第 6 讲　Linux 磁盘和文件系统管理

王晓庆

wangxiaoqing@outlook.com

May 5, 2016

Outline

如何使用 Linux 系统中新添加的磁盘？

1. 对新磁盘进行分区
2. 在分区上创建文件系统 (格式化)
3. 将文件系统挂载到挂载点

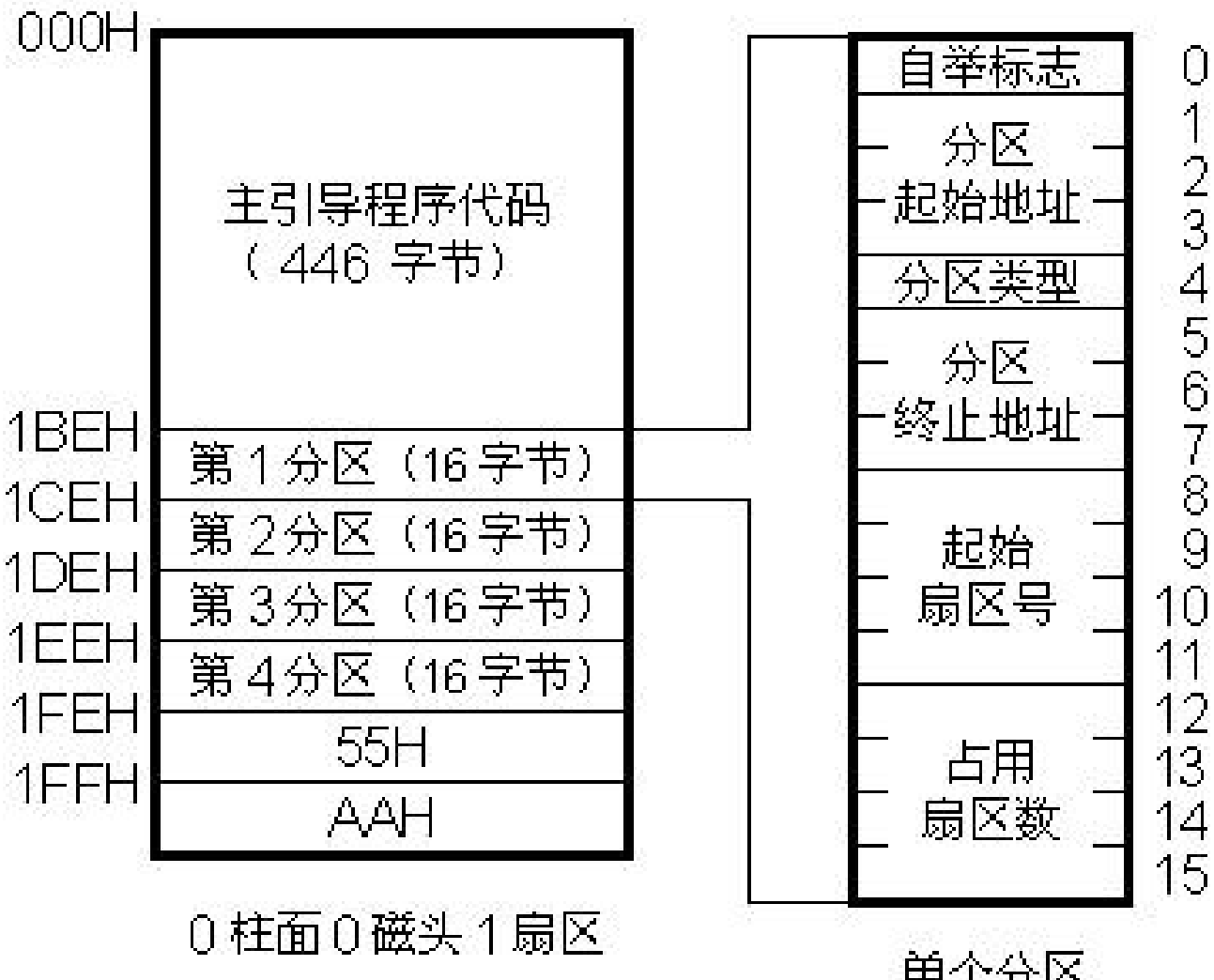
创建和管理磁盘分区

- Linux 磁盘设备文件名
  - IDE 接口设备 (由物理连接来区分，一般最多接 4 个接口)
    1. /dev/hda # 第 1 个 IDE 通道的主设备 (master)
    2. /dev/hdb # 第 1 个 IDE 通道的从设备 (slave)
    3. /dev/hdc # 第 2 个 IDE 通道的主设备 (master)
    4. /dev/hdd # 第 2 个 IDE 通道的从设备 (slave)
  - SCSI/SAS/SATA/USB 接口设备

    IDE 磁盘代号只有一个字母，而 SCSI 磁盘代号可以多达两个字母
    1. /dev/sda # 第 1 块 SCSI 磁盘
    2. /dev/sdad # 第 30 块 SCSI 磁盘

Linux 分区

- 主引导记录 (Master Boot Record, MBR) 和分区表



Linux 分区 (2)

- 查看 MBR 和分区表

```
# 复制 mbr 到文件
dd if=/dev/sda of=mbr.dat bs=512 count=1
# 查看 mbr
hexdump -C mbr.dat
# 查看分区表
hexdump -s 446 -n 64 -e \
'1/1 "%02x "
 3/1 "%02x"
 1/1 " %02x "
 3/1 "%02x" " "
 2/4 "%08x "
 "\n"' mbr.dat
```

Linux 分区 (3)

- 分区表项内容解读

偏移量	长度	定义
0x00	1B	0x00 非活动分区；0x80 活动分区
0x01	1B	起始磁头号
0x02	2B	起始扇区号：0x02 的低 6 位
		起始柱面号：0x02 的高 2 位和 0x03 字节
0x04	1B	文件系统标识：0x83 代表 Linux 分区
0x05	1B	结束磁头号
0x06	2B	结束扇区号：0x06 的低 6 位
		结束柱面号：0x06 的高 2 位和 0x07 字节
0x08	4B	起始逻辑扇区号
0x0C	4B	总扇区数

