

管理员必备的 20 个 Linux 系统监控工具

大多数的 Linux 发行版都装备了很多 Linux 系统监控工具，充分合理的利用这些工具，可以找出系统运行的性能瓶颈，包括硬盘瓶颈、CPU 及内存瓶颈和网络瓶颈。本文介绍的 20 个工具只是最基本的，因此也十分实用。

需要监控 Linux 服务器系统性能吗？尝试下面这些系统内置或附件的工具吧。大多数 Linux 发行版本都装备了大量的监控工具。这些工具提供了能用作取得相关信息和系统活动的量度指标。你能使用这些工具发现造成性能问题可能原因。此次讨论到的工具只是分析和调试服务器下面问题时最基本工具中的一部分。

1.找出瓶颈

2.硬盘（存储）瓶颈

3.CPU 及内存瓶颈

4.网络瓶颈

#1: top - 进程活动

top 提供一个当前运行系统实时动态的视图，也就是正在运行进程。在默认情况下，显示系统中 CPU 使用率最高的任务，并每 5 秒钟刷新一次。

```
top - 04:14:53 up 1 day, 20:07, 5 users, load average: 0.53, 0.69, 0.55
tasks: 187 total, 2 running, 184 sleeping, 0 stopped, 1 zombie
cpu(s): 4.6%us, 0.5%sy, 0.0%ni, 94.8%id, 0.1%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8299944k total, 8828784k used, 279168k free, 221276k buffers
Swap: 1951888k total, 2976k used, 1948912k free, 6454792k cached
```

PID	USER	PP	NI	VT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
8532	vivek	20	0	240m	73m	25m	S	10	0.9	6:21.20	littera-bin
4575	vivek	20	0	227m	123m	10m	S	5	1.5	86:26.20	deluge
29831	vivek	20	0	768m	472m	34m	S	3	5.8	43:02.18	firefox-bin
6873	root	20	0	368m	95m	17m	S	2	1.2	88:44.31	Xorg
7268	vivek	20	0	31412	5240	3820	S	1	0.1	10:26.83	pulseaudio
12592	root	15	-5	0	0	0	S	1	0.0	10:54.52	ntos_wq
32484	vivek	20	0	84512	26m	11m	S	0	0.3	0:07.67	gnome-terminal
1	root	20	0	1900	804	452	S	0	0.0	0:01.09	init
2	root	15	-5	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	RT	-5	0	0	0	S	0	0.0	0:00.02	migration/0
4	root	15	-5	0	0	0	R	0	0.0	0:27.71	ksoftirqd/0
5	root	RT	-5	0	0	0	S	0	0.0	0:00.00	watchdog/0
6	root	RT	-5	0	0	0	S	0	0.0	0:00.01	migration/1
7	root	15	-5	0	0	0	S	0	0.0	0:06.38	ksoftirqd/1
8	root	RT	-5	0	0	0	S	0	0.0	0:00.00	watchdog/1
9	root	RT	-5	0	0	0	S	0	0.0	0:00.02	migration/2
10	root	15	-5	0	0	0	S	0	0.0	0:05.59	ksoftirqd/2
11	root	RT	-5	0	0	0	S	0	0.0	0:00.00	watchdog/2
12	root	RT	-5	0	0	0	S	0	0.0	0:00.03	migration/3
13	root	15	-5	0	0	0	S	0	0.0	0:05.05	ksoftirqd/3
14	root	RT	-5	0	0	0	S	0	0.0	0:00.00	watchdog/3
15	root	15	-5	0	0	0	S	0	0.0	0:00.27	events/0
16	root	15	-5	0	0	0	S	0	0.0	0:00.52	events/1
17	root	15	-5	0	0	0	S	0	0.0	0:00.44	events/2
18	root	15	-5	0	0	0	S	0	0.0	0:00.50	events/3
19	root	15	-5	0	0	0	S	0	0.0	0:00.01	khelper
61	root	15	-5	0	0	0	S	0	0.0	0:00.00	kintegrityd/0
62	root	15	-5	0	0	0	S	0	0.0	0:00.00	kintegrityd/1
63	root	15	-5	0	0	0	S	0	0.0	0:00.00	kintegrityd/2
64	root	15	-5	0	0	0	S	0	0.0	0:00.00	kintegrityd/3
66	root	15	-5	0	0	0	S	0	0.0	0:00.65	kblockd/0
67	root	15	-5	0	0	0	S	0	0.0	0:00.16	kblockd/1
68	root	15	-5	0	0	0	S	0	0.0	0:00.23	kblockd/2
69	root	15	-5	0	0	0	S	0	0.0	0:00.11	kblockd/3
71	root	15	-5	0	0	0	S	0	0.0	0:00.00	kacpid
72	root	15	-5	0	0	0	S	0	0.0	0:00.00	kacpi_notify
153	root	15	-5	0	0	0	S	0	0.0	0:00.00	cqueue
157	root	15	-5	0	0	0	S	0	0.0	0:00.01	kseriod
211	root	15	-5	0	0	0	S	0	0.0	0:02.25	kswapd0

