# 背景

某台机器上安装了2块IDE磁盘和4块SCSI磁盘，IDE硬盘直接连接到主板集成的IDE接口上，SCSI磁盘则是连接到一块PCI接口的SCSI卡上。在没有RAID参与的条件下，系统可以识别出6块磁盘，并且经过文件系统格式化以后，挂载到某个盘符或者目录下，供应用程序读写。

安装RAID后，通过配置，先将2块IDE磁盘做成一个RAID0系统，如果原来每块IDE磁盘是80G，做成RAID0后变成一个160GB的虚拟磁盘。然后将4块SCSI磁盘做成RAID5系统，如果每块SCSI盘大小73GB，则最后是3块磁盘，容量216GB。

因为RAID程序需要使用磁盘上的部分空间来存放一些RAID信息，所以实际容量将会变小，经过RAID程序的处理后，6块磁盘变为2块虚拟磁盘。

# 概述

RAID是廉价冗余磁盘阵列（Redundant Array of Inexpensive Disk）的简称，有时也简称磁盘阵列（Disk Array）。

RAID是一种把多块独立的磁盘（物理磁盘）按不同的方式组合起来形成一个磁盘组，在逻辑上看起来就是一块大的磁盘，从而提供比单个物理磁盘更大的存储容量或更高的存储性能，同时又提供不同级别数据冗余备份的一种技术。

# 级别

把多个物理磁盘通过不同的技术方式组成磁盘阵列，这个的不同技术方式就被称为RAID级别。

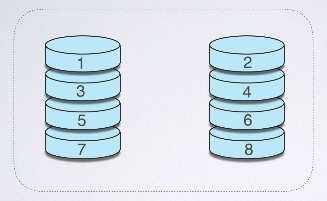
RAID级别一般有：Raid0、Raid1、Raid0+1(也称为Raid10)、Raid5

Raid级别优缺点对比：

|  |  |  |  |
| --- | --- | --- | --- |
| RAID级别 | 优点 | 缺点 | 应用场景 |
| RAID0 | 读写速度最快 | 没有任何冗余 | MySQL Slave(数据库slave节点多个，允许一个故障)，  集群节点RS |
| RAID1 | 100%冗余，镜像  (可以损失一块盘) | 读写性能一般，  成本高 | 单独的，数据重要，不能宕机的业务，监控，系统盘 |
| RAID5 | 具备一定的性能和冗余，可以坏一块盘，  读性能不错 | 写性能不高 | 一般的业务都可以用 |
| RAID10 | 读写速度快，  100%冗余 | 成本高 | 性能和冗余要求都很高业务，数据库主库和存储的主节点 |

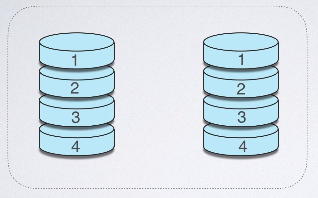
## RAID0

RAID0需要两块硬盘，这样速度快1倍，但是如果一个磁盘坏了则数据全部作废。



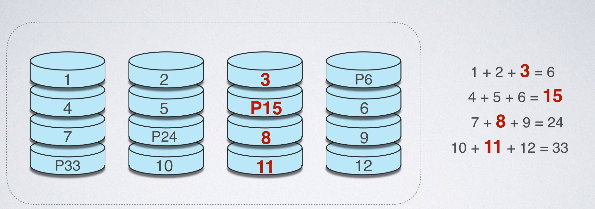
## RAID1

RAID1也需要2块硬盘，一个作为备份盘，安全性提高1倍，但是利用率只有一半。



## RAID5

RAIID5至少需要3块硬盘，一个硬盘的空间用来存放校验码（不是一个磁盘单独作为校验盘，校验信息分散在各个磁盘上，最后校验信息占据1个磁盘的空间大小）。即3块硬盘，能用2块的空间。



## RAID10

# 技术分类

常见的RAID技术分为两种：

基于硬件的RAID技术和基于软件的RAID技术。

## 软件RAID

### 概述

通过硬件阵列卡实现raid具有可靠性高，性能好等特点，但是对于一般的企业而言硬件阵列卡固然好，如果大规模应用的话动辄几千上万的费用也不是他们所能承受的，难道就没有既能保证数据安全，又能减少费用支持的IT方案吗？当然有，软raid就可以实现这一需求。

