20202281rf.

服务器软硬磁盘阵列

一、RAID概述

RAID(Redundant Array of Inexpensive Disks)称为廉价磁盘冗余阵列。RAID的基本思想是把多个便宜的小磁盘组合到一起，组合为一个大磁盘组，使性能达到或超过一个容量巨大、价格昂贵、读写速度快的磁盘。

目前RAID技术主要分为两种：基于硬件（服务器RAID控制器）的技术和基于软件的RAID技术。在Linux系统中通过自带的软件就能模拟实现RAID功能，这样可省去购买昂贵的硬件RAID控制器的费用，便可极大地增强磁盘的IO性能和可靠性。由于是用软件去模拟实现的RAID功能，所以它的配置灵活、管理方便。同时使用软件RAID，还可以实现将几个物理磁盘合并成一个更大的虚拟设备，从而达到性能改进和数据冗余的目的。当然基于硬件的RAID解决方案比基于软件RAID 技术在性能上会胜一筹，具体表现在检测和修复多位错误的能力、错误磁盘自动检测和阵列重建等方面。在本章将详细讲述如何在Linux服务器上创建及维护RAID。

二、RAID级别介绍-----面试必考

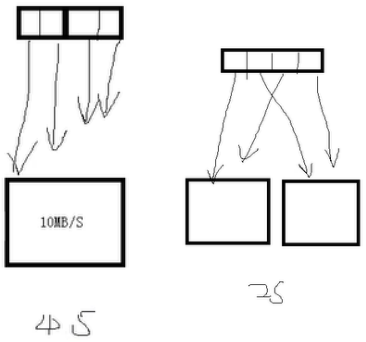
随着RAID技术经过不断的发展，现已有RAID0到RAID6七种基本的RAID级别，同时还有RAID0+RAID1的组合形式，称为RAID10，还有RAID0+RAID5的组合形式，称为RAID50。当然级别并不能代表技术的高低，其中RAID2-RAID4基本上不再使用了。目前这些常用的RAID级别Linux内核都能够支持，本节以Linux2.6以上的内核为例，在Linux2.6内核中的软RAID可支持以下级别：RAID0、RAID1、RAID4、RAID5以及RAID6等。Linux2.6的内核除支持以上几种RAID级别外，还可支持LINEAR（线性模式）的软RAID，线性模式是将两个或更多的磁盘组合到一个物理设备中，磁盘不必具有相同的大小，在写入RAID设备时会首先填满磁盘A，然后是磁盘B，以此类推。

级别	特点	缺点	磁盘数	可用空间	故障磁盘数

REID0	高读写速度	数据的可靠性低	1（通常1+）	硬盘空间总和	0
REID1	可靠性高	低读写速度	2N	总容量的1/2	1
REID4	高读写/可靠	校验盘故障率高	3+	（N-1）*S	1 校验盘固
REID5	高读写/可靠		3+	（N-1）*S	1 校验盘轮
REID6	高读写/可靠	硬盘多	4+	（N-2）*S	2 增加校验
REID1+0	高读写/可靠	硬盘多	（偶数 N>=4）	1/2	2
REID5+0	高读写/可靠	硬盘多	（3N，N>=6）	[N-(N/3)]*S	N/3

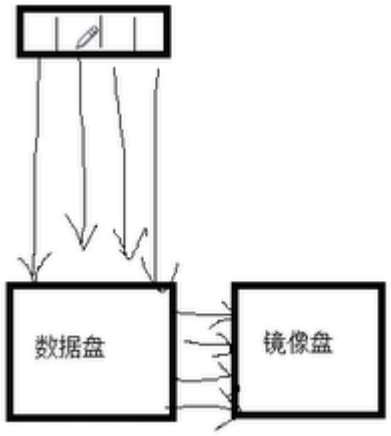
1.RAID 0

也称为条带模式（striped），即把连续的数据分散到多个磁盘上存取。当系统有数据请求就可以被多个磁盘并行的执行，每个磁盘执行属于它自己的那部分数据请求。这种数据上的并行操作可以充分利用磁盘总线的带宽，显著提高磁盘整体存取性能。因为读取和写入是在设备上并行完成的，读取和写入性能将会增加，这通常是运行RAID0的主要原因。但RAID0没有数据冗余，如果其中一个硬盘出现故障，那么将无法恢复任何数据。



2.RAID1

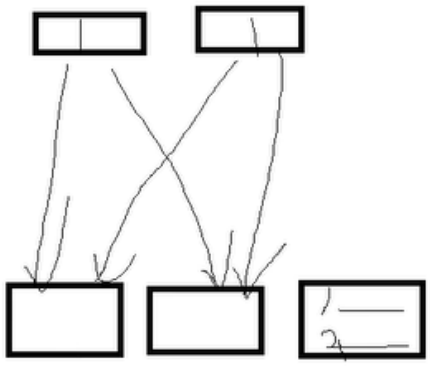
RAID1又称为镜像（Mirroring），一个具有全冗余的模式。RAID1可以用于两个或2xN个磁盘，并使用0块或更多的备用磁盘，每次写数据时会同时写入镜像盘。这种阵列可靠性很高，但其有效容量会减小到总容量的一半，同时这些磁盘的大小应该相等，否则总容量只具有最小磁盘的大小。



3.RAID4（已被淘汰）

创建RAID4需要三块或更多的磁盘，它在一个磁盘上保存校验信息，并以RAID0方式将数据写入其它磁盘。因为一块磁盘是为校验信息保留的，所以阵列的空间大小是（N-1）*S是阵列中最小磁盘的大小。就像在RAID1那样，磁盘的大小应该相等。

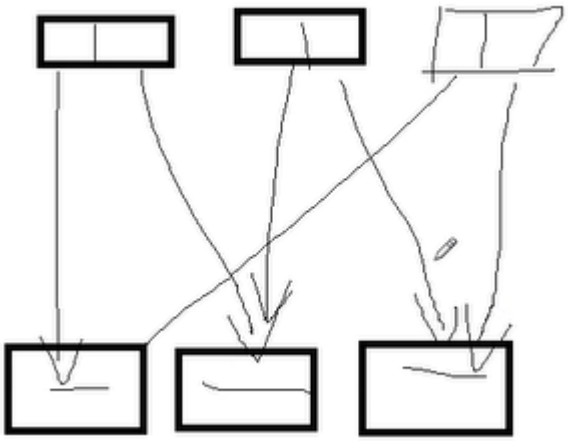
如果一个磁盘出现故障，那么可以使用校验信息及另一个磁盘来重建数据。如果两个磁盘出现故障，那么所有数据都将丢失。不经常使用这个级别的原因是校验信息存储在一个磁盘上。每次写入其它磁盘时，都必须更新这些信息。因此，在大量写入数据时很容易造成校验磁盘的瓶颈，所以目前这个级别的RAID很少使用了。



4.RAID5

在希望结合大量物理磁盘并且仍然保留一些冗余时，RAID5可能是最有用的RAID模式。RAID5可以用在三块或更多的磁盘上，并使用0块或更多的备用磁盘。就像RAID4一样，RAID5设备的大小是（N-1）*S。

RAID5与RAID4之间最大的区别就是校验信息均匀分布在各个驱动器上，这样就避免了 RAID4中出现的瓶颈问题。如果其中一块磁盘出现故障，那么由于有校验信息，所以所有数据仍然可以保持不变。如果可以使用备用磁盘，那么在设备出现故障之后，将立即开始往备用磁盘上同步数据。如果两块磁盘同时出现故障，那么所有数据都会丢失。

