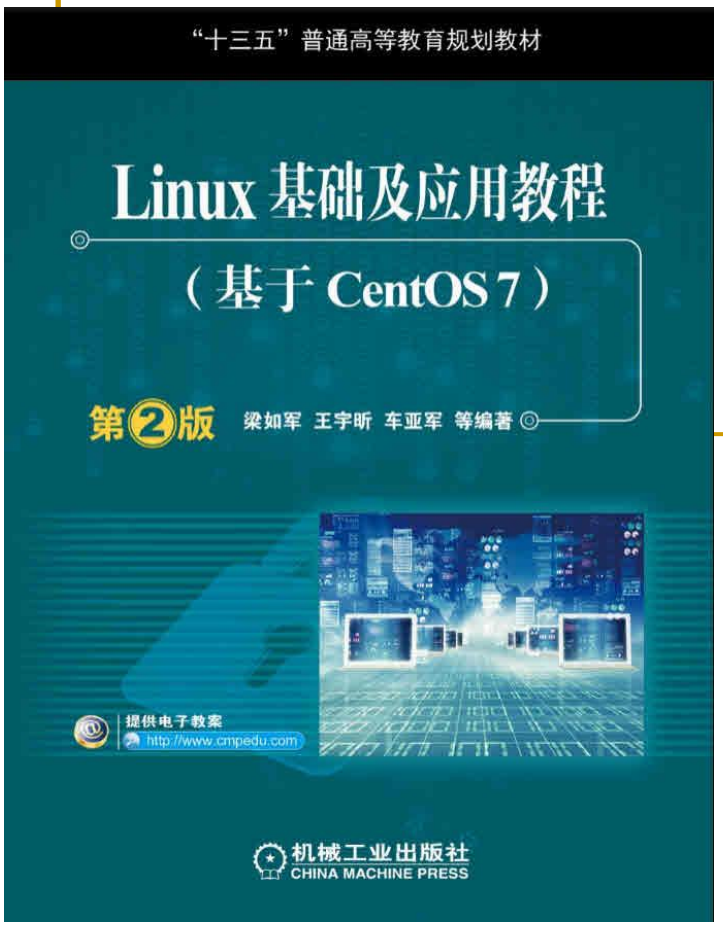


第7章
系统日常维护

主讲人： 梁如军

2015-05-05

本章内容要点

- 理解影响系统性能的因素
- 系统性能分析工具
- Linux的内核参数的修改方法
- 内核管理
- 备份与同步
- 系统启动过程
- 故障排查与修复

梁如军（linuxbooks@126.com）

本章学习目标

- 学会使用top、mpstat、vmstst、iostat工具分析系统性能
- 熟悉系统性能评估标准
- 管理内核模块
- 调整内核参数
- 安全升级内核
- 熟悉系统系统过程
- 掌握修复运行级别和援救环境的使用
- 学会排查和修复常见的故障

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

监视系统性能

梁如军（linuxbooks@126.com）

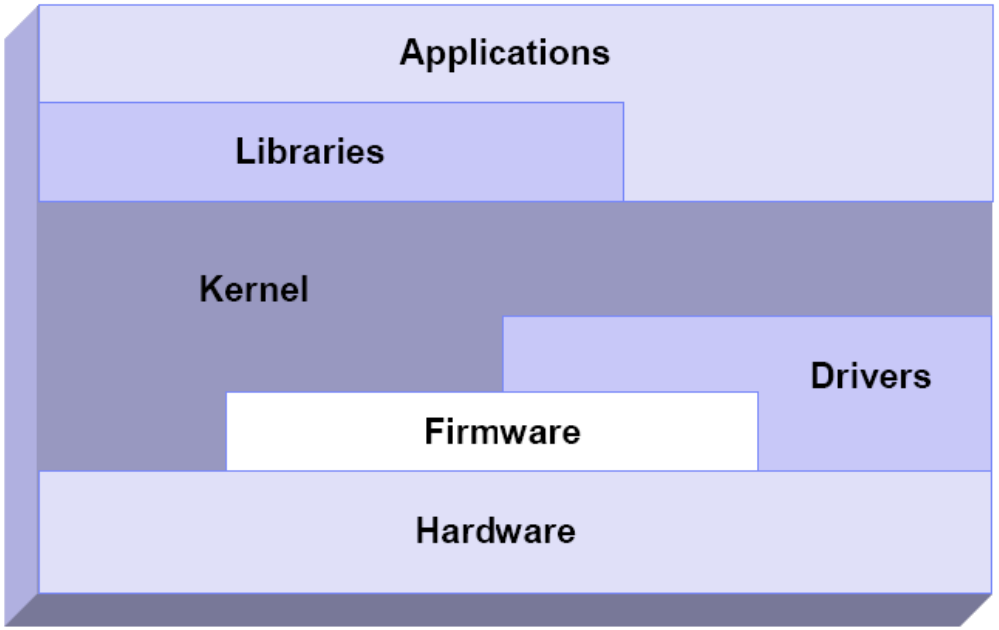
系统性能监视对象

- 容易形成性能瓶颈的监视对象
  - CPU性能
  - 内存性能
  - 磁盘I/O性能
  - 网络I/O带宽

梁如军（linuxbooks@126.com）

影响系统性能的因素

- 影响系统性能的因素众多
  - 硬件
    - CPU
    - 内存
    - IO总线等
  - 操作系统
    - 内核子系统
    - 驱动模块等
  - 应用程序



- 系统调优是一项非常复杂的任务

梁如军（linuxbooks@126.com）

系统性能监视常用工具

- CPU监视工具
  - uptime：显示系统平均负载
  - top：动态显示系统进程任务
  - mpstat：输出CPU的各种统计信息
- 内存监视工具
  - free：显示系统内存的使用
  - vmstat：报告虚拟内存的统计信息
- 磁盘I/O监视工具
  - iostat：输出CPU、I/O系统和磁盘的统计信息
- 网络流量
  - nload：显示当前的网络流量

梁如军（linuxbooks@126.com）

top命令

- 动态显示系统的统计信息和进程的重要信息
  - 统计信息
    - 系统平均负载
    - 进程状态统计
    - CPU使用的统计信息
    - 物理内存和虚拟内存的使用统计信息
  - 进程信息

```
1 PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
2 PID  PPID    TIME+  %CPU %MEM  PR  NI S  VIRT SWAP  RES  UID COMMAND
3 PID %MEM  VIRT SWAP  RES CODE DATA  SHR nFLT nDRT S  PR  NI %CPU COMMAND
4 PID  PPID  UID USER     RUSER    TTY        TIME+  %CPU %MEM S COMMAND
```


梁如军（linuxbooks@126.com）

top命令输出的统计信息(1)

top - 07:01:55 up 14 min, 1 user, load average: 0.03, 0.02, 0.00

- 显示的是uptime命令的输出
  - **07:01:55 —— 当前时间**
  - **up 14 min —— 系统自开机后运行的时间**
  - **1 user —— 当前登录的用户数**
  - **load average: 0.03, 0.02, 0.00**
    - 1分钟系统平均负载，5分钟系统平均负载，15分钟系统平均负载
    - **一段时间内，每个值都应该小于系统中CPU的个数，否则表示系统存在CPU瓶颈**
- 相关的交互命令
  - 交互命令“l”是用于是否显示此信息的乒乓切换开关

top命令输出的统计信息(2)

CentOS
The Community ENTerprise Operating System

Tasks: 98 total, 2 running, 96 sleeping, 0 stopped, 0 zombie

- **进程状态**的输出字段
  - 总进程数（**total**）
  - 正在运行进程数（**running**）
  - 睡眠的进程数（**sleeping**）
  - 停止的进程数（**stopped**）
  - 僵尸进程数（**zombie**）

僵尸进程指的是子进程退出后父进程并没有处理子进程的退出信号，导致子进程变为僵尸进程。

梁如军（linuxbooks@126.com）

top命令输出的统计信息(3)