Linux 分区 (4)

- 分区编号与分区文件名
  - IDE 磁盘：最多 63 个分区
  - SCSI 磁盘：最多 15 个分区
  - 主分区和扩展分区：1 ～ 4
  - 扩展分区中的逻辑分区：5 ～ 15(63)
  - 例：/dev/sda1, /dev/sdb5
- 分区的文件系统类型
  - Linux 支持多种文件系统类型
  - Ext2、Ext3、Ext4、FAT32、FAT16、NTFS、XFS、...
  - swap 交换分区

利用 fdisk 进行磁盘分区

- 查看分区情况

```
fdisk -l        # 按柱面查看分区
fdisk -lu       # 按扇区查看分区
```

- 对磁盘进行分区

```
fdisk /dev/sdb
m   # 获取帮助
n   # 新增分区
d   # 删除分区
p   # 打印分区情况
t   # 更改分区类型标识
w   # 保存退出
q   # 不保存退出
partprobe       # 不重新启动而让内核重新加载分区表
```

创建文件系统

- 使用 mkfs 创建文件系统

```
mkfs -t ext3 /dev/sdb1
mkfs.ext3 /dev/sdb1
```

- 使用 mke2fs 创建文件系统

```
mke2fs /dev/sdb1
-b   # 指定块大小
-j   # 创建 ext3 文件系统
-i   # 指定每多少字节分配一个 i 节点
-N   # 直接指定 i 节点个数
-n   # 预览创建结果，并不真正创建
-m   # 保留块所占百分比
-L   # 指定分区卷标
```

管理文件系统

- 查看文件系统

```
dumpe2fs /dev/sda1
tune2fs -l /dev/sda1
```

- 调整文件系统

```
tune2fs -j /dev/sda1          # 转换为 ext3 文件系统
tune2fs -L data /dev/sda1     # 设置卷标
blkid /dev/sda1               # 查看设备的 UUID(128 位)
tune2fs -U random /dev/sda1   # 设置随机 UUID
tune2fs -U clear /dev/sda1    # 清除 UUID
man tune2fs
```

- 检查/修复文件系统

```
fsck /dev/sdb1 #fsck 应在分区被卸载时运行
```

挂载文件系统

- mount 命令

```
mount [-t type] [-o option] device mountpoint
mount /dev/sda1 /mnt/sda1 # 挂载点一般为空目录
mount -a                  # 挂载/etc/fstab 的文件系统
mount                     # 打印文件系统挂载情况
cat /etc/mtab             # 同上
mount -t ext2 -o noatime,noexec /dev/sda1 /mnt/sda1
date >/mnt/sda1/date1
mount -o remount,ro /mnt/sda1 # 重新挂载已挂载文件系统
date >/mnt/sda1/date2
mount -o remount,rw /mnt/sda1
date >/mnt/sda1/date3
```

卸载文件系统

- umount 命令
  卸载文件系统时要确保没有进程在访问该文件系统。

```
fuser /mnt/sda1     # 查看正在使用目录的进程
lsof /mnt/sda1      # 查看正被进程打开的文件/目录
umount /dev/sda1    # 设备名
umount /mnt/sda1    # 挂载点
```

- /etc/fstab 文件

```
cat /etc/fstab
man 5 fstab
```

访问外部存储设备

- 挂载/卸载光盘

```
mkdir /media/cdrom                  # 创建挂载点
mount /dev/cdrom /media/cdrom       # 挂载光盘
umount /dev/cdrom                   # 卸载光盘
umount /media/cdrom                 # 同上
eject                               # 弹出光驱
eject -t                            # 收回光驱
```

- 挂载/卸载 USB 存储设备

```
mkdir /media/usb                    # 创建挂载点
mount -t vfat /dev/sdb1 /media/usb  # 挂载
umount /media/usb                   # 卸载
```

制作和使用光盘镜像文件

- 从光盘制作镜像文件

```
cp /dev/cdrom cdrom.iso
```

- 从目录制作镜像文件

```
mkisofs -r -o mydir.iso mydir
```

- 挂载光盘镜像文件

```
mkdir /media/iso
mount -o loop cdrom.iso /media/iso
```

交换空间

- 操作系统可在需要时暂时换出部分内存数据至磁盘的交换空间以腾出更多内存空间，或从交换空间将数据换入内存。
- Linux 支持两种形式的交换空间
  1. 交换分区
  2. 交换文件
- Linux 系统最多可以有 32 个交换空间,386 兼容平台上每个交换空间最大不能超过 2GB
- 分配交换空间的建议：以 4MB 或 8MB 为单位，一般为物理内存 1 ～ 2 倍

交换分区

```
# 准备：创建分区/dev/sdb1 并设置其类型为 82(Linux swap)
mkswap /dev/sdb1
free
swapon /dev/sdb1
swapon -s
free
swapoff /dev/sdb1
free
```