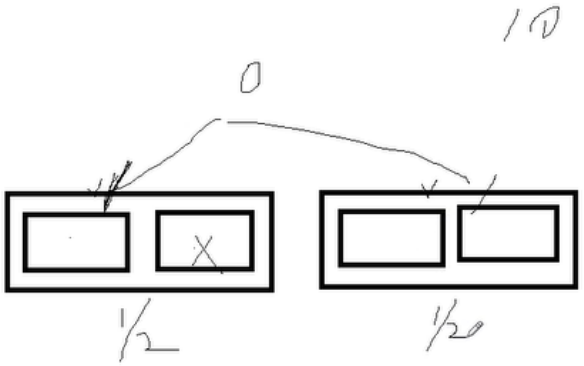


5.RAID6

RAID6是在RAID5基础上扩展而来的。与RAID5一样，数据和校验码都是被分成数据块然后分别存储到磁盘阵列的各个硬盘上。只是RAID6中多增加一块校验磁盘用于备份分布在各个磁盘上的校验码，这样RAID6磁盘阵列就允许两个磁盘同时出现故障，所以RAID6的磁盘阵列最少需要四块硬盘。

6.RAID1+0

N（偶数，N>=4）块盘两两镜像后，在组合成一个RAID0。容量为N/2，N/2块盘同时写入，写速度一般，N块盘同时读取，读速度较快。性能高，可靠性高。



7.RAID5+0

N（N*3 N>=6）块三三镜像后，再组合成一个RAID0，容量为[N-(N/3)]*S,N-(N/3)块磁盘同时写入，N/3块同时读取，读速度较快。性能高，可靠性高。

三、阵列卡介绍（RAID控制器）

阵列卡就是用来实现RAID功能的板卡，通常是由I/O处理器、硬盘控制器、硬盘连接器和缓存等一系列零组件构成。

不同的RAID卡支持的RAID功能不同，例如支持RAID0、RAID1、RAID1、RAID5、RAID6、RAID10等，RAID卡的接口类型：IDE接口、SCSI接口、SATA接口和SAS接口。

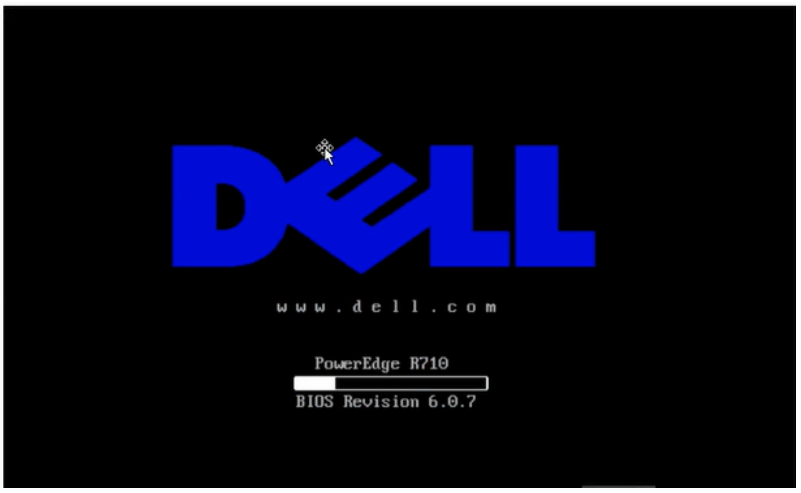
缓存（Cache）是RAID卡与外部总线交换数据的场所，RAID卡先将数据传送到缓存，再由缓存和外边数据总线交换数据。它是RAID卡电路板上的一块存储芯片，与硬盘盘片相比，具有极快的存取速度。

缓存的大小与速度是直接关系到RAID卡的实际传输速度的重要因素，大缓存能够大幅度地提高数据命中率从而提高RAID卡整体性能。不同的RAID卡出厂时配置的内存容量不同，一般为几兆到数百兆容量不等。

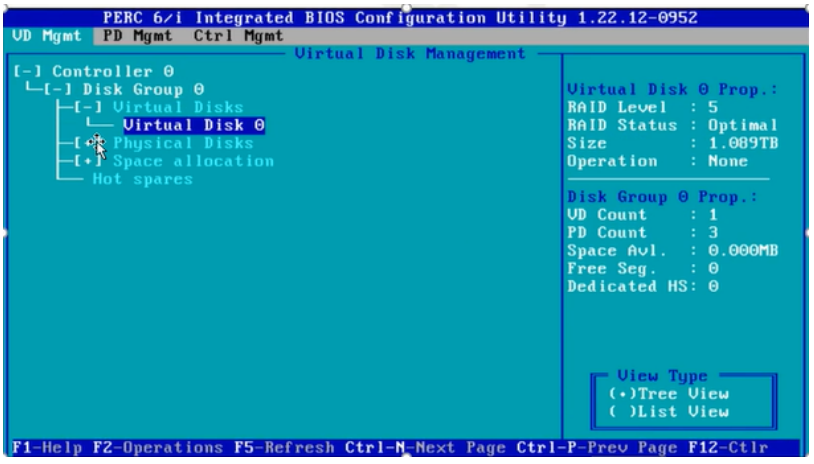
四、硬RAID创建步骤

1.DELL R710服务器案例：

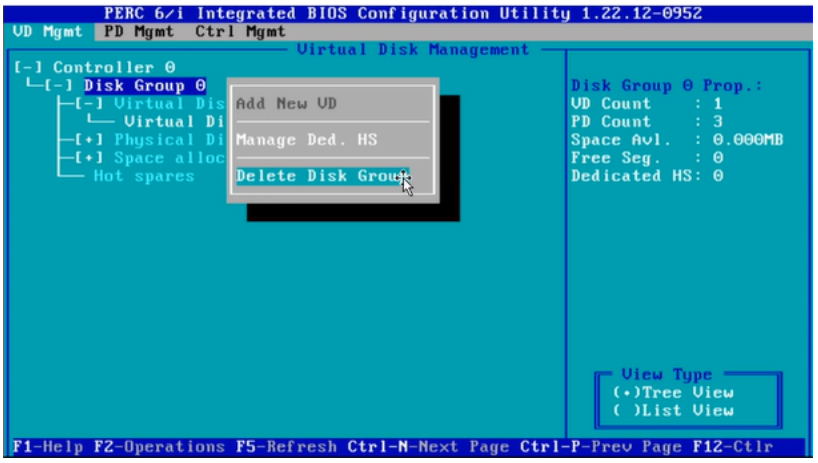
（1）开机等待出现以下界面后看屏幕提示，准备按组合键（DELL：Ctrl+R　Ctrl+H）进入RAID配置界面



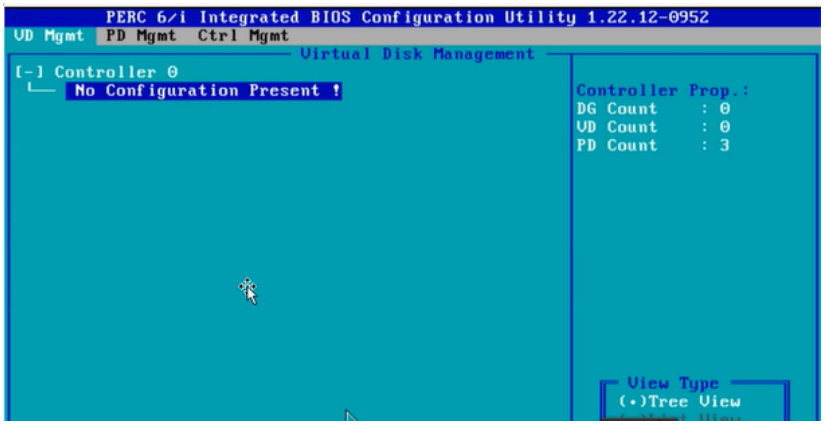
（2）当dell界面过去后屏幕会有相关组合键提示，同时按下Ctrl+r进入RAID配置界面，根据经验建议多按几次，进入到如下的配置界面，在图的右侧可以看到默认为RAID5，磁盘大小等信息。



