Linux系统资源监控命令df free top mpstat vmstat iostat sar

2010-06-08 17:33

|  |
| --- |
| 想用LR监控[**Linux**](http://hi.baidu.com/kingfly666666/blog/item/:;)来着，可是得先在Linux服务器上装一堆乱七八糟的东西，由于本人之前对Linux不太了解，半天也没装上，时间紧迫，只好先弄出一套人工监控Linux系统[**资源**](http://hi.baidu.com/kingfly666666/blog/item/:;)的命令……  衡量CPU性能的指标：  1，用户使用CPU的情况； CPU运行常规用户进程 CPU运行niced process  CPU运行实时进程  2，系统使用CPU情况； 用于I/O管理：中断和驱动 用于内存[**管理**](http://hi.baidu.com/kingfly666666/blog/item/.:;)：页面交换 用户进程管理：进程开始和上下文切换  3，WIO：用于进程等待磁盘I/O而使CPU处于空闲状态的比率。  4，CPU的空闲率，除了上面的WIO以外的空闲时间  5，CPU用于上下文交换的比率  6，nice  7，real-time  8，运行进程队列的长度  9，平均负载  Linux中常用的监控CPU整体性能的工具有：  ♣ mpstat： mpstat 不但能查看所有CPU的平均信息，还能查看指定CPU的信息。  ♣[**vmstat**](http://hi.baidu.com/kingfly666666/blog/item/.:;)：只能查看所有CPU的平均信息；查看cpu队列信息；  ♣[**iostat**](http://hi.baidu.com/kingfly666666/blog/item/.:;): 只能查看所有CPU的平均信息。  ♣[**sar**](http://hi.baidu.com/kingfly666666/blog/item/.:;)： 与mpstat 一样，不但能查看CPU的平均信息，还能查看指定CPU的信息。  ♣ top：显示的信息同ps接近，但是top可以了解到CPU消耗，可以根据用户指定的时间来更新显示。  下面一一介绍：  **一，vmstat**  [root@localhost ~]#vmstat -n 3    (每个3秒刷新一次） procs-----------memory--------------------swap-- ----io---- --system---- ------cpu-------- r b swpd free      buff        cache      si so bi bo in     cs         us   sy   id wa 10 144 186164 105252 2386848 0 0    18 166   83     2          48   21 31 0 20 144 189620 105252 2386848 0 0     0 177   1039 1210   34   10 56 0 00 144 214324 105252 2386848 0 0     0 10    1071 670    32   5    63 0 00 144 202212 105252 2386848 0 0     0 189    1035 558    20   3    77 0 20 144 158772 105252 2386848 0 0     0 203   1065 2832    70 14 15 0  **红色内容标示CPU相关的参数**  **PROC(ESSES)**  **--r: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. 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.**  **SYSTEM --in:每秒产生的中断次数 --cs:每秒产生的上下文切换次数 上面2个值越大，会看到由内核消耗的CPU时间会越大**  **CPU -us:用户进程消耗的CPU时间百分 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（比如PHP/PERL） -sy:内核进程消耗的CPU时间百分比（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等待消耗的CPU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:CPU处于空闲状态时间百分比,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.   解决办法: 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**](http://hi.baidu.com/kingfly666666/blog/item/:;)**语句不合理等等都会造成这样的现象.**  **二，sar sar [options] [-A] [-o file] t [n]  在命令行中，n 和t 两个参数组合起来定义采样间隔和次数，t为采样间隔，是必须有 的参数，n为采样次数，是可选的，默认值是1，-o file表示将命令结果以二进制格式 存放在文件中，file 在此处不是关键字，是文件名。options 为命令行选项，sar命令 的选项很多，下面只列出常用选项：  -A：所有报告的总和。 -u：CPU利用率 -v：进程、I节点、文件和锁表状态。 -d：硬盘使用报告。 -r：内存和交换空间的使用统计。 -g：串口I/O的情况。 -b：缓冲区使用情况。 -a：文件读写情况。 -c：系统调用情况。 -q：报告队列长度和系统平均负载 -R：进程的活动情况。 -y：终端设备活动情况。 -w：系统交换活动。 -x { pid | SELF | ALL }：报告指定进程ID的统计信息，SELF关键字是sar进程本身的统计，ALL关键字是所有系统进程的统计。