<http://molinux.blog.51cto.com/2536040/516008>

**http://blog.51cto.com/image/skin/artType01.jpgLinux阵列 RAID详解**

2011-03-15 17:06:07

标签：[linux](http://blog.51cto.com/tag-linux.html) [raid mdadm mdadm.conf 优化](http://blog.51cto.com/tag-raid%20%20mdadm%20%20mdadm.conf%20优化.html) [休闲](http://blog.51cto.com/tag-休闲.html) [职场](http://blog.51cto.com/tag-职场.html)

原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](http://molinux.blog.51cto.com/2536040/516008) 、作者信息和本声明。否则将追究法律责任。<http://molinux.blog.51cto.com/2536040/516008>

主要内容：（笔记总结--会不定时补充、修改，如有错误，欢迎指正探讨）

**一、 RAID详解  
  二、 mdadm工具介绍  
  三、 创建一个RAID的基本过程  
  四、 磁盘阵列的管理  
  五、 RAID优化**

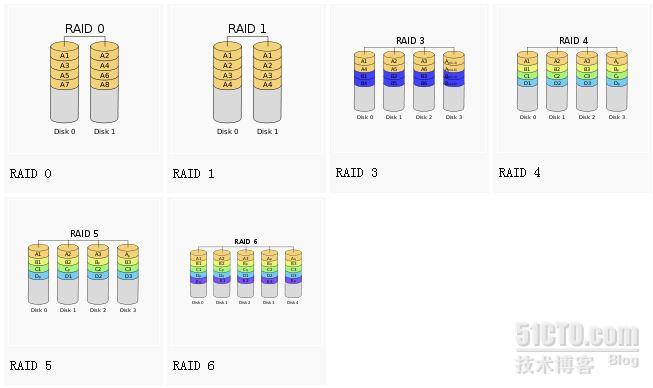
**RAID详解：**

**描述：**RAID：（Redundant Array of indenpensive Disk） 独立磁盘冗余阵列: 磁盘阵列是把多个磁盘组成一个阵列,当作单一磁盘使用,它将数据以分段(striping)的方式储存在不同的磁盘中,存取数据时,阵列中的相关磁盘一起动作,大幅减低数据的存取时间,同时有更佳的空间利用率。磁盘阵列利用的不同的技术,称为RAID level,不同的level针对不同的系统及应用,以解决数据安全的问题。简单来说，RAID把多个硬盘组合成为一个逻辑扇区，因此，操作系统只会把它当作一个硬盘。  
      
  一般高性能的磁盘阵列都是以硬件的形式来达成,进一步的把磁盘存取控制及磁盘阵列结合在一个控制器(RAID controler)或控制卡上,针对不同的用户解决人们对磁盘输输入输出系统的四大要求:   
  (1)增加存取速度,   
  (2)容错(fault tolerance),即安全性   
  (3)有效的利用磁盘利用率;   
  (4)尽量的平衡CPU,内存及磁盘的性能差异,提高主机的整体工作性能。  
  
**可提供的功能：**

   1.冗余(容错)

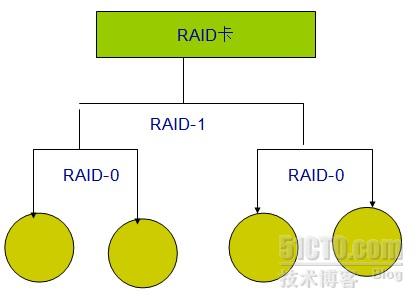
   2.性能提升  
 **RAID分类:**     
 1. 硬件RAID：用RAId接口卡来实现；需要内核支持其驱动，并且该类设备设备显示为SCSI设备，代号为/dev/sd\*  
 2. 软件RAID：用内核中的MD(multiple devices)模块实现，该类设备在/etc/下表示为：md\*；在现在的RH 5版本中使用mdadm工具管理软RAID；（虽然来说可以用软件模拟实现raid，但是相对对磁盘控制的功能及性能不如硬件实现的好，生产环境中最好使用硬件raid  
  
   
**几种常见RAID类型描述：**

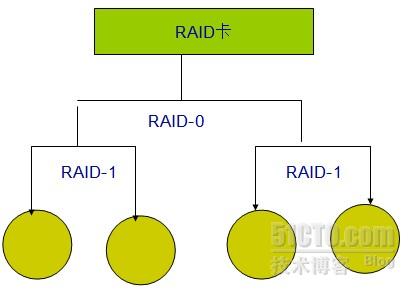
**图示：**

[](http://img1.51cto.com/attachment/201103/172332459.jpg)