交换文件

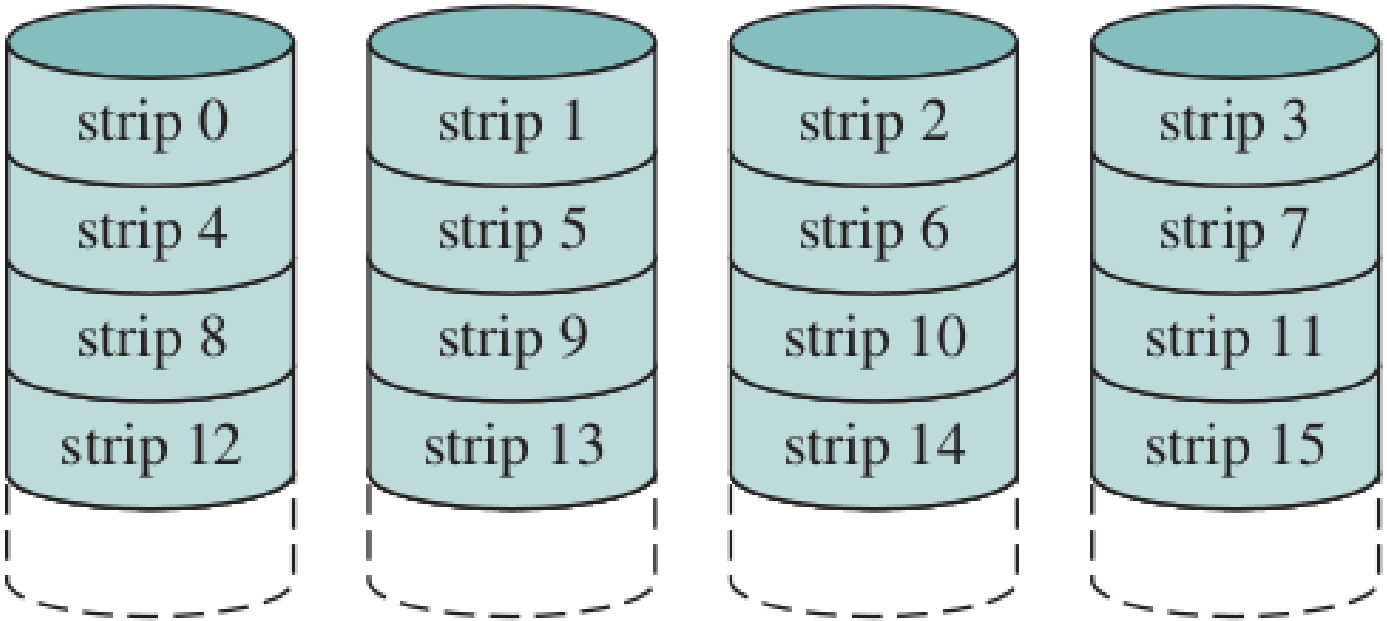
```
dd if=/dev/zero of=/tmp/swapfile bs=1M count=200
mkswap /tmp/swapfile
free
swapon /tmp/swapfile
swapon -s
free
swapoff /tmp/swapfile
rm /tmp/swapfile
```

独立磁盘冗余阵列 (Redundant Array of Independent Disks, RAID)