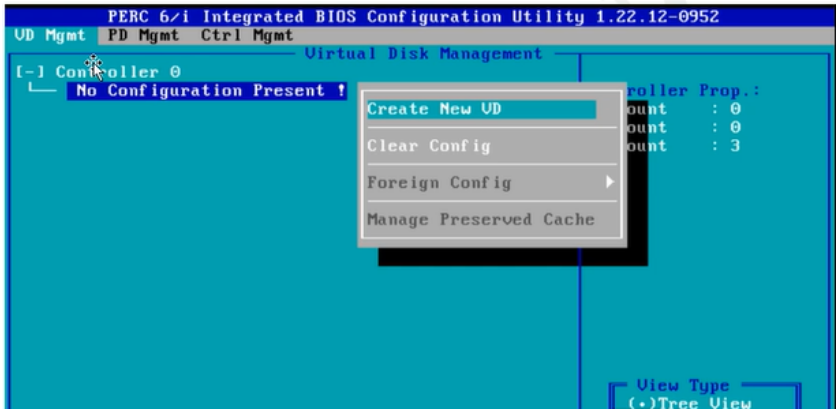
（3）使用上下键移动到第二行的位置按下F2，出现以下窗口，选择第二项（Delete Disk Group）然后回车，删除raid5



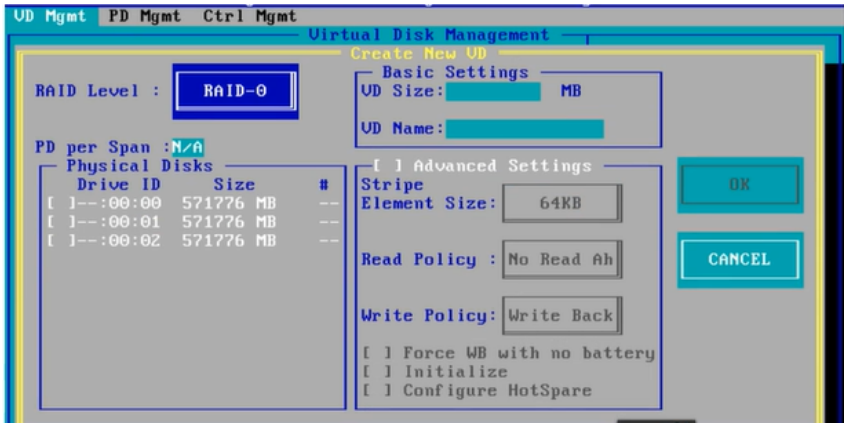
(4)删除RAID5后的状态



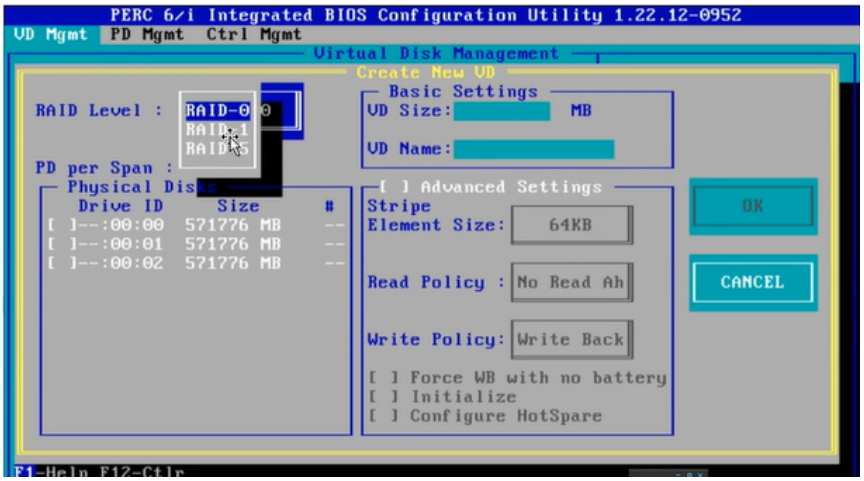
（5）然后按F2选择第一项（Create New VD）创建新的RAID



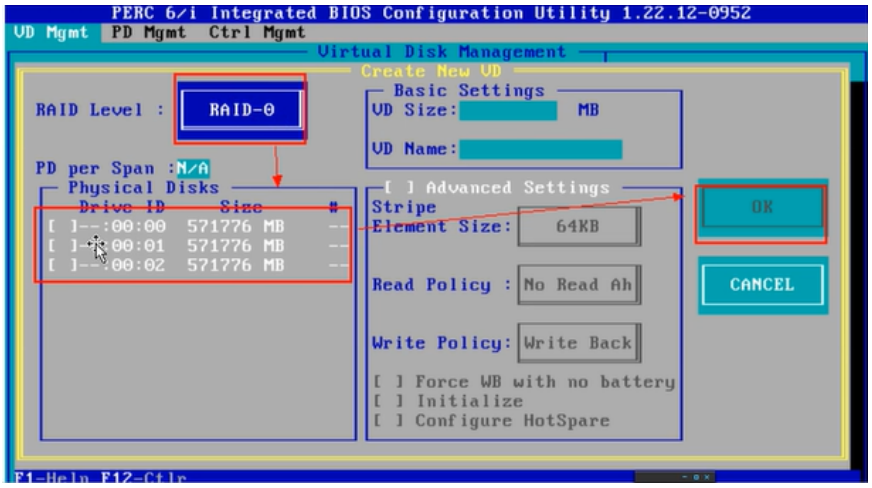
(6)进入到下面的界面，蓝色体的为选中状态，此时光标移动到RAID level那一栏，继续回车



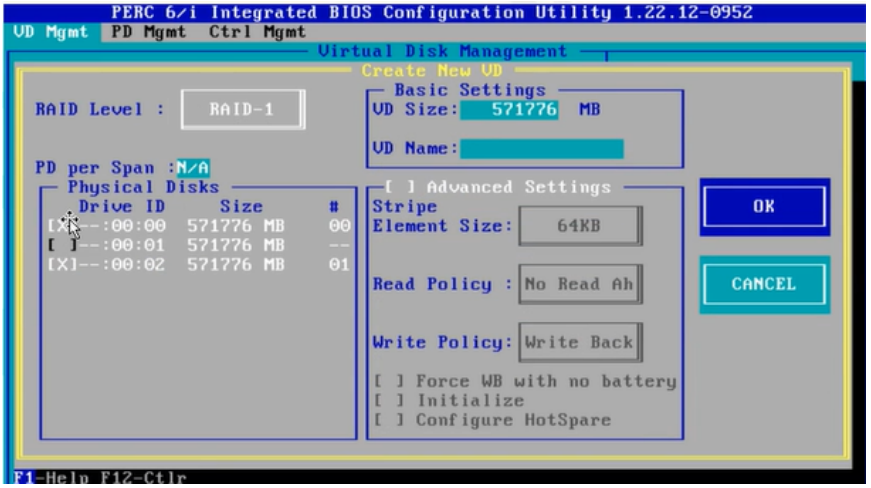
（7）此时出现三个选项，分别是RAID-0、RAID-1、RAID-5 说明只能做raid0和raid1和raid5，raid10需要至少4块磁盘，所以这里没有raid10的选项。



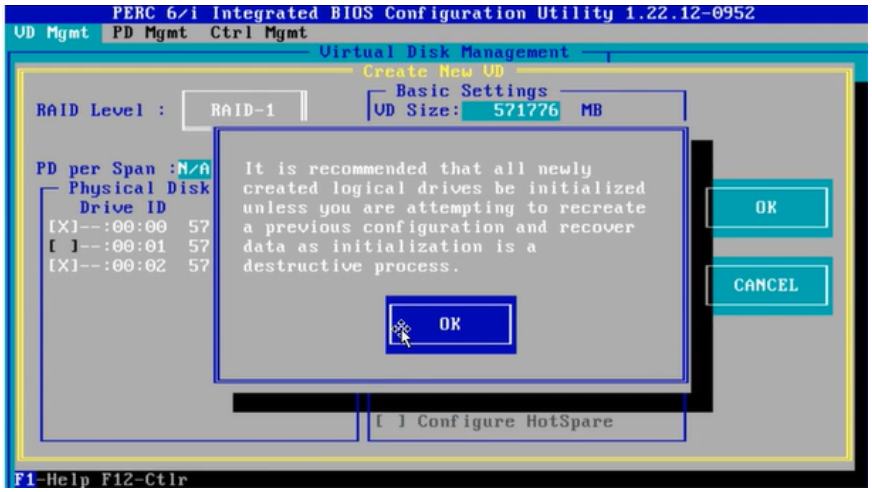
（8）选择需要做的选项比如RAID-0，然后在三块硬盘中选择需要做的硬盘，tab键移动到该位置选择用空格键就可以，最后选择ok