CentOS The Community ENTerprise Operating System

Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

- **CPU使用率**的输出字段
  - %us(user)：用户态进程占用CPU百分比
  - %sy(system)：核心态进程占用CPU百分比
  - %ni(nice)：调整过优先级的用户态进程占用CPU时间的百分比
  - %id(idel)：CPU空闲的百分比
  - %wa(iowait)：等待系统I/O的CPU时间百分比
  - %hi(hard interrupt)： CPU用于处理硬件中断的时间百分比
  - %si(soft interrupt)： CPU用于处理软中断的时间百分比
  - %st(steal)：被虚拟机偷掉的CPU时间百分比（仅用于运行虚拟机的情况）

- 交互命令 **t** 是用于是否显示进程状态统计和**CPU**使用率的乒乓切换开关
- 交互命令 **1** 是用于显示所有**CPU**的平均状态还是每个**CPU**状态的乒乓切换开关

梁如军（linuxbooks@126.com）

top命令输出的统计信息(4)

- 物理内存
  - 总量，使用量，空闲量

Buffers 指的是块设备的读写缓冲区

```
Mem:  1025692k total,  121072k used,  904620k free,   13236k buffers
Swap: 2064376k total,       0k used, 2064376k free,   59436k cached
```

Cached 指的是文件系统本身的页面缓存

- 交换空间
  - 总量，使用量，空闲量

交互命令 m 是用于是否显示系统内存和交换空间信息的乒乓切换开关

buffers和cached都是Linux操作系统底层的机制，目的就是为了加速对磁盘的访问。

梁如军（linuxbooks@126.com）

top命令输出的进程信息

- PID(进程号)， USER（运行用户）
- PR（优先级），NI（任务nice值）
- VIRT（虚拟内存用量），RES（物理内存用量），SHR（共享内存用量）
- S（进程状态）
  - R=运行；S=睡眠；T=跟踪/停止；Z=僵尸
- %CPU（CPU占用比），%MEM（内存占用比）
- TIME+（累计CPU占用时间)

梁如军（linuxbooks@126.com）

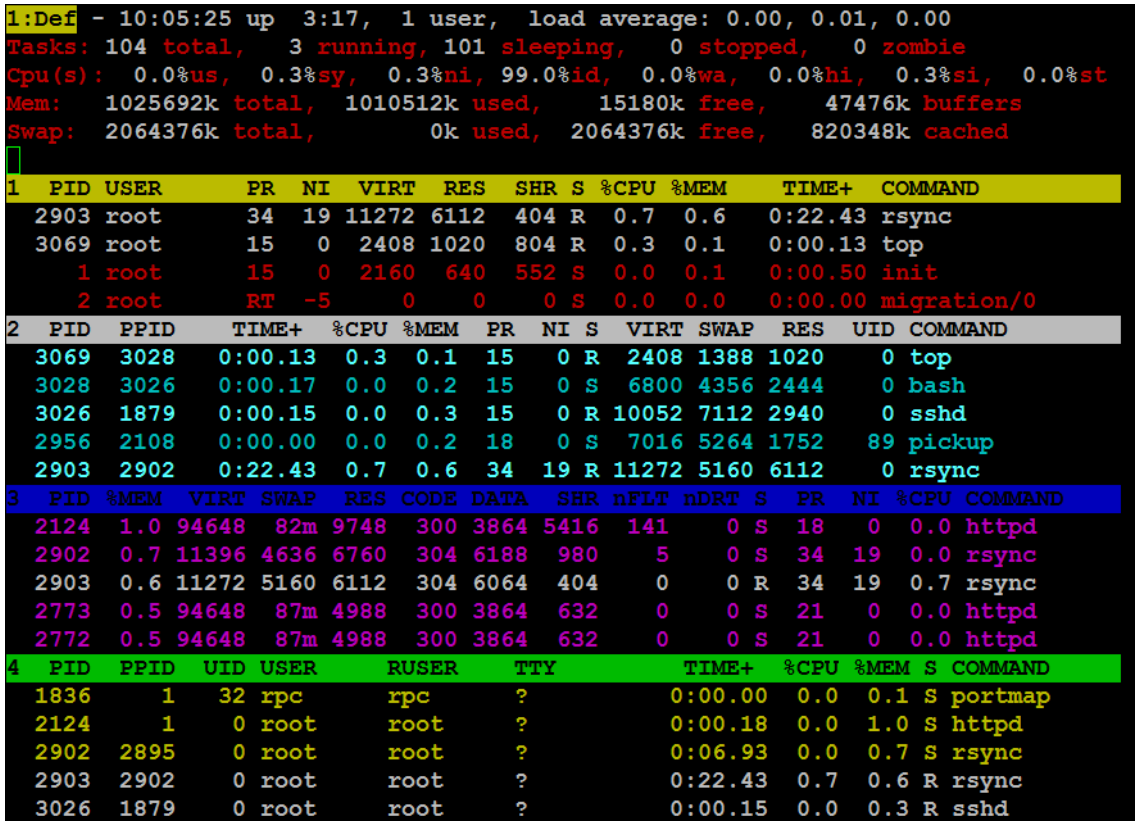
top的交互命令（1）

- <Space>或<Enter>：立即刷新显示
- ?或h：显示帮助信息屏幕
- G[1234]：可以使用G1~G4切换top提供的四种字段方案的显示窗口
- B：加粗加亮显示的乒乓切换开关
- u：显示指定用户的进程（仅匹配EUID）
- U：显示指定用户的进程（匹配RUID、EUID、SUID和UID）
- k：杀死指定的进程（发送进程信号）
- r：重新设置一个进程的优先级别
- d或s：改变两次刷新显示之间的时间间隔，单位为秒
- W：将当前的top设置写入~/.toprc文件中
- q：退出top

梁如军（linuxbooks@126.com）

top的交互命令（2）

- 多窗口显示
  - A：是否在一个界面中同时显示四种字段方案显示窗口的乒乓切换开关
  - a和w：在四种字段方案的显示窗口中移动以确认当前窗口
    - a表示下一个窗口
    - w表示上一个窗口



梁如军（linuxbooks@126.com）

top的交互命令（3）

- 加亮显示行和列
  - x：是否对当前排序字段进行加亮显示的乒乓切换开关
  - y：是否对当前正在运行进程进行加亮显示的乒乓切换开关

```
top - 10:12:59 up  3:25,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 104 total,   2 running, 102 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.3%sy,  0.0%ni, 99.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1025692k total,  1011532k used,    14160k free,    47344k buffers
Swap:  2064376k total,        0k used,  2064376k free,   821328k cached

  PID %MEM  VIRT SWAP  RES CODE DATA  SHR nFLT nDRT S  PR  NI %CPU COMMAND
 2124  1.0 94648  82m 9748  300 3864 5416  141    0 S  18   0  0.0 httpd
 2902  0.7 11396 4636 6760  304 6188  980    5    0 S  34  19  0.0 rsync
 2903  0.6 11272 5160 6112  304 6064  404    0    0 S  34  19  0.0 rsync
 2764  0.5 94648  87m 4988  300 3864  632    0    0 S  16   0  0.0 httpd
 2765  0.5 94648  87m 4988  300 3864  632    0    0 S  16   0  0.0 httpd
 2766  0.5 94648  87m 4988  300 3864  632    0    0 S  16   0  0.0 httpd
 2767  0.5 94648  87m 4988  300 3864  632    0    0 S  16   0  0.0 httpd
 2768  0.5 94648  87m 4988  300 3864  632    0    0 S  21   0  0.0 httpd
 2771  0.5 94648  87m 4988  300 3864  632    0    0 S  21   0  0.0 httpd
 2772  0.5 94648  87m 4988  300 3864  632    0    0 S  21   0  0.0 httpd
 2773  0.5 94648  87m 4988  300 3864  632    0    0 S  21   0  0.0 httpd
 3026  0.3 10212 7260 2952  396  776 2376    0    0 R  15   0  0.0 sshd
 3028  0.2  6800 4356 2444  696 1332 1192    1    0 S  15   0  0.0 bash
```