**1. RAID-0** ：striping（条带模式），至少需要两块磁盘，做RAID分区的大小最好是相同的（可以充分发挥并优势）;而数据分散存储于不同的磁盘上，在读写的时候可以实现并发，所以相对其读写性能最好；但是没有容错功能，任何一个磁盘的损坏将损坏全部数据；  
**2. RAID-1** ：mirroring（镜像卷），至少需要两块硬盘，raid大小等于两个raid分区中最小的容量（最好将分区大小分为一样），可增加热备盘提供一定的备份能力；数据有冗余，在存储时同时写入两块硬盘，实现了数据备份；但相对降低了写入性能，但是读取数据时可以并发，几乎类似于raid-0的读取效率；  
**3. RAID-5** ：需要三块或以上硬盘，可以提供热备盘实现故障的恢复；采用奇偶效验，可靠性强，且只有同时损坏两块硬盘时数据才会完全损坏，只损坏一块硬盘时，系统会根据存储的奇偶校验位重建数据，临时提供服务；此时如果有热备盘，系统还会自动在热备盘上重建故障磁盘上的数据；  
    存储方式：简单来说就是，磁盘阵列的第一个磁盘分段是校验值,第二个磁盘至后一个磁盘再折回第一个磁盘的分段是数据,然后第二个磁盘的分段是校验值,从第三个磁盘再折回第二个磁盘的分段是数据,以此类推,直到放完数据为止。这样数据与校验值的循环分离存储就可以达到一定的故障重建功能；但是raid-5的控制较为复杂，且计算大量的校验码，可能给系统造成额外计算的负担(软raid来说，硬件有自己的数据处理能力)  
  
 **注：**RAID中的容错表示即使磁盘故障,数据仍能保持完整,可让系统存取到正确的数据,而SCSI的磁盘阵列更可在工作中抽换磁盘,并可自动重建故障磁盘的数据。  
   **热备份(hot spare or hot standby driver)**：为了加强容错的功能以及使系统在磁盘故障的情况下能迅速的重建数据,以维持系统的性能,一般的磁盘阵列系统都可使用热备份(hot spare or hot standby driver)的功能,所谓热备份是在建立(configure) 磁盘阵列系统的时候,将其中一磁盘指定为后备磁盘,该磁盘在平常并不操作,但若阵列中某一磁盘发生故障时,磁盘阵列即以后备磁盘取代故障磁盘,并自动将故障磁盘的数据重建(rebuild)在后备磁盘之上,因为反应快速,加上快取内存减少了磁盘的存取, 所以数据重建很快即可完成,对系统的性能影响很小。  
   在任何时候都不要用同一个硬盘上的多个区来做RAID,那样不仅不能提高系统的性能，反而会大大降低整体系统的系能；

   对上面几种常用的RAID类型分析后，可知，RAID-0主要可以提高磁盘性能，RAID-1主要可以实现备份容错（RAID-5由于控制机制复杂在此暂不使用），所以可以在实际生产环境下考虑结合RAID-0和RAID-1来实现磁盘存储系统的高性能、高容错。  
**下面图示的两种raid结合方式解析：**

[](http://img1.51cto.com/attachment/201103/170813606.jpg)

[](http://img1.51cto.com/attachment/201103/170813106.jpg)

          
    **对于一**：底层分别用有两块硬盘的raid0实现高并发，再集合两个raid0组实现冗余；下层的任意一个raid0组中有任意一块硬盘会使改组失效，但是两外一个组仍能提供全部数据；  
    **对于二**：底层用raid-1实现数冗余，上层用raid-2实现高并发，该种结构中任意一个硬盘的故障，不对本组整体数据工作构成破坏性影响；所以感觉该种方案更优异,在实际生产中有部分应用（具体配置过程再文章最后附）；  
  
     
  