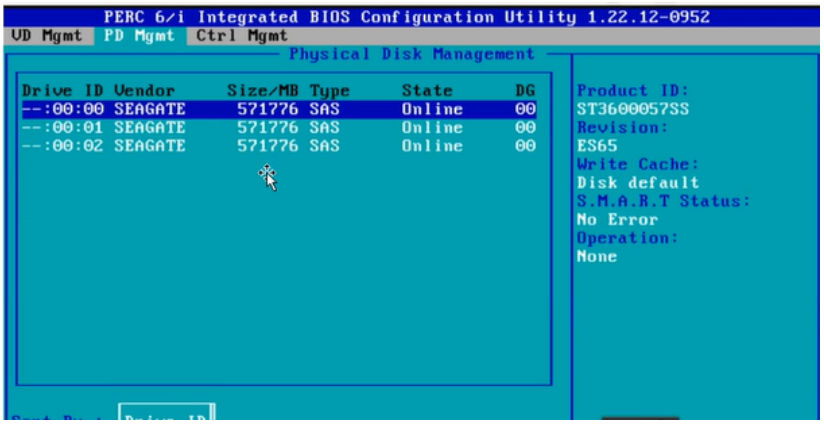
RAID-1只能选择两块硬盘，如下图（RAID-0和RAID-5都可以选择3块）

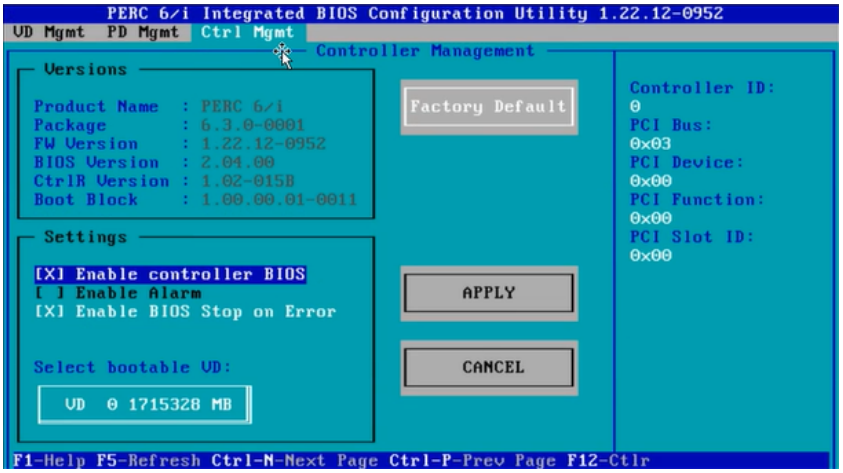


然后再选择OK

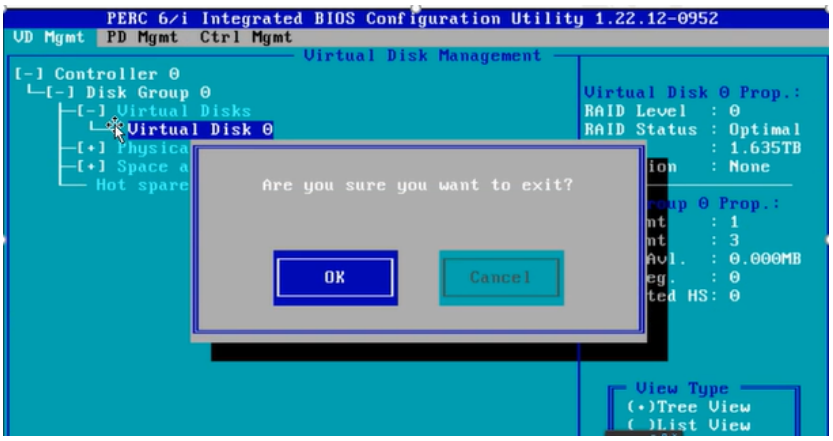


然后按Ctrl+n 可以翻屏查看其他信息





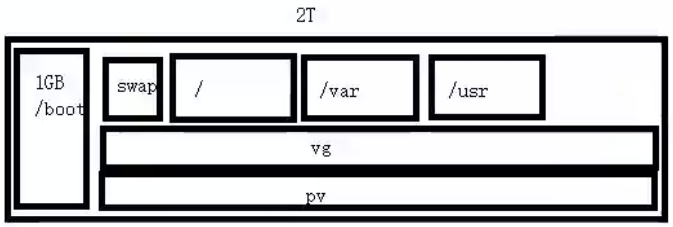
最后按ESC准备保存退出，弹出下图窗口选择OK



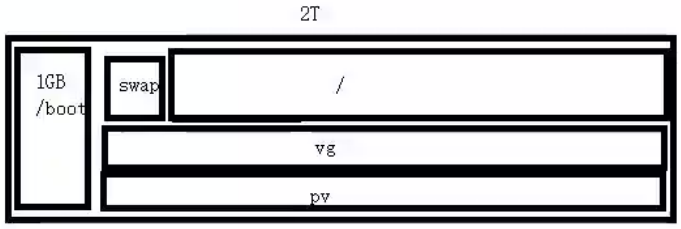
出现以下界面后，根据提示操作就可以了

注意：当拿到一个服务器后，首先做硬RAID，再装系统，然后分区

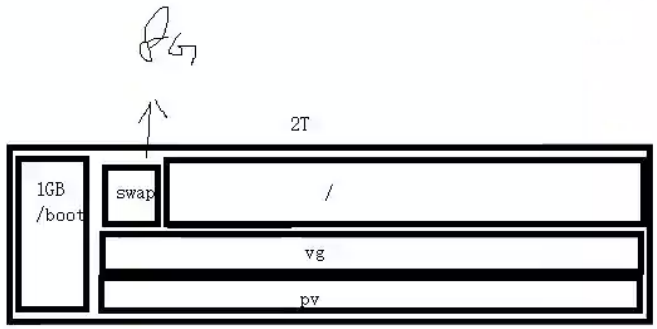
分区：



更简单点，如果没考虑好分区，就都分给根吧



swap一般给8G就可以



2.实际生产环境中RAID的使用

在实际生产环境中，系统硬盘与数据库和应用是分开的，这样有利于系统的维护和对数据应用的使用，磁盘阵列可以在安装系统之前或之后产生（硬RAID之前产生；软RAID之后产生，企业一般是硬RAID），系统会视之为一个(大型)硬盘，而它具有容错及冗余的功

能。磁盘阵列不单只可以加入一个现成的系统，它更可以支持容量扩展，方法也很简单，只需要加入一个新的硬盘并执行一些简单的指令，系统便可以实时利用这新加的容量。

3.RAID的种类及应用

IDE 和SCSI是计算机的两种不同的接口，前者普遍用于PC机，后者一般用于服务器。J基于这两种接口，RAID分为两种类型：基于IDE接口的RAID应用，称为IDE RAID；二基于SCSI接口RAID应用则相应称为SCSI RAID

4.基于不同的架构，RAID的分类

- **软件RAID（软件 RAID）**
- **硬件RAID （硬件 RAID）主板自带的RAID功能**
- **外置RAID（External RAID）RAID卡**

软件RAID很多情况下已经包含在系统之中，并成为其中一个功能，如Windows、Netware及Linux。软件RAID中所有操作皆由中央处理器负责，所以系统资源的利用率会很高，从而使系统性能降低。软件RAID是不需要另外添加任何硬件设备，因为它是靠你的系统，主要是中央处理器的功能提供所有现成的资源。

