconfig查看对应的IP是哪个网卡名称即可。选择好后回车。
在这里 我的是eth1
回车,如下就是实时的带宽占用情况。

注意：如果需要退出本软件，在键盘中按X键即可。
Sql导入导出

首先进入mysql,创建空数据库ceshi, create database ceshi;
使用创建的数据库: use ceshi;
设置字符集: set names utf8;

导入命令:
Mysql -u root -p 库名<.sql文件

导入成功.
导出:

1、导出整个数据库：
mysqldump -u username -p ceshi> /ceshi.sql

说明：username是数据库用户名，ceshi源数据库，回车，输入数据库密码，就进行导入操作了！目标文件路径为/

2、导出数据库中某张表：

mysqldump -u username -p ceshi news> /news.sql
说明：导出数据库ceshi中的news表到目标目录。

vi/vim命令
vim 直接启动vim
Vim test.php 使用vim编辑器打开test.php,如果test.php不存在,则创建并打开.
插入命令:
i 在当前位置生前插入
I 在当前行首插入
a 在当前位置后插入
A 在当前行尾插入
o 在当前行之后插入一行
O 在当前行之前插入一行
查找命令: /text 查找text，按n健查找下一个，按N健查找前一个。
?text 查找text，反向查找，按n健查找下一个，按N健查找前一个。
vim中有一些特殊字符在查找时需要转义 .*[]^%/?~$
:set ignorecase 忽略大小写的查找
:set noignorecase 不忽略大小写的查找
查找很长的词，如果一个词很长，键入麻烦，可以将光标移动到该词上，按*或#键即可以该单词进行搜索，相当于/搜索。而#命令相当于?搜索。
:set hlsearch 高亮搜索结果，所有结果都高亮显示，而不是只显示一个匹配。
:set nohlsearch 关闭高亮搜索显示
:nohlsearch 关闭当前的高亮显示，如果再次搜索或者按下n或N键，则会再次高亮。
:set incsearch 逐步搜索模式，对当前键入的字符进行搜索而不必等待键入完成。
:set wrapscan 重新搜索，在搜索到文件头或尾时，返回继续搜索，默认开启。
举一个高亮查找的例子:
首先set hlsearch 设置查找结果高亮显示
设置成功之后再查找:
查找结果全部高亮显示
替换命令:
ra 将当前字符替换为a，当期字符即光标所在字符。
s/old/new/ 用old替换new，替换当前行的第一个匹配
s/old/new/g 用old替换new，替换当前行的所有匹配
%s/old/new/ 用old替换new，替换所有行的第一个匹配
%s/old/new/g 用old替换new，替换整个文件的所有匹配
:10,20 s/^/ /g 在第10行知第20行每行前面加四个空格，用于缩进。
ddp 交换光标所在行和其下紧邻的一行。
移动命令:
h 左移一个字符
l 右移一个字符，这个命令很少用，一般用w代替。
k 上移一个字符
j 下移一个字符
以上四个命令可以配合数字使用，比如20j就是向下移动20行，5h就是向左移动5个字符，在Vim中，很多命令都可以配合数字使用，比如删除10个字符10x，在当前位置后插入3个
！，3a！，这里的Esc是必须的，否则命令不生效。
w 向前移动一个单词（光标停在单词首部），如果已到行尾，则转至下一行行首。此命令快，可以代替l命令。
b 向后移动一个单词 2b 向后移动2个单词
e，同w，只不过是光标停在单词尾部

ge，同b，光标停在单词尾部。

^ 移动到本行第一个非空白字符上。

移动到本行第一个字符。同0健。

$ 移动到行尾 3$ 移动到下面3行的行尾

gg 移动到文件头。 = [[

G（shift + g） 移动到文件尾。 =]]

f（find）命令也可以用于移动，fx将找到光标后第一个为x的字符，3fd将找到第三个为d的字符。

F 同f，反向查找。

跳到指定行，冒号+行号，回车，比如跳到240行就是 :240回车。另一个方法是行号+G，比如230G跳到230行。

Ctrl + e 向下滚动一行

Ctrl + y 向上滚动一行

Ctrl + d 向下滚动半屏

Ctrl + u 向上滚动半屏

Ctrl + f 向下滚动一屏

Ctrl + b 向上滚动一屏

撤销和重做:

u 撤销（Undo）

U 撤销对整行的操作

Ctrl + r 重做（Redo），即撤销的撤销。

删除命令:

x 删除当前字符

3x 删除当前光标开始向后三个字符

X 删除当前字符的前一个字符。X=dh

dl 删除当前字符， dl=x

dh 删除前一个字符

dd 删除当前行

dj 删除上一行

dk 删除下一行

10d 删除当前行开始的10行。

D 删除当前字符至行尾。D=d$

d$ 删除当前字符之后的所有字符（本行）

kdgg 删除当前行之前所有行（不包括当前行）

jdG（jd shift + g） 删除当前行之后所有行（不包括当前行）

:1,10d 删除1-10行

:11,$d 删除11行及以后所有的行

:1,$d 删除所有行

J(shift + j) 删除两行之间的空行，实际上是合并两行。

复制和粘贴:

yy 拷贝当前行

nyy 拷贝当前后开始的n行，比如2yy拷贝当前行及其下一行。

p 在当前光标后粘贴,如果之前使用了yy命令来复制一行，那么就在当前行的下一行粘贴。

shift+p 在当前行前粘贴

:1,10 co 20 将1-10行插入到第20行之后。

:1,$ co $ 将整个文件复制一份并添加到文件尾部。

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按y即可复制

剪切命令:

正常模式下按v（逐字）或V（逐行）进入可视模式，然后用jklh命令移动即可选择某些行或字符，再按d即可剪切

ndd 剪切当前行之后的n行。利用p命令可以对剪切的内容进行粘贴

:1,10d 将1-10行剪切。利用p命令可将剪切后的内容进行粘贴。

:1, 10 m 20 将第1-10行移动到第20行之后。

退出命令:

:wq 保存并退出

ZZ 保存并退出

:q! 强制退出并忽略所有更改

:e! 放弃所有修改，并打开原来文件。

注释命令:

perl程序中#开始的行为注释，所以要注释某些行，只需在行首加入#

3,5 s/^/#/g 注释第3-5行

3,5 s/^#//g 解除3-5行的注释

1,$ s/^/#/g 注释整个文档。

:%s/^/#/g 注释整个文档，此法更快。
