**服务器性能监控:**

**Nmon介绍**  
Nmon 工具是 IBM 提供的免费的在AIX与各种Linux操作系统上广泛使用的监控与分析工具。该工具可将服务器的系统资源耗用情况收集起来并输出一个特定的文件,并可利用 excel 分析工具nmonanalyser进行数据的统计分析。并且，nmon运行不会占用过多的系统资源，通常情况下CPU利用率不会超过2%。针对不同的操作 系统版本，nmon有相应版本的程序。

nmon是linux下一款非常好用的工具，他有以下功能：

CPU 使用率

内存使用情况

内核统计信息和运行队列信息

磁盘 I/O 速度、传输和读/写比率

文件系统中的可用空间

磁盘适配器

网络 I/O 速度、传输和读/写比率

页面空间和页面速度

CPU 和 AIX 规范

消耗资源最多的进程

IBM HTTP Web 缓存

用户自定义的磁盘组

计算机详细信息和资源

异步 I/O，仅适用于 AIX

工作负载管理器 (WLM)，仅适用于 AIX

IBM TotalStorage® Enterprise Storage Server® (ESS) 磁盘，仅适用于 AIX

网络文件系统 (NFS)

动态 LPAR (DLPAR) 更改，仅适用于面向 AIX 或 Linux 的 pSeries p5 和 OpenPower

还包括一个用来从 nmon 的输出生成图形并创建可以在 Web 站点显示的 .gif 文件的新工具。

**Nmon安装及使用**  
上传nmon\_linux\_14i.tar.gz到/usr/local/src文件夹下。  
执行以下操作：

[root@localhost /]# mkdir -p /usr/local/nmon

[root@localhost /]# cp -a /usr/local/src/nmon\_linux\_14i.tar.gz /usr/local/nmon

[root@localhost /]# /usr/local/nmon

[root@localhost nmon]# tar -zxvf nmon\_linux\_14i.tar.gz

[root@localhost nmon]# ./nmon\_x86\_64\_sles11 //启动

说明：

按C显示CPU信息，再按一次C关闭

按M显示内存

按N显示网络

按D显示硬盘

使用 -f 命令可以将 nmon 的执行结果保存到一个 CSV 文件中，用于日后分析。

更多功能可以输入h查看。

其他说明，如果在任意目录执行 ./nmon即可开启监控功能的话，需要把nmon\_x86\_64\_sles11 目录拷贝到bin目录下，具体操作如下

[root@localhost nmon]# cp nmon\_linux\_x86\_64 nmon

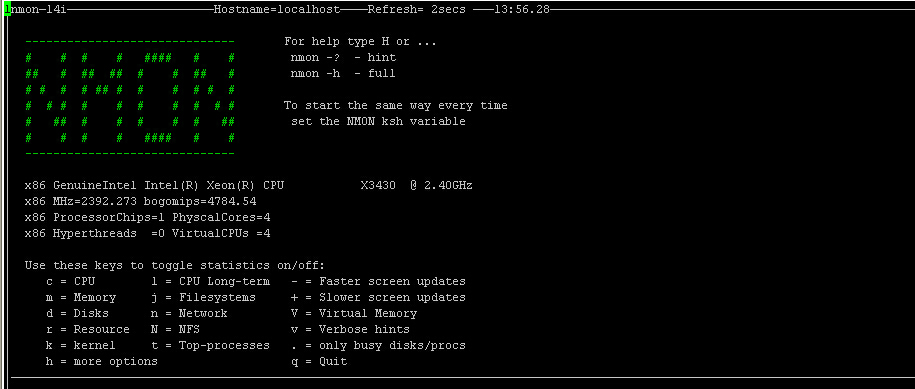
[root@localhost nmon]#cp nmon /bin/nmon

[root@localhost nmon]#cp nmon /usr/bin/nmon

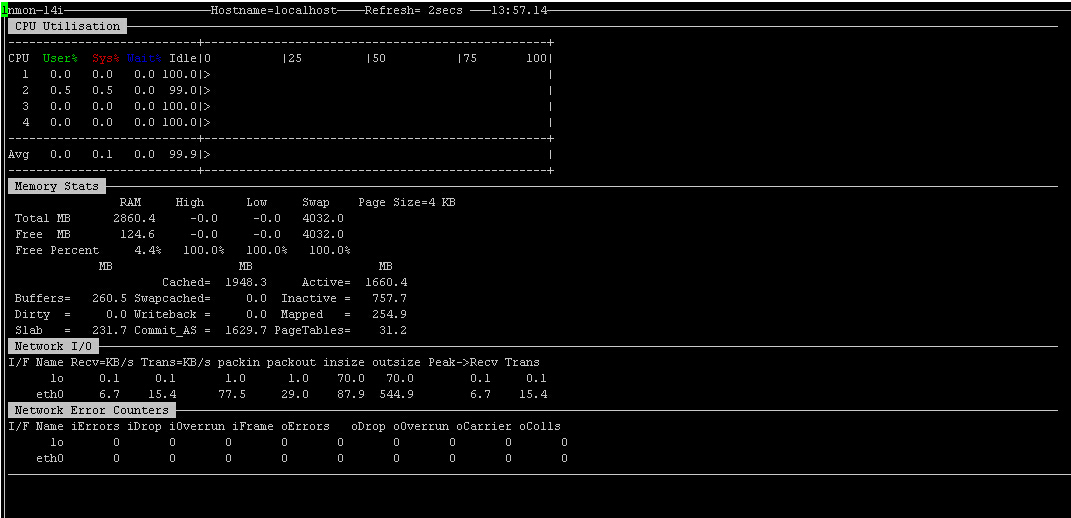
[root@localhost nmon]#cp nmon /usr/local/bin/nmon

