Linux 性能监测工具

Linux 系统出现问题时，我们不仅需要查看系统日志信息，而且还要使用大量的性能监测工具来判断究竟是哪一部分（内存、CPU、硬盘……）出了问题。在 Linux 系统中，所有的运行参数保存在虚拟目录/proc 中，换句话说，我们使用的性能监控工具取到的数据值实际上就是源自于这个目录，当涉及到系统高 估时，我们就可以修改/proc 目录中的相关参数了，当然有些是不能乱改的。下面就让我们了解一下这些常用的性能监控工具。

1、uptime

uptime 命令用于查看服务器运行了多长时间以及有多少个用户登录，快速获知服务器的负荷情况。

uptime 的输出包含一项内容是 load average，显示了最近 1，5，15 分钟的负荷情况。它的值代表等待 CPU 处理的进程数，如果 CPU 没有时间处理这些进程，load average 值会升高；反之则会降低。

load average 的最佳值是 1，说明每个进程都可以马上处理并且没有 CPU cycles 被丢失。对于单 CPU 的机器，1 或者 2 是可以接受的值；对于多路 CPU 的机器，load average 值可能在 8 到 10 之间。

也可以使用 uptime 命令来判断网络性能。例如，某个网络应用性能很低，通过运行 uptime 查看服务器的负荷是否很高，如果不是，那么问题应该是网络方面造成的。

以下是 uptime 的运行实例：

```
9:24am up 19:06, 1 user, load average: 0.00, 0.00, 0.00
```

也可以查看/proc/loadavg 和/proc/uptime 两个文件，注意不能编辑/proc 中的文件，要用 cat 等命令来查看，如：

```
liyawei:~ # cat /proc/loadavg
0.00 0.00 0.00 1/55 5505
```

2、dmesg

dmesg 命令主要用来显示内核信息。使用 dmesg 可以有效诊断机器硬件故障或者添加硬件出现的问题。

另外，使用 dmesg 可以确定您的服务器安装了那些硬件。每次系统重启，系统都会检查所有硬件并将信息记录下来。执行/bin/dmesg 命令可以查看该记录。

dmesg 输入实例：

```
ReiserFS: hda6: checking transaction log (hda6)
ReiserFS: hda6: Using r5 hash to sort names
Adding 1044184k swap on /dev/hda5. Priority:-1 extents:1 across:1044184k
parport_pc: VIA 686A/8231 detected
parport_pc: probing current configuration
parport_pc: Current parallel port base: 0×378
parport0: PC-style at 0×378 (0×778), irq 7, using FIFO
[PCSPP,TRISTATE,COMPAT,ECP]
parport_pc: VIA parallel port: io=0×378, irq=7
lp0: using parport0 (interrupt-driven).
e100: Intel(R) PRO/100 Network Driver, 3.5.10-k2-NAPI
e100: Copyright(c) 1999-2005 Intel Corporation
ACPI: PCI Interrupt 0000:00:0d.0[A] -> GSI 17 (level, low) -> IRQ 169
e100: eth0: e100_probe: addr 0xd8042000, irq 169, MAC addr
00:02:55:1E:35:91
usbcore: registered new driver usbfs
usbcore: registered new driver hub
hdc: ATAPI 48X CD-ROM drive, 128kB Cache, UDMA(33)
Uniform CD-ROM driver Revision: 3.20
```

USB Universal Host Controller Interface driver v2.3

3、top
top 命令显示处理器的活动状况。缺省情况下，显示占用 CPU 最多的任务，并且每隔 5 秒钟做一次刷新。
Process priority 的数值决定了 CPU 处理进程的顺序。LIUNX 内核会根据需要调整该数值的大小。nice value 局限于 priority。priority 的值不能低于 nice value（nice value 值越低，优先级越高）。您不可以直接修改 Process priority 的值，但是可以通过调整 nice level 值来间接地改变 Process priority 值，然而这一方法并不是所有时候都可用。如果某个进程运行异常的慢，可以通过降低 nice level 为该进程分配更多的 CPU。
Linux 支持的 nice levels 由 19 (优先级低)到-20 (优先级高)，缺省值为 0。
执行/bin/ps 命令可以查看到当前进程的情况。