硬件RAID通常是一张PCI卡，你会看到在这卡上会有处理器及内存。因为这卡上的处理器已经可以提供一切RAID所需要的资源，所以不会占用系统资源，从而令系统的表现可以大大提升。硬件RAID可以连接内置硬盘、热插拔背板或外置存储设备。无论连接何种硬盘，控制权都是在RAID卡上，亦即是由系统所操控。在系统里，硬件RAID PCI卡通常都需要安装驱动程序，否则系统会拒绝支持。

外置式RAID也是属于硬件RAID的一种，区别在于RAID卡不会安装在系统里，而是安装在外置的存储设备内。而这个外置的储存设备则会连接到系统的SCSI卡上。系统没有任何的RAID功能，因为它只有一张SCSI卡；所有的RAID功能将会移到这个外置存储里。好处是外置的存储往往可以连接更多的硬盘，不会受到系统机箱的大小所影响。而一些高级的技术，如双机容错，是需要多个服务器外连到一个外置储存上，以提供容错能力。

五、创建软RAID环境描述

操作系统	IP地址	主机名	软件包列表
CentOS Linux release 7.5.1804 (Core)	192.168.200.100	localhost	mdadm

ps:查看Centos版本

```
[root@localhost ~]# cat /proc/version
[root@localhost ~]# uname -a
 [root@localhost ~]# cat /etc/redhat-release
 [root@localhost ~]# getconf LONG_BIT    #位数
```

LINUX服务器中添加1块100G空闲的硬盘，划分4个20G的分区用来练习创建一个RAID及其后维护操作。

1.创建软RAID

在Linux服务器中可通过mdadm工具来创建和维护软RAID的madam在创建和管理软RAID时非常方便，而且灵活。mdadm常用的参数有如下：

-C　　--create: 创建一个软RAID，后面需要标识RAID设备的名称。例如，/dev/md0./dev/md1

-A　　--assemble:加载一个已存在的RAID，后面跟RAID以及设备的名称。

-S　　--detail: 输出指定的RAID设备的详细信息

-l　　--level: 指定RAID配置级别，例如，设置“--level=5”则表示创建阵列的级别是RAID5

-n　　--raid-devices:指定RAID中活动磁盘的数目

-r　　--remove:删除RAID中的某个磁盘

-a　　--add:向RAID中添加磁盘

-x　　--spare-devices：指定备用磁盘数量

-s　　--scan:扫描配置文件/proc/mdstat 来搜索软RAID的配置信息，该参数不能单独使用，需要配合其它参数才能使用。

2.配置过程

```
[root@localhost ~]# rpm -q mdadm
```

创建4个20G的分区：

创建RAID5

创建完分区后，其中/dev/adb4作为备用磁盘，其余为活动磁盘，备用磁盘的作用是一旦某一块磁盘损坏可以立即使用备用磁盘替换。

```
[root@localhost ~]# mdadm --create /dev/md5 --level=5 --raid-devices=3 --spare-devices=1 \
> /dev/sdb[1-4]
```

或者[root@localhost ~]# mdadm -C /dev/md5 -l 5 -n 3 -x 1 /dev/sdb[1-4]

```
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md5 started.
```

其中“--spare-devices=1”表示当前阵列中备用设备只有一块，即作为备用设备的“/dev/sdb4”,若有多块备用设备，则将“--spare-devices”的值设置为相应的数目。

成功创建完成RAID设备后，通过如下命令可以看到RAID的详细信息：

```
[root@localhost ~]# mdadm --detail /dev/md5
或者[root@localhost ~]# mdadm -D /dev/md5
/dev/md5:
           Version : 1.2
     Creation Time : Thu Feb 27 21:38:22 2020
        Raid Level : raid5
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 3
     Total Devices : 4
       Persistence : Superblock is persistent

       Update Time : Thu Feb 27 21:40:09 2020
             State : clean
    Active Devices : 3
   Working Devices : 4
    Failed Devices : 0
     Spare Devices : 1

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : localhost:5  (local to host localhost)
              UUID : 267e73ac:b02b2fc4:951feefd:07f6575c
            Events : 18

    Number   Major   Minor   RaidDevice State
       0       8       17        0      active sync   /dev/sdb1
       1       8       18        1      active sync   /dev/sdb2
       4       8       19        2      active sync   /dev/sdb3
```

```
       3       8       20        -       spare   /dev/sdb4
```

3.创建RAID的配置文件

作用：防止重启后找不到RAID

RAID的配置文件名为“mdadm.conf”,默认是不存在的，所以需要手工创建，该配置文件存在的主要作用是系统启动的时候能够自动加载软RAID，同时也方便日后管理。“mdadm.conf”文件内容包括：由DEVICE选项指定用于软RAID的所有设备，和ARRAY选项所指定阵列的设备名、RAID级别、阵列中活动设备的数目以及设备的UUID号。生成RAID配置文件操作如下：

```
[root@localhost ~]# mdadm -D -s > /etc/mdadm.conf
或者[root@localhost ~]# mdadm --detail -scan > /etc/mdadm.conf
```

但是当前生成"mdadm.conf"文件的内容并不符合所规定的格式，所以也是不生效的，这时需要手工修改改文件内容为如下格式：

```
[root@localhost ~]# vim /etc/mdadm.conf
ARRAY /dev/md5 metadata=1.2 spares=1 name=localhost:5
UUID=267e73ac:b02b2fc4:951feefd:07f6575c auto
=yes
```

如果没有创建RAID的配置文件，那么在每次系统启动后，需要手工加载软RAID才能使用，手工加载软RAID的命令：

```
[root@localhost ~]# mdadm -A /dev/md5 /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4
```

4.创建文件系统

格式化

```
[root@localhost ~]# mkfs.xfs /dev/md5
meta-data=/dev/md5               isize=512    agcount=16, agsize=654720 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=4096   blocks=10475520, imaxpct=25
```

```
         =                       sunit=128    swidth=256 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=4096   blocks=5120, version=2
         =                       sectsz=512   sunit=8 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

挂载：

```
[root@localhost ~]# mkdir /raid
[root@localhost ~]# mount /dev/md5 /raid
[root@localhost ~]# df -Th
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	50G	5.5G	45G	11%	/
devtmpfs	devtmpfs	471M	0	471M	0%	/dev
tmpfs	tmpfs	488M	0	488M	0%	/dev/shm
tmpfs	tmpfs	488M	8.3M	480M	2%	/run
tmpfs	tmpfs	488M	0	488M	0%	/sys/fs/cgroup
/dev/mapper/centos-home	xfs	27G	33M	27G	1%	/home
/dev/sda1	xfs	1014M	157M	858M	16%	/boot
tmpfs	tmpfs	98M	12K	98M	1%	/run/user/42
tmpfs	tmpfs	98M	0	98M	0%	/run/user/0
/dev/md5	xfs	40G	33M	40G	1%	/raid

创建完文件系统后，将该设备挂载上就可正常使用了。如果要创建其它级别的RAID，其步骤和创建RAID5基本都一样，区别在于指定“--level”值得时候，需要将该值设置为相应的级别。

设置开机自动挂载，在/etc/fstab中加入下面一行：

```
UUID="fabf785c-9c3f-49b2-8620-e204f61cc52e"       /raid      xfs      defaults
0 0
```

挂载测试：[root@localhost ~]# umount /dev/md5

[root@localhost ~]# mount /dev/md5

或者：[root@localhost ~]# umount /raid

[root@localhost ~]# mount /raid

5.维护软RAID

```
[root@localhost ~]# umount /raid
```

```
[root@localhost ~]# mount /raid
```

1.模拟故障磁盘

假设其中的“/dev/sdb2”设备出现故障时，更换一个新的磁盘，整个过程的详细说明如下：

在实际中，当软RAID检测到某个磁盘有故障时，会自动标记该磁盘为故障磁盘，并停止对故障磁盘的读写操作，所以这里需要将/dev/sdb2标记为出现故障的磁盘，命令如下：

```
[root@localhost ~]# mdadm /dev/md5 --fail /dev/sdb2      #标记为故障盘
                                    mdadm: set /dev/sdb2 faulty in /dev/md5
```