  
**mdadm工具介绍：**  
  **描述：**  
    mdadm（multiple devices admin）是 linux下标准的的软raid管理工具，是一个模式化工具（在不同的模式下）；程序工作在内存用户程序区，为用户提供RAID接口来操作内核的模块，实现各种功能；  
RedHat已经内置了该工具；官方最新的版本应该是3.2，需要最新版本可以到官方网站下载或 /\* Style Definitions \*/ table.MsoNormalTable {mso-style-name:普通表格; mso-tstyle-rowband-size:0; mso-tstyle-colband-size:0; mso-style-noshow:yes; mso-style-parent:""; mso-padding-alt:0cm 5.4pt 0cm 5.4pt; mso-para-margin:0cm; mso-para-margin-bottom:.0001pt; mso-pagination:widow-orphan; font-size:10.0pt; font-family:"Times New Roman"; mso-fareast-font-family:"Times New Roman"; mso-ansi-language:#0400; mso-fareast-language:#0400; mso-bidi-language:#0400;} <http://www.kernel.org/pub/linux/utils/raid/mdadm/>下载源码包进行编译安装（官网一直打不开o(╯□╰)o）.

**实验环境：**RedHhat5.4 ；mdadm版本为v2.6.9 ；

   可能不同的版本选项等略有变动，使用时请注意对照自己版本的man文档；

**基本语法：**  
  # mdadm [mode] <raid-device> [options] <component-devices>  
  
**目前支持的模式：**  
  LINEAR(线性模式)、RAID0(striping条带模式)、RAID1(mirroring)、 RAID-4、RAID-5、 RAID-6、 RAID-10、 MULTIPATH和FAULTY  
  **LINEAR:**线性模式，该模式不是raid的标准模式，其主要作用是可以实现将几块小的硬盘组合为一块大硬盘来使用；数组存储时一次存满一个硬盘在使用下一个硬盘；对上层来说操作的是一个大硬盘  
  
**模式(7种)：**  
 **Assemble：**装配模式：加入一个以前定义的阵列；可以使挺值得阵列或从其他主机移出的阵列  
 **Build：**  创建：创建一个没有超级块的阵列  
   **Create：** 创建一个新的阵列，每个设备具有超级块  
**Follow or Monitor:** 监控RAID的状态，一般只对RAID-1/4/5/6/10等有冗余功能的模式来使用  
   **Grow：(Grow or shrink)** 改变RAID的容量或阵列中的设备数目；收缩一般指的是数据收缩或重建；  
 **Manage：** 管理阵列(如添加spare盘和删除故障盘)  
 **Incremental Assembly**：添加一个设备到一个适当的阵列。  
   **Misc：**  允许单独对阵列中的某个设备进行操作(如抹去superblocks 或停止阵列)  
   **Auto-detect：** 此模式不作用于特定的设备或阵列，而是要求在Linux内核启动任何自动检测到的阵列。  
  
**OPTIONS：**

**选择一个模式的选项：（Options for selecting a mode）**  
**-A, --assemble：** 加入并开启一个以前定义的阵列  
**-B, --build：**   创建一个没有超级块的阵列(Build a legacy array without superblocks.)  
**-C, --create：**  创建一个新的阵列  
**-F, --follow, --monitor：**选择监控(Monitor)模式  
**-G, --grow：**   改变激活阵列的大小或形态  
**-I, --incremental：** 添加一个单独的设备到合适的阵列，并可能启动阵列  
**--auto-detect：**   请求内核启动任何自动检测到的阵列

**不特定于一个模式的选项：（Options that are not mode-specific）**  
**-c, --config=：** 指定配置文件，缺省为 /etc/mdadm.conf  
**-s, --scan：**  扫描配置文件或 /proc/mdstat以搜寻丢失的信息。默认配置文件：/etc/mdadm.conf  
**-h, --help：**  帮助信息，用在以上选项后，则显示该选项信息  
**-v, --verbose：** 显示细节，一般只能跟 --detile 或 --examine一起使用，显示中级的信息；  
**-b, --brief：**  较少的细节。用于 --detail 和 --examine 选项  
**--help-options**： 显示更详细的帮助  
**-V, --version：** 版本信息  
**-q，--quit：**   安静模式；加上该选项能使mdadm不显示纯消息性的信息，除非那是一个重要的报告；  
  
**create build 或grow时使用的选项:**  
**-n, --raid-devices=:** 指定阵列中活动的device数目，不包括spare磁盘，这个数目只能由--grow修改  
**-x, --spare-devices=：**指定初始阵列的冗余device 数目即spare device数目；  
**-c, --chunk=:** Specify chunk size of kibibytes. 缺省为 64. chunk-size是一个重要的参数,决定了一次向阵列中每个磁盘写入数据的量