图 01.Linux top 命令

常用热键

热键	用途
t	显示摘要信息开关.
m	显示内存信息开关.
A	分类显示系统不同资源的使用大户。有助于快速识别系统中资源消耗多的任务。
f	添加删除所要显示栏位.
o	调整所要显示栏位的顺序.
r	调整一个正在运行的进程 Nice 值.
k	结束一个正在运行的进程.
z	彩色/黑白显示开关

#2:vmstat -系统活动、硬件及系统信息

使用 `vmstat` 命令可以得到关于进程、内存、内存分页、堵塞 IO、traps 及 CPU 活动的信息。

```
# vmstat 3
```

输出样例：

```
procs -----memory----- ---swap-- ----io---- --system--
-----cpu-----
```

[illegible]

显示内存使用详细信息

```
# vmstat -m
```

显示内存活动/不活动的信息

```
# vmstat -a
```

#3: w - 显示谁已登录，他们正在做什么？

w 命令显示系统当前用户及其运行进程的信息。

```
# w username
```

```
# w vivek
```

输出样例：

```
17:58:47 up 5 days, 20:28, 2 users, load average: 0.36, 0.26, 0.24
```

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
root	pts/0	10.1.3.145	14:55	5.00s	0.04s	0.02s	vim /etc/resolv.conf
root	pts/1	10.1.3.145	17:43	0.00s	0.03s	0.00s	w

#4 : uptime - 告诉系统已经运行了多久？

uptime 命令过去只显示系统运行多久。现在，可以显示系统运行多久、当前有多少的用户登录、在过去的 1，5，15 分钟里平均负载时多少。

```
# uptime
```

输入样例：

```
18:02:41 up 41 days, 23:42, 1 user, load average: 0.00, 0.00, 0.00
```

1 可以被认为是最优的负载值。负载是会随着系统不同改变得。单 CPU 系统 1-3 和 SMP 系统 6-10 都是可能接受的。

#5 : ps - 显示进程

ps 命令显示当前运行进程的快照。使用 -A 或 -e 显示所有进程。

```
# ps -A
```

输出样例：

PID	TTY	TIME	CMD
1	?	00:00:02	init
2	?	00:00:02	migration/0
3	?	00:00:01	ksoftirqd/0

```

4 ?      00:00:00 watchdog/0
5 ?      00:00:00 migration/1
6 ?      00:00:15 ksoftirqd/1
....
.....
4881 ?    00:53:28 java
4885 tty1 00:00:00 mingetty
4886 tty2 00:00:00 mingetty
4887 tty3 00:00:00 mingetty
4888 tty4 00:00:00 mingetty
4891 tty5 00:00:00 mingetty
4892 tty6 00:00:00 mingetty
4893 ttyS1 00:00:00 agetty
12853 ?   00:00:00 cifsoplockd
12854 ?   00:00:00 cifsdnotifyd
14231 ?   00:10:34 lighttpd
14232 ?   00:00:00 php-cgi
54981 pts/0 00:00:00 vim
55465 ?    00:00:00 php-cgi
55546 ?    00:00:00 bind9-snmp-stat
55704 pts/1 00:00:00 ps

```

ps 与 top 非常相似，但 ps 提供更多的信息。

输出长格式

```
# ps -Al
```

输出附加全格式（显示进程在执行时传入的参数）

```
# ps -AlF
```

显示进程结构

```
# ps -AlFH
```

在进程后显示线程

```
# ps -Alm
```

打印服务器上所有进程

```
# ps ax
```

```
# ps axu
```

打印进程树

```
# ps -ejH
```

```
# ps axjf
```

```
# pstree
```

打印安全信息

```
# ps -eo euser,ruser,suser,fuser,f,comm,label
```

```
# ps axZ
```

```
# ps -eM
```