### 特点

软件RAID有3个缺点：

1. 占用内存空间
2. 占用CPU资源
3. 软件RAID程序无法将安装有操作系统的那个磁盘分区做成RAID模式

因为RAID程序是运行在操作系统之上的，所以在启动操作系统之前，是无法实现RAID功能的，也就是说系统损坏了，RAID程序也就无法运行了。

### mdadm

mdadm是多磁盘和设备管理Multiple Disk and Device Administration的缩写。它是一个命令行工具，可用于管理Linux上的软件RAID阵列。

基础功能：

1. 创建RAID阵列：

mdadm --create /dev/md/test --homehost=any --metadata=1.0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1

1. 组合（并启动）RAID阵列：

mdadm --assemble /dev/md/test /dev/sda1 /dev/sdb1

1. 停止RAID阵列：

mdadm --stop /dev/md/test

1. 删除RAID阵列：

mdadm --zero-superblock /dev/sda1 /dev/sdb1

1. 检查所有已组合的RAID阵列的状态：

cat /proc/mdstat

### 实践

一台centos 7.2 VM，主机名分别为host1，host1上挂载两块50G的数据盘，我们的实验就在这两块数据盘上完成。

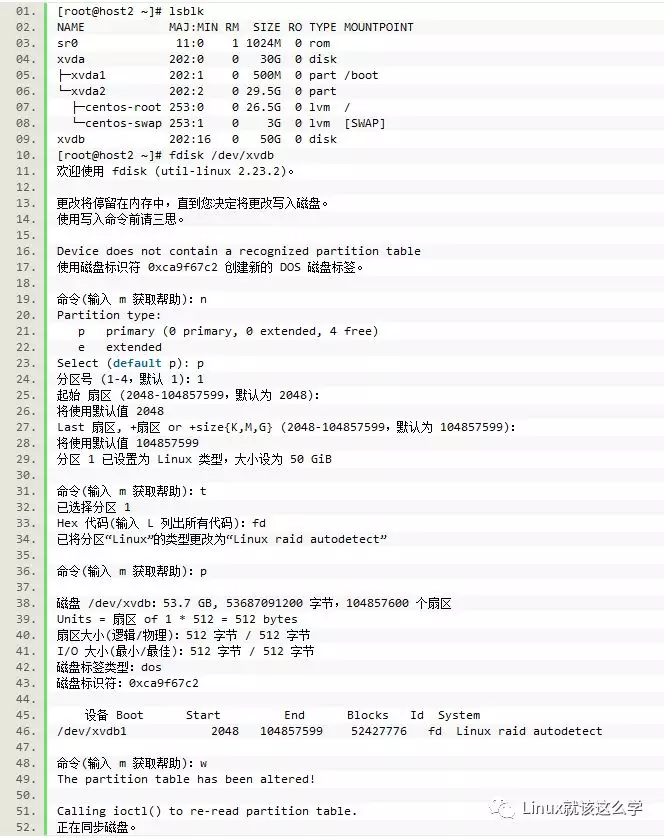
P.S:需要强调的是生产环境下两块组raid的硬盘必须是同品牌同型号同容量的，否则极容易出现软raid失效的情况。

1、确认操作系统是否安装了**mdadm软件**

[root@host1 ~]# rpm -qa |grep mdadm

mdadm-3.3.2-7.el7.x86\_64

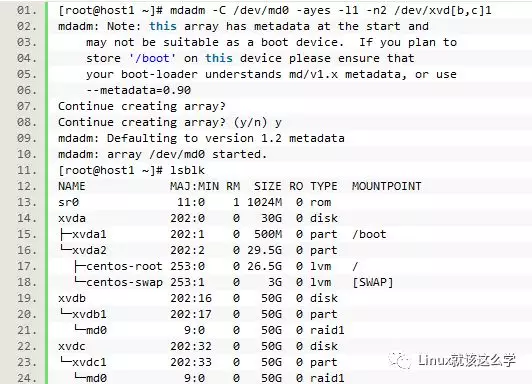
2、对两块数据盘进行分区，并设置分区类型为raid



fdisk命令只能对容量在2T以下的硬盘进行分区，如果超过2T则需要使用parted工具了。

在fdisk下raid的分区类型代码是fd，在parted工具下首先需要使用mklabel将磁盘格式由MBR改成GPT，然后才能使用mkpart命令进行分区，分完区之后使用set设置分区flag为raid即可。

