

第十一章 Linux 系统调优相关工具

本节所讲内容：

- 11.1 系统调优概述
- 11.2 查看 CPU 负载相关工具，找出系统中使用 CPU 最多的进程
- 11.3 查看 Memory 运行状态相关工具，找出系统中使用内存最多的进程
- 11.4 查看 IO 运行状态相关工具，找出系统中对磁盘读写最多的进程
- 11.5 查看 Network 运行状态相关工具，找出系统中使用网络最多的进程
- 11.6 查看系统整体运行状态

11.1 系统调优概述

11.1.1 系统调优

1. 系统的运行状况： CPU -> MEM -> DISK*-> NETWORK -> 应用程序调优
2. 分析是否有瓶颈(依据当前应用需求)
3. 调优(把错误的调正确)

性能优化就是找到系统处理中的瓶颈以及去除这些的过程。

性能优化其实是对 OS 各子系统达到一种平衡的定义,这些子系统包括了:

- CPU
- Memory
- IO
- Network

这些子系统之间关系是相互彼此依赖的，任何一个高负载都会导致其他子系统出现问题。

比如：

大量的网页调入请求导致内存队列的拥塞；

网卡的大吞吐量可能导致更多的 CPU 开销；

大量的 CPU 开销又会尝试更多的内存使用请求；

大量来自内存的磁盘写请求可能导致更多的 CPU 以及 IO 问题；

所以要对一个系统进行优化,查找瓶颈来自哪个方面是关键,虽然看似是某一个子系统出现问题,其实有可能是别的子系统导致的。

调优就像医生看病，因此需要你对服务器所有地方都了解清楚。

当系统出了问题，运行卡，如何快速找出以下进程：

- 1、找出系统中使用 CPU 最多的进程？
- 2、找出系统中使用内存最多的进程？
- 3、找出系统中对磁盘读写最多的进程？
- 4、找出系统中使用网络最多的进程？

11.2 查看 CPU 负载相关工具，找出系统中使用 CPU 最多的进程

11.2.1 查看 CPU 负载相关工具

实战 1：找出系统中使用 CPU 最多的进程？

uptime 命令：

```
[root@localhost ~]# uptime
```

```
13:22:30 up 8 min, 1 users, load average: 0.14, 0.38, 0.25
```

其内容如下：

12:38:33	当前时间
up 8 min	系统运行时间，说明此服务器连续运行 8 分钟
1 user	当前登录用户数
load average: 0.06, 0.60, 0.48	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。

例 1：找出当前系统中，CPU 负载过高的服务器？

服务器 1：load average: 0.15, 0.08, 0.01 1 核

服务器 2：load average: 4.15, 6.08, 6.01 1 核

服务器 3：load average: 10.15, 10.08, 10.01 4 核

答案：服务器 2

如果服务器的 CPU 为 1 核心，则 load average 中的数字 ≥ 3 负载过高，如果服务器的 CPU 为 4 核心，则 load average 中的数字 ≥ 12 负载过高。

经验：单核心，1 分钟的系统平均负载不要超过 3，就可以，这是个经验值。

任务队列的平均长度是什么？

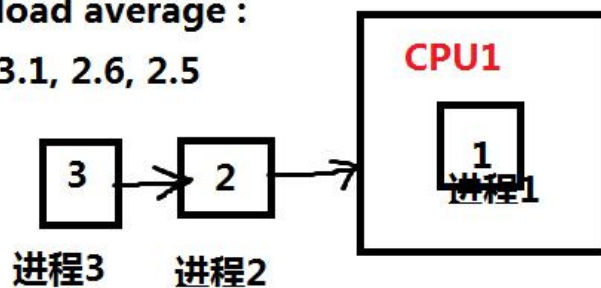
大厅排除买票：



队列数为 3 时，如图：

load average :

3.1, 2.6, 2.5



11.2.2 实战：找出系统中使用 CPU 最多的进程？

方法 1：使用 top 命令

运行 top，找出使用 CPU 最多的进程，按大写的 P，可以按 CPU 使用率来排序显示

Swap: 1023992k total, 0k used, 1023992k free, 2/4516k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2276	root	20	0	163m	29m	7692	R	5.0	2.6	0:27.98	Xorg
3107	root	20	0	637m	15m	10m	S	2.3	1.4	0:05.12	gnome-terminal
3363	root	20	0	15088	1308	956	R	0.7	0.1	0:00.09	top

11.2.3 实战：按照实际使用 CPU，从大到小排序显示所有进程列表

[root@xuegod63 ~]# ps -aux --sort -pcpu | more #按 cpu 降序排序

查看

注：-pcpu 可以显示出进程绝对路径，方便找出木马程序运行的路径。

查看 -pcpu 帮助：

[root@xuegod63 ~]# man ps #然后搜索 pcpu，找到下面内容

pcpu cpu utilization # cpu 利用率

查看 CPU 信息：

[root@xuegod63 ~]# cat /proc/cpuinfo

processor: 0

vendor_id: GenuineIntel

cpu family : 6

model : 42

model name : Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz

stepping : 7

cpu MHz : 2195.039

cache size : 6144 KB

physical id : 0

siblings : 4

core id : 0

cpu cores: 4

apicid : 0

initial apicid : 0

```

fpu      : yes
fpu_exception : yes
cpuid level  : 13
wp       : yes
flags    : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36
clflush dts mmx fxsr sse sse2 ss ht syscall nx rdtscp lm constant_tsc arch_perfmon pebs bts
xtopology tsc_reliable nonstop_tsc aperfmperf unfair_spinlock pni pclmulqdq ssse3 cx16
sse4_1 sse4_2 x2apic popcnt aes xsave avx hypervisor lahf_lm ida arat epb xsaveopt pln pts
dts