[root@localhost nmon]#nmon //就可以启动监控程序 可以直接运行nmon

以上Nmon配置成功后，Nmon工具使用比较简单,在nmon 目录下或者其他任意目录,输入nmon  
[root@localhost nmon]#nmon

[](http://7xlhg3.com1.z0.glb.clouddn.com/uploads/article/20140616/b465c90fffd7733dd1aa24e752d3107a.jpg)

出现此画面，说明已安装成功。  
输入c可显示CPU的信息,“m”对应内存、“n”对应网络,“d”可以查看磁盘信息；“t”可以查看系统的进程信息；“

[](http://7xlhg3.com1.z0.glb.clouddn.com/uploads/article/20140616/d88a6a38ec00b30d91323f468c6ec6ef.jpg)

**数据采集**  
以上实现的是对服务器的实时监控，在实际的性能测试中我们需要把一段时间之内的数据记录下来，可以使用如下命令

[root@localhost nmon]#nmon –f –t –r test –s 30 –c 10

上面命令的含义是：

-f ：按标准格式输出文件名称：<hostname>\_YYYYMMDD\_HHMM.nmon

-t 输出最耗资源的进程

-s ：每隔n秒抽样一次，这里为30秒

-c ：取出多少个抽样数量，这里为10，即监控=10\*30/60=5分钟

test:监控记录的标题

该命令启动后，会在nmon所在目录下生成监控文件，并持续写入资源数据，直至360个监控点收集完成——即监控1小时，这些操作均自动完成，无需手工干 预，测试人员可以继续完成其他操作。如果想停止该监控，需要通过“#ps –ef|grep nmon”查询进程号，然后杀掉该进程以停止监控。

nmon 输出文件的命名方式是服务器名\_日期时间.nmon

其他信息：[root@dhdb ~]# nmon -s 300 -c 288 -f -m /tmp

-s 300:表示每300秒采集一次数据，

-c 288 :表示采集288次，300\*288=86400秒，刚好是1天的数据，这样运行一次这个程序就会生成一个一天的数据文件，

-m /tmp: 表示生成的数据文件的路径

-f ：表示生成的数据文件名中有时间

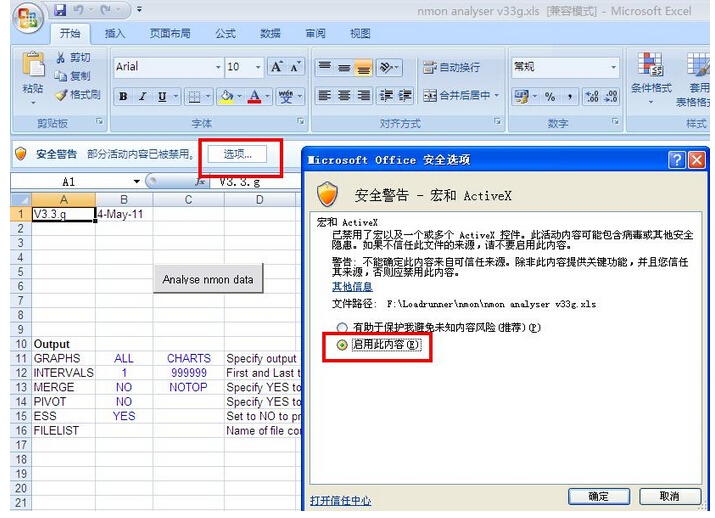
当然我们也可以让其自动监控，每天一个文件夹，每小时一个文件，每分钟或5分采样一次

通过以下命令，将.nmon文件转化成.csv文件，在当前目录生成对应的.csv文件

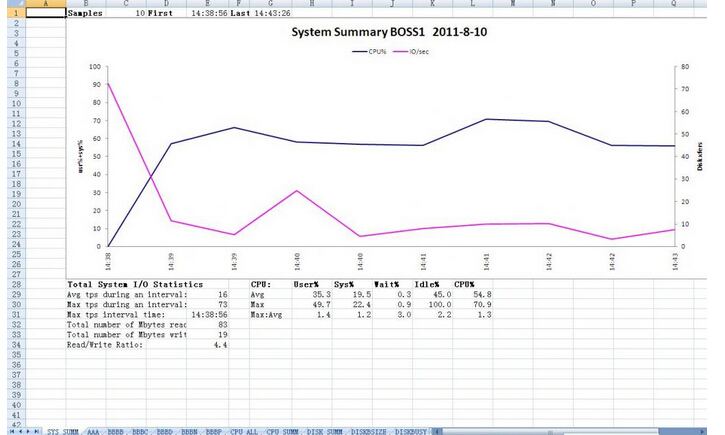
[root@localhost tmp]# sort localhost\_140616\_0737.nmon>localhost\_140616\_0737.csv

将localhost\_140616\_0737.csv文件下载到本地

**nmon analyser工具应用**  
1） 下载后打开nmon analyser v33g.xls。下载文件见附件  
  
2） 调整excel宏的安全级别，调整为最低或者如下操作

[](http://7xlhg3.com1.z0.glb.clouddn.com/uploads/article/20140616/44bdb90762ff806af8e8b99defee2093.jpg)

3） 然后点击Analyser nmon data 按钮,选择下载下来的.csv文件，然后就会转化成.excel文件，生成图形化的文件，例如：

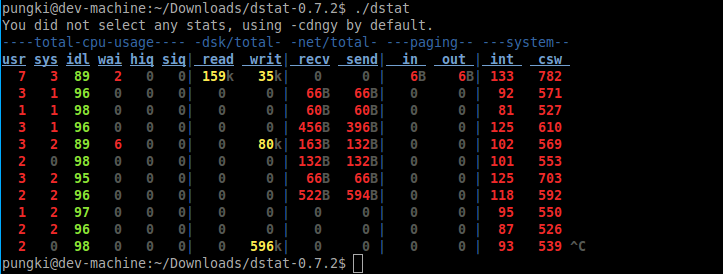
[](http://7xlhg3.com1.z0.glb.clouddn.com/uploads/article/20140616/3269d87a49490bef2b0310e414248a41.jpg)