4、iostat
iostat 由 Red Hat Enterprise Linux AS 发布。同时 iostat 也是 Sysstat 的一部分，可以下载到，网址是 http://perso.wanadoo.fr/sebastien.godard/
执行 iostat 命令可以从系统启动之后的 CPU 平均时间，类似于 uptime。除此之外，iostat 还对创建一个服务器磁盘子系统的活动报告。该报告包含两部分：CPU 使用情况和磁盘使用情况。
iostat 显示实例：
avg-cpu: %user %nice %system %iowait %steal %idle
0.16 0.01 0.03 0.10 0.00 99.71
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
hda 0.31 4.65 4.12 327796 290832
avg-cpu: %user %nice %system %iowait %steal %idle
1.00 0.00 0.00 0.00 0.00 100.00
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
hda 0.00 0.00 0.00 0 0
avg-cpu: %user %nice %system %iowait %steal %idle
0.00 0.00 0.00 0.00 0.00 99.01
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
hda 0.00 0.00 0.00 0 0
CPU 占用情况包括四块内容
%user：显示 user level (applications)时，CPU 的占用情况。
%nice：显示 user level 在 nice priority 时，CPU 的占用情况。
%sys:显示 system level (kernel)时，CPU 的占用情况。
%idle: 显示 CPU 空闲时间所占比例。
磁盘使用报告分成以下几个部分：
Device: 块设备的名字
tps: 该设备每秒 I/O 传输的次数。多个 I/O 请求可以组合为一个，每个 I/O 请求传输的字节数不同，因此可以将多个 I/O 请求合并为一个。
Blk_read/s, Blk_wrtn/s: 表示从该设备每秒读写的数据块数量。块的大小可以不同，如 1024, 2048 或 4048 字节，这取决于 partition 的大小。
例如，执行下列命令获得设备/dev/sda1 的数据块大小：
dumpe2fs -h /dev/sda1 |grep -F “Block size”
输出结果如下
dumpe2fs 1.34 (25-Jul-2003)
Block size: 1024
Blk_read, Blk_wrtn: 指示自从系统启动之后数据块读/写的合计数。
也可以查看这几个文件/proc/stat，/proc/partitions，/proc/diskstats 的内容。

5、vmstat
vmstat 提供了 processes, memory, paging, block I/O, traps 和 CPU 的活动状况
procs ———–memory———- —swap– —-io—- -system– —-cpu——
r b swpd free buff cache si so bi bo in cs us sy id wa st

1 0 0 513072 52324 162404 0 0 2 2 261 32 0 0 100 0 0
0 0 0 513072 52324 162404 0 0 0 0 271 43 0 0 100 0 0
0 0 0 513072 52324 162404 0 0 0 0 255 27 0 0 100 0 0
0 0 0 513072 52324 162404 0 0 0 28 275 51 0 0 97 3 0
0 0 0 513072 52324 162404 0 0 0 0 255 21 0 0 100 0 0
各输出列的含义：
Process
– r: The number of processes waiting for runtime.
– b: The number of processes in uninterruptable sleep.
Memory
– swpd: The amount of virtual memory used (KB).
– free: The amount of idle memory (KB).
– buff: The amount of memory used as buffers (KB).
Swap
– si: Amount of memory swapped from the disk (KBps).
– so: Amount of memory swapped to the disk (KBps).
IO
– bi: Blocks sent to a block device (blocks/s).
– bo: Blocks received from a block device (blocks/s).
System
– in: The number of interrupts per second, including the clock.
– cs: The number of context switches per second.
CPU (these are percentages of total CPU time)
- us: Time spent running non-kernel code (user time, including nice time).
– sy: Time spent running kernel code (system time).
– id: Time spent idle. Prior to Linux 2.5.41, this included IO-wait time.
– wa: Time spent waiting for IO. Prior to Linux 2.5.41, this appeared as zero.

6、sar
sar 是 Red Hat Enterprise Linux AS 发行的一个工具，同时也是 Sysstat 工具集的命令之一，可以从以下网址下载：http://perso.wanadoo.fr/sebastien.godard/
sar 用于收集、报告或者保存系统活动信息。sar 由三个应用组成：sar 显示数据、sar1 和 sar2 用于收集和保存数据。
使用 sar1 和 sar2，系统能够配置成自动抓取信息和日志，以备分析使用。配置举例：
在/etc/crontab 中添加如下几行内容
同样的，你也可以在命令行方式下使用 sar 运行实时报告。如图所示：
从收集的信息中，可以得到详细的 CPU 使用情况(%user, %nice, %system, %idle)、内存页面调度、网络 I/O、进程活动、块设备活动、以及 interrupts/second
liyawei:~ # sar -u 3 10
Linux 2.6.16.21-0.8-default (liyawei) 05/31/07
10:17:16 CPU %user %nice %system %iowait %idle
10:17:19 all 0.00 0.00 0.00 0.00 100.00
10:17:22 all 0.00 0.00 0.00 0.33 99.67
10:17:25 all 0.00 0.00 0.00 0.00 100.00
10:17:28 all 0.00 0.00 0.00 0.00 100.00
10:17:31 all 0.00 0.00 0.00 0.00 100.00
10:17:34 all 0.00 0.00 0.00 0.00 100.00