```

```

bogomips    : 4390.07
clflush size : 64
cache_alignment : 64
address sizes : 40 bits physical, 48 bits virtual
power management:

```

processor: 1

...

processor: 2

...

processor: 3 #说明本机器支持 4 个 CPU

扩展: top 中 VIRT、RES 和 SHR 的含意:

VIRT: virtual memory usage 虚拟内存

- 1、进程“需要的”虚拟内存大小，包括进程使用的库、代码、数据等
- 2、假如进程申请 100m 的内存，但实际只使用了 10m，那么它会增长 100m，而不是实际的使用

量

RES: resident memory usage 常驻内存

- 1、进程当前使用的内存大小，但不包括 swap out
- 2、包含其他进程的共享
- 3、如果申请 100m 的内存，实际使用 10m，它只增长 10m，与 VIRT 相反
- 4、关于库占用内存的情况，它只统计加载的库文件所占内存大小

SHR: shared memory 共享内存

- 1、除了自身进程的共享内存，也包括其他进程的共享内存
- 2、虽然进程只使用了几个共享库的函数，但它包含了整个共享库的大小
- 3、计算某个进程所占的物理内存大小公式: RES - SHR
- 4、swap out 后，它将会降下来

11.2.4 CPU 专用的 mpstat

mpstat ---> CPU 运行情况

```
[root@xuegod63 ~]# rpm -qf `which mpstat`
```

sysstat-9.0.4-18.el6.x86_64 #此软件包，包括很多查看系统状态的软件包

```
[root@xuegod63 ~]# mpstat
```

```
[root@xuegod63 ~]# mpstat
Linux 2.6.32-220.el6.x86_64 (xuegod63.cn)      08/04/2015      _x86_64_      (4 CPU)

09:46:18 PM  CPU      %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
09:46:18 PM  all       0.49    0.01    0.53    0.75    0.00    0.04    0.00    0.00   98.17
[root@xuegod63 ~]#
```

usr:用户空间 CPU 使用占比

nice:低优先级进程使用 CPU 占比。nice 值大于 0。

sys:内核空间 CPU 使用占比

iowait:CPU 等待 IO 占比

irq:CPU 处理硬中断占比

soft:CPU 处理软中断占比

idle:CPU 空闲时间占比

guest 与 steal 与虚拟机有关，暂不涉及。

[root@stu003 ~]# mpstat -P ALL #查看所有 CPU 运行状态

```
[root@xuegod63 ~]# mpstat -P ALL
Linux 2.6.32-220.el6.x86_64 (xuegod63.cn)      08/04/2015      _x86_64_      (4 CPU)

09:45:49 PM  CPU      %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
09:45:49 PM  all       0.49    0.01    0.53    0.75    0.00    0.04    0.00    0.00   98.17
09:45:49 PM   0       0.53    0.02    0.74    0.68    0.00    0.05    0.00    0.00   97.97
09:45:49 PM   1       0.91    0.00    0.50    0.38    0.01    0.09    0.00    0.00   98.11
09:45:49 PM   2       0.39    0.02    0.63    1.77    0.00    0.02    0.00    0.00   97.17
09:45:49 PM   3       0.14    0.00    0.27    0.18    0.00    0.00    0.00    0.00   99.41
```

[root@stu003 ~]# mpstat -P ALL 1 100 #一秒钟刷新一次 连续刷新 100 次

11.3 查看 Memory 运行状态相关工具, 找出系统中使用内存最多的进程

11.3.1 查看 Memory 运行状态相关工具

RHEL 7

[root@xuegod64 ~]# free -h

	total	used	free	shared	buff/cache	available
Mem:	976M	291M	156M	7.3M	527M	479M
Swap:	2.0G	0B	2.0G			

RHEL 6

[root@xuegod64 ~]# free -m

	total	used	free	shared	buffers	cached
Mem:	988	883	105	4	177	308
-/+ buffers/cache:		398	590			
Swap:	2047	0	2047			

注解:

Mem: 物理内存

total	容量总数
used	已使用容量
free	空余容量
shared	
buffers	缓冲, 通常缓冲元数据
cached	缓存, 通常缓存数据

-/+ buffers/cache: 398 (物理内存实际使用) 590 (物理内存实际空余)

