

文件系统

思考：

一个文件的组成：stat 查看文件信息

目录项

inode号：文件的编号

block：真正存放数据的地方，一个文件一般对应着多个block

重要的参数：

文件类型的ext4和xfs

标准分区空间不足的扩容----》raid和lvm

软链接和硬链接

raid：廉价冗余阵列

LVM：逻辑卷管理 (Logical Volume Manager)

文件系统

思考：

• 文件系统是什么？

- - 1 管理文件和目录的一套机制（方法）
 - 2
 - 3 1. 文件存取
 - 4 2. 文件的查找
 - 5 3. 文件的大小--》2T
 - 6 ext3--》不能超过2T
 - 7 fat32 --》4G
 - 8 4. 一个目录可以存放多少个文件
 - 9
 - 10 5. 文件的命名
 - 11 6. 一个分区可以多大
 - 12 7. 一个文件最大可以多大
 - 13 8. 一个分区使用了多少空间，还有多少空间没有使用
 - 14 等等

• 文件系统如何得到？

- - 1 linux系统的分区方案：
 - 2 标准分区
 - 3 1.centos8 经典方案：
 - 4 / 剩下的空间全备给根分区
 - 5 /boot 1000M左右--》用来存放启动系统所需要的文件--》相当于
 - 6 windows里的c盘
 - 7 swap 物理内存的2倍 用来临时存放数据，当物理内存不足
 - 8 /home 单独的分区
 - 9 2.自定义方案：
 - 10 / xfs 剩下的空间
 - 11 /boot ext4 1G 不需要特别大的空间
 - 12 swap swap 物理内存的2倍

```

12      /mysql_data xfs 200G -->专门存放数据库的数据的分区
13      /home xfs 单独
14

```

- **文件系统的使用?**

- **删除一个大文件速度快, 还是删除一个小文件速度快, 或者是一样?**

- 删小文件快, 因为大文件把使用状态改成未使用状态的修改区域要多一些。

```

1 [root@sanchuang jd]# time rm -rf dd.dd
2
3 real    0m0.034s
4 user    0m0.002s
5 sys 0m0.032s
6 [root@sanchuang jd]# time rm -rf lihu.txt
7
8 real    0m0.003s
9 user    0m0.001s
10 sys 0m0.002s

```

- **删除一个文件到底删除了什么?**

- 删除了目录项、把inode映射表中的inode号标记为未使用
- inode没删
- block没删

- ```
1 删除文件后, 要马上停止新建文件操作, 最好是关机--》使用恢复软件去恢复
2 文件系统里有日志--》会记录磁盘的相关操作--》可以找回目录项等信息
3
```

- **某个程序一直往一个文本文件的体制文件里写入内容, 文件路劲为/var/log/abc.log, 这个文件已经接近20G了, 如何清除这个文件里的内容, 释放空间. 但是不允许停止这个程序**

- 使用重定向 > /var/log/abc.log
- 如果文件没有意义, 因为系统内存中的文件描述符映射表有inode的记录, 还是可以找到源文件。

- **分区格式化后会产生哪些东西:**

```

1 - 1.inode table --》inode空间 inode区
2 2.data block --》数据区
3 3.inode映射表 --》记录哪些inode使用了, 哪些没有使用 --》账簿记录了inode区
 里的inode的使用情况
4 4.block映射表 --》记录哪些block使用了, 哪些没有使用 --》账簿记录了block区
 里的inode的使用情况
5 5.superblock --》记录此 file system 的整体信息, 包括inode/block的总量、使
 用量、剩余量, 以及文件系统的格式
6
7