**  **用sar进行CPU利用率的分析**  **#sar -u 2 10 Linux 2.6.18-53.el5PAE (localhost.localdomain) 03/28/2009**  **07:40:17 PM       CPU     %user     %nice   %system   %iowait    %steal     %idle 07:40:19 PM       all         12.44      0.00         6.97          1.74         0.00        78.86 07:40:21 PM       all         26.75      0.00        12.50         16.00      0.00        44.75 07:40:23 PM       all         16.96      0.00         7.98          0.00        0.00       75.06 07:40:25 PM       all         22.50      0.00         7.00          3.25        0.00        67.25 07:40:27 PM       all         7.25        0.00         2.75          2.50        0.00        87.50 07:40:29 PM       all         20.05      0.00         8.56          2.93        0.00        68.46 07:40:31 PM       all         13.97      0.00         6.23          3.49        0.00       76.31 07:40:33 PM       all          8.25        0.00         0.75          3.50        0.00       87.50 07:40:35 PM       all         13.25      0.00         5.75          4.00        0.00       77.00 07:40:37 PM       all         10.03      0.00         0.50          2.51        0.00       86.97 Average:             all         15.15      0.00         5.91          3.99        0.00       74.95**  **在显示内容包括：  %user：CPU处在用户模式下的时间百分比。**  **%nice：CPU处在带NICE值的用户模式下的时间百分比。 %system：CPU处在系统模式下的时间百分比。 %iowait：CPU等待输入输出完成时间的百分比。**  **%steal：管理程序维护另一个虚拟处理器时，虚拟CPU的无意识等待时间百分比。 %idle：CPU空闲时间百分比。**  **在所有的显示中，我们应主要注意%iowait和%idle，%iowait的值过高，表示硬盘存在I/O瓶颈，%idle值高，表示CPU较空闲，如 果%idle值高但系统响应慢时，有可能是CPU等待分配内存，此时应加大内存容量。%idle值如果持续低于10，那么系统的CPU处理能力相对较低， 表明系统中最需要解决的资源是CPU。**  **用sar进行运行进程队列长度分析：**  **#sar -q 2 10 Linux 2.6.18-53.el5PAE (localhost.localdomain) 03/28/2009**  **07:58:14 PM   runq-sz plist-sz   ldavg-1   ldavg-5 ldavg-15 07:58:16 PM         0         493          0.64        0.56        0.49 07:58:18 PM         1         491          0.64        0.56        0.49 07:58:20 PM         1         488          0.59        0.55        0.49 07:58:22 PM         0         487          0.59        0.55        0.49 07:58:24 PM         0         485          0.59        0.55        0.49 07:58:26 PM         1         483          0.78        0.59        0.50 07:58:28 PM         0         481          0.78        0.59        0.50 07:58:30 PM         1         480          0.72        0.58        0.50 07:58:32 PM         0         477          0.72        0.58        0.50 07:58:34 PM         0         474          0.72        0.58        0.50 Average:               0         484         0.68        0.57        0.49**  **runq-sz 准备运行的进程运行队列。 