物理内在实际使用计算: used - buffers/cache

883 - 177 - 308 = 398

物理内存实际空余计算: free + buffers/cache

105 + 177 + 308 = 590

Swap:	交换空间
2047	容量总数
0	已使用容量
2047	空余容量

buffers #缓存从磁盘读出的内容

cached #缓存需要写入磁盘的内容

当物理内存不够用的时候, 内核会把非活跃的数据清空。

通过/proc 目录, 查看非活跃的内存:

/proc 文件系统下的多种文件提供的系统信息不是针对某个特定进程的, 而是能够在整个系统范围的上下文中使用。可以使用的文件随系统配置的变化而变化。

[root@xen ~]# cat /proc/meminfo

MemTotal: 1720320 kB

MemFree: 909000 kB

Buffers: 41500 kB

Cached: 385016 kB

SwapCached: 0 kB

Active: 347340 kB 活跃内存, 指进程一直读写的内存空间

Inactive: 346992 kB 非活跃内存

注: 当内存不够用时, kernel 总是把不活跃的内存交换到 swap 空间。如果 inactive 内存多时, 加 swap 空间可以解决问题, 而 active 多, 则考虑加内存。

HighTotal: 981512 kB

HighFree: 318568 kB

LowTotal: 738808 kB

11.3.2 实战: 找出系统中使用内存最多的进程?

方法 1: 使用 top 命令

运行 top , 然后按下大写的 M 可以按内存使用率来排序显示


```
top - 10:21:17 up 26 min, 3 users, load average: 0.09, 0.22, 0.58
Tasks: 191 total, 2 running, 189 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.9%us, 0.5%sy, 0.0%ni, 98.6%id, 0.1%wa, 0.0%hi, 0.0%si,
Mem: 1164636k total, 556464k used, 608172k free, 31804k buffers
Swap: 1023992k total, 0k used, 1023992k free, 270376k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2276	root	20	0	164m	27m	7692	S	3.6	2.4	0:27.28	Xorg
2614	root	20	0	436m	26m	19m	S	0.3	2.3	0:08.29	vmtoolsd
2598	root	20	0	594m	20m	13m	S	0.0	1.8	0:11.16	nautilus
2638	root	20	0	324m	19m	9836	S	0.0	1.7	0:00.56	python
2107	root	20	0	627m	17m	10m	S	1.2	1.6	0:04.70	gnome-termi

方法 2:

11.3.4 实战: 按照实际使用内存, 从大到小排序显示所有进程列表

```
[root@xuegod63 ~]# ps -aux --sort -rss | more 内存降序排序(去掉减号就是升序)
```

或:

```
[root@xuegod63 ~]# ps -aux --sort -rss > a.log
```

11.4 查看 IO 运行状态相关工具, 找出系统中对磁盘读写最多的进程

11.4.1 I/O 调优相关查看工具

查看系统块大小

RHEL 6 (ext 2.3.4)

```
[root@xuegod63 ~]# tune2fs -l /dev/sda1 | grep size
```

Filesystem features: has_journal ext_attr resize_inode dir_index filetype needs_recovery extent flex_bg sparse_super huge_file uninit_bg dir_nlink extra_isize

Block size: 1024 # 为 1 个字节。一个扇区 512 字节。

RHEL 7

```
[root@xuegod63 ~]# xfs_growfs -l /dev/sda1 | grep bsize
```

data = bsize=4096 blocks=51200, imaxpct=25

naming =version 2 bsize=4096 ascii-ci=0 ftype=1

log =internal bsize=4096 blocks=855, version=2



11.4.2 找出系统中对磁盘读写最多的进程

实战 1: 查看系统中哪个磁盘或分区最繁忙?

对于这个服务器，8 块磁盘中，哪个硬盘最繁忙？哪个分区最繁忙？

通过 iostat 命令查看 IO 是否存在瓶颈

安装 iostat:

```
[root@xuegod63 ~]# rpm -ivh /mnt/Packages/sysstat-10.1.5-12.el7.x86_64.rpm
```

常用参数:

-d 仅显示磁盘统计信息

-k 以 K 为单位显示每秒的磁盘请求数,默认单位块.

-p device | ALL 用于显示块设备及系统分区的统计信息.

```
[root@xuegod63 ~]# iostat -d -k -p /dev/sda
```

```
Linux 2.6.32-220.el6.x86_64 (xuegod63.cn) 03/20/2017 _x86_64_ (4 CPU)
```

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sda	6.64	167.01	18.53	301418	33437
sda1	0.33	1.28	0.00	2317	9
sda2	6.03	164.53	18.52	296949	33428
sda3	0.18	0.80	0.00	1452	0

注:

每列含意:

kB_read/s 每秒从磁盘读入的数据量,单位为 K.

kB_wrtn/s 每秒向磁盘写入的数据量,单位为 K.

kB_read 读入的数据总量,单位为 K.

kB_wrtn 写入的数据总量,单位为 K.

测试: 给磁盘写入一些内容, 写入时尽可能不读磁盘?