梁如军（linuxbooks@126.com）

top的交互命令（4）

- 选择排序字段
  - 快速选择排序字段
    - M：按 %MEM字段排序
    - N：按 PID字段排序
    - P：按 %CPU字段排序
    - T：按 TIME+字段排序
  - 交互式选择排序字段
    - F 或 O
  - 切换排序字段和逆向排序
    - < 或 >
    - R

梁如军（linuxbooks@126.com）

mpstat命令

- 功能：输出每一个 CPU 的运行状况，为多处理器系统中的 CPU 利用率提供统计信息。
- 格式：
  - mpstat [-P { cpu | ALL }] [interval [count]]
  - 其中
    - -P {cpu-id|ALL}：用CPU-ID指定CPU，CPU-ID从0开始
    - interval：为取样时间间隔
    - count：为输出次数

梁如军（linuxbooks@126.com）

mpstat 命令举例

 # mpstat

Linux 2.6.18-194.32.1.el5 (centos1.ls-al.me) 04/29/11

12:56:27	CPU	%user	%nice	%sys	%iowait	%irq	%soft	%steal	%idle	intr/s
12:56:27	all	3.89	0.00	0.76	4.04	0.02	0.12	0.00	91.18	1050.99

 # mpstat -P 0

Linux 2.6.18-194.32.1.el5 (centos1.ls-al.me) 04/29/11

12:56:37	CPU	%user	%nice	%sys	%iowait	%irq	%soft	%steal	%idle	intr/s
12:56:37	0	3.86	0.00	0.75	4.01	0.02	0.12	0.00	91.24	1050.81

 # mpstat 5 10

 # mpstat –P 1 5 10

vmstat命令

- 功能：显示进程队列、内存、交换空间、磁盘I/O、和CPU状态信息。
- 格式：
  - vmstat [-a] [-n] [-S k|K|m|M] [Interval [Count]]
  - 其中：
    - -a：显示活跃和非活跃内存
    - -n：只在开始时显示一次各字段名称
    - -S：使用指定单位显示。k(1000)、K(1024)、m(1000000)、M(1048576) 字节，默认单位为K。
    - interval和count的含义与mpstat一致

梁如军（linuxbooks@126.com）

vmstat命令举例

```
# vmstat 5 2
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 504544 119328 236716    0    0    46    30 1036   52  1  0  97  2  0
 0  0      0 504544 119336 236708    0    0     0     6 1002   28  0  0 100  0  0
```

- procs
  - r 列表示运行和等待CPU时间片的进程数，这个值若长期大于系统CPU的个数，说明CPU资源不足。
  - b 列表示在等待资源的进程数，比如正在等待I/O、或者内存交换等。
- cpu
  - us 列显示了用户态进程消耗的CPU 时间百分比；sy 列显示了核心态进程消耗的CPU时间百分比。
  - us 的值比较高时说明用户进程消耗的CPU时间多，sy 值较高时说明内核消耗的CPU资源很多。
  - **根据经验，us+sy 的参考值为80%，若 us+sy>80% 说明可能存在CPU资源不足。**

梁如军（linuxbooks@126.com）

vmstat命令举例（续）

```
# vmstat 5 2
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 504544 119328 236716    0    0    46    30 1036   52  1  0  97  2  0
 0  0      0 504544 119336 236708    0    0     0     6 1002   28  0  0 100  0  0
```

- memory
  - swpd 列表示切换到内存交换区的内存数量(以k为单位)。
  - 若 swpd 的值不为0，或者比较大，只要si、so的值长期为0，这种情况下一般不用担心，不会影响系统性能。
  - free 列表示当前空闲的物理内存数量(以k为单位)。
  - buff 列表示 buffers cache 的内存数量，一般对块设备的读写才需要缓冲。
  - cache 列表示 page cached 的内存数量，一般作为文件系统 cached，频繁访问的文件都会被 cached
  - 若cache值较大，说明cached的文件数较多，如果此时IO中bi比较小，说明文件系统效率比较好。

梁如军（linuxbooks@126.com）

vmstat命令举例（续2）

```
# vmstat 5 2
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 504544 119328 236716    0    0    46    30 1036   52  1  0  97  2  0
 0  0      0 504544 119336 236708    0    0     0     6 1002   28  0  0 100  0  0
```

- swap
  - si 列表示由磁盘调入内存，也就是内存进入内存交换区的数量。
  - so 列表示由内存调入磁盘，也就是内存交换区进入内存的数量。
  - 一般情况下，si、so的值都为0，**如果si、so的值长期不为0，则表示系统内存不足**。

iostat命令

- 功能：输出CPU和磁盘I/O相关的统计信息。
- 格式：
  - iostat [-c|-d] [-x] [-k|-m] [device | ALL] [interval [count]]
  - 其中：
    - -c：仅显示CPU统计信息。与-d选项互斥
    - -d：仅显示磁盘统计信息。与-c选项互斥
    - -k：以KB为单位显示每秒的磁盘请求数。默认单位为块
    - -m：以MB为单位显示每秒的磁盘请求数。默认单位为块
    - -x：输出扩展信息
    - device：用于指定磁盘设备
    - interval和count的含义与mpstat一致

梁如军（linuxbooks@126.com）

iostat命令举例（1）

```
# iostat -d sda sda3 5 2
Linux 2.6.18-194.32.1.el5 (centos1.ls-al.me)    2011年04月29日

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
sda               4.61        51.09        39.21     577048     442878
sda3              4.07        44.89        39.00     507088     440552

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
sda               0.41         0.00        16.33          0         80
sda3              0.41         0.00        16.33          0         80
```

- tps：每秒钟物理设备的I/O传输总量
- 长期的、超大的数据读写，肯定是不正常的，这种情况一定会影响系统性能。

梁如军（linuxbooks@126.com）

iostat命令举例（2）

```
# iostat -dxk sda sda3
Linux 2.6.18-194.32.1.el5 (centos1.ls-al.me)    2011年04月29日

Device:  rrqm/s  wrqm/s   r/s   w/s   rkB/s   wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sda        0.75    2.44  2.18  2.10   23.58   18.31    19.59     0.40   93.52   2.53   1.08
sda3       0.28    2.42  1.70  2.09   20.72   18.21    20.58     0.40  105.24   2.52   0.95
```

- rrqm/s：每秒发送到设备的读入请求数。
- wrqm/s：每秒发送到设备的写入请求数。
- avgrq-sz：发送到设备的请求的平均大小。
- avgqu-sz：发送到设备的请求的平均队列长度。
- await：表示平均每次设备I/O操作的等待时间(以毫秒为单位)。
- svctm：表示平均每次设备I/O操作的服务时间(以毫秒为单位)。
- %util：表示一秒中有百分之几的时间用于 I/O 操作。

梁如军（linuxbooks@126.com）

iostat命令举例（2续）

```
# iostat -dxk sda sda3
Linux 2.6.18-194.32.1.el5 (centos1.ls-al.me)   2011年04月29日

Device:  rrqm/s  wrqm/s   r/s   w/s   rkB/s   wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sda        0.75     2.44  2.18  2.10   23.58   18.31    19.59     0.40   93.52   2.53   1.08
sda3       0.28     2.42  1.70  2.09   20.72   18.21    20.58     0.40  105.24   2.52   0.95
```

- 正常情况下 svctm 应该小于 await 的值
  - svctm 的大小和磁盘性能有关
  - CPU、内存的负荷也会对 svctm 值造成影响
  - 过多的磁盘请求也会间接的导致 svctm 值的增加
- await 的大小一般取决与 svctm 的值和I/O队列长度以及I/O请求模式
  - 如果 svctm 的值与 await 很接近，表示几乎没有I/O等待，磁盘性能很好
  - 如果 await 的值远高于 svctm 的值，则表示I/O队列等待太长，系统上运行的应用程序将变慢，此时可以通过更换更快的硬盘来解决问题。
- %util 项的值也是衡量磁盘I/O的一个重要指标
  - 如果 %util 接近 100%，表示磁盘产生的I/O请求太多，I/O系统已经满负荷的在工作，该磁盘可能存在瓶颈。