* [nmon\_analyser.zip](http://www.flybi.net/file/download/file_name-bm1vbl9hbmFseXNlci56aXA=__url-aHR0cDovLzd4bGhnMy5jb20xLnowLmdsYi5jbG91ZGRuLmNvbS91cGxvYWRzL2FydGljbGUvMjAxNDA2MTYvMjY3NmFlNjI1N2QyNzFmMjlkZWIyNWQyY2QwNzg1MmU=)

**基于命令行的性能监控工具**

**1. dstat - 多类型资源统计工具**

该命令整合了vmstat，iostat和ifstat三种命令。同时增加了新的特性和功能可以让你能及时看到各种的资源使用情况，从而能够使你对比和整合不同的资源使用情况。通过不同颜色和区块布局的界面帮助你能够更加清晰容易的获取信息。它也支持将信息数据导出到cvs格式文件中，从而用其他应用程序打开，或者导入到数据库中。你可以用该命令来[监控cpu，内存和网络状态随着时间的变化](http://linux.cn/article-3215-1.html" \t "_blank)。



**5. sar - 性能监控和瓶颈检查**

sar 命令可以将操作系统上所选的累积活动计数器内容信息输出到标准输出上。其基于计数值和时间间隔参数的审计系统，会按照指定的时间间隔输出指定次数的监控信息。如果时间间隔参数为设置为0，那么[sar命令将会显示系统从开机到当时时刻的平均统计信息](http://linoxide.com/linux-command/linux-system-performance-monitoring-using-sar-command/" \t "_blank)。有用的命令如下：

# sar -u 2 3

# sar -u -f /var/log/sa/sa05

# sar -P ALL 1 1

# sar -r 1 3

# sar -W 1 3

**6. Saidar - 简单的统计监控工具**

Saidar是一个简单且轻量的系统信息监控工具。虽然它无法提供大多性能报表，但是它能够通过一个简单明了的方式显示最有用的系统运行状况数据。你可以很容易地看到[运行时间、平均负载、CPU、内存、进程、磁盘和网络接口](http://linoxide.com/monitoring-2/monitor-linux-saidar-tool/" \t "_blank)统计信息。

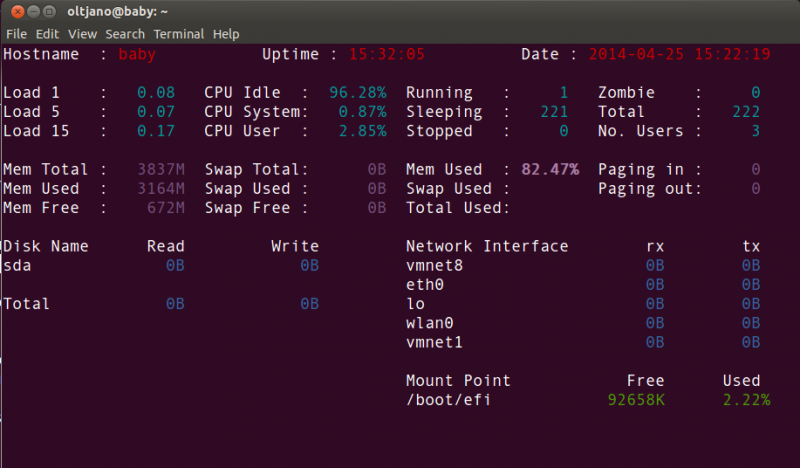
Usage: saidar [-d delay] [-c] [-v] [-h]

-d 设置更新时间（秒）

-c 彩色显示

-v 显示版本号

-h 显示本帮助



**8. Sysdig - 系统进程的高级视图**

Sysdig是一个能够让系统管理员和开发人员以前所未有方式洞察其系统行为的监控工具。其开发团队希望改善系统级的监控方式，通过提供关于存储，进程，网络和内存子系统的统一有序以及粒度可见的方式来进行错误排查，并可以创建系统活动记录文件以便你可以在任何时间轻松分析。

简单例子:

# sysdig proc.name=vim

# sysdig -p"%proc.name %fd.name" "evt.type=accept and proc.name!=httpd"

# sysdig evt.type=chdir and user.name=root

# sysdig -l

# sysdig -L

# sysdig -c topprocs\_net

# sysdig -c fdcount\_by fd.sport "evt.type=accept"

# sysdig -p"%proc.name %fd.name" "evt.type=accept and proc.name!=httpd"

# sysdig -c topprocs\_file

# sysdig -c fdcount\_by proc.name "fd.type=file"

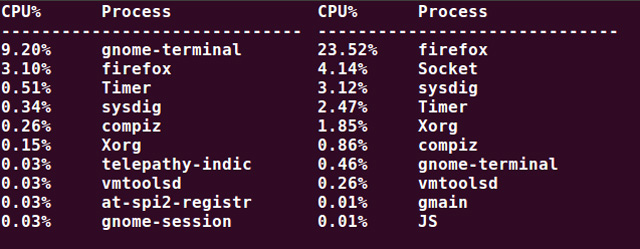
# sysdig -p "%12user.name %6proc.pid %12proc.name %3fd.num %fd.typechar %fd.name" evt.type=open

# sysdig -c topprocs\_cpu

# sysdig -c topprocs\_cpu evt.cpu=0

# sysdig -p"%evt.arg.path" "evt.type=chdir and user.name=root"

# sysdig evt.type=open and fd.name contains /etc



**9. netstat - 显示开放的端口和连接**

它是Linux管理员使用来显示各种网络信息的工具，如查看什么端口开放和什么网络连接已经建立以及何种进程运行在该连接之上。同时它也显示了不同程序间打开的Unix套接字的信息。作为大多数Linux发行版本的一部分，netstat的许多命令在[netstat和它的不同输出](http://linux.cn/article-2434-1.html" \t "_blank)中有详细的描述。最为常用的如下：