```
[root@xuegod63 ~]# dd if=/dev/zero of=a.txt bs=10M count=1000 ; sync
```

读入的数据用/dev/zero, /dev/zero 不会读磁盘的。

sync #把内存中的数据快速写到磁盘上。 只做 dd 不执行 sync, 不容易看不出写入效果

这一招学到手: 1

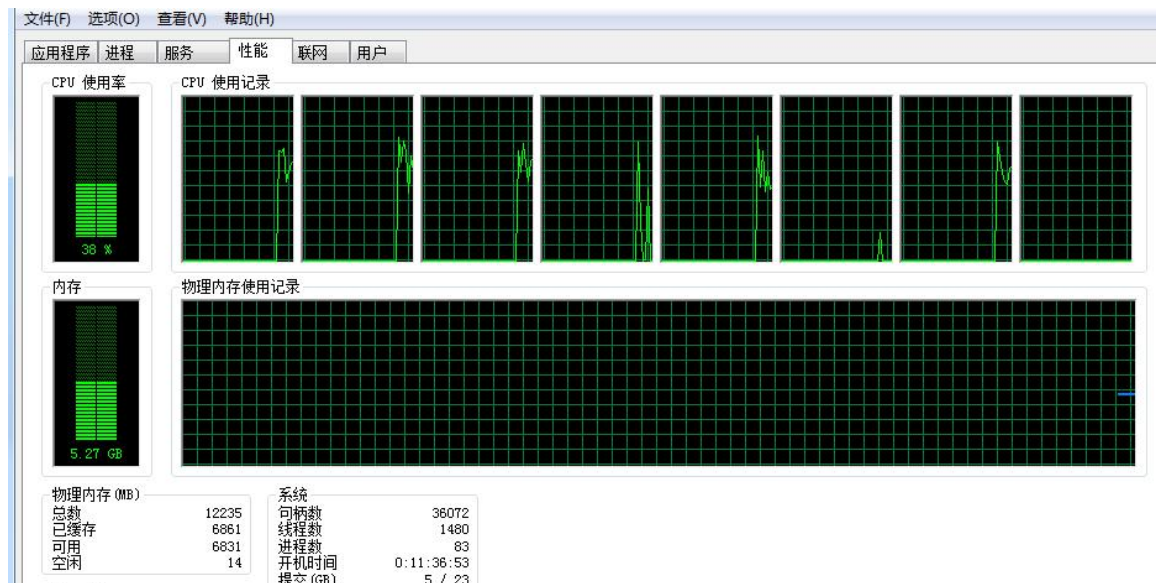
```
[root@xuegod63 ~]# iostat -p sda -dk
```

```
Linux 2.6.32-220.el6.x86_64 (xuegod63.cn) 03/03/2017 _x86_64_ (4 CPU)
```

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sda	6.07	128.02	463.04	298190	1078577
sda1	0.25	0.99	0.00	2317	9
sda2	5.60	126.10	463.04	293721	1078568

11.4.3 实战:

实战场景: 公司的服务器很卡, 查看 CPU 使用率不高, 内存也够用, 但就是卡, 尤其是打开新程序或文件时, 更卡。此时是哪出了问题了?



这时系统的瓶颈在哪里?

硬盘

查看哪个进程使用磁盘读写最多?

iotop 命令, 查看哪个进程使用磁盘读写最多

安装:

```
[root@xuegod63 ~]# rpm -ivh /mnt/Packages/iotop-0.6-2.el7.noarch.rpm
```

参数:

-o, -only 只显示在读写硬盘的程序

-d SEC, -delay=SEC 设定显示时间间隔。刷新时间

退出, 按 q 或 ctrl+C

例:

```
[root@xuegod63 ~]# iotop -o -d 1 #显示正在使用磁盘的进程
```

在另一个终端对磁盘进行大量读操作, 执行:

```
[root@xuegod63 ~]# find /
```

查看结果:

```
[root@xuegod63 ~]# iotop
Total DISK READ: 1458.61 K/s | Total DISK WRITE: 0.00 B/s
  TID  PRIO  USER    DISK READ  DISK WRITE  SWAPIN     IO>     COMMAND
  2654 be/4  root    1458.61 K/s    0.00 B/s    0.00 %    60.62 % find /
    1 be/4  root      0.00 B/s    0.00 B/s    0.00 %     0.00 % init
```

选项:

-o: 只显示有 io 操作的进程

-b: 批量显示, 无交互, 主要用作记录到文件。

-n NUM: 显示 NUM 次, 主要用于非交互式模式。

-d SEC: 间隔 SEC 秒显示一次。

-p PID: 监控的进程 pid。

-u USER: 监控的进程用户

iotop 常用快捷键:

<- / ->: 左右箭头: 改变排序方式, 默认是按 IO 排序。

r: 改变排序顺序。

o: 只显示有 IO 输出的进程。

p: 进程/线程的显示方式的切换。

a: 显示累积使用量。

q: 退出。

11.5 查看 Network 运行状态相关工具, 找出系统中使用网络最多的进程

11.5.1 查看 Network 运行状态相关工具