梁如军（linuxbooks@126.com）

Sysstat——sar与sadc

- sar（System Activity Reporter）是系统活动情况报告的缩写
  - sar是目前Linux上最为全面的系统性能分析工具之一，可以从多方面对系统的活动进行报告
  - 文件的读写情况、系统调用的使用情况、磁盘I/O、CPU效率、内存使用状况、进程活动及IPC有关的活动等
- sadc（System activity data collector ）是系统活动数据收集系统的缩写
  - 收集的数据被写入一个二进制的文件中（ /var/log/sa/saDD ）
  - 它被用做sar工具的后端

梁如军（linuxbooks@126.com）

sar的运作机制

- 守护进程 /etc/init.d/sysstat
  - 每次开机重置系统活动日志
- Cron任务 /etc/cron.d/sysstat
  - 每隔10分钟执行一次/usr/lib/sa/sa1 1 1命令，将信息写入文件/var/log/sa/saDD
  - 每天23:53执行一次/usr/lib/sa/sa2 -A 命令，将当天的汇总信息写入文件/var/log/sa/saDD

梁如军（linuxbooks@126.com）

sar命令

- 怀疑CPU存在瓶颈
  - 可用sar -u和sar -q来查看。
- 怀疑内存存在瓶颈
  - 可用sar -B、sar -r和sar -W来查看。
- 怀疑I/O存在瓶颈
  - 可用sar -b、sar -u和sar -d来查看。
- 怀疑网络存在瓶颈
  - 可用sar -n DEV 和 sar -n EDEV来查看。

系统性能评估

影响因素	评判标准		
	好	坏	糟
CPU	user% + sys%< 70%	user% + sys%= 85%	user% + sys% >=90%
内存	Swap In（si）=0 Swap Out（so）=0	Per CPU with 10 page/s	More Swap In & Swap Out
磁盘	iowait % < 20%	iowait % =35%	iowait % >= 50%

- %user：表示CPU处在用户模式下的时间百分比。
- %sys：表示CPU处在系统模式下的时间百分比。
- %iowait：表示CPU等待输入输出完成时间的百分比。
- swap in：即si，表示虚拟内存的页导入，即从SWAP DISK交换到RAM。
- swap out：即so，表示虚拟内存的页导出，即从RAM交换到SWAP DISK。

梁如军（linuxbooks@126.com）

网络性能评估工具

- nload
  - monitor network traffic and bandwidth usage
  - http://www.roland-riegel.de/nload/
  - EPEL仓库提供RPM包
- iftop
  - **display bandwidth usage on an interface**
  - http://www.ex-parrot.com/~pdw/iftop/
  - EPEL仓库提供RPM包
- iptraf
  - IP Network Statistics Utility
  - http://iptraf.seul.org/

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

内核管理

梁如军（linuxbooks@126.com）

RHEL/CentOS下的内核

- 支持的系统架构
  - **x86-64**：Intel 和 AMD 64 位
  - **x86** ：Intel 和 AMD 32 位
  - **IBM POWER**： >= POWER6 的 64 位
  - **IBM System z**：>= z9
- 内核的RPM包
  - **kernel**：支持单核或多核CPU
  - **kernel-doc**：内核文档

梁如军（linuxbooks@126.com）

显示内核相关的信息

- 显示安装的内核版本信息
  - **rpm -q {redhat,centos}-release**
  - **yum list installed kernel***
- 显示当前运行的内核版本信息
  - **uname -r**
  - **cat /proc/version**
- 显示当前运行系统架构
  - **arch**
  - **uname -m**
- 显示安装的发行版本信息
  - **cat /etc/redhat-release**
  - **lsb_release -d**

梁如军（linuxbooks@126.com）

内核的重要组件

- 内核映像文件
  - 文件保存在 /boot/vmlinuz-$(uname -r) 。
  - 由启动加载器（GRUB）直接加载到内存以便启动内核。
- 内核模块
  - 可根据需要装载或者卸载的内核扩展
  - 包括驱动程序、文件系统、防火墙等等
- 初始化内存盘
  - Bootloader Initialized RAM Disk

梁如军（linuxbooks@126.com）

内核模块

- 内核的众多功能
  - 可以直接编译到内核映像文件
  - 也可以编译为独立的模块
- 内核模块
  - 可以在系统运行期间动态地加载或卸载以改变系统功能
  - 保存在 /lib{,64}/modules/$(uname -r) 目录，因为/lib{,64}存在根文件系统中，因此所有内核模块必须在根文件系统挂载后才能使用
  - 必须为特定的内核版本编译，以Kernel的RPM提供，用户也可以添加第三方模块

梁如军（linuxbooks@126.com）

内核模块工具

- **lsmod：** 列出已装载的模块
  - **# lsmod |grep usb**
- **modprobe：** 装载和卸载模块
  - **# modprobe usb_storage**
  - **# modprobe -r usb_storage**
- **modinfo：** 显示模块的信息
  - **# modinfo usb_storage**

内核模块配置文件 /etc/modprobe.d/*.conf

- 定义模块别名、设置默认的模块执行参数、指定在装载或卸载模块时需要执行的操作
- 配置文件中 可以使用如下四个配置语句
  - alias：用于指定别名。
  - options：用于指定模块运行时的默认参数。
  - install：用于指定加载模块时执行的命令。
  - remove：用于指定卸载模块时执行的命令。

初始化内存盘

- 初始化内存盘提供在引导初期装载的模块
  - 用于内核映像文件 vmlinuz 中没有提供的其他设备的内核驱动模块
  - 文件位于 **/boot/initramfs-$(uname -r).img**
- 由启动加载器（GRUB）直接加载到内存
- 在引导初期，根文件系统挂载之前使用
- 是Linux安装盘、Linux启动盘（CD、USB）、LiveCD的必备部件

使用rpm命令升级内核

- 安装新版内核
  - **# rpm -i kernel***
  - **不要使用 rpm -U 或 rpm -F !**
- 重新启动系统，在GRUB中选择新版内核启动系统
- 测试新版内核，若有任何问题发生可以使用旧版内核重新启动系统
- 当确信新版内核无任何问题时，删除旧版内核
  - **# rpm -e kernel-oldversion**

梁如军（linuxbooks@126.com）

使用yum命令升级内核

- 使用yum命令升级内核
  - **# yum -y update kernel**
  - 主要包含如下操作：
    （1）下载最新版的内核RPM文件
    （2）安装新版的内核RPM文件
    （3）根据/etc/syconfig/kernel的设置自动配置GRUB
- 测试新版内核，若有任何问题发生可以使用旧版内核重新启动系统
- 当确信新版内核无任何问题时，删除旧内核
  - **# yum remove kernel-oldversion**

SYSTEMD与系统启动过程

梁如军（linuxbooks@126.com）

Systemd简介

- Systemd是Linux的系统引导器和服务管理器
- 大大加快了系统启动过程
  - Systemd是C语言编写的经过编译的二进制程序
  - systemd尽可能减少对shell脚本的依赖
  - 提供优秀的框架用以表示系统服务间的依赖关系实现系统初始化时诸多服务的并行启动。

梁如军（linuxbooks@126.com）

Systemd的特性

- Systemd 通过Socket缓存、DBus缓存和建立临时挂载点等方法进一步解决了启动进程之间的依赖，实现了服务并发启动
- Systemd 提供了服务按需启动的能力，使得特定的服务只有在被真正请求的时候才启动
- Systemd 提供了基于依赖关系的服务控制逻辑，即启动一个单元之前先启动其依赖的单元
- Systemd 通过控制组（controller group，Cgroup）跟踪和管理进程的生命周期

Systemd的特性 (续)

- Systemd支持已启动的服务的监控同时支持重启已崩溃的服务
- 支持组件模块的加载和卸载
- 支持低内存使用痕迹以及任务调度能力
- 通过systemd-journald模块记录二进制日志
- 通过systemd-login模块用于控制用户登录
- 支持系统状态的快照和恢复
- 向下兼容SysVinit