plist-sz 进程队列里的进程和线程的数量 ldavg-1 前一分钟的系统平均负载(load average) ldavg-5 前五分钟的系统平均负载(load average) ldavg-15 前15分钟的系统平均负载(load average)**  **顺便说一下load avarage的含义**  **load average可以理解为每秒钟CPU等待运行的进程个数.**  **在Linux系统中，sar -q、uptime、w、top等命令都会有系统平均负载load average的输出，那么什么是系统平均负载呢？ 系统平均负载被定义为在特定时间间隔内运行队列中的平均任务数。如果一个进程满足以下条件则其就会位于运行队列中： - 它没有在等待I/O操作的结果 - 它没有主动进入等待状态(也就是没有调用'wait') - 没有被停止(例如：等待终止) 例如： # uptime 20:55:40 up 24 days, 3:06, 1 user, load average: 8.13, 5.90, 4.94 命令输出的最后内容表示在过去的1、5、15分钟内运行队列中的平均进程数量。 一般来说只要每个CPU的当前活动进程数不大于3那么系统的性能就是良好的，如果每个CPU的任务数大于5，那么就表示这台机器的性能有严重问题。对 于上面的例子来说，假设系统有两个CPU，那么其每个CPU的当前任务数为：8.13/2=4.065。这表示该系统的性能是可以接受的。**  **三，iostat**  **#iostat -c 2 10 Linux 2.6.18-53.el5PAE (localhost.localdomain) 03/28/2009**  **avg-cpu: %user   %nice %system %iowait %steal   %idle 30.10    0.00          4.89         5.63    0.00   59.38**  **avg-cpu: %user   %nice %system %iowait %steal   %idle 8.46       0.00           1.74         0.25    0.00   89.55**  **avg-cpu: %user   %nice %system %iowait %steal   %idle 22.06     0.00          11.28       1.25    0.00   65.41**  **四，mpstat**  **mpstat是Multiprocessor Statistics的缩写，是实时系统**[**监控**](http://hi.baidu.com/kingfly666666/blog/item/:;)**工具。其报告与CPU的一些统计信息，这些信息存放在/proc/stat文件中。在多CPUs系统里，其不但能查看所有CPU的平均状况信息，而且能够查看特定CPU的信息。下面只介绍 mpstat与CPU相关的参数，mpstat的语法如下：**  **mpstat [-P {|ALL}] [internal [count]]**  **参数的含义如下：**  **参数 解释**  **-P {|ALL} 表示监控哪个CPU， cpu在[0,cpu个数-1]中取值**  **internal 相邻的两次采样的间隔时间**  **count 采样的次数，count只能和delay一起使用**  **当没有参数时，mpstat则显示系统启动以后所有信息的平均值。有interval时，第一行的信息自系统启动以来的平均信息。从第二行开始，输出为前一个interval时间段的平均信息。与CPU有关的输出的含义如下：**  **参数 解释 从/proc/stat获得数据**  **CPU 处理器ID**  **user 在internal时间段里，用户态的CPU时间（%） ，不包含 nice值为负 进程 δusr/δtotal\*100**  **nice 在internal时间段里，nice值为负进程的CPU时间（%） δnice/δtotal\*100**  **system 在internal时间段里，核心时间（%） δsystem/δtotal\*100**  **iowait 在internal时间段里，硬盘IO等待时间（%） δiowait/δtotal\*100**  **irq 在internal时间段里，软中断时间（%） δirq/δtotal\*100**  **soft 在internal时间段里，软中断时间（%） δsoftirq/δtotal\*100**  **idle 在internal时间段里，CPU除去等待磁盘IO操作外的因为任何原因而空闲的时间闲置时间 （%） δidle/δtotal\*100**  **intr/s 在internal时间段里，每秒CPU接收的中断的次数 δintr/δtotal\*100**  **CPU总的**[**工作**](http://hi.baidu.com/kingfly666666/blog/item/:;)**时间=total\_cur=user+system+nice+idle+iowait+irq+softirq**  **total\_pre=pre\_user+ pre\_system+ pre\_nice+ pre\_idle+ pre\_iowait+ pre\_irq+ pre\_softirq**  **δuser=user\_cur – user\_pre**  **δtotal=total\_cur-total\_pre**  **其中\_cur 表示当前值，\_pre表示interval时间前的值。上表中的所有值可取到两位小数点。