  （**Chunk :,**可以理解为raid分储数据时每个数据段的大小（通常为32/64/128等这类数字大小）；合理的选择chunk大小非常重要，若chunk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，使得数据的读写只局限于一块硬盘上，这便不能充分发挥RAID并发的优势；如果chunk设置过小，任何很小的I/O指令都 可能引发大量的读写操作，不能良好发挥并发性能，占用过多的控制器总线带宽，也影响了阵列的整体性能。所以，在创建带区时，我们应该根据实际应用的需要，合理的选择带区大小。）

**-z, --size=：**组建RAID1/4/5/6后从每个device获取的空间总数；但是大小必须为chunk的倍数，还需要在每个设备最后给RAID的superblock留至少128KB的大小。  
**--rounding=:** Specify rounding factor for linear array (==chunk size)  
**-l, --level=:** 设定 raid level.raid的几倍  
**--create：**   可用:linear, raid0, 0, stripe, raid1,1, mirror, raid4, 4, raid5, 5, raid6, 6, multipath, mp.  
**--build：**   可用：linear, raid0, 0, stripe.  
**-p, --layout=：**设定raid5 和raid10的奇偶校验规则；并且控制故障的故障模式；其中RAID-5的奇偶校验可以在设置为：:eft-asymmetric, left-symmetric, right-asymmetric, right-symmetric, la, ra, ls, rs.缺省为left-symmetric  
**--parity:**  类似于--layout=  
**--assume-clean:**目前仅用于 --build 选项  
**-R, --run:** 阵列中的某一部分出现在其他阵列或文件系统中时，mdadm会确认该阵列。此选项将不作确认。  
**-f, --force:** 通常mdadm不允许只用一个device 创建阵列，而且此时创建raid5时会使用一个device作为missing drive。此选项正相反  
**-N,--name=:** 设定阵列的名称

**管理模式选项（For Manage mode）:**  
**-a, --add：** 添加列出的设备到一个工作的阵列中；当阵列处于降级状态（故障状态），你添加一个设备，该设备将作为备用设备并且在该备用设备上开始数据重建。  
**-r, --remove：**从阵列中移除列出的设备，并且该设备不能处于活动状态（是冗余盘或故障盘）；  
**-f，--fail：**将列出的设备标记为faulty状态，标记后就可以移除设备；（可以作为故障恢复的测试手段）  
--set-faulty：同上

**监控模式选项（For Monitor mode）：**  
**-m, --mail：** 设置一个mail地址，在报警时给该mail发信；该地址可写入conf文件，在启动阵列是生效  
**-p, --program, --alert：**当检测到一个事件时运行一个指定的程序  
**-y, --syslog:** 设置所有的事件记录于syslog中  
**-t, --test：**  给启动时发现的每个阵列生成test警告信息；该信息传递给mail或报警程序；（以此来测试报警信息是否能正确接收）  
      
**MISC模式选项：**  
  Usage: mdadm options ...  devices ...  
**-Q, --query：** 查看一个device，判断它为一个 md device 或是 一个 md 阵列的一部分  
**-D, --detail：** 打印一个或多个md device 的详细信息  
**-E, --examine**：打印 device 上的 md superblock 的内容  
  
 **创建一个软RAID的基本过程：**

（以三个分区模拟创建一个raid5为操作示例，对于level0/1的How-To不再写出，如果理解原理，配置真的很简单；）  
**1.  生成raid组成：**  
  linux中阵列组成单元是分区，分区可以是整个设备也可以是设备多个分区中的一个；在fdisk分区后需要将分区标志改为Linux raid auto类型；

1. # 分区后如下：
2. Device  Boot      Start         End      Blocks   Id  System
3. /dev/sdb1               1         609     4891761   fd  Linux raid autodetect
4. /dev/sdc1               1         609     4891761   fd  Linux raid autodetect
5. /dev/sdd1               1         609     4891761   fd  Linux raid autodetect

**2.  建立磁盘阵列**

1. # mdadm -C /dev/md0 -a yes -l 5 -n 3 /dev/sd{b,c,d}1
2. mdadm: array /dev/md0 started.
4. -C ：创建一个阵列，后跟阵列名称
5. -l ：指定阵列的级别；
6. -n ：指定阵列中活动devices的数目