梁如军（linuxbooks@126.com）

Systemd的组件

- 核心组件
  - 守护进程**systemd**
  - **systemd-cgls**和**systemd-cgtop**
  - **systemd-analyze**
- 附加组件
  - **systemd-logind**守护进程负责管理用户登录
  - **systemd-journal**守护进程负责记录二进制日志
  - **systemd-udevd**守护进程负责监听内核的udev事件并根据相应的 udev 规则执行匹配的指令

梁如军（linuxbooks@126.com）

Systemd的单元类型

service	.service	描述一个系统服务
socket	.socket	描述一个进程间通信的套接字
device	.device	描述一个由内核识别的设备文件
mount	.mount	描述一个文件系统的挂装点
automount	.automount	描述一个文件系统的自动挂装点
swap	.swap	描述一个内存交换设备或交换文件
path	.path	描述一个文件系统中文件或目录
timer	.timer	描述一个定时器
snapshot	.snapshot	用于保存一个systemd的状态
scope	.scope	以编程方式创建的外部进程
slice	.slice	描述基于Cgroup的一组通过层次组织的管理系统进程
target	.target	描述一组systemd的单元

梁如军（linuxbooks@126.com）

Systemd的激活机制

- 基于Socket激活
- 基于D-Bus激活
- 基于Device激活
- 基于Path激活

Systemd单元配置文件

目录	描述	优先级
/usr/lib/systemd/system/	由安装的RPM包发布的Systemd单元	最低
/run/systemd/system/	在运行时创建的Systemd 单元	高
/etc/systemd/system/	由管理员创建和管理的Systemd单元	最高

Systemd单元的依赖关系

- 需求依赖（Requirement dependence）
  - 使用Requires或Wants配置语句来描述
- 顺序依赖（Ordering dependence）
  - 使用After或Before配置语句来描述
- 冲突依赖（Conflict dependence）
  - 使用Conflicts配置语句来描述

梁如军（linuxbooks@126.com）

显示单元依赖关系

CentOS
The Community ENTerprise Operating System

```
systemctl show --property "Requires" boot.mount
systemctl show --property "Wants"    boot.mount

systemctl show --property "Before"   crond.service
systemctl show --property "After"    crond.service

systemctl show --property "Conflicts" postfix.service
```


梁如军（linuxbooks@126.com）

Systemd单元配置文件语法

- 使用描述性语言
- 通常包括三节内容：
  - 用于描述本单元的相关信息的[Unit]
  - 用于描述本单元的[Install]
  - 用于描述特定单元类型信息的[Service]、[Socket]、[Mount]、[Automount]、[Swap]、[Path]、[Timer]、[Slice]。

梁如军（linuxbooks@126.com）

Systemd单元配置文件举例1


```
# cat /etc/systemd/system/multi-user.target.wants/sshd.service
[Unit]
Description=OpenSSH server daemon
After=network.target sshd-keygen.service
Wants=sshd-keygen.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```


梁如军（linuxbooks@126.com）

Systemd单元配置文件举例2


```
# cat /etc/systemd/system/default.target
[Unit]
Description=Multi-User System
Documentation=man:systemd.special(7)
Requires=basic.target
Conflicts=rescue.service rescue.target
After=basic.target rescue.service rescue.target
AllowIsolate=yes

[Install]
Alias=default.target
```


梁如军（linuxbooks@126.com）

Systemd的目标

- Systemd的目标是一种特殊类型的单元，其单元配置文件的扩展名为.target。
- Systemd 的目标代表一组单元，其目的是通过一个依赖关系链组织其他的系统单元。
- 一个目标的启动代表了一种运行状态，即此目标包含依赖关系的一组单元均已被启动。

梁如军（linuxbooks@126.com）

使用systemctl命令查看目标

- 显示当前已激活的目标
  - systemctl -t target
- 显示当前已加载的所有目标
  - systemctl -at target
- 显示systemd的RPM包安装的所有目标
  - systemctl list-unit-files -t target
- 显示指定目标的依赖关系
  - systemctl list-dependencies <TargetName>.target

梁如军（linuxbooks@126.com）

目标与运行级别

运行级别	Systemd的目标	Systemd模拟SysVinit的目标	说明
0	poweroff.target	runlevel0.target	关机并断电
1	rescue.target	runlevel1.target	单用户或救援Shell模式
2	multi-user.target	runlevel2.target	非图形界面多用户系统
3	multi-user.target	runlevel3.target	非图形界面多用户系统
4	multi-user.target	runlevel4.target	非图形界面多用户系统
5	graphical.target	runlevel5.target	图形界面多用户系统
6	reboot.target	runlevel6.target	重新启动

梁如军（linuxbooks@126.com）

管理默认目标

- 显示默认的目标
  - **systemctl get-default**
- 设置默认的目标（下次启动时生效）
  - **systemctl set-default graphical.target**
- 更改当前的目标（立即生效）
  - **systemctl isolate multi-user.target**

系统启动过程（1）

- 系统固件初始化
  - 1）计算机加电，系统固件（BIOS/UEFI）执行开机自检。
  - 2）系统固件搜索可启动设备。
  - 3）系统固件从磁盘加载启动引导器，然后将系统控制权交给启动引导器。

梁如军（linuxbooks@126.com）

系统启动过程（2）

- 启动引导器GRUB2
  - 1）启动引导器从磁盘加载其配置。
  - 2）启动引导器向用户显示GRUB菜单。
  - 3）当用户选择了启动项或自动超时后，启动引导器从磁盘加载 kernel 和 initramfs 到内存（initramfs是以gzip压缩的cpio归档文件，其中包含启动时所需的所有必要硬件的内核模块以及初始化脚本等）。
  - 4）启动引导器将系统控制权交给内核，并为其传递启动引导器的内核命令行中指定的选项以及initramfs在内存中的位置。

梁如军（linuxbooks@126.com）

系统启动过程（3）

- Linux内核初始化
  - 1）kernel 从 initramfs 启动 systemd 的工作副本 /sbin/init （PID=0）
  - 2）initramfs 的 systemd 执行 **initrd.target**目标的所有单元（包括其依赖的单元）
    - 内核在initramfs中查找所有硬件的驱动程序，随后内核初始化这些硬件
    - initrd-root-fs.target 以只读形式将系统实际的root文件系统挂载到 /sysroot
    - 执行initrd.target 目标的其他相关单元
    - initrd-switch-root.target 切换 root文件系统（从 initramfs的root文件系统切换到系统实际root文件系统），并将控制权交给实际 root文件系统上的 systemd 实例

梁如军（linuxbooks@126.com）

系统启动过程（4）

- 执行本地系统的第一个进程systemd
  - 1）systemd使用系统中安装的 systemd 副本（PID=1）自行重新执行
  - 2）systemd查找系统配置的默认目标或从内核命令行传递的默认目标
  - 3）systemd启动默认目标**default.target**的所有单元并自动解决单元间的依赖关系
    - 若默认目标为 multi-user.target，则最终启用文本登录屏幕
    - 若默认目标为 graphical.target，则最终启用图形登录屏幕

Systemd的相关工具——启动过程性能分析

- 显示内核和普通用户空间启动时所花的时间
  - systemd-analyze **time**
- 列出所有正在运行的单元
  - systemd-analyze **blame**
- 显示在所有系统单元中是否有语法错误
  - systemd-analyze **verify**
- 将整个引导过程写入一个SVG格式文件
  - systemd-analyze **plot** > boot.svg

梁如军（linuxbooks@126.com）

Systemd的相关工具
——查看单元的资源使用情况

- 以递归形式显示systemd利用的Cgroup结构
  - systemd-cgls
- 显示每个Cgroup中的systemd单元的资源使用情况
  - systemd-cgtop

Systemd的相关工具
——使用journalctl命令查看日志

CentOS
The Community ENTerprise Operating System

journalctl	显示journal记录的所有日志
journalctl --since yesterday	显示自昨天以来记录的日志
journalctl -f	动态跟踪显示最新日志信息
journalctl -p err	显示日志级别为err的日志
journalctl -k	显示内核日志
journalctl -b	显示最近一次的启动日志
journalctl -b-1 -p err	显示上次启动时的错误日志
journalctl -u sshd.service	显示systemd指定单元的日志
journalctl _COMM=sshd	显示进程名为sshd的相关日志

梁如军（linuxbooks@126.com）

CentOS
The Community ENTerprise Operating System

备份与同步

梁如军（linuxbooks@126.com）

备份简介

- 什么是备份
  - 备份就是把一个文件系统或其部分文件存储到另外的介质中，以使得通过这些介质中的记录信息可以恢复原有的文件系统或其中的某些文件。
- 备份介质的选择
  - 磁带、硬盘、光盘、软盘
  - 选择备份介质应该从存储容量、可靠性、速度和介质价格之间进行权衡

实施备份应考虑的因素

- 选择备份介质
- 选择备份策略
- 选择要备份的数据
- 选择合适的备份工具
- 选择是否进行远程备份或网络备份
- 备份的自动化（备份周期和备份文件的存放周期）

备份策略

备份方式	备份内容	工作量	恢复步骤	备份速度	恢复速度	优缺点
完全备份	全部内容	大	一次操作	慢	很快	占用空间大，恢复快
增量备份	每次修改后的所有内容	小	多次操作	很快	中	占用空间小，恢复麻烦
差分（累计）备份	自上次完全备份之后修改的所有内容	中	二次操作	快	快	占用空间较小，恢复快

梁如军（linuxbooks@126.com）

备份分类

- 系统备份：实现对操作系统和应用程序的备份
  - 只需要备份不稳定部分
  - 系统数据并不经常发生改变，所以一般只有当系统内容发生变化时才进行
- 用户备份：实现对用户文件的备份
  - 用户的数据变动更加频繁
  - 需要为用户提供一个合理的最近的数据文件的备份
  - 用户备份通常采用增量备份和（或）差分备份策略进行

备份注意事项

CentOS
The Community ENTerprise Operating System

- 确保备份质量
  - 管理员必须经常验证所做的备份。一个没有验证的备份甚至比没有备份更糟。
- 确保介质安全
  - 保持至少一个备份远离源机器。这是为了防止源机器所在地发生灾难，如火灾等。
- 行业最佳经验
  - 提高备份的可靠性，建议将数据备份到多个介质
  - 并备份到分开的不同地理位置
  - 避免依赖于任何一个单独的存储媒体或物理位置

梁如军（linuxbooks@126.com）

备份、同步与快照

- 备份
  - 在备份时保留历史的备份归档，是为了在系统出现错误后能恢复到从前正确的状态。
- 同步
  - 若无需从历史备份恢复到正确状态，而只备份系统最“新鲜”的状态，此时通常称为同步或镜像。
- 快照
  - 核心思想是：对有变化的文件进行复制；对无变化的文件创建硬链接以减少磁盘占用。

备份工具

- 基本的备份工具
  - cp、**tar**、dd
- 常用的备份工具
  - **rsync**（http://rsync.samba.org/）
  - unison（http://www.cis.upenn.edu/~bcpierce/unison/）
  - rdiff-backup（http://www.nongnu.org/rdiff-backup）
  - **rsnapshot**（http://www.rsnapshot.org/）
  - duplicity（http://duplicity.nongnu.org/）
  - bacula（http://www.bacula.org/）

tar

- 基本功能：打包和解包
- 格式： `tar` [选项] 文件或者目录
- 常用选项

-c：创建新的打包文件。

-t：列出打包文件的内容，查看已经打包了哪些文件。

-x：从打包文件中释放文件。

-f：指定打包文件名。

-v：详细列出 tar 处理的文件信息。

-z：用 gzip 来压缩/解压缩打包文件。

-j：用 bzip2 来压缩/解压缩打包文件。

-J：用 xz 来压缩/解压缩打包文件。

梁如军（linuxbooks@126.com）

tar命令举例

CentOS
The Community ENTerprise Operating System

