GNU/Linux
磁盘管理4

磁盘阵列-RAID

RAID

磁盘阵列（Redundant Arrays of independent Disks，RAID），有“独立磁盘构成的具有冗余能力的阵列”之意。

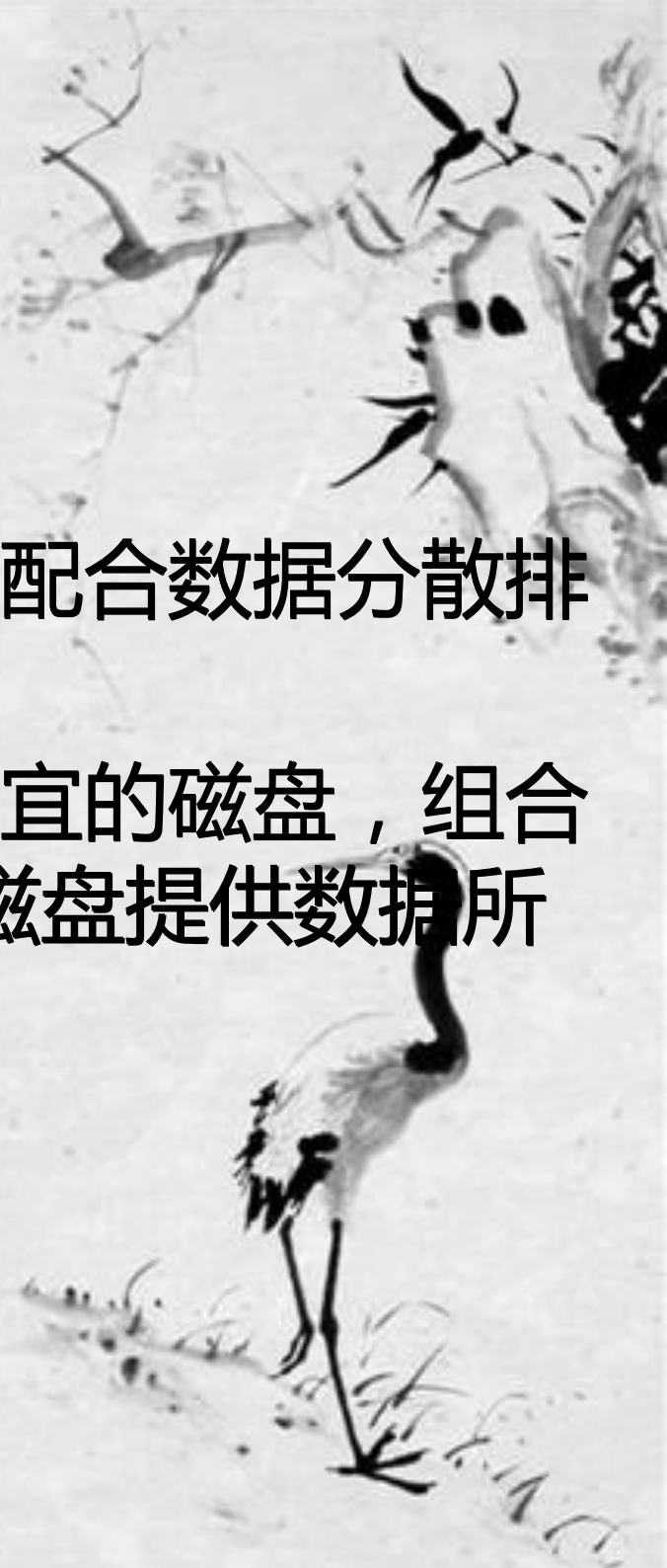
磁盘阵列-RAID

RAID

原理:

利用数组方式来作磁盘组，配合数据分散排列的设计，提升数据的安全性。

磁盘阵列是由很多价格较便宜的磁盘，组合成一个容量巨大的磁盘组，利用个别磁盘提供数据所产生加成效果提升整个磁盘系统效能。



磁盘阵列-RAID

RAID

原理:

利用这项技术，将数据切割成许多区段，分别存放在各个硬盘上。磁盘阵列还能利用同位检查（Parity Check）的观念，在数组中任一颗硬盘故障时，仍可读出数据，在数据重构时，将数据经计算后重新置入新硬盘中。

磁盘阵列-RAID

RAID种类

1. 硬RAID:通过硬件实现RAID功能

2. 软RAID:通过ＯＳ实现RAID功能

磁盘阵列-RAID

常用的RAID类型

RAID0:条带卷

1. 最早出现的RAID模式，即Data Stripping数据分条技术。

2. RAID 0是组建磁盘阵列中最简单的一种形式，只需要2块以上的硬盘即可

3. 成本低，可以提高整个磁盘的性能和吞吐量。

4. RAID 0没有提供冗余或错误修复能力，但实现成本是最低的。

磁盘阵列-RAID

常用的RAID类型

RAID0:

5. 实现方式是把磁盘驱动程序以软件的方式串联在一起创建一个大的卷集。

6. 在使用中电脑数据依次写入到各块硬盘中，它的最大优点就是可以整倍的提高硬盘的容量。如使用了三块80GB的硬盘组建成RAID 0模式，那么磁盘容量就会是240GB。其速度方面，各单独一块硬盘的速度完全相同。

磁盘阵列-RAID

常用的RAID类型

RAID0:

7. 最大的缺点在于任何一块硬盘出现故障，整个系统将会受到破坏，可靠性仅为单独一块硬盘的1/N。

8. 为了解决这一问题，便出现了RAID 0的另一种模式。

即在N块硬盘上选择合理的带区来创建带区集。其原理就是将原先顺序写入的数据被分散到所有的四块硬盘中同时进行读写。四块硬盘的并行操作使同一时间内磁盘读写的速度提升了4倍。

磁盘阵列-RAID

常用的RAID类型

RAID0:

在创建带区集时，合理的选择带区的大小非常重要。如果带区过大，可能一块磁盘上的带区空间就可以满足大部分的I/O操作，使数据的读写仍然只局限在少数的一、两块硬盘上，不能充分的发挥出并行操作的优势。另一方面，如果带区过小，任何I/O指令都可能引发大量的读写操作，占用过多的控制器总线带宽。因此，在创建带区集时，我们应当根据实际应用的需要，慎重的选择带区的大小。

磁盘阵列-RAID

常用的RAID类型

RAID0:

带区集虽然可以把数据均匀的分配到所有的磁盘上进行读写。但如果我们把所有的硬盘都连接到一个控制器上的话，可能会带来潜在的危害。这是因为当我们频繁进行读写操作时，很容易使控制器或总线的负荷 超载。为了避免出现上述问题，建议用户可以使用多个磁盘控制器。最好解决方法还是为每一块硬盘都配备一个专门的磁盘控制器。

10.不能在线添加及删除它的组成分区。

磁盘阵列-RAID

常用的RAID类型

RAID1:磁盘镜像

1. 原理是把一个磁盘的数据镜像到另一个磁盘上。即数据在写入一块磁盘的同时，会在另一块闲置的磁盘上生成镜像文件，在不影响性能情况下最大限度的保证系统的可靠性和可修复性上，只要系统中任何一对镜像盘中至少有一块磁盘可以使用，甚至可以在一半数量的硬盘出现问题时系统都可以正常运行.

磁盘阵列-RAID

常用的RAID类型

RAID1:磁盘镜像

2. 当一块硬盘失效时，系统会忽略该硬盘，转而使用剩余的镜像盘读写数据，具备很好的磁盘冗余能力。虽然这样对数据来讲绝对安全，但是成本也会明显增加，

3.磁盘利用率为50%，以四块80GB容量的硬盘来讲，可利用的磁盘空间仅为160GB。

4. 另外，出现硬盘故障的RAID系统不再可靠，应当及时的更换损坏的硬盘，否则剩余的镜像盘也出现问题，那么整个系统就会崩溃。

磁盘阵列-RAID

常用的RAID类型

RAID1:磁盘镜像

5. 更换新盘后原有数据会需要很长时间同步镜像，外界对数据的访问不会受到影响，只是这时整个系统的性能有所下降。因此，RAID 1多用在保存关键性的重要数据的场合。

6. RAID 1主要是通过二次读写实现磁盘镜像，所以磁盘控制器的负载也相当大，尤其是在需要频繁写入数据的环境中。为了避免出现性能瓶颈，使用多个磁盘控制器就显得很有必要。

磁盘阵列-RAID

常用的RAID类型

RAID1:磁盘镜像

7. 假如只有两块盘组成，则不能在线删除其中的任意一块盘，只能先将其中一块作“破坏”处理，再将之删除。可以在线添加其他硬盘，添加的盘容量应与其他盘相当，否则会报错。

磁盘阵列-RAID

常用的RAID类型

RAID0+1:

1. RAID0与RAID1的结合

2. 保留各自的优点,规避各自的缺点

3. RAID0+1要在磁盘镜像中建立带区集至少4个硬盘。

磁盘阵列-RAID

常用的RAID类型

RAID2:海明校验码

1. 将数据条块化分布于不同的硬盘上，条块单位为位或字节。

2. RAID 2 使用一定的编码技术来提供错误检查及恢复。这种编码技术需要多个磁盘存放检查及恢复信息，使得RAID 2技术实施更复杂。

3. 因此,在商业环境中很少使用。

磁盘阵列-RAID

常用的RAID类型

RAID2:

4. 每个磁盘数据的各个位，由一个数据不同的位运算得到的海明校验码可以保存另一组磁盘上。由于海明码的特点，它可以在数据发生错误的情况下将错误校正，以保证输出的正确。它的数据传送速率相当高，如果希望达到比较理想的速度，那最好提高保存校验码ECC码的硬盘。没有免费的午餐，这里也一样，要利用海明码，必须要付出数据冗余的代价。输出数据的速率与驱动器组中速度最慢的相等。