```

- **通过一个文件名是如何读取到block里的数据:**

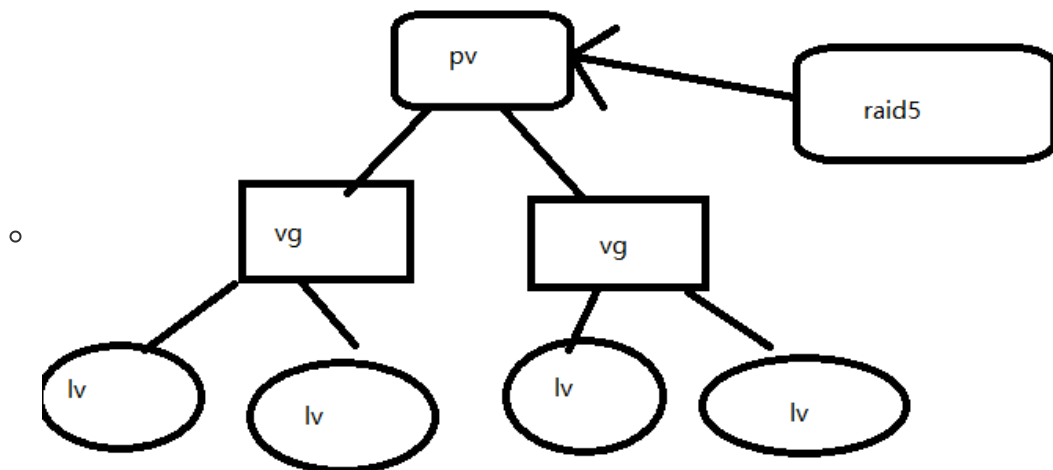
- o stat、ll -i

```

1 [root@sanchuang sanchuang]# mkdir /jd
2 [root@sanchuang sanchuang]# cd /jd/
3 [root@sanchuang jd]# mkdir xiaomi huawei
4 [root@sanchuang jd]# ll -i
5 总用量 0
6 51927840 drwxr-xr-x. 2 root root 6 3月 5 16:43 huawei
7 34270199 drwxr-xr-x. 2 root root 6 3月 5 16:43 xiaomi
8
9
10 [root@localhost lianxi]# echo "Hello world" >> tzk.sh
11 [root@localhost lianxi]# stat tzk.sh
12 文件: tzk.sh
13 大小: 12 块: 8 IO 块: 4096 普通文件
14 设备: 806h/2054d Inode: 100674176 硬链接: 1
15 权限: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
16 环境: unconfined_u:object_r:default_t:s0
17 最近访问: 2020-03-05 07:01:27.541532183 -0500
18 最近更改: 2020-03-05 07:02:01.484760029 -0500
19 最近改动: 2020-03-05 07:02:01.484760029 -0500
20 创建时间: -

```

- **raid是什么? 有哪些类型? raid的作用**
- **lvm是什么? 里面有什么概念? lvm的作用**
  - o 是逻辑卷管理, 有pv、vg、lv, 方便分区的扩容, Linux里的lvm2(软件)+多块磁盘组成
  - o 没有容错功能, 优势是非常方便扩容
  - o 云服务不管raid, 只用设置lvm
- **raid和lvm的区别?**