1. 可以通过并行处理多个独立的 I/O 请求提高读写性能
2. 可以通过增加冗余信息来提高数据存储的可靠性

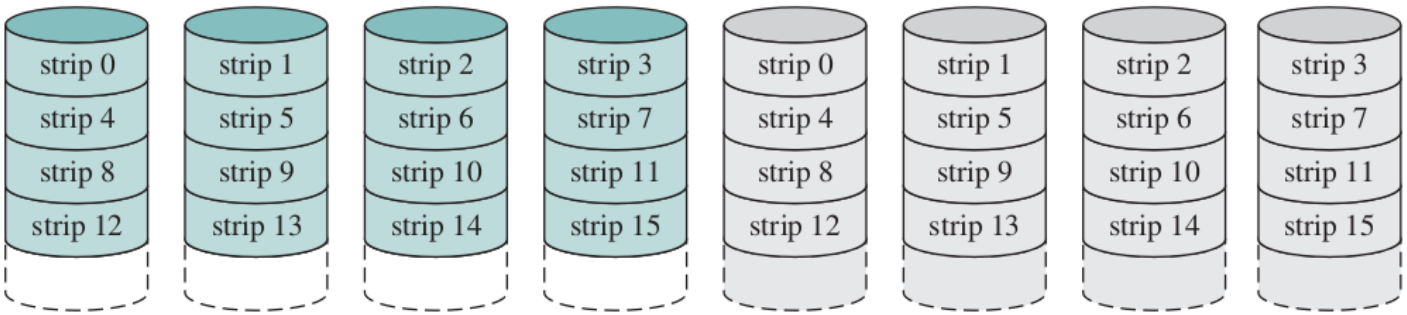
RAID 0

- 非冗余，读写性能好，数据可靠性低于单个磁盘



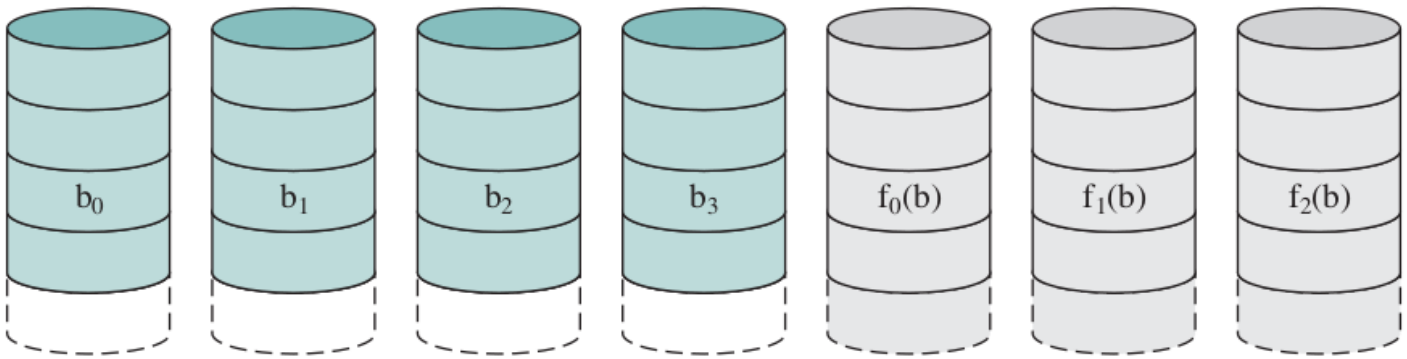
RAID 1

- 镜像，读性能好，写性能与单个磁盘相当，数据可靠性高，成本高



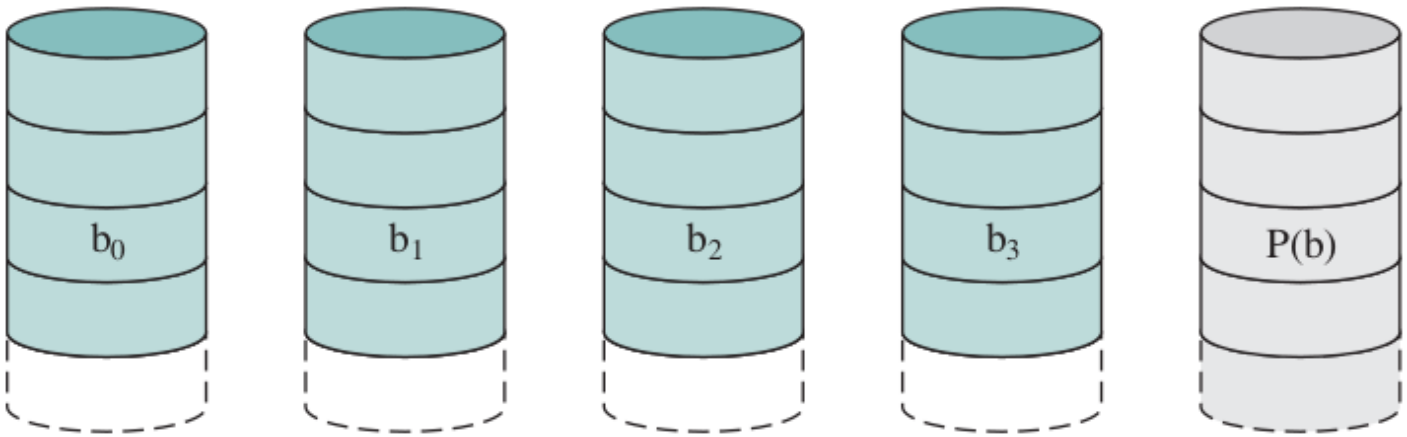
RAID 2

- 并行访问，通过海明码实现冗余，读写性能好，磁盘同步旋转，带检错纠错功能，可靠性高，读写性能好，但一次只能执行一个 I/O 请求



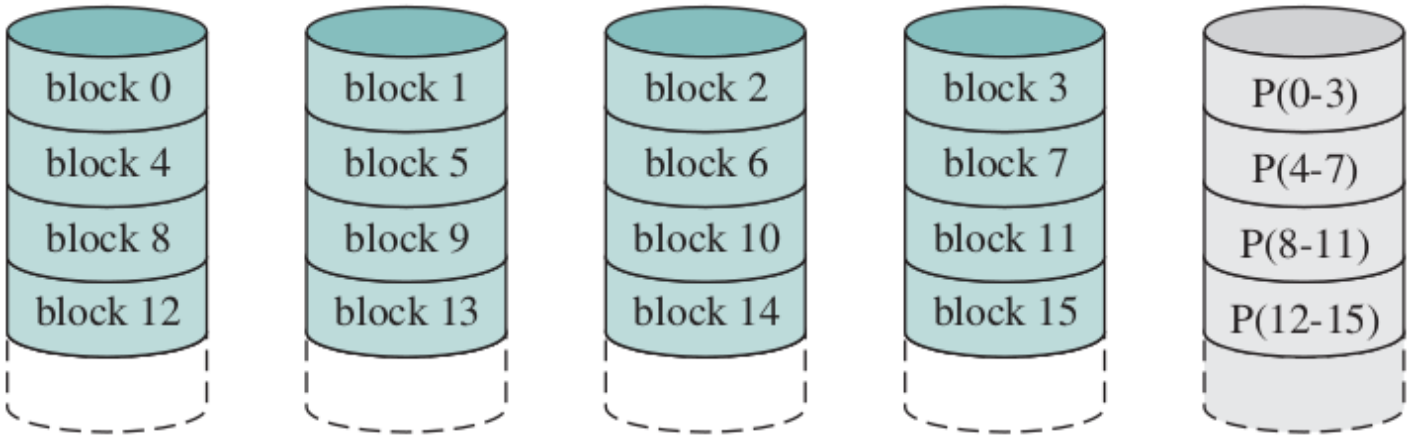
RAID 3

- 并性访问，通过奇偶校验实现冗余，读写性能好，磁盘同步旋转，带检错功能，可靠性高，读写性能好，但一次只能执行一个 I/O 请求



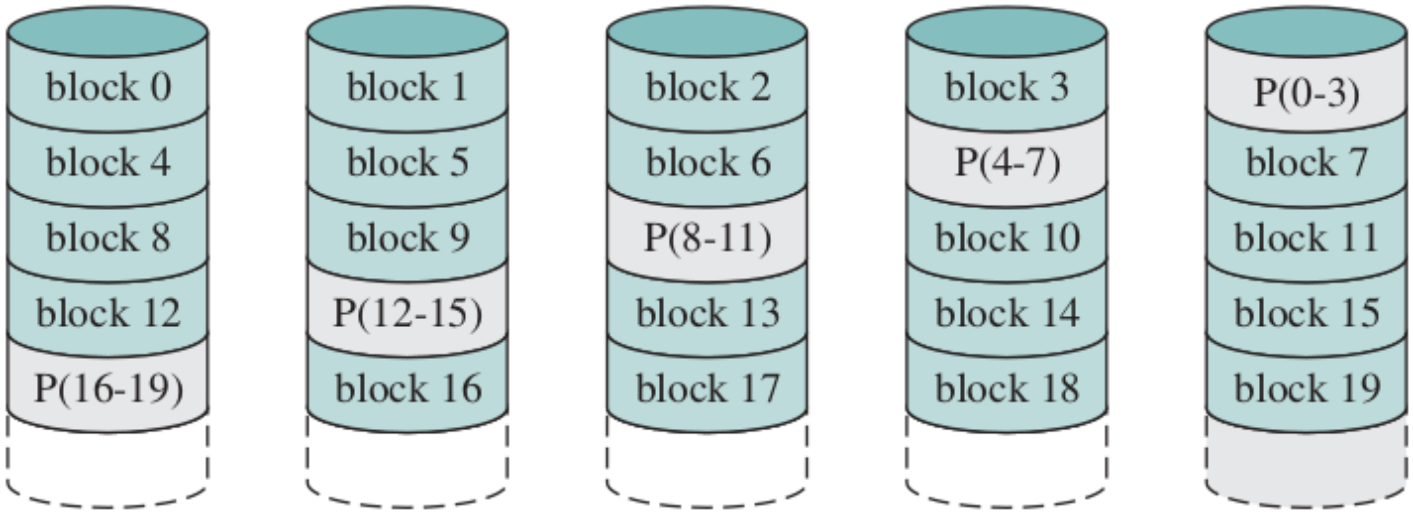
RAID 4

- 独立访问，以块为单位计算奇偶校验块并存放与校验盘，数据可靠性高，读性能好，写性能差（因为每次写都要更新校验盘数据），校验盘成为性能瓶颈



RAID 5

- 在 RAID 4 基础上，把奇偶校验块循环分布在所有磁盘上，从而减轻单个校验盘的性能瓶颈问题，读写性能和可靠性类似于 RAID 4



硬件 RAID 和软件 RAID

- 硬件 RAID
  1. 利用硬件 RAID 控制器来实现，由集成或专用的阵列卡来控制硬盘驱动器
  2. 存取性能和数据保护能力高，但成本也高
  3. Linux 将硬件磁盘阵列看作一块实际的硬盘，其设备名为/dev/sd[a-p]
- 软件 RAID
  1. 利用操作系统提供的软件 RAID 功能来实现
  2. 适用于要求不高的场合，成本低
  3. Linux 将软件磁盘阵列看作多重磁盘设备 (MD)，其设备名为/dev/md0、/dev/md1 等。

利用 mdadm 管理软件 RAID 1 阵列 (1)

- 创建 RAID 1 阵列

```
#1. 创建两个相同大小的 RAID 分区, 设置分区 id 为 fd
#2. 建立 RAID 设备
mdadm --create /dev/md0 --level 1 --raid-devices=2 \
/dev/sdb1 /dev/sdc1
#3. 设置 mdadm 配置文件/etc/mdadm.conf
DEVICE /dev/sdb1 /dev/sdc1
ARRAY /dev/md0 devices=/dev/sdb1,/dev/sdc1
#4. 建立文件系统
mkfs -t ext3 /dev/md0
#5. 挂载 RAID 1 设备
mkdir /mnt/raid1
mount /dev/md0 /mnt/raid1
```

利用 mdadm 管理软件 RAID 1 阵列 (2)

- 管理 RAID 1 阵列

```