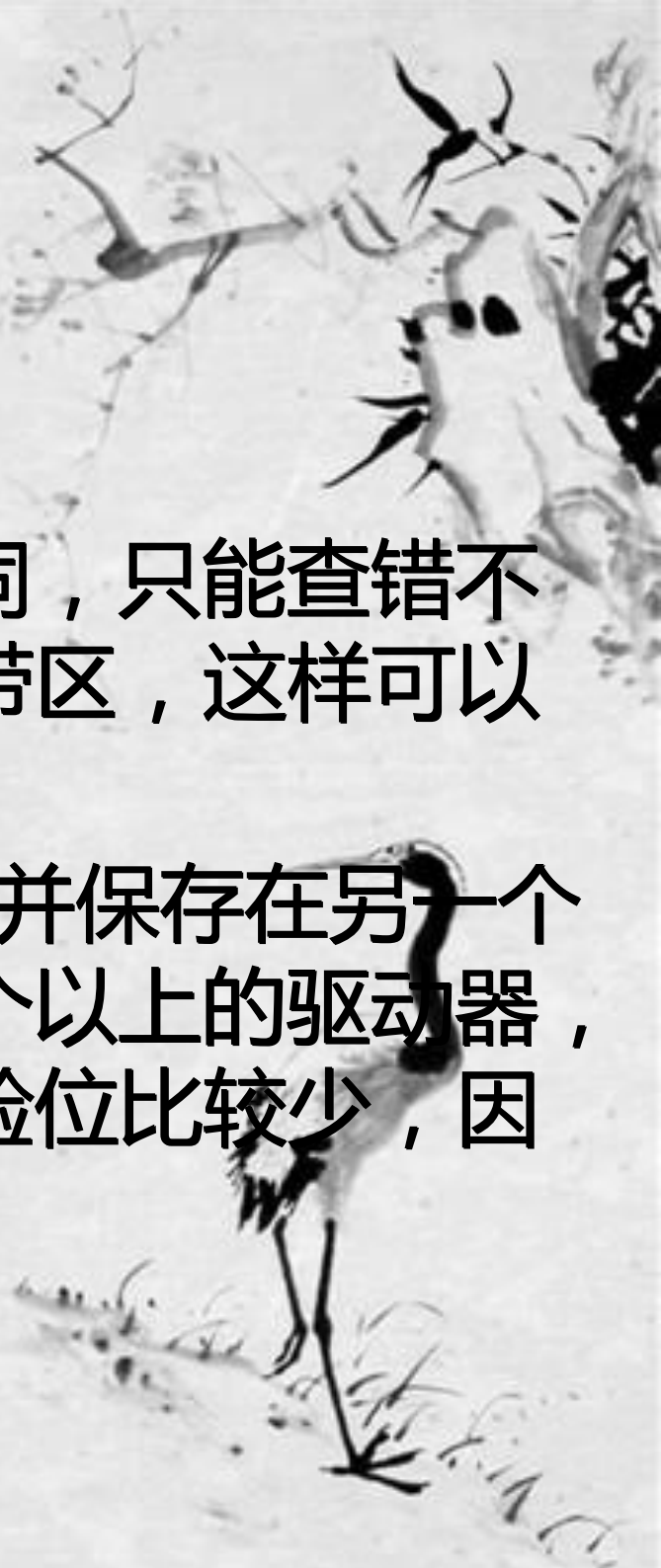
磁盘阵列-RAID

常用的RAID类型

RAID3:带奇偶校验码的并行传送

1. 每这种校验码与RAID2不同，只能查错不能纠错。它访问数据时一次处理一个带区，这样可以提高读取和写入速度。

2. 校验码在写入数据时产生并保存在另一个磁盘上。需要实现时用户必须要有三个以上的驱动器，写入速率与读出速率都很高，因为校验位比较少，因此计算时间相对而言比较少。



磁盘阵列-RAID

常用的RAID类型

RAID3:带奇偶校验码的并行传送

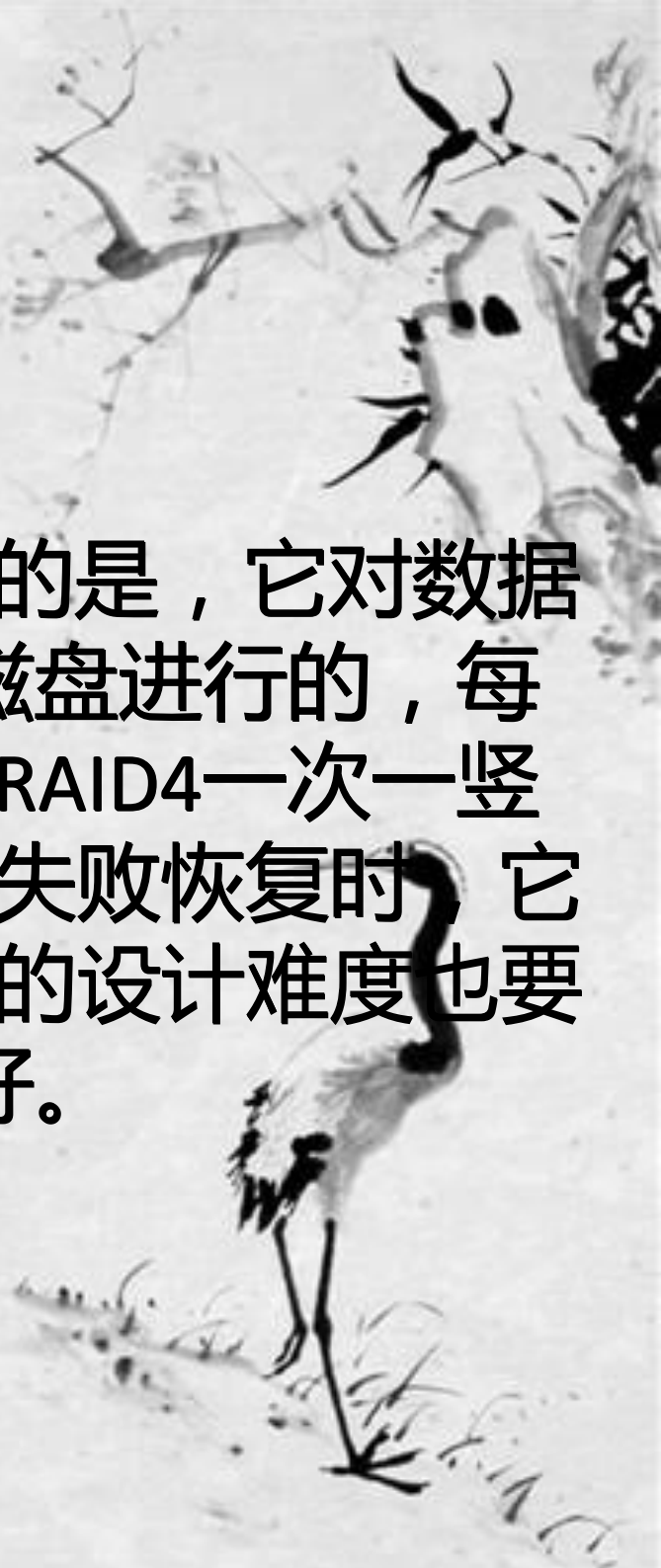
3. 用软件实现RAID控制将是十分困难的，控制器的实现也不是很容易。它主要用于图形（包括动画）等要求吞吐率比较高的场合。不同于RAID 2，RAID 3使用单块磁盘存放奇偶校验信息。如果一块磁盘失效，奇偶盘及其他数据盘可以重新产生数据。如果奇偶盘失效，则不影响数据使用。RAID 3对于大量的连续数据可提供很好的传输率，但对于随机数据，奇偶盘会成为写操作的瓶颈。