由于RAID5设置了一个备用设备，所以当有标记为故障磁盘的时候，备用磁盘会自动顶替故障磁盘工作，阵列也能够在短时间内实现重建。通过“/proc/mdstat”文件可查看到当前阵列的状态，如下：

```
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md5 : active raid5 sdb3[4] sdb4[3] sdb2[1](F) sdb1[0]
      41908224 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>

[root@localhost ~]# mdadm -D /dev/md5
/dev/md5:
           Version : 1.2
     Creation Time : Thu Feb 27 21:38:22 2020
        Raid Level : raid5
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 3
     Total Devices : 4
       Persistence : Superblock is persistent

       Update Time : Fri Feb 28 00:20:39 2020
             State : clean
```

```
    Active Devices : 3
   Working Devices : 3
    Failed Devices : 1
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : localhost:5  (local to host localhost)
              UUID : 267e73ac:b02b2fc4:951feefd:07f6575c
            Events : 37

    Number   Major   Minor   RaidDevice State
       0       8       17        0      active sync   /dev/sdb1
       3       8       20        1      active sync   /dev/sdb4
       4       8       19        2      active sync   /dev/sdb3

       1       8       18        -      faulty   /dev/sdb2
```

```
md5 : active raid5 sdb3[4] sdb4[3] sdb2[1](F) sdb1[0]
      41928704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [U_U]
      [==>..................]  recovery = 11.9% (2503664/20964352) finish=10.0min
speed=30455K/sec

unused devices: <none>
```

当一个设备出现故障或被标记故障时，相应设备的方括号后将被标以（F），如“sdb2[1](F)”，其中“[3/2]”的第一位数表示阵列所包含的设备数，第二位数表示活动的设备数，因为目前有一个故障设备，所以第二位数为2；这时的阵列以降级模式运行，虽然该阵列仍然可用，但是不具有数据冗余；而“[U_U]”表示当前阵列可以正常使用的设备是/dev/sdb1和/dev/sdb3，如果是设备“/dev/sdb1”出现故障时，则将变成[_UU]。

重建完数据后，再次查看阵列状态时，就会发现当前RAID设备又恢复了正常，如下：

```
[root@localhost ~]# cat /proc/mdstat
```

```
Personalities : [raid6] [raid5] [raid4]
md5 : active raid5 sdb2[5](S) sdb3[4] sdb4[3] sdb1[0]
      41908224 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

6.移除故障磁盘

```
[root@localhost ~]# mdadm /dev/md5 --remove /dev/sdb2
mdadm: hot removed /dev/sdb2 from /dev/md5
[root@localhost ~]# mdadm -D /dev/md5
/dev/md5:
           Version : 1.2
     Creation Time : Thu Feb 27 21:38:22 2020
        Raid Level : raid5
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 3
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Fri Feb 28 00:31:13 2020
             State : clean
    Active Devices : 3
   Working Devices : 3
    Failed Devices : 0
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : localhost:5  (local to host localhost)
              UUID : 267e73ac:b02b2fc4:951feefd:07f6575c
            Events : 38
```

```
    Number   Major   Minor   RaidDevice State
       0       8       17        0      active sync   /dev/sdb1
       3       8       20        1      active sync   /dev/sdb4
       4       8       19        2      active sync   /dev/sdb3
```

7.添加新硬盘

```
[root@localhost ~]# mdadm /dev/md5 --add /dev/sdb2
mdadm: added /dev/sdb2
[root@localhost ~]# mdadm -D /dev/md5
/dev/md5:
           Version : 1.2
     Creation Time : Thu Feb 27 21:38:22 2020
        Raid Level : raid5
        Array Size : 41908224 (39.97 GiB 42.91 GB)
     Used Dev Size : 20954112 (19.98 GiB 21.46 GB)
      Raid Devices : 3
     Total Devices : 4
       Persistence : Superblock is persistent

       Update Time : Fri Feb 28 00:31:52 2020
             State : clean
    Active Devices : 3
   Working Devices : 4
    Failed Devices : 0
     Spare Devices : 1

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : localhost:5  (local to host localhost)
              UUID : 267e73ac:b02b2fc4:951feefd:07f6575c
            Events : 39
```

```
    Number   Major   Minor   RaidDevice State
       0       8       17        0      active sync   /dev/sdb1
       3       8       20        1      active sync   /dev/sdb4
       4       8       19        2      active sync   /dev/sdb3

       5       8       18        -      spare   /dev/sdb2
```

8.测试

```
[root@localhost ~]# yum info hdparm
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
可安装的软件包
名称    ：hdparm
架构    ：x86_64
版本    ：9.43
发布    ：5.el7
大小    ：83 k
源    ：xxx
简介    ：  A utility for displaying and/or setting hard disk parameters
网址    ：http://sourceforge.net/projects/hdparm/
协议    ：  BSD
描述    ：  Hdparm is a useful system utility for setting (E)IDE hard drive
        ：parameters.  For example, hdparm can be used to tweak hard drive
        ：performance and to spin down hard drives for power conservation.


[root@localhost ~]# rz
[root@localhost ~]# ls hdparm-9.56.tar.gz
hdparm-9.56.tar.gz
[root@localhost ~]# tar -xf hdparm-9.56.tar.gz
[root@localhost ~]# yum -y install hdparm
[root@localhost ~]# hdparm -t /dev/md5
/dev/md5:
 Timing buffered disk reads: 1802 MB in  3.00 seconds = 600.48 MB/sec
[root@localhost ~]# hdparm -t /dev/sda
```

```
/dev/sda:
 Timing buffered disk reads: 1232 MB in  3.01 seconds = 409.64 MB/sec
```

磁盘配额

一、磁盘配额（Quota）概述

在Linux系统中，由于是多人多任务环境，所以会有很多人共同使用一个硬盘空间的情况发生，如果其中少数几个使用者大量的占用硬盘空间的话，那势必会压缩其他用户的使用权力，因此，管理员应该适当的限制硬盘的容量给用户，以妥善分配系统资源。

再如当Linux根据分区的磁盘空间耗尽时，Linux操作系统将无法再建立新的文件（包括程序运行的临时文件），从而出现服务器程序崩溃，系统无法启动等故障，为了避免服务器中出现类似的磁盘空间不足问题，可以启用磁盘配额功能，对用户在指定文件系统（分区）中使用的磁盘空间，文件数量进行限制，防止个别用户恶意或无意占用大量磁盘空间，从而保持系统存储空间的稳定性和持续可用性。

CentOS系统的内核中已经包含了Linux文件系统的磁盘配额功能。系统中xfs文件系统配置和管理磁盘配额的工具由xfsprogs软件包的xfs_quota配额管理程序提供。