## 一个文件的组成: stat 查看文件信息

### 目录项

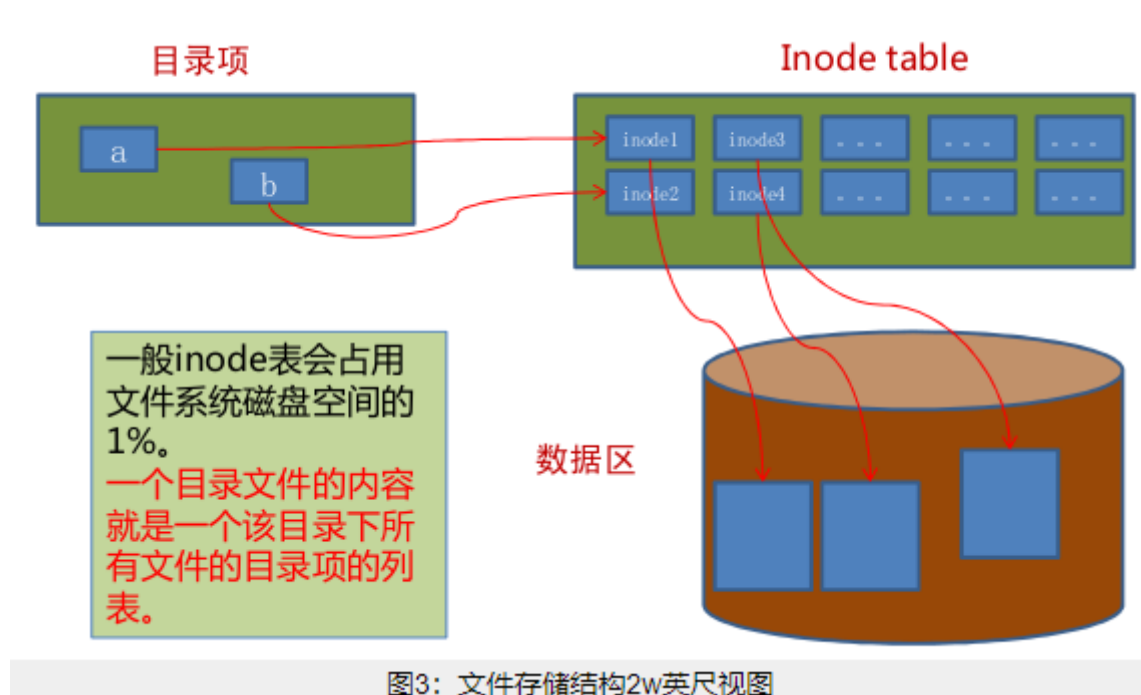
- 文件名+inode号 ==> 一个文件对应着一条目录项



图4: 目录项结构

inode号: 文件的编号

**block**: 真正存放数据的地方, 一个文件一般对应着多个block



## 重要的参数:

- block
- superblock: ---》在分区格式化的时候会显示
- inode

## 文件类型的ext4和xfs

- mkfs.ext4      mkfs.xfs
- 查看文件系统信息: dumpe2fs(ext4)      xfs\_info(xfs)

## 标准分区空间不足的扩容----》raid和lvm

- 1 [root@localhost lianxi]# df -Th|tail -5
- 2 /dev/sda6      xfs      53G   1.7G   52G      4% /
- 3 /dev/sda2      xfs      196G   1.4G   194G      1% /mysql\_data
- 4 /dev/sda3      xfs      49G   381M   49G      1% /home
- 5 /dev/sda1      ext4      976M   129M   781M     15% /boot
- 6 tmpfs      tmpfs      181M      0   181M      0% /run/user/0

- /backup ---》专门存放备份数据的

## 软链接和硬链接

- 软链接

- - 1 软连接 相当于\*\**快捷方式*
  - 2 软连接可以对目录进行链接
  - 3 删除链接文件后，软连接失效
  - 4 软连接创建会形成一个新的iNode号

```

1 -bash: vim: 未找到命令
2 [root@localhost lianxi]# vi lihu.txt
3 [root@localhost lianxi]# ll
4 总用量 8
5 -rw-r--r--. 1 root root 14 3月 5 22:13 lihu.txt
6 -rw-r--r--. 1 root root 12 3月 5 07:02 tzk.sh
7 [root@localhost lianxi]# ln -s lihu.txt lh.txt
8 [root@localhost lianxi]# ll
9 总用量 8
10 lrwxrwxrwx. 1 root root 8 3月 5 22:14 lh.txt -> lihu.txt
11 -rw-r--r--. 1 root root 14 3月 5 22:13 lihu.txt
12 -rw-r--r--. 1 root root 12 3月 5 07:02 tzk.sh