7、KDE System Guard
KDE System Guard (KSysguard) 是 KDE 图形方式的任务管理和性能监视工具。监视本地及远程客户端/服务器架构体系的中的主机。

8、free
/bin/free 命令显示所有空闲的和使用的内存数量，包括 swap。同时也包含内核使用的缓存。

total used free shared buffers cached
Mem: 776492 263480 513012 0 52332 162504
-/+ buffers/cache: 48644 727848
Swap: 1044184 0 1044184

9、Traffic-vis

Traffic-vis 是一套测定哪些主机在 IP 网进行通信、通信的目标主机以及传输的数据量。并输出纯文本、HTML 或者 GIF 格式的报告。

注：Traffic-vis 仅仅适用于 SUSE LINUX ENTERPRISE SERVER。

如下命令用来收集网口 eth0 的信息：

traffic-collector -i eth0 -s /root/output_traffic-collector

可以使用 killall 命令来控制该进程。如果要将报告写入磁盘，可使用如下命令：

killall -9 traffic-collector

要停止对信息的收集，执行如下命令：killall -9 traffic-collector

注意，不要忘记执行最后一条命令，否则会因为内存占用而影响性能。

可以根据 packets, bytes, TCP 连接数对输出进行排序，根据每项的总数或者收/发的数量进行。

例如根据主机上 packets 的收/发数量排序，执行命令：

traffic-sort -i output_traffic-collector -o output_traffic-sort -Hp

如要生成 HTML 格式的报告，显示传输的字节数，packets 的记录、全部 TCP 连接请求和网络中每台服务器的信息，请运行命令：

traffic-tohtml -i output_traffic-sort -o output_traffic-tohtml.html

如要生成 GIF 格式（600X600）的报告，请运行命令：

traffic-togif -i output_traffic-sort -o output_traffic-togif.gif -x 600 -y 600

GIF 格式的报告可以方便地发现网络广播，查看哪台主机在 TCP 网络中使用 IPX/SPX 协议并隔离网络，需要记住的是，IPX 是基于广播包的协议。如果 我们需要查明例如网卡故障或重复 IP 的问题，需要使用特殊的工具。例如 SUSE LINUX Enterprise Server 自带的 Ethereal。

技巧和提示：使用管道，可以只需执行一条命令来产生报告。如生成 HTML 的报告，执行命令：

cat output_traffic-collector | traffic-sort -Hp | traffic-tohtml -o output_traffic-tohtml.html

如要生成 GIF 文件，执行命令：

cat output_traffic-collector | traffic-sort -Hp | traffic-togif -o output_traffic-togif.gif -x 600 -y 600

10、pmap

pmap 可以报告某个或多个进程的内存使用情况。使用 pmap 判断主机中哪个进程因占用过多内存导致内存瓶颈。

pmap

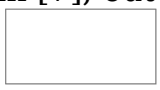
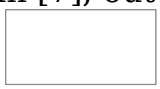
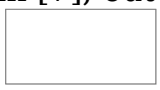
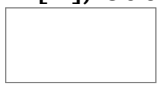
liyawei:~ # pmap 1
1: init
START SIZE RSS DIRTY PERM MAPPING
08048000 484K 244K 0K r-xp /sbin/init
080c1000 4K 4K 4K rw-p /sbin/init
080c2000 144K 24K 24K rw-p [heap]
bfb5b000 84K 12K 12K rw-p [stack]
ffffe000 4K 0K 0K —p [vdso]
Total: 720K 284K 40K
232K writable-private, 488K readonly-private, and 0K shared