查看使用 Vivek 用户名运行的进程

```
# ps -U vivek -u vivek u
```

设置自定义输出格式

```
# ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm
```

```
# ps axo stat,euid,ruid,tt,tpgid,sess,pgrp,ppid,pid,pcpu,comm
```

```
# ps -eopid,tt,user,fname,tmout,f,wchan
```

只显示 Lighttpd 的进程 ID

```
# ps -C lighttpd -o pid=
```

或者

```
# pgrep lighttpd
```

或者

```
# pgrep -u vivek php-cgi
```

显示 PID 为 55977 的进程名称

```
# ps -p 55977 -o comm=
```

找出消耗内存最多的前 10 名进程

```
# ps -auxf | sort -nr -k 4 | head -10
```

找出使用 CPU 最多的前 10 名进程

```
# ps -auxf | sort -nr -k 3 | head -10
```

#6:free - 内存使用情况

free 命令显示系统中空闲的、已用的物理内存及 swap 内存,及被内核使用的 buffer。

```
# free
```

输出样例：

total	used	free	shared	buffers	cached	
Mem:	12302896	9739664	2563232		0	523124 5154740
-/+ buffers/cache:		4061800	8241096			
Swap:	1052248		0	1052248		

#7:iostat - CPU 平均负载，硬盘活动

iostat 命令可报告中央处理器（CPU）的统计信息，各种设备、分区及网络文件系统输入/输出的统计信息。

```
# iostat
```

输出样例：

Linux 2.6.18-128.1.14.el5 (www03.nixcraft.in) 06/26/2009

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
3.50    0.09    0.51    0.03    0.00   95.86

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
sda                 22.04         31.88         512.03    16193351    260102868
sda1                 0.00          0.00          0.00         2166         180
sda2                 22.04         31.87         512.03    16189010    260102688
sda3                 0.00          0.00          0.00         1615          0
```

#8:sar - 搜集和报告系统活动

sar 命令用来搜集、报告和储存系统活动信息。查看网路计数器，输入：

```
# sar -n DEV | more
```

显示最近 24 小时网络计数器

```
# sar -n DEV -f /var/log/sa/sa24 | more
```

你亦可以用 sar 显示实时情况

```
# sar 4 5
```

输出样例：

Linux 2.6.18-128.1.14.el5 (www03.nixcraft.in) 06/26/2009

```
06:45:12 PM    CPU   %user   %nice  %system  %iowait  %steal
%idle
```

```
06:45:16 PM    all    2.00    0.00    0.22    0.00    0.00    97.78
```

```
06:45:20 PM    all    2.07    0.00    0.38    0.03    0.00    97.52
```

```
06:45:24 PM    all    0.94    0.00    0.28    0.00    0.00    98.78
```

```
06:45:28 PM    all    1.56    0.00    0.22    0.00    0.00    98.22
```

```
06:45:32 PM    all    3.53    0.00    0.25    0.03    0.00    96.19
```

```
Average:      all    2.02    0.00    0.27    0.01    0.00    97.70
```

#9:mpstat - 多处理器使用率

mpstat 命令可以显示所有可用处理器的使用情况，处理器编号从 0 开始。mpstat -P ALL 显示每个处理器的平均使用率。

```
# mpstat -P ALL
```

输出样例：

Linux 2.6.18-128.1.14.el5 (www03.nixcraft.in) 06/26/2009

```
06:48:11 PM CPU   %user   %nice   %sys %iowait    %irq   %soft %steal
%idle   intr/s
```

```
06:48:11 PM all    3.50    0.09    0.34    0.03    0.01    0.17    0.00
95.86   1218.04
```

```
06:48:11 PM    0    3.44    0.08    0.31    0.02    0.00    0.12
0.00    96.04   1000.31
```

```
06:48:11 PM    1    3.10    0.08    0.32    0.09    0.02    0.11
0.00    96.28    34.93
```

```
06:48:11 PM    2    4.16    0.11    0.36    0.02    0.00    0.11
0.00    95.25    0.00
```

```
06:48:11 PM    3    3.77    0.11    0.38    0.03    0.01    0.24
0.00    95.46    44.80
```

```
06:48:11 PM    4    2.96    0.07    0.29    0.04    0.02    0.10
0.00    96.52    25.91
```

```
06:48:11 PM    5    3.26    0.08    0.28    0.03    0.01    0.10
0.00    96.23    14.98
```

```
06:48:11 PM    6    4.00    0.10    0.34    0.01    0.00    0.13
0.00    95.42    3.75
```

```
06:48:11 PM    7    3.30    0.11    0.39    0.03    0.01    0.46
0.00    95.69    76.89
```

#10: pmap - 进程的内存使用

pmap 命令可以显示进程的内存映射，使用这个命令可以找出造成内存瓶颈的原因。