**3.  查看阵列状态**

1. [root@bogon ~]# cat /proc/mdstat
2. Personalities : [raid6] [raid5] [raid4]
3. md0 : active raid5 sdd1[2] sdc1[1] sdb1[0]
4. 9783296 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]
5. unused devices: <none>

  通过cat /proc/mdstat信息查看所有运行的RAID阵列的状态，在第一行中首先是MD的设备名md0，active和inactive选项表示阵列是否能读/写，接着是阵列的RAID级别raid5，后面是属于阵列的块设备，方括号[]里的数字表示设备在阵列中的序号，(S)表示其是热备盘，(F)表示这个磁盘是 faulty状态。下一行中首先是阵列的大小，用块数来表示；后面有chunk-size的大小，然后是layout类型，不同RAID级别的 layout类型不同，[3/3] [UUU]表示阵列有3个磁盘并且3个磁盘都是正常运行的，而[2/3]和[\_UU] 表示阵列有3个磁盘中2个是正常运行的，下划线对应的那个位置的磁盘是faulty状态的。

1. 查看阵列的详细信息：
2. [root@bogon ~]# mdadm --detail /dev/md0
3. /dev/md0:
4. Version : 0.90
5. Creation Time : Tue Mar 15 08:17:52 2011
6. Raid Level : raid5
7. Array Size : 9783296 (9.33 GiB 10.02 GB)
8. Used Dev Size : 4891648 (4.67 GiB 5.01 GB)
9. Raid Devices : 3
10. Total Devices : 3
11. Preferred Minor : 0
12. Persistence : Superblock is persistent
14. Update Time : Tue Mar 15 08:20:25 2011
15. State : clean
16. Active Devices : 3
17. Working Devices : 3
18. Failed Devices : 0
19. Spare Devices : 0
21. Layout : left-symmetric  校验规则
22. Chunk Size : 64K
24. UUID : e0d929d1:69d7aacd:5ffcdf9b:c1aaf02d
25. Events : 0.2
27. Number   Major   Minor   RaidDevice State
28. 0       8       17        0      active sync   /dev/sdb1
29. 1       8       33        1      active sync   /dev/sdc1
30. 2       8       49        2      active sync   /dev/sdd1

**4. mdadm.conf配置：**  
   mdadm.conf是该软件的默认配置文件，主要作用是方便跟踪软RAID的配置，尤其是可以配置监视和事件上报选项。其中写入每一个阵列组成的详细情况，用于在下次开启阵列后依据此文件重新装配（assemble）开启阵列，否则就需要在开启时手动定义阵列中的成员；当然是推荐创建该文件，防止出现意外情况，具体的详细配置及示例可以参看man文档# man mdadm.conf

1. [root@bogon ~]# echo "DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 " >> /etc/mdadm.conf
2. [root@bogon ~]# mdadm -Ds >> /etc/mdadm.conf
3. [root@bogon ~]# echo "MAILADDR mospiral@gmail.com" >> /etc/mdadm.conf
4. MAILADDR指定出问题时监控系统发邮件的地址
6. # 格式如下：
7. DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1
8. ARRAY /dev/md0 level=raid5 num-devices=3 metadata=0.90 UUID=e0d929d1:69d7aacd:5ffcdf9b:c1aaf02d
9. MAILADDR mospiral@gmail.com
10. #DEVICE行指明：依据该配置文件开启阵列时，去查找那些设备的超级快信息；若没有该行，
11. 就去搜索mtab中所有设备分区的超级快信息；所以改行可以不写，但是只要写上，以后添加spare
12. 设备时就需要同时修改改行信息；
13. #ARRAY 行指明raid的名称，级别uuid等基本信息
14. #可以添加诸如MAILADDR及PROGRAM等指定monitor状态下的监控报警信息；

**磁盘阵列的管理：**  
  可以在manage模式下对磁盘进行各种管理工作；

**给raid-5新增一个spare盘：**

1. [root@bogon ~]# mdadm -a /dev/md0 /dev/sda5
2. mdadm: added /dev/sda5
4. 此时查看状态：
5. [root@bogon ~]# cat /proc/mdstat
6. Personalities : [raid6] [raid5] [raid4]
7. md0 : active raid5 sda5[3](S) sdd1[2] sdc1[1] sdb1[0]
8. 9783296 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]
10. unused devices: <none>