```
[root@xuegod63 ~]# netstat -antup
```

11.5.2 找出系统中使用网络最多的进程?



windows 中有流量显示, Linux 有吗?

双击后, 可以查到每个进程使用的流量。

电脑管家-网络优化				
流量监控 测试网速 网络修复				
当前有 32 个程序正在或已访问过网络, 下载速度: 9.6KB/S, 上传速度: 0.2KB/S				
名称	↓ 下载速度	↑ 上传速度	占用程度	操作
▶ 运行的程序				
et.exe	0KB/S	0KB/S	低	禁用网络
wpscloudsvr.exe	0KB/S	0KB/S	低	禁用网络
qqprotect.exe	0KB/S	0KB/S	低	禁用网络
vpxclient.exe	0KB/S	0KB/S	低	禁用网络
foxmail.exe	0KB/S	0KB/S	低	禁用网络
itrus.exe	0KB/S	0KB/S	低	禁用网络
查看更多(22项) ✓				
已禁用的程序				
kagent.exe	0KB/S	0KB/S	无	恢复连接

只是对 Linux 下的开源工具做了一个封装。

11.5.3 实战：使用 nload 监控总体带宽使用情况

rz 上传 nload.rpm 到 centos7 上

```
[root@xuegod63 ~]# rpm -ivh nload-0.7.4-1.el7.x86_64.rpm
```

或

```
[root@iZ25b6alxstZ ~]# yum install nload -y    (在 centos 上, 配置好 epel 源)
```

```
[root@xuegod63 ~]# nload    #开始监控
```

然后另一个终端上运行:

```
[root@xuegod63 ~]# rpm -ivh
```

```
/mnt/Packages/httpd-tools-2.4.6-67.el7.centos.x86_64.rpm
```

```
[root@xuegod63 ~]# ab -n 1000 -c 2 http://www.baidu.com/index.html    #产生一些测试
```

数据

查看结果:

```
Device ens33 [192.168.1.64] (1/4):
=====
Incoming:
#####|###|#####|###|
#####.#####
#####.#####
#####.##### Curr: 9.73 MBit/s
#####.##### Avg: 4.29 MBit/s
#####.##### Min: 568.00 Bit/s
#####.##### Max: 11.42 MBit/s
#####.##### Ttl: 627.54 MByte
Outgoing:
#####|###|#####|###|
#####.#####
#####.#####
#####.##### Curr: 230.27 kBit/s
#####.##### Avg: 115.05 kBit/s
#####.##### Min: 3.88 kBit/s
#####.##### Max: 374.00 kBit/s
#####.##### Ttl: 20.29 MByte
```

当前速度为: 9.73MBit/s 那么公司使用的带宽是几 M 的?

当前带宽就是: 10M

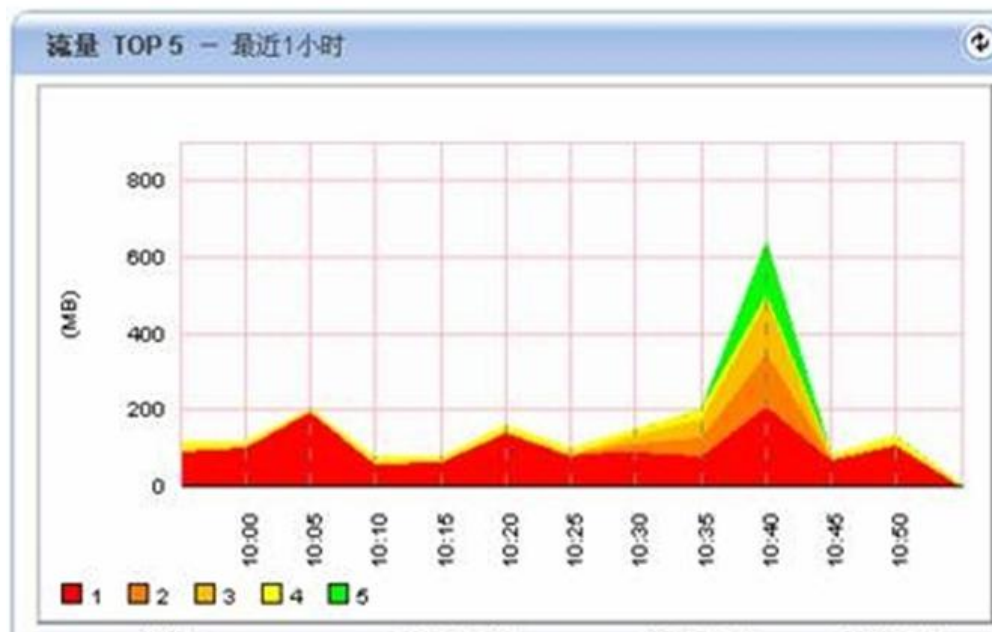
10MBit (位) - > 真正下载速度为:

1 MB (字节)

Incoming: 下载 Outgoing : 上传

11.5.4 实战 2: 使用 nethogs 找出使用带宽最多的进程

实战情景: 突然今天 11:00 左右, 运维人员接到 IDC 机房电话, 说你的网站流量对外输出很多, 这样应该怎么办?



nethogs 相当于:

电脑管家-网络优化

流量监控 测试网速 网络修复

当前有 32 个程序正在或已访问过网络, 下载速度: 9.6KB/S, 上传速度: 0.2KB/S

名称	↓ 下载速度	↑ 上传速度	占用程度	操作
▶ 运行的程序				
et.exe	0KB/S	0KB/S	低	禁用网络
wpscloudsvr.exe	0KB/S	0KB/S	低	禁用网络
qqprotect.exe	0KB/S	0KB/S	低	禁用网络
vpclient.exe	0KB/S	0KB/S	低	禁用网络
foxmail.exe	0KB/S	0KB/S	低	禁用网络
itrus.exe	0KB/S	0KB/S	低	禁用网络
查看更多(22项) ✓				
已禁用的程序				
kagent.exe	0KB/S	0KB/S	无	恢复连接

上传到 nethogs 到 linux 上:

```
[root@xuegod63 ~]# rpm -ivh nethogs-0.8.5-1.el7.x86_64.rpm
```

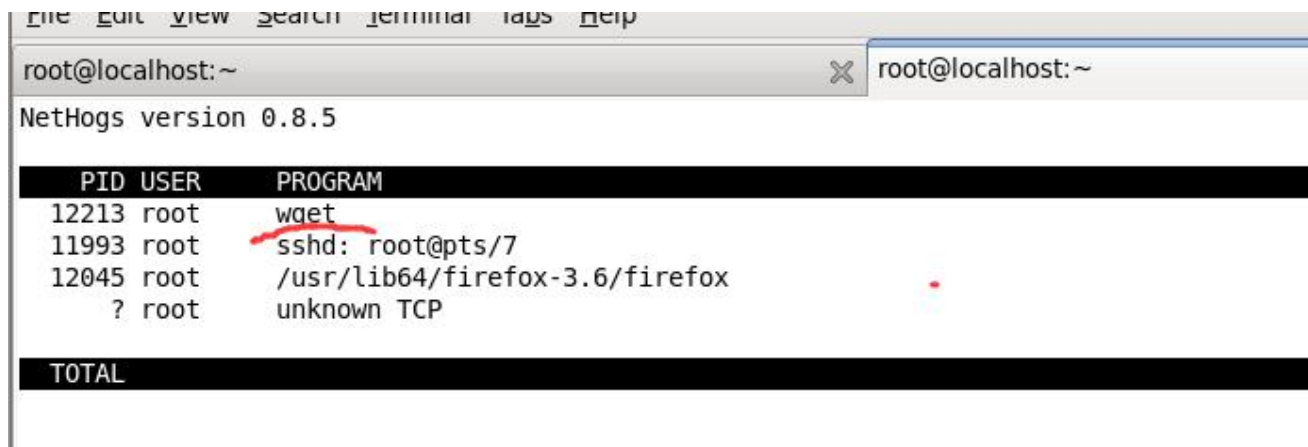
```
[root@xuegod63 ~]# nethogs
```

在另一个终端生成一些数据:

```
[root@xuegod63 ~]# wget
```

http://issuecdn.baidu.com/issue/netdisk/yunguanjia/BaiduNetdisk_5.5.3.exe

查看结果：



NetHogs version 0.8.5

PID	USER	PROGRAM
12213	root	wget
11993	root	sshd: root@pts/7
12045	root	/usr/lib64/firefox-3.6/firefox
?	root	unknown TCP

TOTAL

11.6 查看系统整体运行状态

11.6.1 查看内存及系统整体运行状态：

vmstat：命令是最常见的 Linux/Unix 监控工具，可以展现给定时间间隔的服务器的状态值，包括服务器的 CPU 使用率，MEM 内存使用，VMSwap 虚拟内存交换情况，IO 读写情况。

使用 **vmstat** 可以看到整个机器的 CPU，内存，IO 的使用情况，而不是单单看到各个进程的 CPU 使用率和内存使用率。比 **top** 命令节省资源。

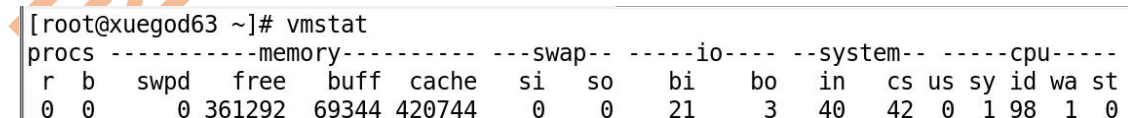
注：当机器运行比较慢时，建议大家使用 **vmstat** 查看运行状态，不需要使用 **top**，因 **top** 使用资源比较多。

例：

```
[root@xuegod63 ~]# vmstat
```

```
procs-----memory-----swap--io-----system--cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 0 361292 69344 420744 0 0 21 3 40 42 0 1 98 1 0
```

如图：