$ netstat | head -20

$ netstat -r

$ netstat -rC

$ netstat -i

$ netstat -ie

$ netstat -s

$ netstat -g

$ netstat -tapn

**10. tcpdump - 洞察网络封包**

tcpdump可以用来查看网络连接的封包内容。它显示了传输过程中封包内容的各种信息。为了使得输出信息更为有用，它允许使用者通过不同的过滤器获取自己想要的信息。可以参照的例子如下：

# tcpdump -i eth0 not port 22

# tcpdump -c 10 -i eth0

# tcpdump -ni eth0 -c 10 not port 22

# tcpdump -w aloft.cap -s 0

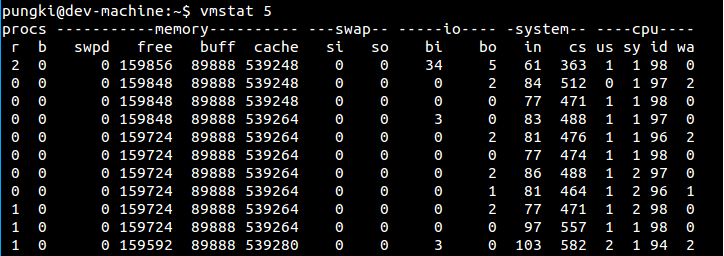
# tcpdump -r aloft.cap

# tcpdump -i eth0 dst port 80

你可以文章“[在topdump和捕捉包](http://linoxide.com/linux-how-to/network-traffic-capture-tcp-dump-command/" \t "_blank)”中找到详细描述。

**11. vmstat - 虚拟内存统计信息**

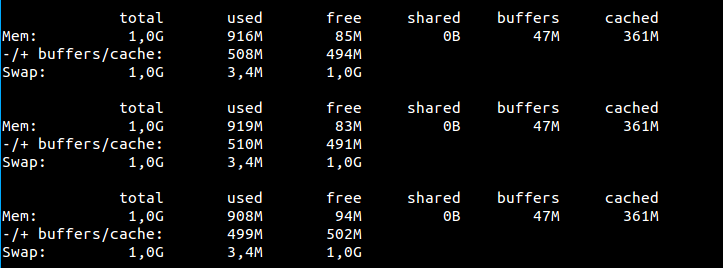
vmstat是虚拟内存(virtual memory statistics)的缩写，作为一个内存监控工具，它收集和显示关于内存，进程，终端和分页和I/O阻塞的概括信息。作为一个开源程序，它可以在大部分Linux发行版本中找到，包括Solaris和FreeBSD。它用来诊断大部分的内存性能问题和其他相关问题。



M更多信息 参考 [vmstat命令文章](http://linux.cn/article-2472-1.html" \t "_blank)。

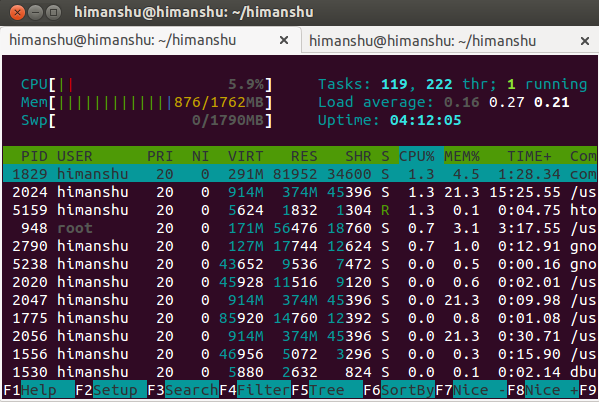
**12. free - 内存统计信息**

free是另一个能够在终端中显示内存和交换空间使用的命令行工具。由于它的简易，它经常用于快速查看内存使用或者是应用于不同的脚本和应用程序中。在这里你可以看到[这个小程序的许多应用](http://linux.cn/article-2443-1.html)。几乎所有的系统管理员日常都会用这个工具。:-)



**13. Htop - 更加友好的top**

Htop基本上是一个top改善版本，它能够以更加多彩的方式显示更多的统计信息，同时允许你采用不同的方式进行排序，它提供了一个用户友好的接口。



你可以在文章“[关于htop和top的比较](http://linux.cn/article-3141-1.html" \t "_blank)”中找到更多的信息 。

**15. lsof - 列表显示打开的文件**

lsof命令，意为“list open files”, 用于在许多类Unix系统中显示所有打开的文件及打开它们的进程。在大部分Linux发行版和其他类Linux操作系统中系统管理员用它来检查不同的进程打开了哪些文件。

# lsof +p process\_id

# lsof | less

# lsof –u username

# lsof /etc/passwd

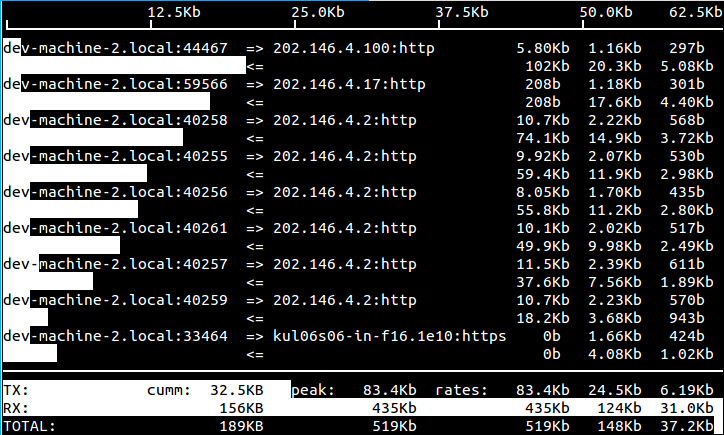
# lsof –i TCP:ftp

# lsof –i TCP:80

你可以找到 更多例子 在[lsof 文章](http://linux.cn/article-4099-1.html" \t "_blank)