3、使用mdadm命令创建raid1



可以看到创建时有个Note的提示，是说软raid不能当启动设备，这就是软raid比较鸡肋的地方了。

mdadm -C /dev/md0 -ayes -l1 -n2 /dev/xvd[b,c]1

命令说明：

-C创建阵列;

-a同意创建设备，如不加此参数时必须先使用mknod 命令来创建一个RAID设备，不过推荐使用-a yes参数一次性创建;

-l阵列模式;

-n阵列中活动磁盘的数目，该数目加上备用磁盘的数目应该等于阵列中总的磁盘数目;

/dev/md0阵列的设备名称，如果还有其他阵列组可以以此类推；

创建完成后可以使用cat /proc/mdstat查看阵列状态：

下图中第一次查看的时候提示resync完成了95.7%，第二次查询的时候两块盘才真正同步完。

也可以使用mdadm -D /dev/md0查看阵列组的状态

4、创建md0的配置文件

echo DEVICE /dev/sd{a,b}1 >> /etc/mdadm.conf

mdadm -Evs >> /etc/mdadm.conf

mdadm运行时会自动检查/etc/mdadm.conf 文件并尝试自动装配，因此第一次配置raid后可以将信息导入到/etc/mdadm.conf 中。

5、使用/dev/md0

在/dev/md0上创建文件，然后挂载进行使用。

mkfs.ext4 /dev/md0

## 硬件RAID

### 概述

RAID卡是一种利用独立硬件实现RAID功能的方法。对于硬件RAID操作系统无法感知，只能通过厂家提供的RAID卡管理软件来查看卡上所连接的物理磁盘。而且，**在配置RAID卡的时候，也不能在操作系统上完成，而必须进入这个硬件完成**（或者在操作系统改下通过RAID卡配置工具来设置），一般的RAID卡都是在开机自检的时候，进入他的ROM配置程序来配置各种RAID功能的。

**注：**在我们安装操作系统的时候，有时候找不到磁盘，这就是以为该服务器默认的是使用了硬件RAID，我们不去设置的话系统感知不到这个盘，我们必须在进入开机前设置RAID的功能才可以捕获这些磁盘。

RAID卡可以集成到SCSI卡或者主板的南桥（在南桥的称为板载RAID卡）。操作系统出了RAID卡驱动外不用安装任何额外的软件。

带CPU的RAID卡就是一个小型计算机系统，有自己的CPU、内存、ROM、总线和IO接口，只不过这个小计算机系统是为大计算机系统服务的。

SCSI RAID卡上一定要包含SCSI控制器，因为其后端链接的是SCSI物理磁盘，前端链接的是主机的PCI总线，所以一定要有一个PCI总线控制器来维护PCI总线的仲裁、数据发送、接收等功能。还需要有一个ROM，一般都是用Flash芯片，其中存放着初始化RAID卡必须的代码以及实现RAID功能所需的代码。

RAM的作用首先是数据缓存，提高性能，其次作为RAID卡上的CPU执行RAID计算所需要的内存空间。

### RAID与SCSI卡

RAID卡与SCSI卡的区别就在于RAID功能，其他的没有什么大的区别。如果RAID卡上有多个SCSI通道，那么就称为多通道RAID卡。目前，SCSI RAID卡最高有4通道的，其后端可以接入4条SCSI总线，所以最多可连接64个SCSI设备（16位总线）。

增加了RAID功能后，SCSI就成了RAID代码的傀儡，RAID让他干嘛就干嘛，SCSI对其下面的磁盘情况了解，那么RAID就通过它获取底层磁盘的情况，按照ROM中所设置的选项，比如RAID类型、条带大小等，对RAID程序代码做相应的调整，操控它的傀儡SCSI控制器向主机报告“虚拟”的逻辑盘，而不是所有的物理盘。

### LUN

条带化之后，RAID程序代码就操控SCSI控制器向OS层驱动程序代码提交一个虚拟化之后所谓的“虚拟盘”或者“逻辑盘”，称之为LUN。

LUN是SCSI ID更细一级的地址号，每个SCSI ID（Target ID）下面还可以有更多的LUN ID（视ID字段的长度而定），对于大型磁阵而言，可以生成几百或者几千个虚拟磁盘，为每个虚拟磁盘分配一个SCSI ID是不可能的，因为每个SCSI总线最多接入16个设备，LUN就是下一级的寻址ID，后来LUN概念转换为软件生成的虚拟磁盘，统一称为“卷”。