```

## • 硬链接

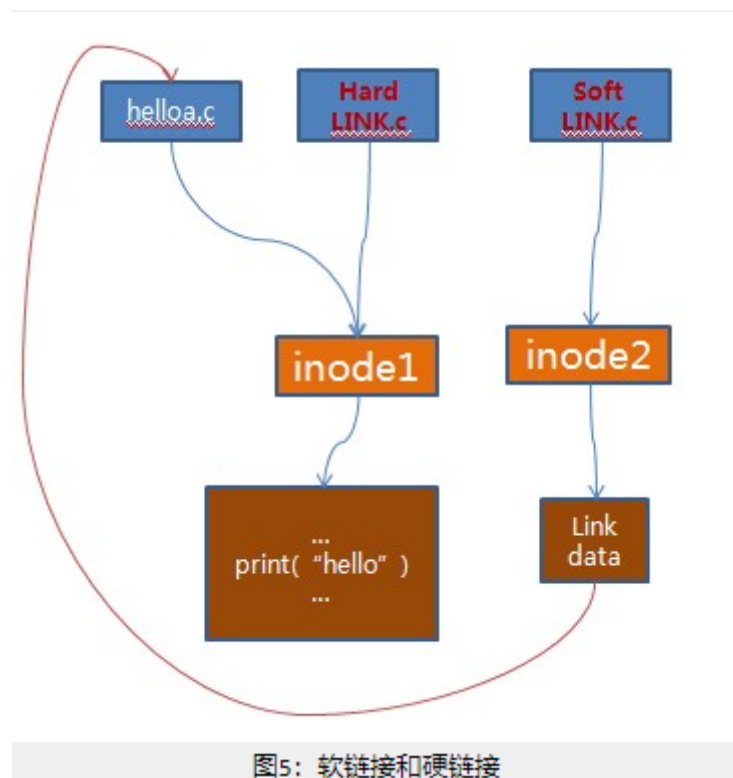
- - 1 硬链接以\*\*文件副本\*\*的形式存在，不占用实际空间。
  - 2 硬链接不可以对目录进行链接。
  - 3 删除链接文件后，硬链接不失效。
  - 4 硬链接的iNode号跟原来的文件inode号一致。
  - 5 \*\*硬链接\*\*起到了节省资源的作用

```

1 [root@localhost lianxi]# vi tanzikun.txt
2 [root@localhost lianxi]# ln tanzikun.txt tzk.txt
3 [root@localhost lianxi]# ll
4 总用量 8
5 -rw-r--r--. 2 root root 14 3月 5 22:18 tanzikun.txt
6 -rw-r--r--. 2 root root 14 3月 5 22:18 tzk.txt
7 [root@localhost lianxi]# ll -i
8 总用量 8
9 100674211 -rw-r--r--. 2 root root 14 3月 5 22:18 tanzikun.txt
10 100674211 -rw-r--r--. 2 root root 14 3月 5 22:18 tzk.txt

```

•



## raid: 廉价冗余阵列

- 类型:

- raid0: **条带卷**

- 没有容错功能，因为是均匀分配地写入磁盘
    - 磁盘利用率100%
    - 读写速度快
    - 原理：假如raid0由三块构成，从理论上讲，三块硬盘的并行操作使同一时间内磁盘读写速度提升了3倍。但由于总线带宽等多种因素的影响，实际的提升速率肯定会低于理论值，但是，大量数据并行传输与串行传输比较，提速效果显著显然毋庸置疑。

- raid1: **镜像卷mirror**

- 有容错功能，每块磁盘都相同地写入全部数据
    - 2块磁盘组成
    - 磁盘利用率50%

- raid5: **至少3块磁盘**

- 有容错功能，有一块校验的磁盘，推出（用算法）受损的那一块磁盘
    - 只能坏一块磁盘
    - 磁盘利用率 $(n-1)/n$

- raid6:

- 有2块校验磁盘，可以最多坏2块
    - 磁盘利用率 $(n-x)/n$

- raid10:

- 条带加镜像

-

# 8种RAID存储模式 性能强大

支持RAID 0/1/3/5/10,clone,large,normal 8种存储模式存储巨量游戏、照片时调用更方便，满足各种存储需求

| Raid模式     | 最少磁盘数                                                                                 | 容量                                                                                  | 安全性                                                                                 | 速度                                                                                    |
|------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Raid 5(推荐) | 3    |    |   |    |
| Raid 0     | 2    |    |    |    |
| Raid 1     | 仅2   |    |   |    |
| Raid 3     | 3    |    |   |    |
| Raid 10    | 4    |    |   |    |
| Clone      | 2    |    |   |    |
| Large      | 2    |    |    |    |
| Normal     | 1  |  |  |  |