```
[root@xuegod63 ~]# vmstat
procs-----memory-----swap--io-----system--cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 0 361292 69344 420744 0 0 21 3 40 42 0 1 98 1 0
```

每一列参数作用：

r 运行状态的进程个数。展示了正在执行和等待 **cpu** 资源的任务个数。当这个值超过了 **cpu** 个数，就会出现 **cpu** 瓶颈。

b 不可中断睡眠 正在进行 **i/o** 等待 - - 阻塞状态的进程个数 进程读取外设上的数据，等待时

free 剩余内存，单位是 **KB**

buffers #内存从磁盘读出的内容

cached #内存需要写入磁盘的内容

si swpin swap 换入到内存

so swapout 内存换出到 swap 换出的越多, 内存越不够用

bi blockin 从硬盘往内存读。单位是块。把磁盘中的数据读入内存

bo blockout 从内存拿出到硬盘 (周期性的有值) 写到硬盘

#判断是读多还是写多, 是否有 i/o 瓶颈

in 系统的中断次数, cpu 调度的次数多

cs 每秒的上下文切换速度

CPU 上下文切换 - 程序在运行的时候, CPU 对每个程序切换的过程。

更多内存参以一下表格: vmstat 每个字段含义说明

类别	项目	含义	说明
Procs (进程)	r	等待执行的任务数	展示了正在执行和等待 cpu 资源的任务个数。当这个值超过了 cpu 个数, 就会出现 cpu 瓶颈。
	B	等待 IO 的进程数量	
Memory (内存)	swpd	正在使用虚拟的内存大小, 单位 k	
	free	空闲内存大小	
	buff	已用的 buff 大小, 对块设备的读写进行缓冲	
	cache	已用的 cache 大小, 文件系统的 cache	
	inact	非活跃内存大小, 即被标明可回收的内存, 区别于 free 和 active	具体含义见: 概念补充 (当使用 -a 选项时显示)
	active	活跃的内存大小	具体含义见: 概念补充 (当使用 -a 选项时显示)
Swap	si	每秒从交换区写入内存的大小 (单位: kb/s)	
	so	每秒从内存写到交换区的大小	
IO	bi	每秒读取的块数 (读磁盘)	现在的 Linux 版本块的大小为 1024bytes
	bo	每秒写入的块数 (写磁盘)	
system	in	每秒中断数, 包括时钟中断	这两个值越大, 会看到由内核消耗的 cpu 时间会越多
	cs	每秒上下文切换数	
CPU (以百分比表示)	Us	用户进程执行消耗 cpu 时间(user time)	us 的值比较高时, 说明用户进程消耗的 cpu 时间多, 但是如果长期超过 50% 的使用, 那么我们就该考虑优化程序算法或其他措施了
	Sy	系统进程消耗 cpu 时间(system time)	sy 的值过高时, 说明系统内核消耗的 cpu 资源多, 这个不是良性的表现, 我们应该检查原因。

	ld	空闲时间(包括 IO 等待时间)	
	wa	等待 IO 时间	Wa 过高时, 说明 io 等待比较严重, 这可能是由于磁盘大量随机访问造成的, 也有可能是磁盘的带宽出现瓶颈。

11.6.2 使用 sar 命令记录系统一段时间的运行状态

[root@xuegod63 ~]# rpm -ivh /mnt/Packages/sysstat-10.1.5-12.el7.x86_64.rpm

或:

[root@xuegod63 ~]# yum -y install sysstat

sysstat 工具包----- 可以把检查到的信息保存下来! 存在/var/log/sa 目录下, 文件名为 sa 当前日期

sar 默认显示每 10 分钟统计一次状态信息 (从装 sysstat 包开始)

sar 命令行的常用格式:

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:

下面只列出常用选项:

-A: 所有报告的总和。

-n: 网络接口的情况。

-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: 没有使用的内存页面和硬盘块。

-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。

-R: 进程的活动情况。

-y: 终端设备活动情况。

-w: 系统交换活动。

-o 文件名: 打印到屏幕并将采样结果以二进制形式存入当前目录下的文件中。

-f 文件名: 查看之前保存的二进制文件。

-d 显示磁盘

-d 1 100 必须得指定次数

-c 每秒创建进程的个数

-i 1 10 指定时间间隔

-P 查看 cpu

-r 查看内存

-w 每秒上下文切换次数

-o /cpu.sar 保存并显示

-f cpu.sar 读取

例一：使用命令行 `sar -u t n`

例如，每 2 秒采样一次，连续采样 5 次，观察 CPU 的使用情况，并将采样结果以二进制形式存入当前目录下的文件 `cpu.sar` 中，需键入如下命令：

`[root@xuegod63 ~]# sar -u 2 5 -o cpu.sar` # 屏幕显示以下内容，同时内容也会写到 `cpu.sar` 中

Linux 2.6.32-220.el6.x86_64 (xuegod63.cn) 08/04/2015 _x86_64_ (4 CPU)

10:44:45 PM	CPU	%user	%nice	%system	%iowait	%steal	%idle
10:44:47 PM	all	0.14	0.00	2.34	0.14	0.00	97.39
10:44:49 PM	all	1.08	0.00	1.33	0.36	0.00	97.23
10:44:51 PM	all	2.54	0.00	2.16	0.00	0.00	95.29
10:44:53 PM	all	0.25	0.00	1.52	0.13	0.00	98.10
10:44:55 PM	all	1.52	0.00	1.65	0.28	0.00	96.56
Average:	all	1.11	0.00	1.79	0.18	0.00	96.92

在显示内容包括：

%usr: CPU 处在用户模式下的时间百分比。

%sys: CPU 处在系统模式下的时间百分比。

%wio: CPU 等待输入输出完成时间的百分比。

%idle: CPU 空闲时间百分比。

在所有的显示中，我们应主要注意 %wio 和 %idle，%wio 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。

实例 2：查看二进制文件 `cpu.sar` 中的内容

`[root@xuegod63 ~]# sar -u -f cpu.sar`

Linux 2.6.32-220.el6.x86_64 (xuegod63.cn) 08/04/2015 _x86_64_ (4 CPU)

08:21:43 PM	CPU	%user	%nice	%system	%iowait	%steal	%idle
08:21:45 PM	all	0.00	0.00	0.72	2.03	0.00	97.25
08:21:47 PM	all	0.00	0.00	0.60	0.00	0.00	99.40
08:21:49 PM	all	0.13	0.00	0.75	0.00	0.00	99.12
08:21:51 PM	all	0.00	0.00	0.75	0.00	0.00	99.25
08:21:53 PM	all	0.12	0.00	0.75	0.00	0.00	99.13

注：sar 命令即可以实时采样，又可以对以往的采样结果进行查询。

11.6.3 查看 sar 的计划任务：

`[root@xuegod63 ~]# vim /etc/cron.d/sysstat`

Run system activity accounting tool every 10 minutes

Run system activity accounting tool every 10 minutes

*/10 * * * * root /usr/lib64/sa/sa1 1 1