# 模拟某成员磁盘发生故障
mdadm /dev/md0 --fail /dev/sdc1
# 从 RAID 1 阵列中移除故障成员
mdadm /dev/md0 --remove /dev/sdc1
# 准备一块要替换的磁盘，并将新磁盘加入到阵列中
mdadm /dev/md0 --add /dev/sdd1
# 查看阵列实时信息
cat /proc/mdstat
mdadm --detail /dev/md0
```

利用 mdadm 管理软件 RAID 5 阵列 (1)

- 创建 RAID 5 阵列

```
#1. 准备 4 个阵列成员 (创建 RAID 分区)
#2. 创建 RAID 设备：系统默认只有 md0 设备, 其他需自行创建
ls -l /dev/md0 # 查看 md 设备的类型和主次设备号
mknod /dev/md1 b 9 1 # 创建设备文件
#3. 建立 RAID 5 设备
mdadm --create /dev/md1 --level=5 \
--raid-devices=3 --spare-devices=1 /dev/sdd[5-8]
mdadm --detail /dev/md1
#4. 设置 mdadm 配置文件/etc/mdadm.conf
DEVICE /dev/sdd5 /dev/sdd6 /dev/sdd7 /dev/sdd8
ARRAY /dev/md1 devices=/dev/sdd5,/dev/sdd6,/dev/sdd7,/d
mkfs.ext3 /dev/md1 #5. 建立文件系统
mkdir /mnt/raid5
mount /dev/md1 /mnt/raid5 #6. 挂载 RAID 5 设备
```

利用 mdadm 管理软件 RAID 5 阵列 (2)

- 管理 RAID 5 阵列

```
# 利用备用盘重建 RAID 5
mdadm /dev/md1 --fail /dev/sdd6
mdadm --detail /dev/md1
# 可看到备用盘自动参与重建阵列，而故障盘成为备用磁盘，而且
# 注意：要等待 RAID 重建完毕，再替换故障磁盘
# 将故障磁盘移除并加入新磁
mdadm /dev/md1 --remove /dev/sdd6
mdadm /dev/md1 --add /dev/sde1
mdadm --detail /dev/md1
```

利用 mdadm 管理软件 RAID

- 启用/停用/监控 RAID 设备

```
# 停止 RAID 设备 (停止前要先卸载)
mdadm --stop /dev/md0
# 启动 RAID 设备
mdadm --assemble --scan /dev/md0
# 监控 RAID 设备
mdadm --monitor --mail=admin@abc.com \
--delay=180 /dev/md0
# 将监控任务转入后台执行
nohup mdadm --monitor --mail=admin@abc.com \
--delay=180 /dev/md0
```