```
$ tar -cvf myball.tar somedirname
$ tar -tf myball.tar
$ tar -xvf myball.tar
$ tar -zcvf myball.tar.gz somedirname
$ tar -ztf myball.tar.gz
$ tar -zxvf myball.tar.gz
$ tar -jcvf myball.tar.bz2 somedirname
$ tar -jtf myball.tar.bz2
$ tar -jxvf myball.tar.bz2
$ tar -Jcvf myball.tar.xz somedirname
$ tar -Jtf myball.tar.xz
$ tar -Jxvf myball.tar.xz
```

使用tar进行备份

- tar命令的完整格式
  - tar 选项 <-cf 备份文件或设备> <备份路径>
  - tar 选项 <-xf 备份文件或设备> [-C 恢复路径]
- 常用选项

  -M ： 分卷处理

  -p ： 保留权限

  -T filename ： 指定备份文件列表

  -N DATE ： 备份指定日期之后修改的文件

梁如军（linuxbooks@126.com）

tar备份举例(1)

```
# tar -zcvpf /backups/full-backup.tar.gz /home /etc

# tar -zcvpf /backups/full-backup.tar.gz -C / \
  $(ls /| egrep -v "backups|mnt|media|dev|lost+found|proc")

# tar -zcvpf /backups/full-backup.tar.gz -C / \
   --exclude=mnt --exclude=media --exclude=dev \
   --exclude=proc  --exclude=backups  \
   --exclude=*/lost+found  --exclude=var/spool/squid
```

tar备份举例(2)

```
tar -zcvpf /backups/full-backup_$(date +%F).tar.gz /home
tar -zcvpf /backups/full-backup_$(date +%Y%m%d-%H%M).tar.gz /home

tar -N 2014-01-29 -zcvpf /backups/inc-backup_$(date +%F).tar.gz /home
tar -N $(date -d yesterday "+%F") \
     -zcvpf /backups/inc-backup_$(date +%F).tar.gz /home