**16. iftop - 类似top的了网络连接工具**

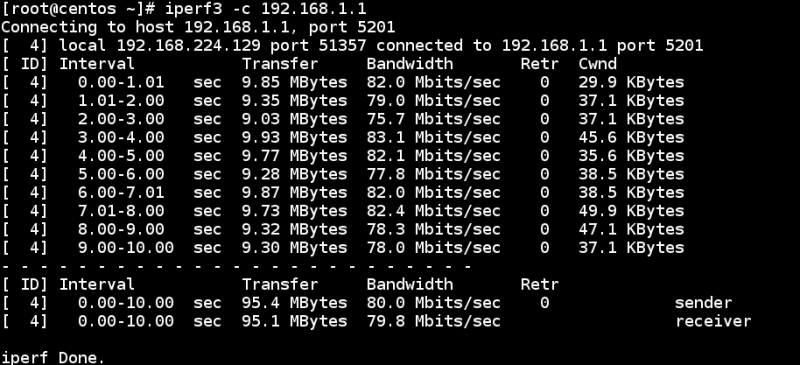
iftop是另一个基于网络信息的类似top的程序。它能够显示当前时刻按照带宽使用量或者上传或者下载量排序的网络连接状况。它同时提供了下载文件的预估完成时间。



更多信息可以参考[网络流量iftop文章](http://linux.cn/article-1843-1.html)。

**17. iperf - 网络性能工具**

iperf是一个网络测试工具，能够创建TCP和UDP数据连接并在网络上测量它们的传输性能。它支持调节关于时间，协议和缓冲等不同的参数。对于每一个测试，它会报告带宽，丢包和其他的一些参数。



如果你想用使用这个工具，可以参考这篇文章： [如何安装和使用iperf](http://linoxide.com/monitoring-2/install-iperf-test-network-speed-bandwidth/)

**18. Smem - 高级内存报表工具**

Smem是最先进的Linux命令行工具之一，它提供关于系统中已经使用的和共享的实际内存大小，试图提供一个更为可靠的当前内存使用数据。

$ smem -m

$ smem -m -p | grep firefox

$ smem -u -p

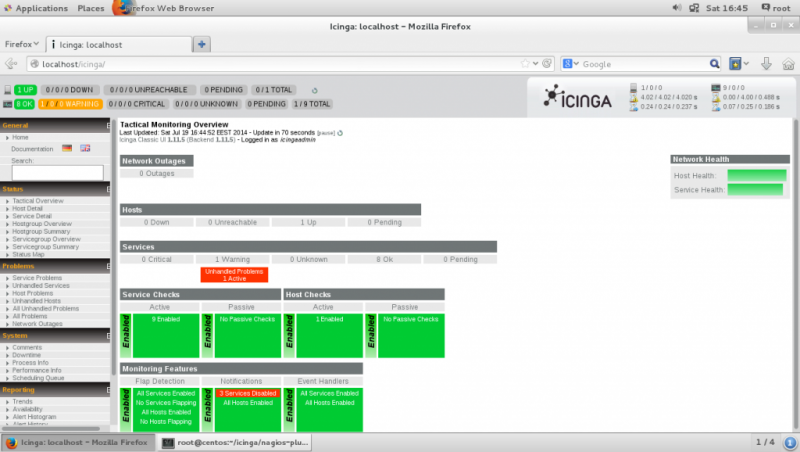
$ smem -w -p

参考我们的文章：[Smem更多的例子](http://linoxide.com/tools/memory-usage-reporting-smem/" \t "_blank)

**图形化或基于Web的性能工具**

**19. Icinga - Nagios的社区分支版本**

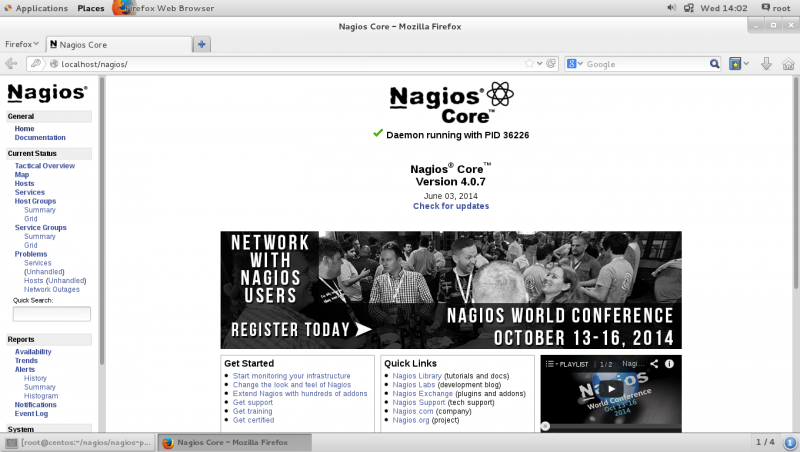
Icinga是一个开源免费的网络监控程序，作为Nagios的分支，它继承了前者现有的大部分功能，同时基于这些功能又增加了社区用户要求已久的功能和补丁。



更多信息请参考[安装和配置lcinga文章](http://linoxide.com/monitoring-2/install-configure-icinga-linux/)。