利用 mdadm 管理软件 RAID

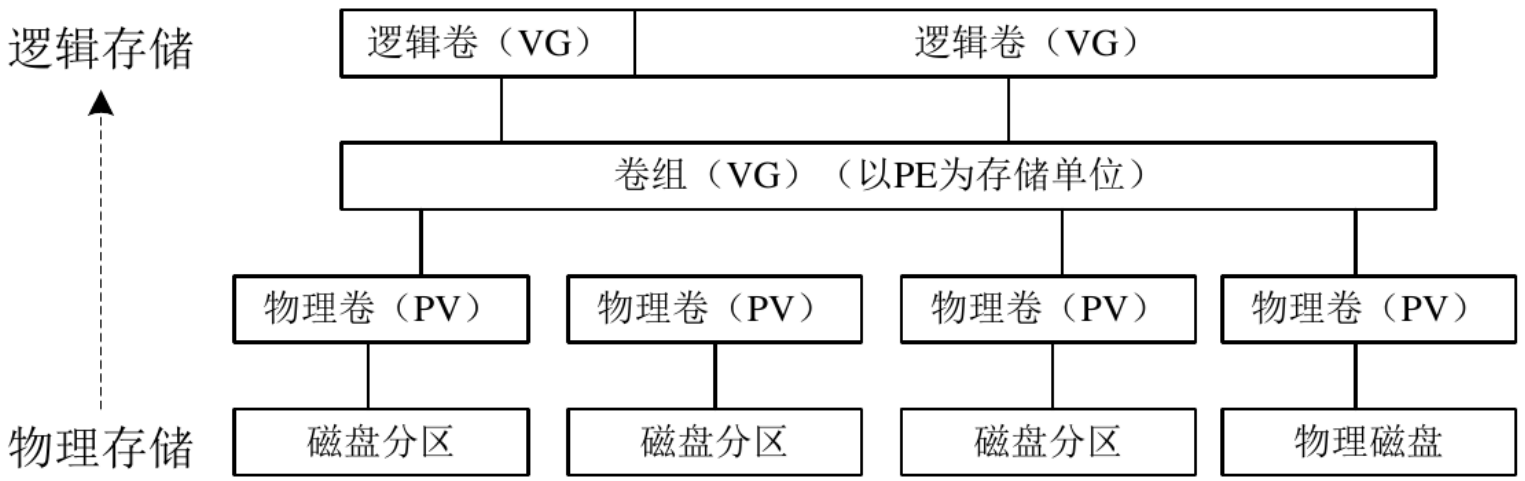
- 删除 RAID 多重磁盘设备
  每个 md 设备只能被建立一次，如果创建命令 (mdadm –create) 出错，将造成该 md 设备无法使用，这时需要按以下步骤先删除该错误的 md 设备，然后才能重新创建它。

```
#1. 停用 RAID 设备
mdadm --stop /dev/md0
#2. 清空每个组成分区的超级块
mdadm --zero-superblock /dev/sdb1
mdadm --zero-superblock /dev/sdc1
```

简介

- 服务器大多处于要求具备高可用性的运行环境，调整磁盘存储空间有时不允许重新引导系统，逻辑卷可以在系统处于运行状态时扩充和缩减，可为管理员提供磁盘存储管理的灵活性。

逻辑卷体系结构


逻辑存储
逻辑卷（VG）
逻辑卷（VG）
卷组（VG）（以PE为存储单位）
物理卷（PV）
物理卷（PV）
物理卷（PV）
物理卷（PV）
物理存储
磁盘分区
磁盘分区
磁盘分区
物理磁盘

逻辑卷管理工具

- 创建逻辑卷步骤
  1. 创建物理卷 (PV)
  2. 创建卷组 (VG)：卷组以大小相等的区域 (PE) 为单位分配存储容量
  3. 创建逻辑卷 (LV)
- lvm2：提供了一组工具用于管理逻辑卷

功能	物理卷	卷组	逻辑卷
扫描检测	pvscan	vgscan	lvscan
显示详细信息	pvdisplay	vgdisplay	lvdisplay
创建	pvcreate	vgcreate	lvcreate
删除	pvremove	vgremove	lvremove
扩充		vgextend	lvextend(lvresize)
缩减		vgreduce	lvreduce(lvresize)
改变属性	pvchange	vgchange	lvchange

建立逻辑卷 (1)

- 1. 建立物理卷

```
# 创建磁盘分区 (100 柱面大小), 并将分区的 ID 设为 8e
pvcreate /dev/sdb2 /dev/sdb3        # 建立物理卷
pvscan                              # 扫描物理卷信息
pvs                                 # 查看物理卷简要信息
pvdisplay                           # 查看物理卷详细信息
```

- 2. 建立卷组

```
vgcreate -s 16M vg1 /dev/sdb[23] # 创建卷组
#-s 指定 PE 大小, 默认为 4M, 最小为 1K, 且必须为 2 的次幂
vgs                                 # 显示卷组简要信息
vgdisplay                           # 显示卷组详细信息
```

建立逻辑卷 (2)

- 3. 建立逻辑卷

```
lvcreate -l 7500 -n lv1 vg1          # 创建逻辑卷
lvcreate -L 1200M -n lv1 vg1         # 同上
#-l 指定 PE 数量, -L 指定容量 (k,m,g,t, 不区分大小写)
lvs                                  # 显示逻辑卷简要信息
lvdisplay                            # 显示逻辑卷详细信息
```

- 4. 使用逻辑卷

```
mkfs -t ext3 /dev/vg1/lv1            # 建立文件系统
mkdir /mnt/lvm1
mount /dev/vg1/lv1 /mnt/lv1          # 挂载逻辑卷
mount /dev/mapper/vg1-lv1 /mnt/lv1   # 同上
# 创建文件测试
dd if=/dev/zero of=/mnt/lv1/bigfile bs=1M count=800
ls /mnt/lv1
df -h /mnt/lv1
```

动态调整逻辑卷容量 (1)

- 放大：先放大逻辑卷，再放大文件系统

```
pvcreate /dev/sdb5                  # 新建物理卷
vgextend vg1 /dev/sdb5              # 扩充卷组
vgdisplay vg1                       # 显示卷组信息
lvdisplay /dev/vg1/lv1              # 显示逻辑卷信息
lvresize -l +50  /dev/vg1/lv1       # 扩充逻辑卷
lvresize -L +800M /dev/vg1/lv1      # 扩充逻辑卷
lvdisplay /dev/vg1/lv1              # 显示卷组信息
df -h /mnt/lv1                      # 查看逻辑卷空间情况
resize2fs /dev/vg1/lv1              # 扩充文件系统
df -h /mnt/lv1                      # 查看逻辑卷空间情况
```