**  **#mpstat -P ALL 2 10 Linux 2.6.18-53.el5PAE (localhost.localdomain) 03/28/2009**  **10:07:57 PM CPU   %user   %nice    %sys %iowait    %irq   %soft %steal   %idle    intr/s 10:07:59 PM all   20.75    0.00   10.50    1.50    0.25    0.25    0.00   66.75   1294.50 10:07:59 PM    0   16.00    0.00    9.00    1.50    0.00    0.00    0.00   73.50   1000.50 10:07:59 PM    1   25.76    0.00   12.12    1.52    0.00    0.51    0.00   60.10    294.00**  **五，top**  [**top**](http://hi.baidu.com/kingfly666666/blog/item/.:;)**命令和ps命令的基本作用是相同的，显示系统当前的进程和**[**其它**](http://hi.baidu.com/kingfly666666/blog/item/:;)**状 况；但是top是 一个动态显示过程，即可以通过用户按键来不断刷新当前状态。如果在前台执行该命令，它将独占前台，直到用户终止该程序为止。比较准确的说，top命令提供 了实时的对系统处理器的状态监视。它将显示系统中CPU最“敏感”的任务列表。该命令可以按CPU使用。内存使用和执行时间对任务进行排序；而且该命令的 很多特性都可以通过交互式命令或者在个人定制文件中进行设定。在后面的介绍中将把命令参数和交互命令分开讲述。 #top**  **top - 22:19:58 up 24 days, 4:31, 1 user, load average: 2.08, 1.88, 1.15 Tasks: 545 total,   7 running, 538 sleeping,   0 stopped,   0 zombie Cpu(s): 23.2%us, 6.3%sy, 0.0%ni, 69.7%id, 0.5%wa, 0.2%hi, 0.2%si, 0.0%st Mem:   4141608k total, 3992588k used,   149020k free,   249092k buffers Swap: 6289352k total,      128k used, 6289224k free, 2604200k cached**  **PID USER      PR NI VIRT RES SHR S %CPU %MEM    TIME+ COMMAND                                                                2746 mysql     15   0 147m 45m 4752 S   24 1.1   6630:02 mysqld                                                                 26630 nobody    25   0 25364 10m 3696 S    7 0.3   0:00.20 httpd                                                                  24892 nobody    15   0 27012 12m 4044 S    4 0.3   0:00.65 httpd                                                                  26619 nobody    15   0 27272 12m 3728 S    3 0.3   0:00.10 httpd                                                                  26626 nobody    23   0 25620 10m 3704 S    3 0.3   0:00.09 httpd                                                                  26634 nobody    21   0 29716 14m 3576 R    3 0.3   0:00.09 httpd                                                                  26628 nobody    25   0 25620 10m 3728 S    3 0.3   0:00.08 httpd                                                                  25448 nobody    15   0 26156 11m 4212 S    2 0.3   0:00.40 httpd                                                                  23843 nobody    16   0 25900 11m 4116 S    1 0.3   0:00.54 httpd**  **统计信息区 前五行是系统整体的统计信息。第一行是任务队列信息，同 uptime 命令的执行结果。其内容如下：**  **22:19:58当前时间 up 24 days, 4:31系统运行时间，格式为时:分 1 user当前登录用户数 load average: 2.08, 1.88, 1.15系统负载，即任务队列的平均长度。 三个数值分别为 1分钟、5分钟、15分钟前到现在的平均值。**  **第二、三行为进程和CPU的信息。当有多个CPU时，这些内容可能会超过两行。内容如下：**  **Tasks:545total 进程总数 7 running 正在运行的进程数 538sleeping 睡眠的进程数 0 stopped 停止的进程数 0 zombie 僵尸进程数 Cpu(s):23.2%us用户空间占用CPU百分比 6.3%sy内核空间占用CPU百分比 0.0% ni 用户进程空间内改变过优先级的进程占用CPU百分比 69.7%id空闲CPU百分比 0.5%wa等待输入输出的CPU时间百分比 0.2%hi 0.2%si 0.0%st**  **最后两行为内存信息。