**20. Nagios - 最为流行的监控工具**

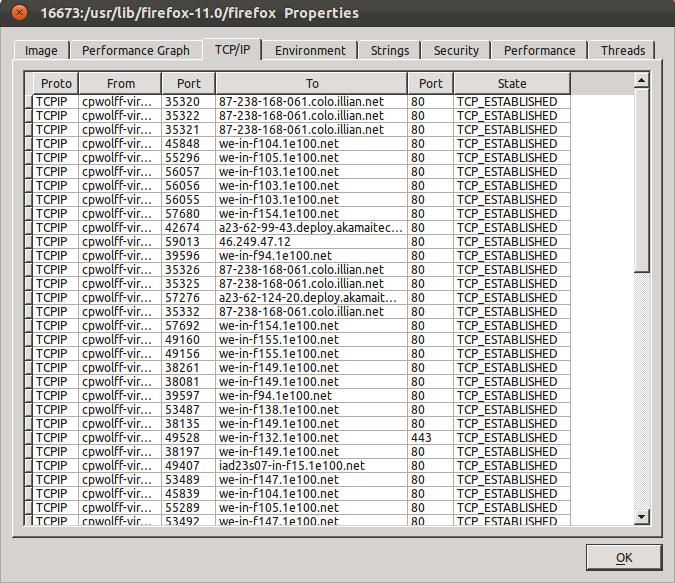
作为在Linux上使用最为广泛和最为流行的监控方案，它有一个守护程序用来收集不同进程和远程主机的信息，这些收集到的信息都通过功能强大的web界面进行呈现。



你可以在文章“[如何安装nagios](http://linux.cn/article-2436-1.html)”里面找到更多的信息

**21. Linux process explorer - Linux下的procexp**

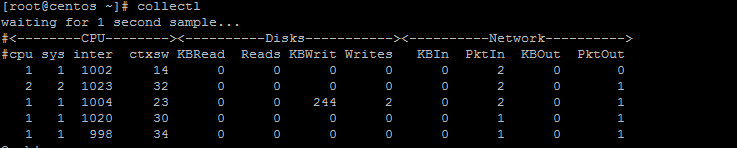
Linux process explorer是一个Linux下的图形化进程浏览工具。它能够显示不同的进程信息，如进程数，TCP/IP连接和每一个进程的性能指标。作为Windows下procexp在Linux的替代品，是由Sysinternals开发的，其目标是比top和ps提供更好用户体验。



查看 [linux process explorer 文章](http://sourceforge.net/projects/procexp/" \t "_blank)获取更多信息。

**22. Collectl - 性能监控工具**

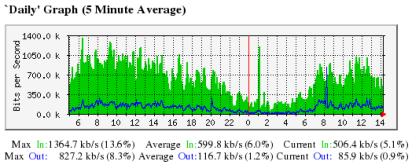
你可以既可以通过交互的方式使用这个性能监控工具，也可以用它把报表写到磁盘上，并通过web服务器来访问。它以一种易读易管理的格式，显示了CPU，磁盘，内存，网络，网络文件系统，进程，slabs等统计信息。



更多 关于[Collectl的文章](http://linux.cn/article-3154-1.html" \t "_blank)。

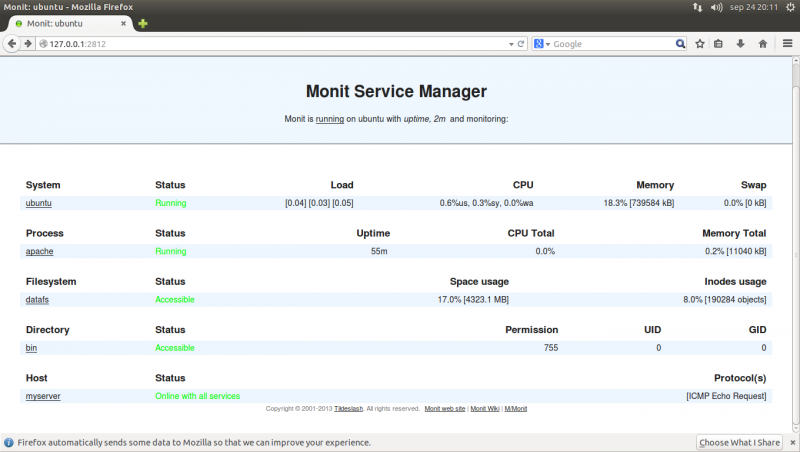
**23. MRTG - 经典网络流量监控图形工具**

这是一个采用rrdtool的生成图形的流量监控工具。作为最早的提供图形化界面的流量监控工具，它被广泛应用在类Unix的操作系统中。查看我们关于[如何使用MRTG](http://linoxide.com/tools/multi-router-traffic-grapher/" \t "_blank)的文章获取更多关于安装和配置的信息。



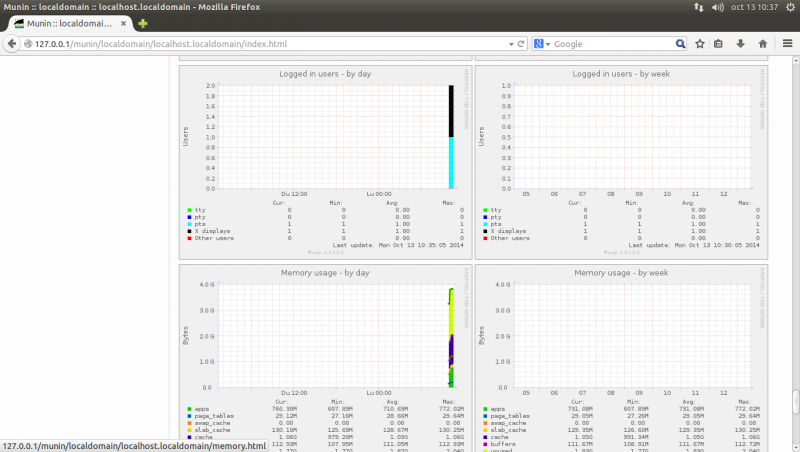
**24. Monit - 简单易用的监控工具**

Monit是一个用来监控进程，系统加载，文件系统和目录文件等的开源的Linux工具。你能够让它自动化维护和修复，也能够在运行错误的情景下执行特定动作或者发邮件报告提醒系统管理员。如果你想要用这个工具，你可以查看[如何使用Monit的文章](http://linoxide.com/monitoring-2/monit-linux/" \t "_blank)。