0通道RAID卡又称为RAID子卡，0通道的意思是这块卡的后端没有SCSI通道。

磁阵上会有多个外部SCSI接口，这个接口是为了链接多台主机使用的，每个由盘阵RAID控制器生成的逻辑磁盘，可以通过设置只分配（Assign/Map）到其中一个口。比兔LUN1被分配到了1号口，那么链接2号口的主机就看不到，月可以一个LUN分配给多个口，但是很容易造成数据不一致，除非使用集群文件系统或者高可用性系统软件的额参与。

# RAID与LVM

## 概述

LVM（Logic Volumn Management）逻辑卷管理，它的最大用途是可以灵活的管理磁盘的容量，让磁盘分区可以随意放大或缩小，便于更好的应用磁盘的剩余空间，如果过于强调性能与备份，那么还是应该使用RAID功能，而不是LVM。

LVM是在磁盘的分区之上，创建一个逻辑层，以方便系统管理整个硬盘分区系统，LVM最先由IBM开发，在AIX系统上实现，OS/2操作系统与HP-UX也支持这个功能。

简单地讲，LVM就是将几个物理分区或磁盘通过软件组合在一起，看起来像是一个独立的大磁盘，如果要用这块大磁盘，就要在将它分割为可以使用的分区（LV），可以在这个可使用的分区上任意切割，增加及缩小分区容量。我们知道分区上的文件系统受到block（块）大小的限制，同理，LVM的磁盘大小也是受PE的限制。

## 特点

LVM的优点：

1. 灵活性：当使用逻辑卷，文件系统可以分布在多个磁盘上，大小不会受物理磁盘的限制，因为你能够指定多个磁盘和分区组成一个逻辑卷；
2. 可调整的存储池：可以在系统运行状态下动态拓展或缩小逻辑卷的大小，而无需重新格式化或者重新分区，还可以增加新磁盘到LVM的存储池中；
3. 在线数据重分布：可以在线地把数据从一个盘移动到另一个盘，或者改变数据在磁盘上的分布位置；
4. 方便的设备命名：卷组可以使用宿主机名称，LV可以使用虚拟服务器的主机名；
5. 磁盘条带化：可以实现并发读写；
6. 镜像卷；
7. 快照卷。

LVM的缺点：

1. 在从卷组中移出一个磁盘时必须使用reduceer，否则会出问题；
2. 性能和备份差。

## RAID和LVM区别

LVM：灵活的管理磁盘的容量，有一定的冗余和性能功能，但很弱；

RAID：更侧重性能和数据安全。

# 条带化

## 概述

RAID把数据分成条带，一个条带横跨所有数据磁盘，每个磁盘上存储条带的一部分，称为sagment，也称为条带深度。一个条带包含的扇区或块的个数，称为条带长度。

RAID向操作系统提供的是卷，是连续的扇区（或者叫LBA）空间地址，而卷向文件系统所展示的是分区，相对于卷，分区就是将大片连续地址再次切开。文件系统用扇区组成所谓“簇”，或者叫cluster，或者叫分配单元，fs存放数据只以簇为单位，而不会出现“读入或者写出半个簇”这种io命令。而卷之下的各层都可以以单个扇区为单位了，扇区是存储系统最小的io单位。