```
# pmap -d PID
```

显示 PID 为 47394 进程的内存信息。

```
# pmap -d 47394
```

输出样例：

```
47394: /usr/bin/php-cgi
```

Address	Kbytes	Mode	Offset	Device	Mapping
0000000000400000	2584	r-x--	0000000000000000	008:00002	php-cgi
0000000000886000	140	rw---	0000000000286000	008:00002	php-cgi
00000000008a9000	52	rw---	00000000008a9000	000:00000	[anon]
0000000000aa8000	76	rw---	00000000002a8000	008:00002	php-cgi
0000000000f678000	1980	rw---	0000000000f678000	000:00000	[anon]
000000314a600000	112	r-x--	0000000000000000	008:00002	ld-2.5.so
000000314a81b000	4	r----	000000000001b000	008:00002	ld-2.5.so
000000314a81c000	4	rw---	000000000001c000	008:00002	ld-2.5.so


```

000000314aa00000 1328 r-x-- 0000000000000000 008:00002 libc-
2.5.so
000000314ab4c000 2048 ----- 0000000000014c000 008:00002 libc-
2.5.so
.....
.....
..
00002af8d48fd000 4 rw--- 00000000000006000 008:00002 xsl.so
00002af8d490c000 40 r-x-- 00000000000000000 008:00002
libnss_files-2.5.so
00002af8d4916000 2044 ----- 000000000000a000 008:00002
libnss_files-2.5.so
00002af8d4b15000 4 r---- 0000000000009000 008:00002 libnss_files-
2.5.so
00002af8d4b16000 4 rw--- 000000000000a000 008:00002
libnss_files-2.5.so
00002af8d4b17000 768000 rw-s- 0000000000000000 000:00009 zero
(deleted)
00007fffc95fe000 84 rw--- 00007fffffea000 000:00000 [ stack ]
ffffffff600000 8192 ----- 0000000000000000 000:00000 [ anon ]
mapped: 933712K writeable/private: 4304K shared: 768000K

```

最后一行非常重要：

* mapped: 933712K 内存映射所占空间大小

* writeable/private: 4304K 私有地址空间大小

* shared: 768000K 共享地址空间大小

#11 和 #12: netstat 和 ss - 网络相关信息

netstat 可以显示网络链接、路由表信息、接口统计信息、伪装链接和多播成员 (multicast memberships), ss 命令用来显示网络套接字信息，它允许显示类似 netstat 一样的信息。关于 ss 和 netstat 使用，可参考下列资源。

#13: iptraf - 网络实时信息

iptraf 是一个可交互式的 IP 网络监控工具。它可以生成多种网络统计信息包括：TCP 信息、UDP 数量、ICMP 和 OSPF 信息、以太网负载信息、节点状态、IP 校验错误等。有下面几种信息格式：

- 不同网络 TCP 链接传输量
- 不同网络接口 IP 传输量
- 不同协议网络传输量
- 不同 TCP/UDP 端口和不同包大小网络传输量
- 不同第二层地址网络传输量

IPTraf
Statistics for eth1

	Total Packets	Total Bytes	Incoming Packets	Incoming Bytes	Outgoing Packets	Outgoing Bytes
Total:	9492	5046545	4710	741047	4782	4305498
IP:	9492	4907016	4710	668466	4782	4238550
TCP:	9190	4856215	4564	658065	4626	4198150
UDP:	287	49658	144	10261	143	39397
ICMP:	15	1143	2	140	13	1003
Other IP:	0	0	0	0	0	0
Non-IP:	0	0	0	0	0	0

Total rates:	66.4 kbits/sec	Broadcast packets:	0
	27.8 packets/sec	Broadcast bytes:	0
Incoming rates:	17.8 kbits/sec		
	14.4 packets/sec		
Outgoing rates:	48.6 kbits/sec	IP checksum errors:	0
	13.4 packets/sec		

图 02：一般接口信息：不同网络接口 IP 传输量

IPTraf

IPTraf	Source Host (Port)	Pkt Size	Min Size	State	Interface
13.222.232.20:45359	--A--	52	6573	eth1	eth1
74.86.48.99:22	--A--	1500	190	eth1	eth1
74.86.48.99:80	--A--	570	137	eth1	eth1
74.86.48.99:80	--A--	587	31	eth1	eth1