磁盘阵列-RAID

常用的RAID类型

RAID4:带奇偶校验码的独立磁盘结构

1. RAID4和RAID3很象，不同的是，它对数据的访问是按数据块进行的，也就是按磁盘进行的，每次是一个盘。RAID3是一次一横条，而RAID4一次一竖条。它的特点和RAID3也挺象，不过在失败恢复时，它的难度可要比RAID3大得多了，控制器的设计难度也要大许多，而且访问数据的效率不怎么好。



磁盘阵列-RAID

常用的RAID类型

RAID5:分布式奇偶校验的独立磁盘结构

1. 它的奇偶校验码存在于所有磁盘上。RAID5的读出效率很高，写入效率一般，块式的集体访问效率不错。因为奇偶校验码在不同的磁盘上，所以提高了可靠性。但是它对数据传输的并行性解决不好，而且控制器的设计也相当困难。

磁盘阵列-RAID

常用的RAID类型

RAID5:分布式奇偶校验的独立磁盘结构

2. RAID 3 与RAID 5相比，重要的区别在于RAID 3每进行一次数据传输，需涉及到所有的阵列盘。而对于RAID 5来说，大部分数据传输只对一块磁盘操作，可进行并行操作。在RAID 5中有“写损失”，即每一次写操作，将产生四个实际的读/写操作，其中两次读旧的数据及奇偶信息，两次写新的数据及奇偶信息。

3. RIAD5至少需要 3 块磁盘,分区小大尽量一致

磁盘阵列-RAID

常用的RAID类型

RAID6:带有两种分布存储的奇偶校验码的独立磁盘结构

1. 是对RAID5的扩展，主要是用于要求数据绝对不能出错的场合。当然了，由于引入了第二种奇偶校验值，所以需要N+2个磁盘，同时对控制器的设计变得十分复杂，写入速度也不好，用于计算奇偶校验值和验证数据正确性所花费的时间比较多，造成了不必须的负载。价格昂贵。