动态调整逻辑卷容量 (2)

- 缩小：先缩小文件系统，再缩小逻辑卷 (操作需谨慎，注意事先备份数据)

```
dh -h /mnt/lv1                    # 查看逻辑卷空间情况
umount /mnt/lv1                   # 卸载逻辑卷
e2fsck -f /dev/vg1/lv1            # 检查逻辑卷
resize2fs /dev/vg1/lv1 1G         # 缩小文件系统
lvreduce -L 1G /dev/vg1/lv1       # 缩小逻辑卷
lvs                               # 查看逻辑卷简要信息
mount /dev/vg1/lv1 /mnt/lv1       # 重新挂载逻辑卷
dh -h /mnt/lv1                    # 查看逻辑卷空间情况
```

动态调整卷组

- 动态替换物理卷

```
# 替换
pvcreate /dev/sdb6              # 创建新物理卷
vgextend vg1 /dev/sdb6          # 向卷组添加物理卷
vgs -o+pv_used                  # 查看物理卷信息
pvmove /dev/sdb2 /dev/sdb6      # 转移物理卷数据
vgreduce vg1 /dev/sdb2          # 从卷组中删除物理卷
vgs -o+pv_used
ls /mnt/lv1
# 恢复
pvextend vg1 /dev/sdb2
pvmove /dev/sdb6 /dev/sdb2
vgreduce vg1 /dev/sdb6
vgs -o+pv_used
ls /mnt/lv1
```

删除 lvm

- 要删除逻辑卷并恢复磁盘分区，可执行建立逻辑卷的逆过程：

```
#1. 卸载逻辑卷
umount /mnt/lv1
#2. 删除逻辑卷
lvremove /dev/vg1/lv1
#3. 停用卷组
vgchange -a n vg1
#4. 删除卷组
vgremove vg1
#5. 删除物理卷
pvremove /dev/sde[23]
#6. 使用 fdisk 将分区 ID 改回 83
```

磁盘配额

- 多个用户共享同一磁盘空间时，为防止某用户 (组) 占用过多磁盘空间，可通过设置磁盘配额 (Disk Quota) 对其可用存储空间进行限制。
- Linux 磁盘配额软件 (quota) 的特点
  1. 只能对分区进行设置，不能对目录进行设置
  2. 可以针对用户设置，也可针对组设置
  3. 磁盘配额只适用于普通用户或组，root 用户不受其限制
- 在 Linux 系统中磁盘配额的限制项目有两种类型
  1. 磁盘容量限制：限制用户能够使用的磁盘块数 (block)，实际应用中多使用此类型
  2. 文件数量限制：限制用户能够使用的索引节点数 (inode)，即文件数

启用和管理磁盘配额

```
#1. 修改/etc/fstab 配置文件
/dev/sdb1 /mnt/sdb1 ext3 defaults,usrquota,grpquota 0 0
#2. 重新挂载分区
mount -o remount /mnt/sdb1
mount
#3. 扫描分区并生成磁盘配额信息文件
quotacheck -cvuga    #/etc/fstab 中设定了配额选项的分区
quotacheck -cvug /mnt/sdb1 # 指定分区
ls /mnt/sdb1
#4. 启用磁盘配额
quotaon -a
quotaon /mnt/sdb1
#5. 编辑用户/组配额
edquota mike
edquota -g stu
```

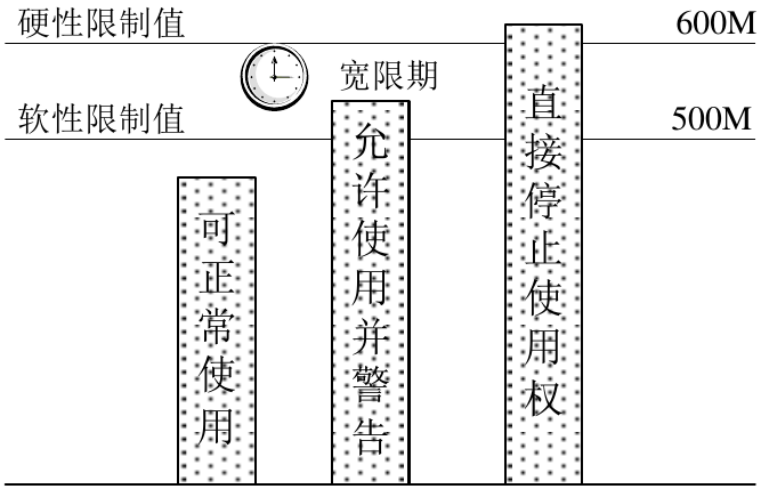
设置磁盘配额

- 配置磁盘配额限制值
  - 硬限制 (hard)
  - 软限制 (soft)
  - 软限制宽限期

```
edquota -t
```

- 复制配额设置

```
# 复制 mike 的配额设置
edquota -p mike mary bob
```



备份内容

- 系统数据备份
  备份操作系统和应用程序，以便在系统崩溃后能恢复系统运行。
- 用户数据备份
  备份用户的数据文件，以便用户恢复自己丢失或损坏的数据。
- 用户的数据变动相对频繁，因此，用户备份应该比系统备份更加频繁。可采用自动定期运行某个程序的方法来备份数据。