102.08.150.211:11990	--A--	52	65535	eth1	eth1
75.126.168.152:80	--A--	52	99	CLOSED	eth1
157.127.124.15:62884	--A--	52	65535	eth1	eth1
74.86.48.99:80	--A--	52	40	CLOSED	eth1
74.86.48.99:80	--A--	40	6432	DONE	eth1
41.219.299.181:1834	--A--	484	17520	eth1	eth1
75.126.168.152:80	--A--	52	68	CLOSED	eth1
102.08.150.211:11990	--A--	52	65535	eth1	eth1
72.223.24.98:117089	--A--	684	16656	eth1	eth1
74.86.48.99:80	--A--	1420	20	eth1	eth1
213.47.93.180:59494	--A--	48	4125	eth1	eth1
75.126.168.153:80	--A--	52	40	CLOSED	eth1
74.86.47.178:60077	--A--	40	272	CLOSED	eth1
74.86.48.99:80	--A--	52	27	CLOSED	eth1
76.79.231.39:39049	--A--	0	0	eth1	eth1
74.86.48.99:80	--A--	52	41	eth1	eth1
17.43.20.16:65102	--A--	52	65535	eth1	eth1
74.86.48.99:80	--A--	655	27	eth1	eth1
158.230.100.182:45664	--A--	46	65535	eth1	eth1
74.86.48.99:80	--A--	0	0	eth1	eth1
24.223.36.194:64223	--A--	46	4880	eth1	eth1
74.86.48.99:80	--A--	0	39	CLOSED	eth1
74.86.48.99:80	--A--	52	31	eth1	eth1
60.177.214.106:48403	--A--	0	0	eth1	eth1
50.92.50.94:26515	--A--	54	110	eth1	eth1
74.86.48.99:80	--A--	1492	31	eth1	eth1
17.43.20.16:65106	--A--	499	65535	eth1	eth1

TCP: 92 out/1205

UDP (255 bytes) from 75.126.168.152:53 to 66.182.66.65:30459 on eth1

ICMP dest unreachable (port) (283 bytes) from 195.222.26.1 to 75.126.168.152 on eth1

UDP (74 bytes) from 64.193.228.2:45500 to 75.126.168.152:53 on eth1

UDP (300 bytes) from 75.126.168.152:53 to 64.193.228.2:45500 on eth1

UDP (74 bytes) from 208.186.134.101:32594 to 75.126.168.152:53 on eth1

UDP (280 bytes) from 75.126.168.152:53 to 208.186.134.101:32594 on eth1

UDP (74 bytes) from 154.186.235.5:25828 to 75.126.168.152:53 on eth1

UDP (280 bytes) from 75.126.168.152:53 to 154.186.235.5:25828 on eth1

UDP (64 bytes) from 213.228.63.14:26454 to 75.126.168.152:53 on eth1

UDP (280 bytes) from 75.126.168.152:53 to 213.228.63.14:26454 on eth1

UDP (64 bytes) from 66.87.68.164:24927 to 75.126.168.152:53 on eth1

UDP (319 bytes) from 75.126.168.152:53 to 66.87.68.164:24927 on eth1

UDP (74 bytes) from 24.25.4.51:2592 to 75.126.168.152:53 on eth1

UDP (280 bytes) from 75.126.168.152:53 to 24.25.4.51:2592 on eth1

UDP (54 bytes) from 213.228.63.26:3686 to 75.126.168.152:53 on eth1

UDP (286 bytes) from 75.126.168.152:53 to 213.228.63.26:3686 on eth1

UDP (72 bytes) from 218.240.240.179:40302 to 75.126.168.152:53 on eth1

UDP (327 bytes) from 75.126.168.152:53 to 218.240.240.179:40302 on eth1

Bottom: stopped time: 9:08

eth1: 92 out/1205

up/down/up/down: scroll Ltr/Rtr: ctrl scroll W:chp actv win S:port TCP X:exit

TCP flow rates: 150.00 kbits/sec

Active

图 03：不同网络 TCP 链接传输量

#14：tcpdump：详细的网络流量分析

tcpdump 是一个简单网络流量转储工具，然而要使用好需要对 TCP/IP 协议非常熟悉。例如要显示关于 DNS 的网络流量，输入：