**模拟硬盘故障：**

|  |
| --- |
|  |

1. [root@bogon ~]# mdadm -f /dev/md0  /dev/sdd1
2. mdadm: set /dev/sdd1 faulty in /dev/md0
4. # 此时查看状态，发现概念刚才的热备盘已经顶替了故障盘的位置，
5. # 并且进度条显示数据重建过程：
6. [root@bogon ~]# cat /proc/mdstat
7. Personalities : [raid6] [raid5] [raid4]
8. md0 : active raid5 sda5[3] sdd1[4](F) sdc1[1] sdb1[0]
9. 9783296 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU\_]
10. [>....................]  recovery =  1.3% (66912/4891648) finish=6.0min speed=13382K/sec
12. unused devices: <none>

**热移除故障的硬盘：**

1. [root@bogon ~]# mdadm -r /dev/md0  /dev/sdd1
2. mdadm: hot removed /dev/sdd1
3. [root@bogon ~]# cat /proc/mdstat
4. Personalities : [raid6] [raid5] [raid4]
5. md0 : active raid5 sda5[3] sdc1[1] sdb1[0]
6. 9783296 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU\_]
7. [===>.................]  recovery = 16.1% (792136/4891648) finish=4.6min speed=14828K/sec
9. unused devices: <none>

  对于有冗余的raid形式，在单一磁盘故障时一般能在一段时间内重建数据；但是数据量非常大时，重建会非常缓慢，且重建过程系统压力比较大，此时需要多关注系统负载，防止重建过程出现错误；在热移除故障盘一户，也需要尽快的换上新硬盘，并且添加spare盘；在故障修复，重建重建之后，需要重新生成配置文件，防止在下次开启时，按照最初的定义模式开启；

**停止RAID：**

1. # 指定停止某个阵列
2. [root@bogon ~]# mdadm -S /dev/md0
4. # 停止配置文件中定义的所有阵列
5. [root@bogon ~]# mdadm -Ss
6. mdadm: stopped /dev/md0
7. # -s –scan去查询配置文件或没有配置文件时查询mdstat中的所有阵列

**开启RAID：**

1. [root@bogon ~]# mdadm -As
2. mdadm: /dev/md0 has been started with 2 drives (out of 3).
3. # -s –scan：根据配置文件开启所有的阵列，此时由于故障恢复过，
4. # 但是由于未重建配置文件，阵列我不能加载上新添加的硬盘；（需要停掉，重新手动指定）
6. [root@bogon ~]# cat /proc/mdstat
7. Personalities : [raid6] [raid5] [raid4]
8. md0 : active raid5 sdb1[0] sdc1[1]
9. 9783296 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU\_]
10. unused devices: <none>
12. 若此时没有配置文件，就需要手动指定设备名称：
13. [root@bogon ~]# mdadm -A  /dev/md0 /dev/sdb1 /dev/sdc1  /dev/sda5
14. mdadm: /dev/md0 has been started with 3 drives.
15. # 注：将一个raid设备添加入md阵列后，md的信息会写入到该设备分区的superblock中；
16. # 在手动装配时；mdadm工具会自动验证阵列配置是否合法，并且做出相应的动作；

  若新接手一个raid，没有配置文件，或忘记设备的准确组成，就需要按分区逐个检查是否是raid设备及其他信息，然后根据信息装配阵列：

1. [root@bogon ~]# mdadm -E /dev/sdb1
2. /dev/sdb1:
3. Magic : a92b4efc
4. Version : 0.90.00
5. UUID : e0d929d1:69d7aacd:5ffcdf9b:c1aaf02d
6. Creation Time : Tue Mar 15 08:17:52 2011
7. Raid Level : raid5
8. Used Dev Size : 4891648 (4.67 GiB 5.01 GB)
9. Array Size : 9783296 (9.33 GiB 10.02 GB)
10. Raid Devices : 3
11. Total Devices : 3
12. Preferred Minor : 0
14. Update Time : Tue Mar 15 09:25:10 2011
15. State : clean
16. Active Devices : 3
17. Working Devices : 3
18. Failed Devices : 0
19. Spare Devices : 0
20. Checksum : b0cd088f - correct
21. Events : 8
23. Layout : left-symmetric
24. Chunk Size : 64K
26. Number   Major   Minor   RaidDevice State
27. this     0       8       17        0      active sync   /dev/sdb1
29. 0     0       8       17        0      active sync   /dev/sdb1
30. 1     1       8       33        1      active sync   /dev/sdc1
31. 2     2       8        5        2      active sync   /dev/sda5
32. # 该处显示出的是该分区superblock中包含的md信息；没有配置文件时，可以依据该信息装配md；