```
[root@localhost ~]# rpm -q xfsprogs
xfsprogs-4.5.0-15.el7.x86_64
[root@localhost ~]# rpm -ql xfsprogs | grep xfs_quota
/usr/sbin/xfs_quota
/usr/share/man/man8/xfs_quota.8.gz
```

二、磁盘配额概念

1.磁盘配额的作用范围

xfs_quota设置的的磁盘配额功能，只在指定的的文件系统（分区）内有效，用户使用其他为设置配额的文件系统时，将不受限制。

2.磁盘配额的限制对象

xfs_quota主要针对系统中指定的用户账号、组账号进行限制，没有被设置限额的用户或组将不受影响。对组账号设置配额后，组内所有用户使用的磁盘容量、文件数量的总和不能超过限制。

3.磁盘配额的限制类型

硬盘容量：限制用户能够使用的磁盘数据块（block）大小，也就是限制磁盘空间大小，默认单位为KB。

文件数量：限制用户能够拥有的文件个数。quota通过限制节点的数量来实现对文件数量的限制。

4.磁盘配额的限制方法

软限制：制定一个软性的配额数值（如480MB、180个文件），在固定的宽限期内（默认为7天）允许用户暂时超过这个限制，但系统会给出警告信息。

硬限制：制定一个硬性的配额数值（如500MB、200个文件），是绝对禁止用户超过这个限制值的，当达到硬限制值时，系统也会给出警告并禁止继续写入数据。硬限制的配额应大于相应的软限制的值，否则软限制将失效。

三、磁盘配额管理

1.以支持磁盘配额功能的方式挂载文件系统：

除了内核和xfs_quota软件支持外，实施磁盘配额功能还有一个前提条件，即指定的分区必须已经挂载且支持磁盘配额功能。

让分区支持磁盘配额选项：

usrquota选项：支持对用户的磁盘配额；

grpquota选项：支持对组的磁盘配额；

示例：

在配置磁盘配额过程中，可以使用带“-o userquota grp qupquota”选项的mount命令重新挂载指定的分区以便增加对用户、组配额功能。需要注意的是xfs文件系统只有首次挂载时才启动磁盘配额功能。所以不能使用“-o remount”挂载选项。

```
[root@localhost ~]# mkdir /data
[root@localhost ~]# mount -o usrquota,grpquota /dev/myvg/mylv /data
[root@localhost ~]# mount | grep myvg
/dev/mapper/myvg-mylv on /data type xfs
(rw,relatime,seclabel,attr2,inode64,usrquota,grpquota)
[root@localhost ~]# chmod 777 /data
```

若需要在每次系统开机后自动以支持磁盘配额的功能方式挂载该分区，可以将usrquot和grpquota选项参数写到/etc/fstab配置文件中。

```
[root@localhost ~]# chmod 777 /data
[root@localhost ~]# vim /etc/fstab
[root@localhost ~]# blkid /dev/myvg/mylv
/dev/myvg/mylv: UUID="ac2d2823-b9f6-4550-bf6b-5b8762d20e67" TYPE="xfs"
[root@localhost ~]# vim /etc/fstab
UUID="ac2d2823-b9f6-4550-bf6b-5b8762d20e67"   /data  xfs   defaults     0 0
[root@localhost ~]# mount -a
```

2.编辑用户和组的账号配额设置

配额设置是实现磁盘配额功能中最重要的环节，使用xfs_quota命令设置磁盘容量，文件数的软，硬限制等数值。

1.格式

xfs_quota -x -c ’limit -u bsoft=N bhard=N isoft=N ihard=N 用户’ 挂载点

-x 表示启动专家模式

-c 表示直接调用管理命令，不加-c命令会执行失败并切入到xfs_quota>交互式工作模式

bsoft=N 磁盘容量软限制

bhard=N 磁盘容量硬限制

isoft=N 文件数量软限制

ihard=N 文件数量硬限制

例如：对用户crushlinux用户设置磁盘配额：磁盘容量软限制为80M，硬限制为100M；文件数量软限制为80M个，硬限制为100个；

```
[root@localhost ~]# useradd crushlinux
[root@localhost ~]# echo "123456" | passwd --stdin crushlinux
更改用户 crushlinux 的密码 。
passwd：所有的身份验证令牌已经成功更新。
```

注意：0表示无限制

```
[root@localhost ~]# xfs_quota -x -c ’limit -u bsoft=80M bhard=100M isoft=80 ihard=100 crushlinux’ /data
xfs_quota: invalid user name: crushlinux/data
xfs_quota: invalid user name: crushlinux/data
```

```
xfs_quota: invalid user name: crushlinux/data
xfs_quota: invalid user name: crushlinux/data
xfs_quota: invalid user name: crushlinux/data
```

仅限制磁盘容量

```
[root@localhost ~]# xfs_quota -x -c 'limit -u bsoft=80M bhard=100M crushlinux'
/data
```

仅限制文件容量

```
[root@localhost ~]# xfs_quota -x -c 'limit -u isoft=80 ihard=100 crushlinux'
/data
```

2.查看文件容量限制

```
[root@localhost ~]# xfs_quota -c 'quota -uv crushlinux' /data
Disk quotas for User crushlinux (1001)
Filesystem                         Blocks      Quota      Limit       Warn/Time
Mounted on
/dev/mapper/myvg-mylv
                                        0      81920     102400    00
[--------]        /data
```

3.-i 选项查看文件数量限制

```
[root@localhost ~]# xfs_quota -c 'quota -i -uv crushlinux' /data
Disk quotas for User crushlinux (1001)
Filesystem              Files      Quota      Limit     Warn/Time      Mounted
on
/dev/mapper/myvg-mylv
                            0         80        100       00 [-------
-]        /data
```

4.-g 选项指定组账户（限制组内所有用户共同使用）

```
[root@localhost ~]# xfs_quota -x -c 'limit -g bsoft=80M bhard=100M isoft=80
ihard=100 crushlinux' /data
[root@localhost ~]# xfs_quota -c 'quota -gv crushlinux' /data
Disk quotas for Group crushlinux (1001)
```

```
Filesystem              Blocks      Quota      Limit          Warn/Time
Mounted on
/dev/mapper/myvg-mylv
                                      0       81920      102400       00 [-------
-]       /data
```

5.以crushlinux用户登录系统进行测试

验证磁盘配额功能

dd命令是一个设备转换和复制命令

- if　　指定输入设备（或文件）
- of　　指定输出设备（或文件）
- bs　　指定读取数据块大小
- count 指定读取数据块的数量

```
[crushlinux@localhost ~]$ dd if=/dev/zero of=/data/quotafile（文件名随便）
bs=1M count=81
记录了81+0 的读入
记录了81+0 的写出
84934656字节(85 MB)已复制，0.21712 秒，391 MB/秒
[crushlinux@localhost ~]$ dd if=/dev/zero of=/data/quotafile bs=1M count=100
记录了100+0 的读入
记录了100+0 的写出
104857600字节(105 MB)已复制，0.237532 秒，441 MB/秒
[crushlinux@localhost ~]$ dd if=/dev/zero of=/data/quotafile bs=1M count=101
dd: 写入"/data/quotafile" 出错: 超出磁盘限额
记录了101+0 的读入
记录了100+0 的写出
104857600字节(105 MB)已复制，0.107489 秒，976 MB/秒

记录了80+0 的读入
记录了80+0 的写出
83886080字节(84 MB)已复制，0.0954633 秒，879 MB/秒
[crushlinux@localhost ~]$ cd /data/
[crushlinux@localhost data]$ touch {1..80}.txt
[crushlinux@localhost data]$ ls
10.txt  17.txt  24.txt  31.txt  39.txt  46.txt  53.txt  60.txt  68.txt  75.txt
9.txt
```

```
11.txt  18.txt  25.txt  32.txt  3.txt   47.txt  54.txt  61.txt  69.txt  76.txt
123     19.txt  26.txt  33.txt  40.txt  48.txt  55.txt  62.txt  6.txt   77.txt
12.txt  1.txt   27.txt  34.txt  41.txt  49.txt  56.txt  63.txt  70.txt  78.txt
13.txt  20.txt  28.txt  35.txt  42.txt  4.txt   57.txt  64.txt  71.txt  79.txt
14.txt  21.txt  29.txt  36.txt  43.txt  50.txt  58.txt  65.txt  72.txt  7.txt
15.txt  22.txt  2.txt   37.txt  44.txt  51.txt  59.txt  66.txt  73.txt  80.txt
16.txt  23.txt  30.txt  38.txt  45.txt  52.txt  5.txt   67.txt  74.txt  8.txt
[crushlinux@localhost data]$ touch 81.txt
[crushlinux@localhost data]$ touch {82..100}.txt
touch: 无法创建"100.txt": 超出磁盘限额
[crushlinux@localhost data]$ ls
10.txt  19.txt  28.txt  37.txt  46.txt  55.txt  64.txt  73.txt  82.txt  91.txt
11.txt  1.txt   29.txt  38.txt  47.txt  56.txt  65.txt  74.txt  83.txt  92.txt
123     20.txt  2.txt   39.txt  48.txt  57.txt  66.txt  75.txt  84.txt  93.txt
12.txt  21.txt  30.txt  3.txt   49.txt  58.txt  67.txt  76.txt  85.txt  94.txt
13.txt  22.txt  31.txt  40.txt  4.txt   59.txt  68.txt  77.txt  86.txt  95.txt
14.txt  23.txt  32.txt  41.txt  50.txt  5.txt   69.txt  78.txt  87.txt  96.txt
15.txt  24.txt  33.txt  42.txt  51.txt  60.txt  6.txt   79.txt  88.txt  97.txt
16.txt  25.txt  34.txt  43.txt  52.txt  61.txt  70.txt  7.txt   89.txt  98.txt
17.txt  26.txt  35.txt  44.txt  53.txt  62.txt  71.txt  80.txt  8.txt   99.txt
18.txt  27.txt  36.txt  45.txt  54.txt  63.txt  72.txt  81.txt  90.txt  9.txt
```

6.若想同时查看磁盘容量和文件输出的报告可结合-i与-b选项