**25. Munin - 为服务器提供监控和提醒服务**

作为一个网络资源监控工具，**Munin能够帮助分析资源趋势和查看薄弱环节以及导致产生性能问题的原因。开发此软件的团队希望它能够易用和用户体验友好。该软件是用Perl开发的，并采用rrdtool来绘制图形，使用了web界面进行呈现。开发人员推广此应用时声称当前已有500多个监控插件可以“即插即用**\*”。

更多信息可以在[关于Munin的文章](http://linoxide.com/ubuntu-how-to/install-munin/)。

**性能测试这种测试方式在发生过程中，其中一个过渡性的[工作](http://blog.csdn.net/zhuff0418/article/details/6906862" \t "_blank)，就是对执行过程中的问题，进行定位，对功能的定位，对负载的定位，最重要的，当然就是问题中说的“瓶颈”，接触性能测试不深，更非专家，自己的理解，瓶颈产生在以下几方面：**

　　1、网络瓶颈，如带宽，流量等形成的网络环境

　　2、应用服务瓶颈，如中间件的基本配置，CACHE等

　　3、系统瓶颈，这个比较常用：应用服务器，**[数据库](http://blog.csdn.net/zhuff0418/article/details/6906862" \t "_blank)**服务器以及客户机的CPU，内存，硬盘等配置

　　4、数据库瓶颈，以ORACLE为例，SYS中默认的一些参数设置

　　5、应用程序本身瓶颈，这个是测试过程中最需要去关注的，需要测试人员和开发人员配合执行，然后定位

逐步细化分析，先可以监控一些常见衡量CPU，内存，磁盘的性能指标，进行综合分析，然后根据所测系统具体情况，进行初步问题定位，然后确定更详细的监控指标来分析。

　　怀疑内存不足时：

　　方法1：

　　【监控指标】：Memory Available MBytes ，Memory的Pages/sec， page read/sec， Page Faults/sec

　　【参考值】：

　　如果 Page Reads/Sec 比率持续保持为 5，表示可能内存不足。

　　Page/sec 推荐00-20（如果服务器没有足够的内存处理其工作负荷，此数值将一直很高。如果大于80，表示有问题）。

        方法2：根据Physical Disk 值分析性能瓶颈

　　【监控指标】：Memory Available MBytes ，Pages read/sec，%Disk Time 和 Avg.Disk Queue Length

　　【参考值】：%Disk Time建议阈值90%

　　当内存不足时，有点进程会转移到硬盘上去运行，造成性能急剧下降，而且一个缺少内存的系统常常表现出很高的CPU利用率，因为它需要不断的扫描内存，将内存中的页面移到硬盘上。

**怀疑内存泄漏时**

　　【监控指标】：Memory Available MBytes ，Process\Private Bytes和Process\Working Set，PhysicalDisk/%Disk Time

　　【说明】：

　　Windows资源监控中，如果Process\Private Bytes计数器和Process\Working Set计数器的值在长时间内持续升高，同时Memory\Available bytes计数器的值持续降低，则很可能存在内存泄漏。内存泄漏应该通过一个长时间的，用来研究分析当所有内存都耗尽时，应用程序反应情况的测试来检验。

　　CPU分析

　　【监控指标】：

　　System %Processor Time CPU，Processor %Processor Time CPU

　　Processor%user time 和Processor%Privileged Time

　　system\Processor Queue Length

　　Context Switches/sec 和%Privileged Time

　　【参考值】：

　　System\%Total processor time不持续超过90%，如果服务器专用于SQL Server，可接受的最大上限是80-85% ，合理使用的范围在60%至70%。

　　Processor %Processor Time小于75%

　　system\Processor Queue Length值，小于CPU数量的总数+1

**CPU瓶颈问题**

　　1、System\%Total processor time如果该值持续超过90%，且伴随处理器阻塞，则说明整个系统面临着处理器方面的瓶颈.

　　注：在某些多CPU系统中，该数据虽然本身并不大，但CPU之间的负载状况极不均衡，此时也应该视作系统产生了处理器方面的瓶颈.

　　2、排除内存因素，如果Processor %Processor Time计数器的值比较大，而同时网卡和硬盘的值比较低，那么可以确定CPU 瓶颈。（内存不足时，有点进程会转移到硬盘上去运行，造成性能急剧下降，而且一个缺少内存的系统常常表现出很高的CPU利用率，因为它需要不断的扫描内存，将内存中的页面移到硬盘上。）

**造成高CPU使用率的原因：**

　　频繁执行程序，复杂运算操作，消耗CPU严重

　　数据库查询语句复杂，大量的 where 子句，order by， group by 排序等，CPU容易出现瓶颈

　　内存不足，IO磁盘问题使得CPU的开销增加

　　磁盘I/O分析

　　【监控指标】：PhysicalDisk/%Disk time，PhysicalDisk/%Idle Time，Physical Disk\ Avg.Disk Queue Length， Disk sec/Transfer

　　【参考值】：%Disk Time建议阈值90%

　　Windows资源监控中，如果% Disk Time和Avg.Disk Queue Length的值很高，而Page Reads/sec页面读取操作速率很低，则可能存在磁盘瓶径。

　　Processor%Privileged Time该参数值一直很高，且如果在 Physical Disk 计数器中，只有%Disk time 比较大，其他值都比较适中，硬盘可能会是瓶颈。若几个值都比较大， 那么硬盘不是瓶颈。若数值持续超过80%，则可能是内存泄露。如果 Physical Disk 计数器的值很高时该计数器的值（Processor%Privileged Time）也一直很高， 则考虑使用速度更快或效率更高的磁盘子系统。

　　Disk sec/Transfer 一般来说，该数值小于15ms为最好，介于15-30ms之间为良好，30-60ms之间为可以接受，超过60ms则需要考虑更换硬盘或是硬盘的RAID方式了.