```
# tcpdump -i eth1 'udp port 53'
```

显示所有进出 80 端口 IPv4 HTTP 包，也就是只打印包含数据的包。例如：SYN、FIN 包和 ACK-only 包输入：

```
# tcpdump 'tcp port 80 and (((ip[2:2] - ((ip[0]&0xf)<<2)) - ((tcp[12]&0xf0)>>2)) != 0)'
```

显示所有到的 FTP 会话，输入：

```
# tcpdump -i eth1 'dst 202.54.1.5 and (port 21 or 20)'
```

显示所有到 192.168.1.5 的 HTTP 会话

```
# tcpdump -ni eth0 'dst 192.168.1.5 and tcp and port http'
```

用 Wireshark 浏览转储文件中的详细信息，输入：

```
# tcpdump -n -i eth1 -s 0 -w output.txt src or dst port 80
```

#15: strace - 系统调用

追踪系统调用和型号，这对于调试 Web 服务器和其他服务器非常有用。了解怎样追踪进程和他功能。

#16: /proc 文件系统 - 各种内核信息

/proc 目录下文件提供了很多不同硬件设备和内核的详细信息。更多详情参见 Linux kernel /proc。一般 /proc 例如：

```
# cat /proc/cpuinfo
```

```
# cat /proc/meminfo
```

```
# cat /proc/zoneinfo
```

```
# cat /proc/mounts
```

#17: Nagios - 服务器及网络监控

Nagios 是一款非常流行的系统及网络监控软件。你可以轻松监控所有的主机、网络设备及服务。它能在发生故障和重新恢复后发送警讯。FAN 是 "Fully Automated Nagios" 的缩写。FAN 的目标就是由 Nagios 社群提供 Nagios 的安装。为了使安装 Nagios 服务器更加容易，FAN 提供一个标准 ISO 格式的光盘镜像。此发行版中还会包含一组增强用户使用体验的工具。