11、strace

strace 截取和记录系统进程调用，以及进程收到的信号。是一个非常有效的检测、指导和调试工具。系统管理员可以通过该命令容易地解决程序问题。

使用该命令需要指明进程的 ID(PID)，例如：

strace -p
strace –p 2582

rt_sigprocmask(SIG_SETMASK, [], NULL, ) = 0
read(7, “\”\\\”\\\\\\\”\\\\\\\\\\\\\\\”\\\\\\\\\\\\\\\\\\\\\\\”…, 16384) = 321
write(3, “}H\331q\37\275$\271\t\311M\304$\317~)R9\330Oj\304\257\327″…, 360) = 360
select(8, [3 4 7], [3], NULL, NULL) = 2 (in [7], out [3])
rt_sigprocmask(SIG_BLOCK, [CHLD], [], ) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, ) = 0
read(7, “\”\\\”\\\\\\\”\\\\\\\\\\\\\\\”\\\\\\\\\\\\\\\\\\\\\\\”…, 16384) = 323
write(3, “\204\303\27$\35\206\\\306VL\370\5R\200\226\2\320^\253\253″…, 360) = 360
select(8, [3 4 7], [3], NULL, NULL) = 2 (in [7], out [3])
rt_sigprocmask(SIG_BLOCK, [CHLD], [], ) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, ) = 0
read(7, “\”\\\”\\\\\\\”\\\\\\\\\\\\\\\”\\\\\\\\\\\\\\\\\\\\\\\”…, 16384) = 323
write(3, “\243\207\204\277Cw162\2ju=\205\’L\352?0J\256I\376\32″…, 360) = 360
select(8, [3 4 7], [3], NULL, NULL) = 2 (in [7], out [3])
rt_sigprocmask(SIG_BLOCK, [CHLD], [], ) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, ) = 0
read(7, “\”\\\”\\\\\\\”\\\\\\\\\\\\\\\”\\\\\\\\\\\\\\\\\\\\\\\”…, 16384) = 320
write(3, “6\270S\3i\310\334\301\253!ys\324\’\234%\356\305\26\233″…, 360) = 360
select(8, [3 4 7], [3], NULL, NULL) = 2 (in [7], out [3])
rt_sigprocmask(SIG_BLOCK, [CHLD], [], ) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, ) = 0

12、ulimit
ulimit 内置在 bash shell 中，用来提供对 shell 和进程可用资源的控制
liyawei:~ # ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
file size (blocks, -f) unlimited
pending signals (-i) 6143
max locked memory (kbytes, -l) 32
max memory size (kbytes, -m) unlimited
open files (-n) 1024
pipe size (512 bytes, -p) 8
POSIX message queues (bytes, -q) 819200
stack size (kbytes, -s) 8192
cpu time (seconds, -t) unlimited
max user processes (-u) 6143

virtual memory (kbytes, -v) unlimited
file locks (-x) unlimited
-H 和-S 选项指明所给资源的软硬限制。如果超过了软限制，系统管理员会收到警告信息。硬限制指在用户收到超过文件句炳限制的错误信息之前，可以达到的最大值。
例如可以设置对文件句炳的硬限制：ulimit -Hn 4096
例如可以设置对文件句炳的软限制：ulimit -Sn 1024
查看软硬值，执行如下命令：
ulimit -Hn
ulimit -Sn
例如限制 Oracle 用户. 在/etc/security/limits.conf 输入以下行:
soft nofile 4096
hard nofile 10240
对于 Red Hat Enterprise Linux AS，确定文件/etc/pam.d/system-auth 包含如下行
session required /lib/security/$ISA/pam_limits.so
对于 SUSE LINUX Enterprise Server，确定文件/etc/pam.d/login 和/etc/pam.d/sshd 包含如下行：
session required pam_limits.so
这一行使这些限制生效。

13、mpstat
mpstat 是 Sysstat 工具集的一部分，下载地址是
http://perso.wanadoo.fr/sebastien.godard/
mpstat 用于报告多路 CPU 主机的每颗 CPU 活动情况，以及整个主机的 CPU 情况。
例如，下边的命令可以隔 2 秒报告一次处理器的活动情况，执行 3 次
mpstat 2 3
liyawei:~ # mpstat 2 3
Linux 2.6.16.21-0.8-default (liyawei) 05/31/07

如下命令每隔 1 秒显示一次多路 CPU 主机的处理器活动情况，执行 3 次
mpstat -P ALL 1 3
liyawei:~ # mpstat -P ALL 1 10
Linux 2.6.16.21-0.8-default (liyawei) 05/31/07
10:23:31 CPU %user %nice %sys %iowait %irq %soft %steal %idle intr/s