mdadm

На этой странице рассматриваются вопросы создания и обслуживания программного RAID-массива в операционной системе Linux.

XG-SCALE

Содержание

[убрать]

[править] mdadm

Управление программным RAID-массивом в Linux выполняется с помощью программы **mdadm**.

У программы **mdadm** есть несколько режимов работы.

Assemble (сборка)
: Собрать компоненты ранее созданного массива в массив. Компоненты можно указывать явно, но можно и не указывать — тогда выполняется их поиск по суперблокам.

Build (построение)
: Собрать массив из компонентов, у которых нет суперблоков. Не выполняются никакие проверки, создание и сборка массива в принципе ничем не отличаются.

Create (создание)
: Создать новый массив на основе указанных устройств. Использовать суперблоки размещённые на каждом устройстве.

Monitor (наблюдение)
: Следить за изменением состояния устройств. Для RAID0 этот режим не имеет смысла.

Grow (расширение или уменьшение)
: Расширение или уменьшение массива, включаются или удаляются новые диски.

Incremental Assembly (инкрементальная сборка)
: Добавление диска в массив.

Manage (управление)
: Разнообразные операции по управлению массивом, такие как замена диска и пометка как сбойного.

Misc (разное)
: Действия, которые не относятся ни к одному из перечисленных выше режимов работы.

Auto-detect (автооообнаружение)
: Активация автоматически обнаруживаемых массивов в ядре Linux.

Формат вызова

```
mdadm [mode] [array] [options]
```

Режимы:

- `-A`, `--assemble` — режим сборки
- `-B`, `--build` — режим построения
- `-C`, `--create` — режим создания
- `-F`, `--follow`, `--monitor` — режим наблюдения
- `-G`, `--grow` — режим расширения
- `-I`, `--incremental` — режим инкрементальной сборки

[править] Настройка программного RAID-массива

Рассмотрим как выполнить настройку RAID-массива 5 уровня на трёх дисковых разделах. Мы будем использовать разделы:

```
/dev/hde1
/dev/hdf2
/dev/hdg1
```

В том случае если разделы иные, не забудьте использовать соответствующие имена файлов.

[править] Создание разделов

Нужно определить на каких физических разделах будет создаваться RAID-массив. Если разделы уже есть, нужно найти свободные (*fdisk -l*). Если разделов ещё нет, но есть неразмеченное место, их можно создать с помощью программ *fdisk* или *cfdisk*.

Просмотреть какие есть разделы:

```
%# fdisk -l

Disk /dev/hda: 12.0 GB, 12072517632 bytes
255 heads, 63 sectors/track, 1467 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1        13    104391   83  Linux
/dev/hda2            14       144   1052257+  83  Linux
/dev/hda3           145       209    522112+  82  Linux swap
/dev/hda4           210      1467  10104885    5  Extended
/dev/hda5           210       655   3582463+  83  Linux
...
...
/dev/hda15         1455      1467    104391   83  Linux
```

Просмотреть, какие разделы куда смонтированы, и сколько свободного места есть на них (размеры в килобайтах):

```
%# df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/hda2              1035692    163916    819164  17% /
/dev/hda1               101086      8357     87510   9% /boot
/dev/hda15              101086      4127     91740   5% /data1
...
...
...
/dev/hda7              5336664    464228   4601344  10% /var
```

[править] Размонтирование

Если вы будете использовать созданные ранее разделы, обязательно размонтируйте их. RAID-массив нельзя создавать поверх разделов, на которых находятся смонтированные файловые системы.

```
%# umount /dev/hde1
%# umount /dev/hdf2
%# umount /dev/hdg1
```

[править] Изменение типа разделов

Желательно (но не обязательно) изменить тип разделов, которые будут входить в RAID-массив и установить его равным FD (Linux RAID autodetect). Изменить тип раздела можно с помощью fdisk.

Рассмотрим как это делать на примере раздела `/dev/hde1`.

```
%# fdisk /dev/hde
The number of cylinders for this disk is set to 8355.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
```

```
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):
```

Use FDISK Help

Now use the fdisk m command to get some help:

```
Command (m for help): m
...
...
p   print the partition table
q   quit without saving changes
s   create a new empty Sun disklabel
t   change a partition's system id
...
...
Command (m for help):
```

Set The ID Type To FD

```
Partition /dev/hde1 is the first partition on disk /dev/hde.
Modify its type using the t command, and specify the partition number
and type code.
You also should use the L command to get a full listing
of ID types in case you forget.

Command (m for help): t
Partition number (1-5): 1
Hex code (type L to list codes): L

...
...
...
16  Hidden FAT16    61   SpeedStor       f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  fe  LANstep
1b  Hidden Win95 FA 65  Novell Netware  ff  BBT
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help):
```

Make Sure The Change Occurred

Use the p command to get the new proposed partition table:

```
Command (m for help): p

Disk /dev/hde: 4311 MB, 4311982080 bytes
16 heads, 63 sectors/track, 8355 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

Device Boot    Start       End    Blocks   Id  System
/dev/hde1             1      4088   2060320+  fd  Linux raid autodetect
/dev/hde2          4089      5713    819000   83  Linux
/dev/hde4          6608      8355    880992    5  Extended
/dev/hde5          6608      7500    450040+  83  Linux
/dev/hde6          7501      8355    430888+  83  Linux

Command (m for help):
```

```
    Save The Changes

    Use the w command to permanently save the changes to disk /dev/hde:

    Command (m for help): w
    The partition table has been altered!

    Calling ioctl() to re-read partition table.

    WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
    The kernel still uses the old table.
    The new table will be used at the next reboot.
    Syncing disks.
```

Аналогичным образом нужно изменить тип раздела для всех остальных разделов, входящих в RAID-массив.