2. 同时带有区带奇偶效验和块奇偶校验

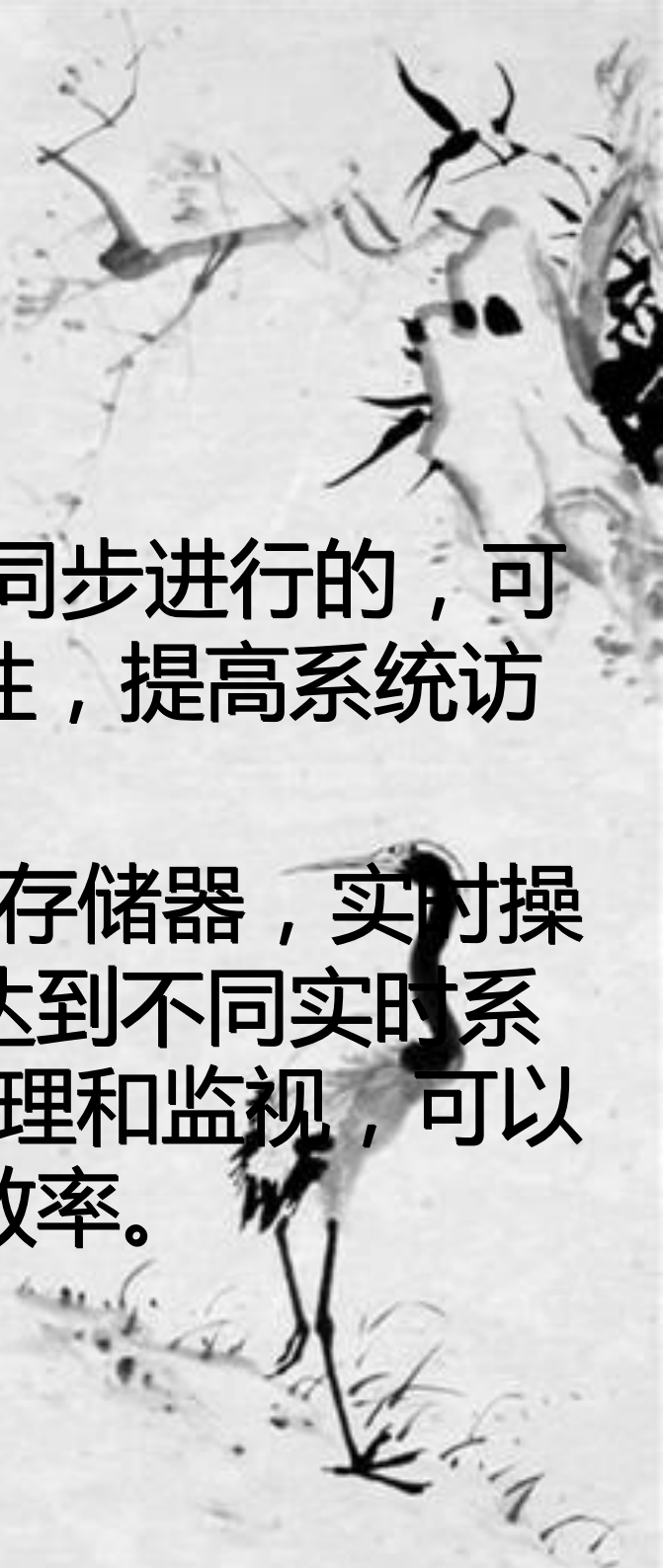
磁盘阵列-RAID

常用的RAID类型

RAID7:优化的高速数据传送磁盘结构

1. RAID7所有的I/O传送均是同步进行的，可以分别控制，这样提高了系统的并行性，提高系统访问数据的速度；

2. 每个磁盘都带有高速缓冲存储器，实时操作系统可以使用任何实时操作芯片，达到不同实时系统的需要。允许使用SNMP协议进行管理和监视，可以对校验区指定独立的传送信道以提高效率。



磁盘阵列-RAID

常用的RAID类型

RAID7:优化的高速数据传送磁盘结构

3. 可以连接多台主机，因为加入高速缓冲存储器，当多用户访问系统时，访问时间几乎接近于0。

4. 由于采用并行结构，因此数据访问效率大大提高。需要注意的是它引入了一个高速缓冲存储器，这有利有弊，因为一旦系统断电，在高速缓冲存储器内的数据就会全部丢失，因此需要和UPS一起工作。