备份原理

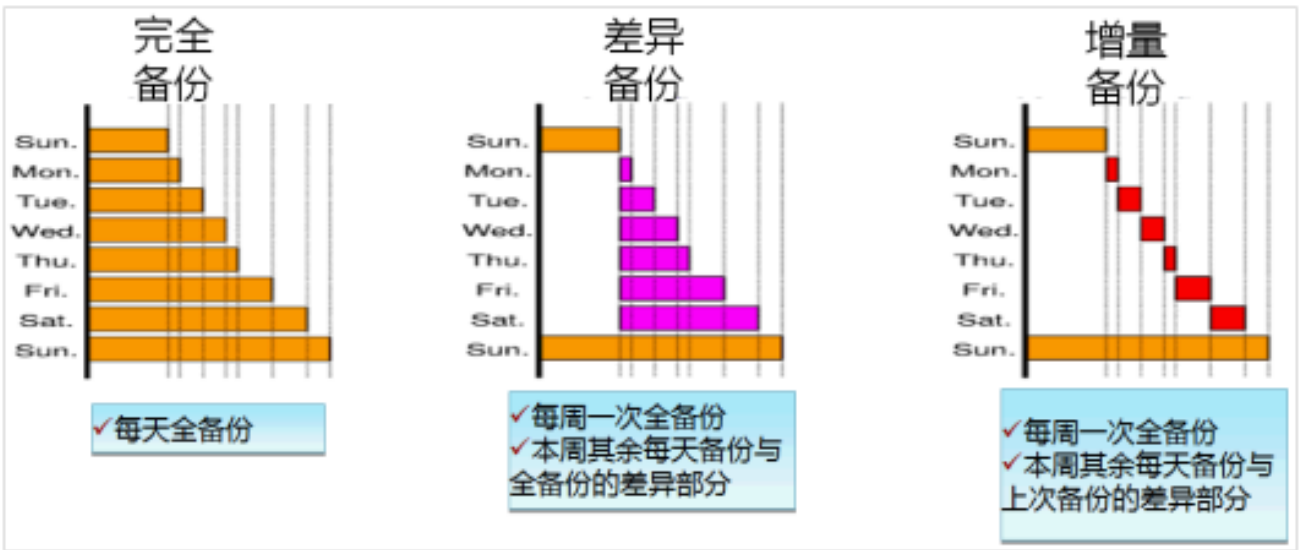
- 备份标识
  - 当一个文件被备份后，该文件会获得一个“已备份”属性，而当文件属性和内容发生变化时，文件的“已备份”属性将被清除。当备份时，就是依据文件的“已备份”属性来确定要备份哪些文件的。

备份方式

- 完全备份 (Full Backup)
  - 对系统进行一次全面的备份，并将所有文件标识为“已备份”
  - 所需时间最长，但恢复时间最短，操作最方便
- 增量备份 (Incremental Backup)
  - 只对上一次备份后增加的和修改过的数据进行备份，并添加“已备份”标识
  - 可快速完成备份，但可靠性较差，恢复较麻烦
- 差异备份 (Differential Backup)
  - 只对上一次完全备份后增加和修改或的数据进行备份，但不添加“已备份”标识
  - 所需时间较短，且恢复较为方便，适合各种场合

备份规划

- 单纯的完全备份
  - 适合数据量不大或者数据变动频率很高的情况
- 完全备份结合增量备份
  - 以较长周期进行完全备份，其间则进行较短周期增量备份
  - 如每周一次完全备份结合每日一次增量备份
- 完全备份结合差异备份
  - 以较长周期进行完全备份，其间则进行较短周期差异备份
  - 如每周一次完全备份结合每日一次差异备份



数据备份操作

- tar 备份
- dump 和 restore 工具
- 光盘备份

dump 和 restore 工具 (1)

- 使用 dump 和 restore 实现备份和恢复
  - dump 能备份任何类型的文件甚至设备，支持完全、增量和差异备份

    注意：dump 只能对分区进行增量备份和差异备份，对目录只能进行完全备份
  - dump 需要指定一个备份级别，它是 0-9 之间的一个整数
  - 级别 0 代表完全备份
  - 级别 n(n>0) 代表基于最近一次低于级别 n 的备份进行差异备份

dump 和 restore 工具 (2)

- 使用 dump 实现完全备份

```
dump -0S /dev/sda1   # 计算完全备份/dev/sda1 所需空间
# 完全备份/boot 分区
dump -0u -f /tmp/boot.dump /boot
dump -0uj -f /tmp/boot.dump.bz2 /boot
#-u 表示更新数据库文件/etc/dumpdates
#    (记录文件日期、存储级别、文件系统等信息)
#-f 指定备份文件
#-j 用 bzip2 进行压缩
dump -W   # 查看分区备份情况
```

dump 和 restore 工具 (3)

- 使用 dump 实现完全备份和增量备份

```
dump -0u -f /tmp/boot0.dump /boot
dump -1u -f /tmp/boot1.dump /boot
dump -2u -f /tmp/boot2.dump /boot
dump -3u -f /tmp/boot3.dump /boot
```

- 使用 dump 实现完全备份和差异备份

```
dump -0u -f /tmp/boot0.dump /boot
dump -1u -f /tmp/boot1.dump /boot
dump -1u -f /tmp/boot2.dump /boot
dump -1u -f /tmp/boot3.dump /boot
```

dump 和 restore 工具 (4)

- 使用 restore 恢复备份

```
# 比较备份文件与文件系统之间的差异
restore -C f /tmp/boot.dump
# 恢复之前，浏览备份文件的数据
restore -tf /tmp/boot.dump
# 恢复备份 (restore 不支持覆盖式还原，应还原至其他位置)
mkdir bootbak
cd bootbak
restore -rf /tmp/boot.dump
#-r 表示重建
# 进入交互式恢复模式
restore -if /tmp/boot.dump
```

光盘备份

- 建立光盘映像文件

```
mkisofs -r -o /tmp/boot.iso /boot
#-r 支持长文件名, -o 指定输出文件
dd if=/dev/sda1 of=/tmp/boot.iso
```

- 刻录光盘

```
cdrecord -scanbus dev=ATA # 获取刻录机设备识别号
cdrecord -v dev=ATA:0,1,0 boot.iso  # 刻录
cdrecord -v -eject speed=8 dev=ATA:0,1,0 boot.iso # 刻录
#-v 输出尽可能多的校验信息
#-eject 刻录完毕后弹出光盘
#speed=8 指定刻录机的速度
```