**删除阵列：**

**若需要彻底清除这个阵列：**

1. [root@bogon ~]# umount /dev/md0
2. mdadm -Ss /dev/md0
3. [root@bogon ~]# mdadm --zero-superblock /dev/sd{b,c,d}1
4. # --zero-superblock 加上该选项时，会判断如果该阵列是否包
5. # 含一个有效的阵列超级快，若有则将该超级块中阵列信息抹除。
6. [root@bogon ~]# rm /etc/mdadm.conf

RAID优化：

**（1） 设置stride值**

 The stride is the software RAID device's chunk-size in filesystem blocks.For example,with an ext3 filesystem that will have an 4KB block size on a RAID device with a chunk-size of 64KB, the stride should be set to 16:（翻译的很纠结，就贴上教材原文了。）

1. mk2fs -j -b 4096 -E stride=16 /dev/md0
2. # 设置时，需要用-E选项进行扩展

 设定良好的stride值，可以在后期使用时，减少写入数据时对数据块计算的负担，从而提高RAID性能；

附：**RAID 1-0双层架构的方法：**

**首先创建两个底层RAID-1**

1. [root@bogon ~]# mdadm -C /dev/md0 -a yes -l 1 -n 2 /dev/sd[bc]1
2. mdadm: array /dev/md0 started.
3. [root@bogon ~]# mdadm -C /dev/md1 -a yes -l 1 -n 2 /dev/sd[bc]2
5. mdadm: array /dev/md1 started.

**用两个RAID-1实现上层RAID-0：**

1. [root@bogon ~]# mdadm -C /dev/md2 -a yes -l 0 -n 2 /dev/md[01]
2. mdadm: array /dev/md2 started.

**查看阵列状态：**

1. [root@bogon ~]# cat /proc/mdstat
2. Personalities : [raid6] [raid5] [raid4] [raid0] [raid1]
3. md2 : active raid0 md0[0] md1[1]
4. 9783232 blocks 64k chunks
6. md1 : active raid1 sdb2[0] sdc2[1]
7. 4891712 blocks [2/2] [UU]
9. md0 : active raid1 sdb1[0] sdc1[1]
10. 4891648 blocks [2/2] [UU]
12. unused devices: <none>

**创建配置文件：**

1. [root@bogon ~]# mdadm -Ds > /etc/mdadm.conf

**停止与开启阵列：**

1. [root@bogon ~]# mdadm -Ss
2. mdadm: stopped /dev/md2
3. mdadm: stopped /dev/md1
4. mdadm: stopped /dev/md0
5. [root@bogon ~]# mdadm -As
6. mdadm: /dev/md0 has been started with 2 drives.
7. mdadm: /dev/md1 has been started with 2 drives.
8. mdadm: /dev/md2 has been started with 2 drives.
9. ##上述关闭与开启过程，系统能只能识别层级，关闭先关闭上层，
10. ##后开启上层；防止了冲突；

  /\* Style Definitions \*/ table.MsoNormalTable {mso-style-name:普通表格; mso-tstyle-rowband-size:0; mso-tstyle-colband-size:0; mso-style-noshow:yes; mso-style-parent:""; mso-padding-alt:0cm 5.4pt 0cm 5.4pt; mso-para-margin:0cm; mso-para-margin-bottom:.0001pt; mso-pagination:widow-orphan; font-size:10.0pt; font-family:"Times New Roman"; mso-fareast-font-family:"Times New Roman"; mso-ansi-language:#0400; mso-fareast-language:#0400; mso-bidi-language:#0400;}

参考：

  软RAID0的技术概要及实现 : <http://www.linuxsir.org/main/node/279>

  维基百科--RAID :  <http://zh.wikipedia.org/zh-cn/Raid>

<http://www.linuxidc.com/Linux/2009-12/23394p6.htm>

本文出自 “[Mo](http://molinux.blog.51cto.com/)” 博客，请务必保留此出处<http://molinux.blog.51cto.com/2536040/516008>