RAID7的价格也非常昂贵。

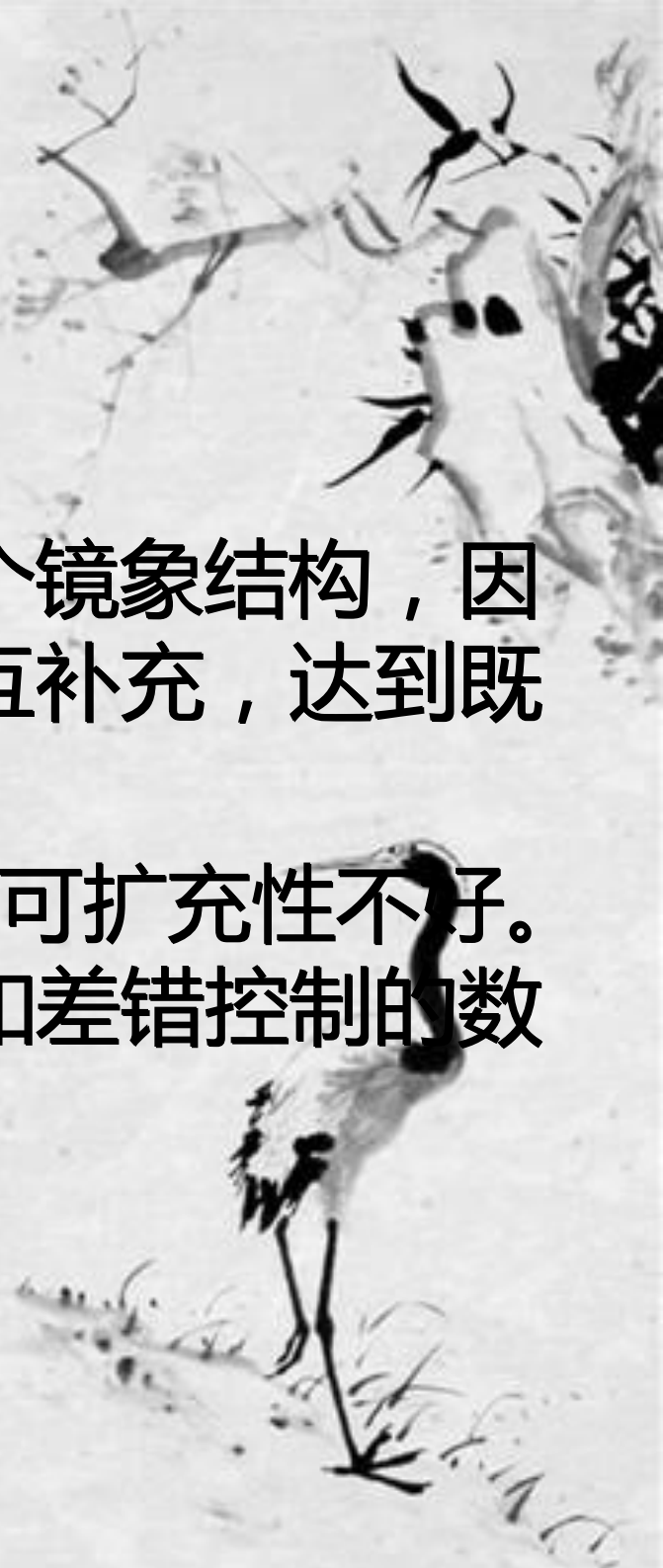
磁盘阵列-RAID

常用的RAID类型

RAID10：高可靠性与高效磁盘结构

1. 此结构即是一个带区结构加一个镜象结构，因为两种结构各有优缺点，因此可以相互补充，达到既高效又高速的目的。

2. 大这种新结构的价格较高，可扩充性不好。主要用于数据容量不大，但要求速度和差错控制的数据库中。



磁盘阵列-RAID

常用的RAID类型

RAID01与RAID10的区别：

我们都以四块硬盘做RIAD 来细说他们的区别：

1. RAID10 是先做镜像后做条带。在这种情况下，如果只是坏掉其中的一个硬盘，对RAID组的影响都不是非常大，只要不是同时坏掉其中的一个硬盘和他的镜像盘，RAID组都不会崩溃。

磁盘阵列-RAID

常用的RAID类型

RAID01与RAID10的区别：

我们都以四块硬盘做RIAD 来细说他们的区别：

2. RAID01 是先将四块硬盘中横向上两两做条带，然后再纵向上做镜像。简而言之：先条带后镜像。这种情况下如果两个条带上有任意两块硬盘坏掉了，则整个RAID组都将崩溃了。不管 发生介质损坏的两块硬盘是否是镜像盘。

磁盘阵列-RAID

常用的RAID类型

RAID01与RAID10的区别：

我们都以四块硬盘做RIAD 来细说他们的区别：

3. RAID10 和 RAID01 在读和写的性能上没有太大的差别，从发生故障的概率上看，RAID01 是远大于RAID10 的。一般情况下选择RAID10的情况会多些。

磁盘阵列-RAID

常用的RAID类型

RAID53：高效数据传送磁盘结构

越到后面的结构就是对前面结构的一种重复和再利用，这种结构就是RAID3和带区结构的统一，因此它速度比较快，也有容错功能。但价格十分高，不易于实现。这是因为所有的数据必须经过带区和按位存储两种方法，在考虑到效率的情况下，要求这些磁盘同步真是很艰苦的。

磁盘阵列-RAID

RAID工具

命令:mdadm

功能:创建/管理/删除RAID

语法格式:

mdadm [模式] <raiddevice> [选项] <组成设备>

磁盘阵列-RAID

模式:

assemble：将以前定义的某个阵列加入当前在用阵列。

create：创建一个新的阵列，每个device 具有 superblocks

manage： 管理阵列，比如 add 或 remove

misc：允许单独对阵列中的某个 device 做操作，比如抹去superblocks 或 终止在用的阵列。

follow or Monitor:监控 raid 1,4,5,6 和 multipath 的状态

grow：改变raid 容量或 阵列中的 device 数目

磁盘阵列-RAID

选项:

-A：加入一个以前定义的阵列

-C：创建一个新的阵列

-D：打印一个或多个 md device 的详细信息

-E：打印 device 上的 md superblock 的内容

-h：帮助信息，用在以上选项后，则显示该选项信息

磁盘阵列-RAID

选项:

-v：显示细节

-b：较少的细节。用于-D和-E选项

-f, --force

-c：指定配置文件，缺省为 /etc/**mdadm/mdadm**.conf

-s：扫描配置文件或 /proc/mdstat以搜寻丢失的信息。

磁盘阵列-RAID

create 或 build 使用的选项:

-c:指定块大小,单位 kb. 缺省为 64.

-l,:设定RAID级别.

-n=:指定阵列中可用 device 数目。

-x：指定初始阵列的热备盘数目。

-a, --auto{=no,yes,md,mdp,part,p}{NN}：创建RAID同时是否创建设备。

磁盘阵列-RAID

实现raid1

创建RAID:

1. 创建硬盘分区,并指定为Linux raid autofdetect(fd)类型

#fdisk /dev/sdb

#fdisk /dev/sdc

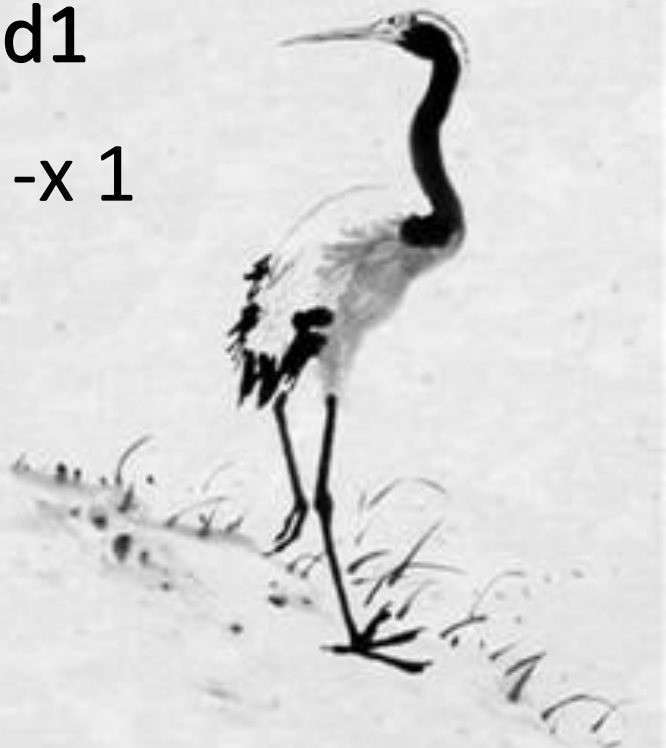
磁盘阵列-RAID

实现raid1

创建RAID:

1. 创建raid设备md1,RAID级别1,可用设备2个,热备设备1个.分别为/dev/sdb1,/dev/sdc1,/dev/sdd1

#mdadm –Cv /dev/md1 –a yes –l 1 –n 2 -x 1 /dev/sd{b,c,d}1



磁盘阵列-RAID

实现raid1

创建RAID:

2. 创建RAID1

注：sdb1及sdc1共同组成RAID1.

磁盘阵列-RAID

实现raid1

3. 格式化RAID阵列

```
# mkfs.ext4  /dev/md1
```

4. 查看RAID的创建过程：

```
mdadm –D /dev/md1
```

磁盘阵列-RAID

实现raid1

5. 生成mdadm.conf配置文件

1)将RAID1的所属设备增加至配置文件

#echo “DEVICE /dev/sdb1 /dev/sdc1” >> /etc/mdadm.conf

2)将通过查看RAID设备的参数增加至配置文件

#mdadm -Ds >> /etc/mdadm.conf

磁盘阵列-RAID

实现raid1

5. 生成mdadm.conf配置文件

#cat /etc/mdadm.conf

DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1

ARRAY /dev/md1 level=raid1 num-devices=2

UUID=dade41ff:9df3d1b7:ba30ec8a:696dd689

磁盘阵列-RAID

实现raid1

6. 创建/dev/md0的挂载点，写入/etc/fstab文件

7. /etc/fstab文件的正确性

#mount -a

磁盘阵列-RAID

实现raid 5

创建RAID:

1. 创建raid设备md5,RAID级别5,可用设备3个,热备设备1个.分别为/dev/sdb1,/dev/sdc1,/dev/sdd1

#mdadm –C /dev/md5 –a yes –l 5 –n 3 –x 1 /dev/sd{b,c,d,e}1

磁盘阵列-RAID

实现raid5

创建RAID:

2. 创建RAID5

注：sdb1、sdc1、sdd1共同组成RAID5,sde1为热备盘。并确认处于umount状态

磁盘阵列-RAID

实现raid5

3. 格式化RAID阵列

```
# mkfs.ext4  /dev/md5
```

4. 查看RAID的创建过程：

```
mdadm –D /dev/md5
```

磁盘阵列-RAID

实现raid5

5. 生成mdadm.conf配置文件

1)将RAID1的所属设备增加至配置文件

#echo “DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 ” >>

/etc/mdadm.conf

2)将通过查看RAID设备的参数增加至配置文件

#mdadm -Ds >> /etc/mdadm.conf

磁盘阵列-RAID

实现raid5

5. 生成mdadm.conf配置文件

#cat /etc/mdadm.conf

DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1

ARRAY /dev/md5 level=raid5 num-devices=3

UUID=dade41ff:9df3d1b7:ba30ec8a:696dd689

磁盘阵列-RAID

实现raid5

6. 创建/dev/md5的挂载点，写入/etc/fstab文件

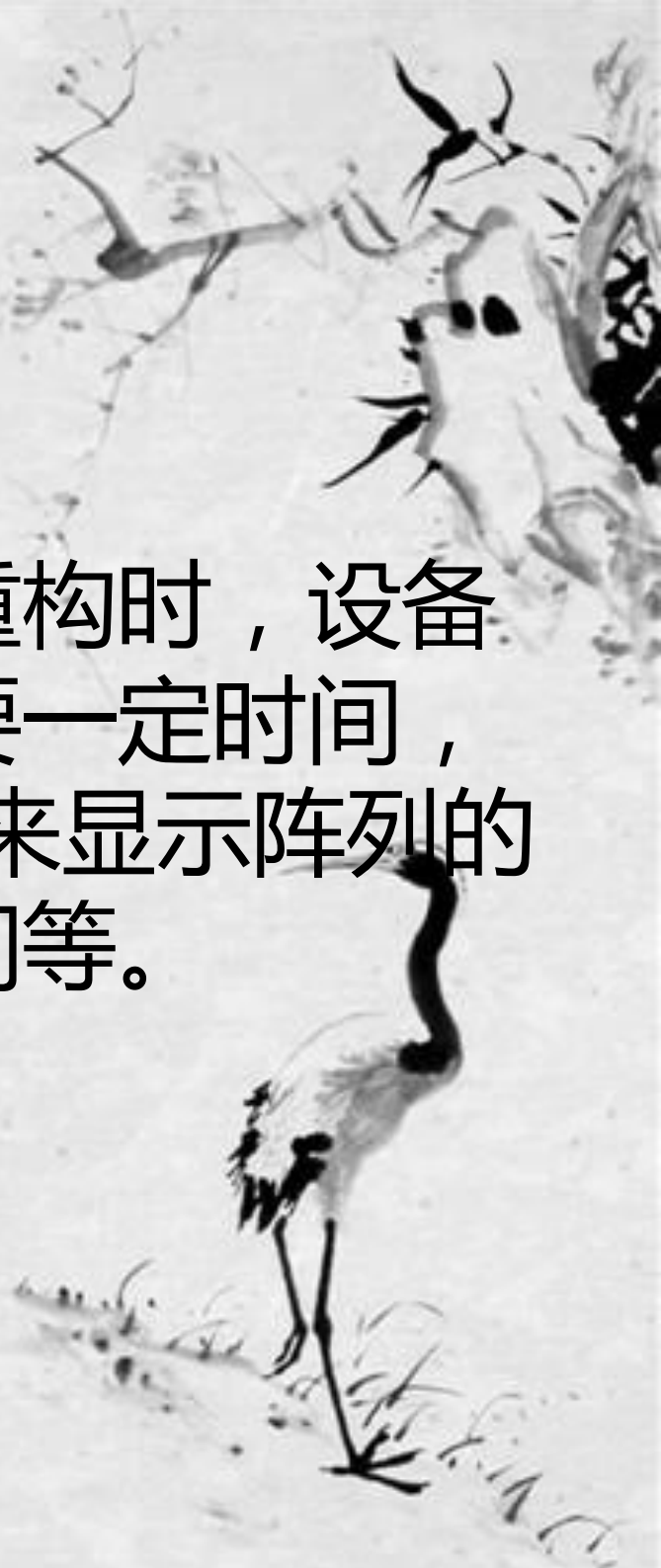
7. mount –a测试/etc/fstab文件的正确性。

磁盘阵列-RAID

查看阵列状态

简介：

当创建一个新阵列或者阵列重构时，设备需要进行同步操作，这一过程需要一定时间，可以通过查看/proc/mdstat文件，来显示阵列的当前状态以及同步进度、所需时间等。



磁盘阵列-RAID

查看阵列状态

查看数据同步过程：

more /proc/mdstat

```
Personalities : [raid6]  [raid5]  [raid4]
md5 : active raid5 sdd1[4] sde1[3](S) sdc1[1] sdb1[0]
75469842 blocks level 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices:<none>
```

磁盘阵列-RAID

可以很清楚地看出当前阵列的状态，各部分所代表的意思如下：“[3/3]”中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，如果有一个设备损坏，则第二位数将减1；“[UUU]”标记当前阵列可以正常使用的设备情况，现假设/dev/sdb1出现故障，则该标记将变成[_UU]，这时的阵列以降级模式运行，即该阵列仍然可用，但是不再具有任何冗余；“sdc1[2]”指阵列所包含的设备数为n，若方括号内的数值小于n，则表示该设备为活动设备，若数值大于等于n，则该设备为备份设备，当一个设备出现故障的时候，相应设备的方括号后将被标以(F)。

磁盘阵列-RAID

故障模拟：

1. 将/dev/sdb1标记为已损坏的设备

```
# mdadm /dev/md5 -f /dev/sdb1
```

2. 移除损坏的设备

```
# mdadm /dev/md5 -r /dev/sdb1
```

3. 查看sdb1是否存在RAID列表中

```
#mdadm  -D  /dev/md5
```

磁盘阵列-RAID

故障模拟：

3.将新设备添加到阵列

1)清空/dev/sdb1的superblcok标记(修复)

#mdadm --misc --zero-superblock -f /dev/sdb1

2) 把新的/dev/sdb1添加至RAID列表中

mdadm /dev/md5 -a /dev/sdb1

*如果想把/dev/sdb1添加上去，需要先删除/dev/sdb1.

磁盘阵列-RAID

阵列常用维护命令：

1.启动阵列

mdadm -As /dev/md5

该命令指启动/dev/md0阵列，其中“-A”指装载一个已存在的阵列；“-s”指查找mdadm.conf文件中的配置信息，并以此为依据启动阵列。

#mdadm -As

该命令指启动mdadm.conf文件中的所有阵列。

#mdadm -A /dev/md5 /dev/sd{b,c,d,e}1

如果没有创建mdadm.conf文件则可以采用上面这种启动方式。事实上，自RHEL5.4以后，即使停掉RAID，在缺少mdadm.conf文件的情况下，系统重新启动时也会自动激活RAID。

磁盘阵列-RAID

阵列常用维护命令：

2、停止(删除)阵列

umount /挂载点

mdadm -S /dev/md5

3、显示指定阵列的详细信息

mdadm -D /dev/md5

磁盘阵列-RAID

阵列常用维护命令：

4. 修改/etc/mdadm.conf、/etc/fstab等配置文件，把相关的地方去掉

5. 最后，用fdisk对磁盘进行重新分区即可。

磁盘阵列-RAID

监控RAID

mdadm的监控模式(mdmonitor)提供一些实用的功能，可以使用下列命令来监控/dev/md5，delay参数指时间间隔(秒)，紧急事件和严重的错误会及时发送给系统管理员

```
# mdadm --monitor --mail=snow@localhost --delay=300 /dev/md5
```

或

```
# nohup mdadm  --monitor --mail=snow@localhost --delay=300 /dev/md0 &
```

可添加到配置文件中以方便未来重启备用

```
#echo "MAILADDR  youremailaddress@example.com" >> /etc/mdadm.conf
#mdadm -As /dev/md5
```