date +%F>/backups/last-full/full-backup-date
tar -N $(cat/backups/last-full/full-backup-date)\
    -zcvpf /backups/inc-backup_$(date +%F).tar.gz /home
```


梁如军（linuxbooks@126.com）

使用 tar 恢复文件

- 恢复全部文件
- 恢复指定文件
- 文件的恢复顺序

rsync（remote synchronize）

- rsync 是一个远程数据同步工具
  - 可通过LAN/WAN同步不同主机上的文件或目录
  - 可以同步本地硬盘中的不同文件或目录
- rsync 使用所谓的 **rsync算法** 进行数据同步
  - 同步若干新文件时：只复制有变化的文件
  - 同步原有文件时：只复制文件的变化部分
  - 参考 How Rsync Works A Practical Overview
- rsync 目前由 http://rsync.samba.org 维护

rsync 的基本特性

- 可以镜像保存整个目录树和文件系统
- 可以很容易做到保持原来文件的权限、时间、软硬链接等
- 无须特殊权限即可安装
- 优化的流程，文件传输效率高
- 可以使用 rsh、ssh 方式来传输文件，当然也可以通过直接的 socket 连接
- 支持匿名传输，以方便进行网站镜象

梁如军（linuxbooks@126.com）

rsync 使用的两种方式

- 远程Shell方式
  - 可以使用rsh、ssh等远程Shell，默认使用ssh
  - 用户验证由远程Shell负责
- C/S方式
  - 客户连接远程 rsync 服务器
  - rsync 服务器默认监听 **873** 端口
  - 用户验证由 rsync 服务器负责
    - rsync 服务器也可配置为匿名访问
  - 访问rsync服务器时可使用URL（**rsync://host**）

梁如军（linuxbooks@126.com）

rsync 命令

- 同步本地文件或目录
  - **rsync [OPTION...] SRC... [DEST]**
- 将远程文件或目录同步到本地（拉）
  - **rsync [OPTION...] [USER@]HOST:SRC... [DEST]**
- 将本地文件或目录同步到远程（推）
  - **rsync [OPTION...] SRC... [USER@]HOST:DEST**

rsync 命令的常用选项

选项	说明
-a，--archive	归档模式，等价于 -rlptgoD（不包括-H）
-r，--recursive	对子目录以递归模式处理
-l，--links	保持符号链接文件
-H，--hard-links	保持硬链接文件
-p，--perms	保持文件权限
-t，--times	保持文件时间信息
-g，--group	保持文件属组信息
-o，--owner	保持文件属主信息（仅 root 可用）
-D	保持设备文件和特殊文件（仅 root 可用）

梁如军（linuxbooks@126.com）

rsync 命令的常用选项续

选项	说明
-e，--rsh=COMMAND	指定远程Shell程序，RHEL/CentOS默认为ssh
-z，--compress	在传输文件时进行压缩处理
--delete	删除那些接收端还保留而发送端已经不存在的文件
--delete-after	接收者在传输之后才进行删除操作
--exclude=*PATTERN*	指定排除不需要传输的文件匹配模式
--include=*PATTERN*	指定需要传输的文件匹配模式
-P	等价于--partial --progress
--partial	保留因故没有完全传输的文件，以加快随后的再次传输
--progress	在传输时显示传输过程
-v，--verbose	详细输出模式
-q，--quiet	精简输出模式

梁如军（linuxbooks@126.com）

rsync 命令应用举例（1）

- 将整个 /home 目录及其子目录同步到 /backups

  # rsync -a --delete /home /backups

- 将 /home 目录下的所有内容同步到 /backups/home.0

  # rsync -a --delete /home/ /backups/home.0

- 执行“推”复制同步

  [root@soho ~]# rsync /etc/hosts centos5:/etc/hosts

- 执行“拉”复制同步

  [root@centos5 ~]# rsync soho:/etc/hosts /etc/hosts

梁如军（linuxbooks@126.com）

rsync 命令应用举例（2）

- 执行“推”复制同步用户的环境文件

  [osmond@soho ~]$ rsync ~/.bash* centos5:

- 执行“拉”复制同步用户的环境文件

  [osmond@cnetos5 ~]$ rsync soho:~/.bash* .

- 执行“推”复制同步站点根目录

  [osmond@soho ~]$ rsync -avz --delete /var/www
  root@192.168.0.101:/var/www

- 执行“拉”复制同步站点根目录

  [osmond@cnetos5 ~]$ rsync -avz --delete
  root@192.168.0.55:/var/www /var/www

梁如军（linuxbooks@126.com）

rsync 命令应用举例（3）

- 从匿名 rsync 服务器同步 CentOS 的 yum 仓库
  - 同步到本地 /var/ftp/yum/distr/centos/ 目录
  - 不同步SRPMS、x86_64和isos 目录

```
# rsync -aqzH --delete --delay-updates \
  --exclude=SRPMS/  --exclude=x86_64/ \
  --exclude=isos/ \
  rsync://mirror.centos.net.cn/centos/6.6  \
  /var/ftp/yum/distr/centos/
```


梁如军（linuxbooks@126.com）

配置 rsync 服务

1. 首先要选择服务器启动方式
   - 对于负荷较重的 rsync 服务器应该使用独立运行方式
   - 对于负荷较轻的 rsync 服务器可以使用 xinetd 运行方式
2. 创建配置文件 rsyncd.conf
3. 对于非匿名访问的 rsync 服务器还要创建认证口令文件

部署 rsync 服务器的两种方法

- 在生产服务器上同时运行 rsync 服务
  - rsync 服务以只读方式提供要备份的数据，从而避免破坏生产服务器上的数据
  - 根据需要，可以配置一个或多个（为了避免风险）主机作为备份主机
  - 在每个备份主机上以“拉”的方式从生产服务器将数据同步到备份主机
- 在备份服务器上运行 rsync 服务
  - 备份服务器实际上是个数据仓库，他集中收集了网络中所有要备份的主机的数据
  - 备份服务器上运行的 rsync 服务以读写方式提供备份空间
  - 根据需要，可以配置一个或多个（为了避免风险）备份服务器
  - 在每台要备份的主机（包括生产服务器）上以“推”的方式将备份数据写入备份服务器

梁如军（linuxbooks@126.com）

使用rsnapshot工具

- /etc/rsnapshot.conf

```
// 确定备份目录
snapshot_root   /.snapshots/
// 确定备份间隔
interval    hourly  6           // 保留6个基于小时的备份快照
interval    daily   7           // 保留7个基于天的备份快照
interval    weekly  4           // 保留4个基于周的备份快照
interval    monthly 3           // 保留3个基于月的备份快照
// 确定备份内容
backup  /etc/          localhost/
backup  /home/         localhost/
backup  /var/www/      localhost/
backup  /usr/local/    localhost/
backup  /root/         localhost/
```


梁如军（linuxbooks@126.com）

使用rsnapshot工具（2）

- /etc/cron.d/rsnapshot

```
0 */4        * * *          root    /usr/bin/rsnapshot hourly
30 3         * * *          root    /usr/bin/rsnapshot daily
0  3        * * 1          root    /usr/bin/rsnapshot weekly
30 2         1 * *          root    /usr/bin/rsnapshot monthly
```


梁如军（linuxbooks@126.com）

周期性同步与实时同步

- 周期性同步
  - 安排cron任务实施周期性同步
- 实时同步
  - 监控文件系统变化，一旦发现文件有变化就立即同步。

梁如军（linuxbooks@126.com）

Inotify与lsyncd

- Inotify是一种基于内核的文件变化通知机制
- inotify-tools（EPEL仓库提供）是在shell 环境中使用inotify 功能的一套辅助工具
- lsyncd（Live Syncing (Mirror) Daemon）是一个使用lua语言编写的轻量级的在线镜像解决方案。
  - 守护进程lsyncd收集几秒钟内的inotify事件，然后创建一个或多个进程将本地文件系统上的变化同步到远程（或本地的其他目录）。默认是以 rsync方式同步，同时支持rsync+ssh方式。

梁如军（linuxbooks@126.com）

使用lsyncd实现实时同步

- 安装（EPEL）
  - # yum install lsyncd
- 配置（/etc/lsyncd.conf）