　　---------------------------------------------

　　Average Transaciton Response Time（事务平均响应时间）随着测试时间的变化，系统处理事务的速度开始逐渐变慢，这说明应用系统随着投产时间的变化，整体性能将会有下降的趋势

　　Transactions per Second（每秒通过事务数/TPS）当压力加大时，点击率/TPS曲线如果变化缓慢或者有平坦的趋势，很有可能是服务器开始出现瓶颈

　　Hits per Second（每秒点击次数）通过对查看“每秒点击次数”，可以判断系统是否稳定。系统点击率下降通常表明服务器的响应速度在变慢，需进一步分析，发现系统瓶颈所在。

　　Throughput（吞吐率）可以依据服务器的吞吐量来评估虚拟用户产生的负载量，以及看出服务器在流量方面的处理能力以及是否存在瓶颈。

　　Connections（连接数）当连接数到达稳定状态而事务响应时间迅速增大时，添加连接可以使性能得到极大提高（事务响应时间将降低）

　　Time to First Buffer Breakdown（Over Time）（第一次缓冲时间细分（随时间变化））可以使用该图确定场景或会话步骤运行期间服务器或网络出现问题的时间。

**碰到过的性能问题：**

　　1. 在高并发的情况下，产生的处理失败（比如：数据库连接池过低，服务器连接数超过上限，数据库锁控制考虑不足等）

　　2. 内存泄露（比如：在长时间运行下，内存没有正常释放，发生宕机等）

　　3. CPU使用偏离（比如：高并发导致CPU使用率过高）

　　4. 日志打印过多，服务器无硬盘空间

**如何定位这些性能问题：**

　　1. 查看系统日志，日志是定位问题的不二法宝，如果日志记录的全面，很容易通过日志发现问题。

　　比如，系统宕机时，系统日志打印了某方法执行时抛出out of memory的错误，我们就可以顺藤摸瓜，很快定位到导致内存溢出的问题在哪里。

　　2. 利用性能监控工具，比如：JAVA开发B/S结构的项目，可以通过JDK自带的Jconsole，或者JProfiler，来监控服务器性能，Jconsole可以远程监控服务器的CPU，内存，线程等状态，并绘制变化曲线图。

　　利用Spotlight可以监控数据库使用情况。

　　我们需要关注的性能点有：CPU负载，内存使用率，网络I/O等

　　3. 工具和日志只是手段，除此之外，还需要设计合理的性能测试场景

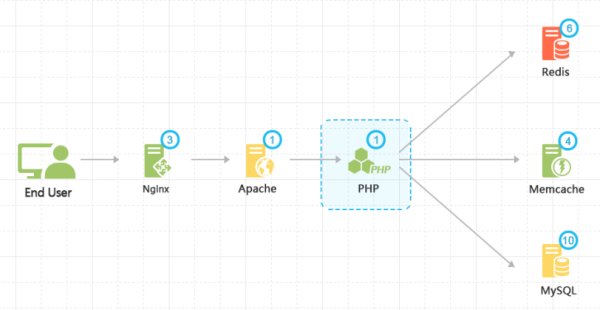
　　具体场景有：性能测试，负载测试，压力测试，稳定性测试，浪涌测试等

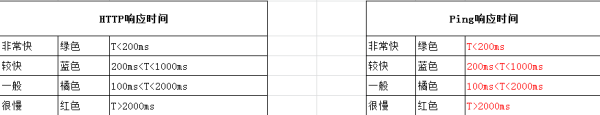
　　好的测试场景，能更加快速的发现瓶颈，定位瓶颈

　　4. 了解系统参数配置，可以进行后期的性能调优

        最后要说的是：做性能测试的时候，我们一定要确保瓶颈不要发生在我们自己的测试脚本和测试工具上。

性能问题服务器、主机、应用、代码、数据库、网络等各种影响，所以要定位问题是不容易的。最好是使用监测的工具，从IT环境的整体布局来考虑，定位到性能的瓶颈。zabbix、newrelix,国内的监控宝都可以考虑。

&amp;lt;img src=&quot;https://pic4.zhimg.com/cbb4e690554d971cad94287bc6f6fde7\_b.png&quot; data-rawwidth=&quot;812&quot; data-rawheight=&quot;419&quot; class=&quot;origin\_image zh-lightbox-thumb&quot; width=&quot;812&quot; data-original=&quot;https://pic4.zhimg.com/cbb4e690554d971cad94287bc6f6fde7\_r.png&quot;&amp;gt;当然，如果知道一些数据标准的话，对定位性能瓶颈是非常有帮助的。还是可以从第三方的工具中获得一些标准。比如这个：当然，如果知道一些数据标准的话，对定位性能瓶颈是非常有帮助的。还是可以从第三方的工具中获得一些标准。比如这个：

&amp;lt;img src=&quot;https://pic3.zhimg.com/2dd80a93014524ddac29020610936512\_b.png&quot; data-rawwidth=&quot;874&quot; data-rawheight=&quot;168&quot; class=&quot;origin\_image zh-lightbox-thumb&quot; width=&quot;874&quot; data-original=&quot;https://pic3.zhimg.com/2dd80a93014524ddac29020610936512\_r.png&quot;&amp;gt;

这张表是我从监控宝抄下来的有关响应时间的阈值。从一定程度上可以反应用户的容忍程度，若测试是发现响应时间超出了这个范围，就是你的瓶颈了，改做相应的改善。