```
# 0 * * * root /usr/lib64/sa/sa1 600 6 &
# Generate a daily summary of process accounting at 23:53
53 23 * * * root /usr/lib64/sa/sa2 -A
```

生成的日志位置:

```
# ls /var/log/sa
```

读取日志:

```
[root@xuegod63 ~]# ls /var/log/sa    #只要安装 sar 后就会定期收集系统信息
sa01  sa19
```

-n: 网络接口的情况。

```
[root@xuegod64 ~]# sar -n DEV -f /var/log/sa/sa19
```

```
Linux 3.10.0-693.el7.x86_64 (localhost.localdomain)    2017年12月19日    _x86_64_    (2 CPU)

00时59分22秒      LINUX RESTART

01时00分01秒      IFACE  rxpck/s   txpck/s   rxkB/s   txkB/s   rxcmp/s   txcmp/s   rxcst/s
01时10分02秒      lo      0.33      0.33      0.03      0.03      0.00      0.00      0.00
01时10分02秒      virbr0-nic  0.00      0.00      0.00      0.00      0.00      0.00      0.00
01时10分02秒      virbr0    0.00      0.00      0.00      0.00      0.00      0.00      0.00
01时10分02秒      ens33    4.52      0.00      0.33      0.00      0.00      0.00      0.00
01时20分01秒      lo      0.11      0.11      0.01      0.01      0.00      0.00      0.00
01时20分01秒      virbr0-nic  0.00      0.00      0.00      0.00      0.00      0.00      0.00
01时20分01秒      virbr0    0.00      0.00      0.00      0.00      0.00      0.00      0.00
01时20分01秒      ens33    3.50      0.00      0.28      0.00      0.00      0.00      0.00
平均时间:      lo      0.22      0.22      0.02      0.02      0.00      0.00      0.00
平均时间:      virbr0-nic  0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间:      virbr0    0.00      0.00      0.00      0.00      0.00      0.00      0.00
平均时间:      ens33    4.01      0.00      0.30      0.00      0.00      0.00      0.00

01时21分44秒      LINUX RESTART
[root@xuegod64 ~]#
```

```
# sar -r -f /var/log/sa/sa27
```

```
# sar -b -f /var/log/sa/sa27
```

```
# sar -f /var/log/sa/sa27
```

动态查看:

```
sar 1    #默认显示 CPU 状态, 1 秒显示一次
```

```
sar 1 10
```

```
sar -P 0 1 10
```

sar 最大特点是可以监控所有状态

```
sar -r 1 查看内存
```

```
sar -n ALL 查看所有
```

```
sar -b 1 10    缓冲区使用情况, 每秒刷新一次, 查看 10 次
```

```
sar -l irq#
```

```
sar -l ALL 1 10
```

```
sar -r -f /tmp/file    -n    -r    -b    -m
```

```
sar -s 15:00:00 -e 15:30:00
```

```
sar -s 15:00:00 -e 15:30:20 -f /var/log/sa/sa01  
sar -r -s 15:00:00 -e 15:30:20 -f /var/log/sa/sa01
```

总结:

- 11.1 系统调优概述
- 11.2 查看 CPU 负载相关工具，找出系统中使用 CPU 最多的进程
- 11.3 查看 Memory 运行状态相关工具，找出系统中使用内存最多的进程
- 11.4 查看 IO 运行状态相关工具，找出系统中对磁盘读写最多的进程
- 11.5 查看 Network 运行状态相关工具，找出系统中使用网络最多的进程
- 11.6 查看系统整体运行状态