```
[root@localhost ~]# xfs_quota -x -c 'report -abi'
User quota on /data (/dev/mapper/myvg-mylv)
                                Blocks
Inodes
User ID              Used       Soft       Hard      Warn/Grace           Used
Soft       Hard      Warn/ Grace
---------- -------------------------------------------------- ----------
---------- ---------- --------------------
root                          4          0          0          00 [--------]
3          0          0                  00 [--------]
crushlinux        81920      81920     102400      00 [--------]            100
80        100            00  [6 days]
```

```
Group quota on /data (/dev/mapper/myvg-mylv)
                                Blocks
Inodes
Group ID          Used       Soft       Hard     Warn/Grace           Used
Soft       Hard            Warn/ Grace
     ---------- ---------------------------------------------- --------------
---------------------------------- ---------------
 root                        4          0              0          00 [--------]
3          0          0                  00 [--------]
crushlinux         81920      81920      102400      00 [--------]          100
80         100             00  [6 days]
```

四、基于ext4文件系统的磁盘配额

课外扩展：https://www.cnblogs.com/sddai/p/11111895.html

1、以支持磁盘配额功能的方式挂载文件系统：

```
[root@localhost ~]# fdisk /dev/sdb
分区 1 已设置为 Linux 类型，大小设为 5 GiB
命令(输入 m 获取帮助)：w
[root@localhost ~]# partprobe /dev/sdb
[root@localhost ~]# mkfs.ext4 /dev/sdb1
[root@localhost ~]# mkdir /data
[root@localhost ~]# mount /dev/sdb1 /data
[root@localhost ~]# mount -o remount,usrquota,grpquota /data
```

remount 只有ext4可以用，xfs只能重新挂载用

```
[root@localhost ~]# mount | grep /data
[root@localhost ~]# chmod 777 /data
```

注意：用户身份的切换，root用户权限没及时切换造成的报错

2、检测磁盘配额并生成配额文件：

除了内核和quota软件支持外，实施磁盘配额功能还有一个前提条件，即指定的分区必须已经挂载且支持磁盘配额功能。

让分区支持磁盘配额选项：

- usrquota选项：支持对用户的磁盘配额；
- grpquota选项：支持对组的磁盘配额；

quotacheck命令：

-a：表示扫描所有分区，查看是否有较早版本的配额文件；

-c：常与u，g选项配合使用，创建用户、组的配额文件；

[root@localhost ~]# quotacheck -ugcv /data

下面的现象是正常的

quotacheck: Your kernel probably supports journaled quota but you are not using it. Consider switching

to journaled quota to avoid running quotacheck after an unclean shutdown.quotacheck: Scanning /dev/sdb1 [/data] done

quotacheck: Cannot stat old user quota file /data/aquota.user: 没有那个文件或目录. Usage will not be su

btracted.quotacheck: Cannot stat old group quota file /data/aquota.group: 没有那个文件或目录. Usage will not be

subtracted.quotacheck: Cannot stat old user quota file /data/aquota.user: 没有那个文件或目录. Usage will not be su

btracted.quotacheck: Cannot stat old group quota file /data/aquota.group: 没有那个文件或目录. Usage will not be

subtracted.quotacheck: Checked 3 directories and 0 files

quotacheck: Old file not found.

quotacheck: Old file not found.

[root@localhost ~]# ls /data

aquota.group aquota.user lost+found

3、编辑用户和组的账号的配额设置：

edquota命令：

结合-u和-g选项可用于编辑用户或组的配额设置；

在edquota的编辑界面中，各字段分别代表的含义如下：

Filesystem：表示本行配置记录对应的文件系统，即配额的作用范围；

blocks：表示用户已经使用的磁盘容量，单位为KB，该值由edquota自动计算；

soft：第3列的soft对应为磁盘容量的软限制数值；

hard：第4列的hard对应为磁盘容量的硬限制数值；

inodes：表示用户当前已经拥有的文件数量，（i节点数）有edquota自动计算；

soft：第6列的soft对应为文件数量的软限制数值；

hard：第7列的hard对应为文件数量的硬限制数值；

```
[root@localhost ~]# useradd crushlinux
[root@localhost ~]# echo "123456" | passwd --stdin crushlinux
[root@localhost ~]# edquota -u crushlinux
Disk quotas for user crushlinux (uid 1000):
  Filesystem                   blocks       soft       hard     inodes     soft
hard
  /dev/sdb1                         0          80000     100000       0          80
100
```

4、开启或者取消磁盘配额功能：

- quotaon： 开启磁盘配额
- quotaon -p 查看是否开启磁盘配额;
- quotaoff： 关闭磁盘配额

注意是用root进行开启/关闭

```
[root@localhost ~]# quotaon -ug /data
```

5、通过crushlinux用户登录系统，测试配额功能

```
[root@localhost ~]# su - crushlinux
[crushlinux@localhost ~]$ touch /data/{1..80}
[crushlinux@localhost ~]$ touch /data/81
sdb1: warning, user file quota exceeded.
[crushlinux@localhost ~]$ touch /data/{82..100}
[crushlinux@localhost ~]$ touch /data/101
sdb1: write failed, user file limit reached.
touch: 无法创建"/data/101": 超出磁盘限额

[crushlinux@localhost ~]$ rm -rf /data/{50..100}
[crushlinux@localhost ~]$ dd if=/dev/zero of=/data/testfile bs=1M count=80M
sdb1: warning, user block quota exceeded.
sdb1: write failed, user block limit reached.
dd: 写入"/data/testfile" 出错: 超出磁盘限额
记录了98+0 的读入
记录了97+0 的写出
```

102400000字节(102 MB)已复制，2.67873 秒，38.2 MB/秒

[crushlinux@localhost ~]$ dd if=/dev/zero of=/data/testfile bs=1M count=100M

sdb1: warning, user block quota exceeded.

sdb1: write failed, user block limit reached.

dd: 写入"/data/testfile" 出错: 超出磁盘限额

记录了98+0 的读入

记录了97+0 的写出

102400000字节(102 MB)已复制，0.419942 秒，244 MB/秒

6、使用quota命令查看用户crushlinux的磁盘使用情况：

[root@localhost ~]# quota -u crushlinux

Disk quotas for user crushlinux (uid 1000):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/sdb1	100000*	80000	100000	7days	50	80	100	

7、使用repquota命令查看/data文件系统的配额使用情况报告：

[root@localhost ~]# repquota /data

*** Report for user quotas on device /dev/sdb1

Block grace time: 7days; Inode grace time: 7days

Block limits

File limits

User used soft hard grace used soft hard grace

---

root -- 20 0 0 2 0 0

crushlinux +- 100000 80000 100000 6days 50 80 100

8、设置过期时间

[root@localhost ~]# edquota -t

Grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem	Block grace period	Inode grace period
/dev/sdb1	7days	7days