```
sync{default.rsyncssh, source="/var/www", host="192.168.0.252",
  targetdir="/var/www/"}
sync{default.rsyncssh, source="/var/lib/tomcat/webapps",
  host="192.168.0.252", targetdir="/var/lib/tomcat/webapps/"}
```

- 启动
  - # systemctl start lsyncd
  - # systemctl enable lsyncd

CentOS
The Community ENTerprise Operating System

故障排查与修复

梁如军（linuxbooks@126.com）

故障排查概述

- 主机故障排查
  - 系统启动故障
  - 文件系统故障
  - 用户登录故障
  - 软件包故障
- 网络故障排查
  - 本机网络配置
  - 与互联网的连接配置
  - 本机的服务无法被访问

梁如军（linuxbooks@126.com）

GRand Unified Bootloader

- GRUB 是一款与操作系统无关的启动加载器
- 提供了交互操作界面和命令行界面
- 支持多种文件系统的类型的访问
  - 可从 ext 2/3/4, XFS, FAT等文件系统引导
  - GRUB device
    - **‘hd0,msdos1’** → /dev/sda1 （MSDOS类型分区表）
    - **‘hd0,gpt1’** → /dev/sda1 （GPT类型分区表）
- 在启动过程中可读取GRUB的配置文件
- 支持多种内核的可执行文件格式
- 支持无盘系统、支持 口令保护

梁如军（linuxbooks@126.com）

GRUB的配置文件

- GRUB 2 在加载时会读取其配置文件：
  - 对于 BIOS 固件系统：/boot/grub2/grub.cfg
  - 对于 UEFI 固件系统：/boot/EFI/redhat/grub2/grub.cfg
- 配置文件不能手工编辑，而是由 **grub2-mkconfig** 工具生成的，生成时会参考：
  - 位于/boot目录中的kernel和initramfs文件
  - 位于/etc/default/grub的自定义设置文件
  - 位于/etc/grub.d/目录下的模板文件

梁如军（linuxbooks@126.com）

GRUB的操作界面

- 在引导屏中，可以：
  - 使用空格键选择，用上/下方向键移动
  - 在菜单编辑模式（’e’）修改现有配置段
  - 键入 ‘c’ 进入 GRUB 的命令行模式
    - 启动 grub.conf 中没有配置的其他操作系统
    - 显示系统信息
    - 按 TAB 查看可用的命令
- 在Shell环境下
  - **grub2-mkconfig 生成grub的配置文件**
  - **grub2-install 重新安装grub**

Systemd用于系统修复的目标

项目	rescue.target	emergency.target
启动目标	rescue.target ├─**rescue.service** └─**sysinit.target** ├─cryptsetup.target ├─local-fs.target └─swap.target	emergency.target └─**emergency.service**
Shell	rescue.service 提供sulogin	emergency.service 提供sulogin
Root 文件系统	以**读写**方式挂装Root文件系统	以**只读**方式挂装Root文件系统
从当前环境切换进入	**systemctl isolate rescue.target** 或 **systemctl rescue**	**systemctl isolate emergency.target** 或 **systemctl emergency**
通过GRUB编辑启动项进入	在 linux16 一行的行末添加 **systemd.unit=rescue.target**，并按Ctrl-x启动	在 linux16 一行的行末添加 **systemd.unit=emergency.target**，并按Ctrl-x启动

梁如军（linuxbooks@126.com）

系统援救环境

- 在RHEL/CentOS的安装程序Anaconda中提供了一种援救环境（rescue environment）
- 解决在执行systemd守护进程之前发生的故障
  - 也可以修复目标**rescue/emergency**能解决的故障
- 使用安装光盘启动系统
  - 进入Troubleshooting菜单
  - 选择 Rescue installed system
- 进入援救环境之后便可以使用其提供的各种工具对系统进行修复。

梁如军（linuxbooks@126.com）

进入系统援救环境（续）

- 自动挂载硬盘中的文件系统到当前Linux系统中的“/mnt/sysimage”目录中

```
Your system is mounted under the /mnt/sysimage directory.
When finished please exit from the shell and your system will reboot.

sh-3.1# ls /mnt/sysimage/
bin   dev  halt  lib         media  mnt  proc  sbin     srv  tmp  var
boot  etc  home  lost+found  misc   opt  root  selinux  sys  usr
sh-3.1# _
```

- 切换根环境到本地硬盘系统
  - **# chroot /mnt/sysimage**

梁如军（linuxbooks@126.com）

修复/etc/fstab丢失故障

- 首先进入援救环境
- 手动查找并挂载根分区
  - 查找逻辑卷：
    - **# lvm vgscan**
  - 激活指定的逻辑卷
    - **# lvm vgchange -ay /dev/VolGroup00**
- 恢复或重建fstab配置文件
- 重新启动系统

梁如军（linuxbooks@126.com）

修复文件系统故障

- 故障原因
  - 非正常关机、突然断电、设备读写失误等
  - 文件系统的超级块（super-block）信息被破坏
- 故障现象
  - 无法向分区中读取或写入数据
  - 启动后提示“Give root password for maintenance”
- 解决思路
  - 根据提示输入root口令，进入修复状态
  - 使用fsck命令进行修复
    - **# fsck -yt ext3 /dev/sdbX**

梁如军（linuxbooks@126.com）

修复磁盘资源耗尽故障

- 故障原因
  - 磁盘空间已被大量的数据占满，空间耗尽
  - 虽然还有可用空间，但文件数i节点耗尽
- 故障现象
  - 无法写入新的文件，提示“…：设备上没有空间”
  - 部分程序无法运行，甚至系统无法启动
- 解决思路
  - 清理磁盘空间，删除无用、冗余的文件
  - 转移或删除占用大量i节点的琐碎文件
  - 进入单用户模式、急救模式进行修复
  - 为用户设置**磁盘限额**

梁如军（linuxbooks@126.com）

修复用户登录故障

- 若普通用户口令丢失，超级用户可以使用passwd命令为用户重新设置
- 若超级用户口令丢失，可以为内核传递参数**rd.break**中断启动过程，重新设置root口令。
- 若用户账号过期，超级用户可以使用chage命令为用户重新设置期限。
- 进一步检查PAM的登录配置
  - /etc/pam.d/login
  - /etc/pam.d/system-auth

梁如军（linuxbooks@126.com）

修复root口令丢失故障

- 在内核初始化后中断系统systemd的执行
  - 可以为内核传递**rd.break**参数
  - 提供一个无需root口令登录的调试Shell
  - 可以修复系统systemd的错误，也可以用来重置root口令

```
switch_root:/# mount -o remount,rw /sysroot
switch_root:/# chroot /sysroot
sh-4.2# passwd root
sh-4.2# exit
switch_root:/# exit
```


梁如军（linuxbooks@126.com）

修复root口令丢失故障

梁如军（linuxbooks@126.com）

修复软件包故障

- 故障原因
  - 非正常关机、误删除运行中的程序文件
  - RPM数据文件被误写或删除
- 故障现象
  - 不能正常查询rpm包信息
  - 无法安装、升级或卸载软件包等
- 解决思路
  - 重建RPM数据库
    - **rpm --rebuilddb 或 rpm --initdb**

梁如军（linuxbooks@126.com）

排查网络配置故障

- 排除非自身因素
- 查看本机IP地址
- 检测与网关的连接
- 检测与互联网的连接
- 测试域名解析
- 测试与远程主机的连接

梁如军（linuxbooks@126.com）

本机的服务无法被访问

- 首先确定服务是否已经启动
- 服务的主配置文件中的访问控制配置
- xinetd的访问控制配置（若此服务以xinetd启动）
- TCP Wappers（若此服务支持TCP Wappers访问控制）
- IPTABLES防火墙规则
- PAM配置（与系统账户相关的服务，如SSH服务、基于本地用户的FTP服务等）
- SELINUX配置（若系统启用了SELINUX）

梁如军（linuxbooks@126.com）

本章思考题

- 常用的系统监视工具有哪些？如何判断系统性能的优劣？
- 使用系统监视工具如何判断CPU、内存和磁盘I/O的瓶颈？
- 内核的功能？内核的主要组件？什么是内核模块？
- 如何显示系统装载的内核模块、显示指定模块的信息、动态装载/卸载内核模块？
- 如何修改内核参数？sysctl的功能？
- 简述Linux的启动过程。比较Systemd的目标与SysVinit的运行级别。

梁如军（linuxbooks@126.com）

本章思考题（续2）

- 什么是备份？简述三种备份策略的不同。常用的备份工具有哪些？
- 简述Inotify机制的作用？如何利用Inotify机制实现实时同步？
- 什么是GRUB？其功能如何？GRUB有哪几种操作界面？
- 简述系统故障排查的方法和步骤。
- 如何进入rescue.target和emergency.target？能实施哪些故障修复？
- 如何进入initramfs 调试 shell？能实施哪些故障修复？
- 什么是系统援救环境？如何进入？能实施哪些故障修复？

梁如军（linuxbooks@126.com）

本章实验

- 学会使用top、mpstat、vmstst、iostat工具分析系统性能。
- 学会设置内核支持的最大文件句柄数并开启包转发功能。
- 学会显示和管理systemd的目标。
- 学会使用rsnapshot实现备份，学会使用lsyncd实现实时同步。
- 学会使用GRUB的操作界面，设置GRUB口令保护。
- 学会进入initramfs 调试 shell并重新设置root口令。

进一步学习

- 添加自己的usb驱动，使用dracut命令重新生成initramfs文件。
- 学习类top在线监控命令（htop、ftop、iftop、iotop、nettop等）的使用。
- 学习sar的运行机理及其命令的使用。
- 学习单机监视工具Monitorix和Monit的配置和使用（EPEL仓库有提供）。
- 学习监视工具Cacti 、Nagios和Zabbix的配置和使用（EPEL仓库有提供）。