RAID提供一片连续扇区地址，假如文件系统选择了扇区1024-1027号这4个连续的扇区作为这个2k文件的存放空间，则文件系统写入这个文件的时候，卷接受到这个指令之后，会将这段扇区号码传送给raid驱动，raid驱动接受之后会将号码传送给raid芯片进行地址翻译（硬raid卡）或者直接在驱动层面进行地址翻译（软raid卡），地址翻译这里的意思是将这段连续的地址映射到实际的物理硬盘地址，因为raid提供给上层的是虚拟磁盘/lun，一个lun可以分步在多个物理硬盘上。地址翻译的过程一定要查询stripe也就是条带映射表，当初你怎么分的，此时就会影响翻译之后实际的硬盘扇区地址。再回来说你的raid5的128KB条带，128KB条带=磁盘数量乘以每个磁盘上组成这个条带的segment大小，也就是说一个条带把排列的多个磁盘横向切成了一条一条的，硬盘本身相当于竖条，而横条和竖条切开之后形成的小格子就是segment，也叫条带深度，stripedepth。比如8个盘的raid5系统，其中一块用于存放parity，128KB条带除以8等于16KB，也就是说segment=条带深度=16KB=每个磁盘上贡献一个条带所使用的空间。再回来说2k的文件写入，这个情况下，地址翻译会将2k的地址翻译为“磁盘m上的n到磁盘m上的n+3号扇区”，当然也可能是“磁盘x上的y到磁盘a上的b”，总之地址落在物理硬盘上的哪个区域，条带中央/条带边缘/单个物理硬盘/多个物理硬盘，文件系统是不知道的，由raid层面决定。这种不知道称为“盲”，现在大多文件系统都盲，但是也有不盲或者半盲的。我们在raid层面来设计条带，分步等，实际上都是盲操作，效果不会很大，包括raid5本身的并发io特点，也是盲并发。所谓并发，拿你刚才的例子，某个io需要写入2k的数据，如果地址翻译结果为这4个扇区落在一个segment里，则这个io只会占用一个数据盘，外加需要占用parity区域，如果此时还有一个io需要写入2k数据，而这次的4个扇区落在了另一块数据盘上，而它需要的parity数据恰好与前一个io及其所需要的parity区域不在同一个盘上，则这两个io可以并行操作，4块磁盘同时读写。由于这种并发是基于“恰好”的，所以raid5提供的是盲并发，要实现不盲的并发只能靠上层文件系统。

## 影响因素

# 特点

磁盘阵列可以把多个磁盘驱动器通过不同的连接方式连接在一起工作，大大提高了读取速度，同时把磁盘系统的可靠性提高到接近无措的境界，使其可靠性极高。

使用RAID的直接好处：

1. 提升数据**安全性**；
2. 提升**数据读写**性能；
3. 提供更大的单一逻辑磁盘数据容量存储。

# 配置

## RAID1

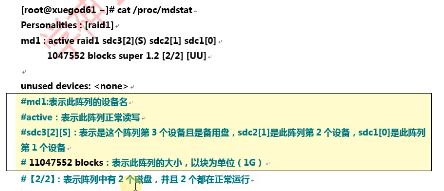
### 创建RAID1

环境：创建分区：sde1，sde2，sde3（大小1G）

注：RAID1至少2块磁盘，热备盘1块，共3块磁盘。

操作：

1. 创建分区（物理盘）：fdisk /dev/sdc（n->p->+1G->…->w）
2. 查看分区：ll /dev/sdc\*
3. 创建RAID1：mdadm –C –v /dev/md1 –l 1 –n 2 –x 1 /dev/sdc{1,2,3}
4. 查看阵列信息：cat /proc/mdstat 或 mdadm –Ds



1. 生成配置文件：mdadm –Ds > /ect/mdadm.conf
2. 创建RAID分区：fdisk /dev/md1
3. 建立文件系统：mkfs.xfs /dev/md1p1
4. 创建挂载点：mount /raid1 /dev/md1p1

### 添加1G热备盘

### 模拟磁盘故障，自动顶替故障盘

操作步骤：

1. 设置故障盘sdc1：

mdadm –f /dev/md1 /dev/sdc1

1. 每隔1s查看状态，观察sdc3备用盘能够自动顶替故障盘：

watch –n 1 cat /proc/mdstat

1. 稍等片刻阵列重新重建成功，此时/dev/sdc3后面的【S】消息，即成功顶替故障盘
2. 移除故障盘/dev/sdb1

### 卸载阵列

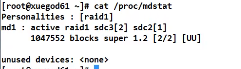
操作步骤：

1、卸载故障盘

mdadm -r /dev/md1 /dev/sdc1

注：移出RAID md1下面的故障盘sdc1。

2、然后使用cat查看状态：



3、重新生成配置文件（否则重启还是原来的配置）：

mdadm –Ds > /etc/dmadm.conf

## RAID5

### 创建RAID5

实验环境：sde1、sde2、sde3、sde5、sde6个主分区，每个1G

操作步骤：

### 停止阵列，重新激活阵列

### 添加新的1G热备盘，拓展阵列容量

## RAID10