## 硬盘组合模式改变方法

使用本产品前，请先备份硬盘资料，因为使用RAID模式时会初始化虚拟硬盘卷完全删除硬盘之前的数据。当重新更换模式时，可对照表或者示意图来设定硬盘模式

## LVM：逻辑卷管理（Logical Volume Manager）

- NAS服务器：实现共享存储给其他的服务器们。==》有raid功能
- 相对传统的分区方式的优势：
  - 空间利用率高、可以在线扩容、数据备份方便、可随意定义逻辑卷卷标
- lv|vg|pv
  - 对/mysql\_data扩容

```

○ 1 [root@mysql ~]# df -Th
2 文件系统 类型 容量 已用 可用 已用% 挂载点
3 /dev/mapper/centos-root xfs 17G 1.2G 16G 7% /
4 devtmpfs devtmpfs 899M 0 899M 0%
5 /dev
6 tmpfs tmpfs 911M 0 911M 0%
7 /dev/shm
8 tmpfs tmpfs 911M 9.6M 902M 2%
9 /run
10 tmpfs tmpfs 911M 0 911M 0%
11 /sys/fs/cgroup
12 /dev/mapper/alimei-tzk_mysql xfs 99G 33M 99G 1%
13 /mysql_data
14 /dev/sda1 xfs 1014M 142M 873M 14%
15 /boot
16 tmpfs tmpfs 183M 0 183M 0%
17 /run/user/0

```

#### ○ 创建pv

```

○ 1 [root@mysql ~]# pvcreate /dev/sdb
2 WARNING: dos signature detected on /dev/sdb at offset 510. wipe it?
3 [y/n]: y
4 Wiping dos signature on /dev/sdb.
5 Physical volume "/dev/sdb" successfully created.

```

#### ○ 将创建的/dev/sdb的pv扩充到alimei (vg)

```

○ 1 [root@mysql ~]# vgscan
2 Reading volume groups from cache.
3 Found volume group "centos" using metadata type lvm2
4 Found volume group "alimei" using metadata type lvm2
5 [root@mysql ~]# vgextend alimei /dev/sdb
6 Volume group "alimei" successfully extended
7

```

#### ○ 最后将扩充的vg在映射到/mysql\_data对应的lv (/dev/alimei/tzk\_mysql) 上, 再在对/mysql\_data扩充

```

○ 1 [root@mysql ~]# lvextend -L +100G /dev/alimei/tzk_mysql
2 Size of logical volume alimei/tzk_mysql changed from 99.00 GiB
3 (25344 extents) to 199.00 GiB (50944 extents).
4 Logical volume alimei/tzk_mysql successfully resized.
5 [root@mysql ~]# xfs_growfs /mysql_data/
6 meta-data=/dev/mapper/alimei-tzk_mysql isize=512 agcount=4,
7 agsize=6488064 blks
8 data = sectsz=512 attr=2, projid32bit=1
9 crc=1 finobt=0 spinodes=0
10 data = bsize=4096 blocks=25952256,
11 imaxpct=25
12 sunit=0 swidth=0 blks
13 naming =version 2 bsize=4096 ascii-ci=0 ftype=1
14 log =internal bsize=4096 blocks=12672,
15 version=2

```



```

12 = sectsz=512 sunit=0 blks, lazy-
count=1
13 realtime =none extsz=4096 blocks=0, rtextents=0
14 data blocks changed from 25952256 to 52166656
15 [root@mysql ~]# df -Th
16 文件系统 类型 容量 已用 可用 已用% 挂载点
17 /dev/mapper/centos-root xfs 17G 1.2G 16G 7% /
18 devtmpfs devtmpfs 899M 0 899M 0%
/dev
19 tmpfs tmpfs 911M 0 911M 0%
/dev/shm
20 tmpfs tmpfs 911M 9.6M 902M 2%
/run
21 tmpfs tmpfs 911M 0 911M 0%
/sys/fs/cgroup
22 /dev/mapper/alimeimei-tzk_mysql xfs 199G 33M 199G 1%
/mysql_data
23 /dev/sda1 xfs 1014M 142M 873M 14%
/boot
24 tmpfs tmpfs 183M 0 183M 0%
/run/user/0

```

- 修改/etc/fstab挂载信息，实现开机自动挂载

```

1 | /dev/alimeimei/tzk_mysql /mysql_data xfs defaults 0 0

```

- 卸载（一般不使用）

- - 1 卸载不要的lv
  - 2
  - 3 千万不要在没有关闭LVM的情况下删除分区
  - 4 正确关闭LVM的步骤
  - 5 umount卸载lvm文件系统
  - 6 lvremove删除逻辑卷
  - 7 vgremove删除卷组
  - 8 pvremove删除物理卷
  - 9 清除/etc/fstab中LVM的挂载记录