#18: Cacti - 基于 Web 的监控工具

Cacti 是一套完成的网络图形化解决方案，基于 RRDTool 的资料存储和图形化功能。Cacti 提供一个快速的轮询器、进阶的图形化模板、多种数据采集方法和用户管理功能。这些功能都拥有非常友好易用的界面，确保可以部署在一个包含数百台设备的复杂网络中。它提供关于网络、CPU、内存、已登录用户、Apache、DNS 等信息。

#19: KDE System Guard

KSysguard 是在 KDE 桌面下一个网络化的系统监控工具。这个工具可以通过 SSH 会话运行。它提供很多功能，例如可以监控本机和远程主机的客户端/服务器架构，前端图形界面使用所谓传感器得到信息并展现出来。传感器返回的可以是一个简单的数值或是一组表格的信息。针对不同的信息类型，提供一个或多个显示。这些显示被组织多个工作表中，可以工作表可以独体储存和加载。所以，KSysguard 不只是一个简单的任务管理器，还是一个可以控制多台服务器的强大工具。

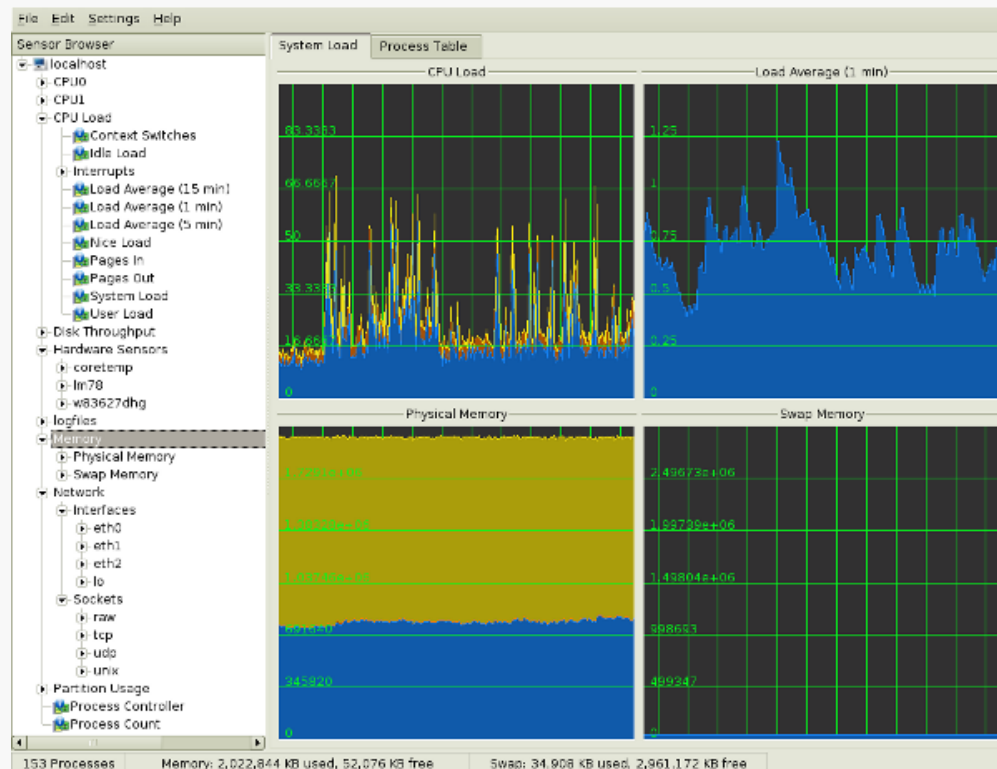


图 05 : KDE System Guard

#20:Gnome System Monitor

System Monitor 可以显示系统基本信息、监控系统进程、系统资源及文件系统使用率。你也可以使用 System Monitor 监控和修改系统行为。尽管没有 KDE System Guard 功能强大，但其提供的基本信息对于入门用户还是非常有用的。

- * 显示关于计算机硬件和软件的各种基本信息。
- * Linux 内核版本
- * GNOME 版本
- * 硬件
- * 安装的内存
- * 处理器及其速度
- * 系统状态
- * 当前可用的硬盘空间

- * 进程
- * 内存及交换空间
- * 网络使用率
- * 文件系统
- * 所有挂载的文件系统及其基本信息

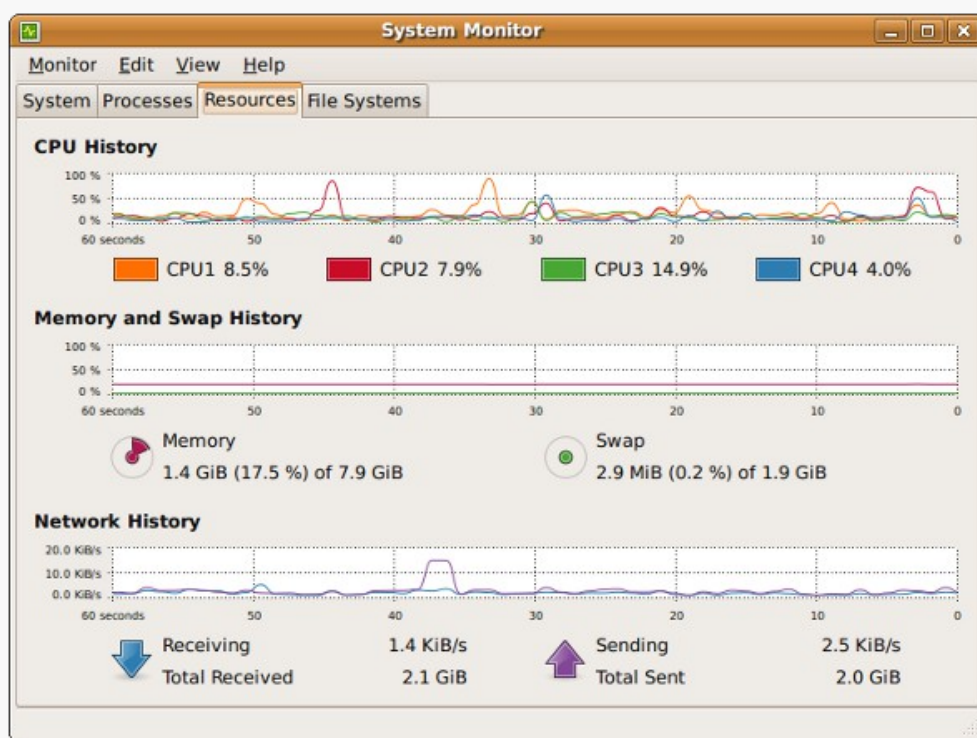


图 06 : The Gnome System Monitor application

本文来源 :

<http://hi.baidu.com/imlidapeng/blog/item/76cc8b15bf38265af2de32cc.html>