内容如下：**  **Mem:4141608k total物理内存总量 3992588k used使用的物理内存总量 149020k free空闲内存总量 249092k buffers用作内核缓存的内存量 Swap:6289352k total交换区总量 128k used使用的交换区总量 6289224k free空闲交换区总量 2604200k cached缓冲的交换区总量。 内存中的内容被换出到交换区，而后又被换入到内存，但使用过的交换区尚未被覆盖， 该数值即为这些内容已存在于内存中的交换区的大小。 相应的内存再次被换出时可不必再对交换区写入。**  **进程信息区 统计信息区域的下方显示了各个进程的详细信息。首先来认识一下各列的含义。**  **序号 列名 含义 a PID 进程id b PPID 父进程id c RUSER Real user name d UID 进程所有者的用户id e USER 进程所有者的用户名 f GROUP 进程所有者的组名 g TTY 启动进程的终端名。不是从终端启动的进程则显示为 ? h PR 优先级 i NI nice值。负值表示高优先级，正值表示低优先级 j P 最后使用的CPU，仅在多CPU环境下有意义 k %CPU 上次更新到现在的CPU时间占用百分比 l**[**TIME**](http://hi.baidu.com/kingfly666666/blog/item/.:;)**进程使用的CPU时间总计，单位秒 m TIME+ 进程使用的CPU时间总计，单位1/100秒 n %MEM 进程使用的物理内存百分比 o VIRT 进程使用的虚拟内存总量，单位kb。VIRT=SWAP+RES p SWAP 进程使用的虚拟内存中，被换出的大小，单位kb。 q RES 进程使用的、未被换出的物理内存大小，单位kb。RES=CODE+DATA r CODE 可执行**[**代码**](http://hi.baidu.com/kingfly666666/blog/item/.:;)**占用的物理内存大小，单位kb s DATA 可执行代码以外的部分(数据段+栈)占用的物理内存大小，单位kb t SHR 共享内存大小，单位kb u nFLT 页面错误次数 v nDRT 最后一次写入到现在，被修改过的页面数。 w S 进程状态。 D=不可中断的睡眠状态 R=运行 S=睡眠 T=跟踪/停止 Z=僵尸进程 x COMMAND 命令名/命令行 y WCHAN 若该进程在睡眠，则显示睡眠中的系统函数名 z Flags 任务标志，参考 sched.h**  **默认情况下仅显示比较重要的 PID、USER、PR、NI、VIRT、RES、SHR、S、%CPU、%MEM、TIME+、COMMAND 列。可以通过下面的快捷键来更改显示内容。**  **更改显示内容 通过 f 键可以选择显示的内容。按 f 键之后会显示列的列表，按 a-z 即可显示或隐藏对应的列，最后按回车键确定。**  **按 o 键可以改变列的显示顺序。按小写的 a-z 可以将相应的列向右移动，而大写的 A-Z 可以将相应的列向左移动。最后按回车键确定。**  **按大写的 F 或 O 键，然后按 a-z 可以将进程按照相应的列进行排序。而大写的 R 键可以将当前的排序倒转。**  **命令使用**  **1． 工具（命令）名称 top 2．工具（命令）作用 显示系统当前的进程和**[**其他**](http://hi.baidu.com/kingfly666666/blog/item/:;)**状 况； top是一个动态显示过程,即可以通过用户按键来不断刷新当前状态.如果在前台执行该命令,它将独占前台,直到用户终止该程序为止. 比较准确的说,top命令提供了实时的对系统处理器的状态监视.它将显示系统中CPU最“敏感”的任务列表.该命令可以按CPU使用.内存使用和执行时间 对任务进行排序；而且该命令的很多特性都可以通过交互式命令或者在个人定制文件中进行设定. 3．环境设置 在Linux下使用。 4．使用方法 4．1使用格式 top [-] [d] [p] [q] [c] [C] [S] [s] [n] 4．2参数说明 d 指定每两次屏幕信息刷新之间的时间间隔。当然用户可以使用s交互命令来改变之。 p 通过指定监控进程ID来仅仅监控某个进程的状态。 q该选项将使top没有任何延迟的进行刷新。如果调用程序有超级用户权限，那么top将以尽可能高的优先级运行。 S 指定累计模式 s 使top命令在安全模式中运行。这将去除交互命令所带来的潜在危险。 i 使top不显示任何闲置或者僵死进程。 c 显示整个命令行而不只是显示命令名 4.3其他 下面介绍在top命令执行过程中可以使用的一些交互命令。从使用角度来看，熟练的掌握这些命令比掌握选项还重要一些。这些命令都是单字母的，如果在命令行选项中使用了s选项，则可能其中一些命令会被屏蔽掉。 Ctrl+L 擦除并且重写屏幕。 h或者? 显示帮助画面，给出一些简短的命令总结说明。 k 终止一个进程。系统将提示用户输入需要终止的进程PID，以及需要发送给该进程什么样的信号。一般的终止进程可以使用15信号；如果不能正常结束那就使用信号9强制结束该进程。默认值是信号15。在安全模式中此命令被屏蔽。 i 忽略闲置和僵死进程。这是一个开关式命令。 q 退出程序。 r 重新安排一个进程的优先级别。系统提示用户输入需要改变的进程PID以及需要设置的进程优先级值。输入一个正值将使优先级降低，反之则可以使该进程拥有更高的优先权。默认值是10。 S 切换到累计模式。 s 改变两次刷新之间的延迟时间。系统将提示用户输入新的时间，单位为s。如果有小数，就换算成m s。输入0值则系统将不断刷新，默认值是5 s。需要注意的是如果设置太小的时间，很可能会引起不断刷新，从而根本来不及看清显示的情况，而且系统负载也会大大增加。 f或者F 从当前显示中添加或者删除项目。 o或者O 改变显示项目的顺序。 l 切换显示平均负载和启动时间信息。 m 切换显示内存信息。 t 切换显示进程和CPU状态信息。 c 切换显示命令名称和完整命令行。 M 根据驻留内存大小进行排序。 P 根据CPU使用百分比大小进行排序。 T 根据时间/累计时间进行排序。 W 将当前设置写入~/.toprc文件中。这是写top配置文件的推荐方法。** |