[править] Создание RAID-массива

Создание RAID-массива выполняется с помощью программы **mdadm** (ключ `--create`). Мы воспользуемся опцией `--level`, для того чтобы создать RAID-массив 5 уровня. С помощью ключа `--raid-devices` укажем устройства, поверх которых будет собираться RAID-массив.

```
    %# mdadm --create --verbose /dev/md0 --level=5  --raid-devices=3
/dev/hde1 /dev/hdf2 /dev/hdg1

    mdadm: layout defaults to left-symmetric
    mdadm: chunk size defaults to 64K
    mdadm: /dev/hde1 appears to contain an ext2fs file system
        size=48160K  mtime=Sat Jan 27 23:11:39 2007
    mdadm: /dev/hdf2 appears to contain an ext2fs file system
        size=48160K  mtime=Sat Jan 27 23:11:39 2007
    mdadm: /dev/hdg1 appears to contain an ext2fs file system
        size=48160K  mtime=Sat Jan 27 23:11:39 2007
    mdadm: size set to 48064K
    Continue creating array? y
    mdadm: array /dev/md0 started.
```

Если вы хотите сразу создать массив, где диска не хватает (degraded) просто укажите слово `missing` вместо имени устройства. Для RAID5 это может быть только один диск; для RAID6 — не более двух; для RAID1 — сколько угодно, но должен быть как минимум один рабочий.

[править] Проверка правильности сборки

Убедиться, что RAID-массив проинициализирован корректно можно просмотрев файл `/proc/mdstat`. В этом файле отражается текущее состояние RAID-массива.

```
    %# cat /proc/mdstat
    Personalities : [raid5]
    read_ahead 1024 sectors
    md0 : active raid5 hdg1[2] hde1[1] hdf2[0]
        4120448 blocks level 5, 32k chunk, algorithm 3 [3/3] [UUU]
```

```
unused devices: <none>
```

Обратите внимание на то, как называется новый RAID-массив. В нашем случае он называется `/dev/md0`. Мы будем обращаться к массиву по этому имени.

[править] Создание файловой системы поверх RAID-массива

Новый RAID-раздел нужно отформатировать, т.е. создать на нём файловую систему. Сделать это можно при помощи программы из семейства **mkfs**. Если мы будем создавать файловую систему ext3, воспользуемся программой **mkfs.ext3**. :

```
%# mkfs.ext3 /dev/md0
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
36144 inodes, 144192 blocks
7209 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
18 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Имеет смысл для лучшей производительности файловой системы указывать при создании количество дисков в рейде и количество блоков файловой системы которое может поместиться в один страйп (chunk), это особенно важно при создании массивов уровня RAID0,RAID5,RAID6,RAID10. Для RAID1 (mirror) это не имеет значения так как запись идет всегда на один device, а в других типах рейдов дата записывается последовательно на разные диски порциями соответствующими размеру stripe.Например если мы используем RAID5 из 3 дисков, с дефолтным размером страйпа в 64К и используем файловую систему ext3 с размером блока в 4К то можно вызывать команду mkfs.ext вот так:

```
%# mkfs.ext3 -b 4096 -E stride=16,stripe-width=32 /dev/md0
```

stripe-width обычно расчитывается как stride * N (N это дата диски в массиве - например в RAID5 - два дата диска и один parity) Для неменее популярной файловой системы XFS надо указывать не количество блоков файловой системы соответствующих размеру stripe в массиве, а непосредственно размер самого страйпа

```
%# mkfs.xfs -d su=64k,sw=3 /dev/md0
```

[править] Создание конфигурационного файла mdadm.conf

Система сама не запоминает какие RAID-массивы ей нужно создать и какие компоненты в них входят. Эта информация находится в файле `mdadm.conf`.

Строки, которые следует добавить в этот файл, можно получить при помощи команды

```
mdadm --detail --scan --verbose
```

Вот пример её использования:

```
%# mdadm --detail --scan --verbose
ARRAY /dev/md0 level=raid5 num-devices=4
UUID=77b695c4:32e5dd46:63dd7d16:17696e09
devices=/dev/hde1,/dev/hdf2,/dev/hdg1
```

Если файла `mdadm.conf` ещё нет, можно его создать:

```
%# echo "DEVICE partitions" > /etc/mdadm/mdadm.conf
%# mdadm --detail --scan --verbose | awk '/ARRAY/ {print}' >>
/etc/mdadm/mdadm.conf
```

[править] Создание точки монтирования для RAID-массива

Поскольку мы создали новую файловую систему, вероятно, нам понадобится и новая точка монтирования. Назовём её `/raid`.

```
%# mkdir /raid
```

[править] Изменение /etc/fstab

Для того чтобы файловая система, созданная на новом RAID-массиве автоматически монтировалась при загрузке, добавим соответствующую запись в файл `/etc/fstab` хранящий список автоматически монтируемых при загрузке файловых систем.

```
/dev/md0      /raid     ext3    defaults    1 2
```

Если мы объединяли в RAID-массив разделы, которые использовались раньше, нужно отключить их монтирование: удалить или закомментировать соответствующие строки в файле `/etc/fstab`. Закомментировать строку можно символом #.

```
#/dev/hde1       /data1        ext3    defaults        1 2
#/dev/hdf2       /data2        ext3    defaults        1 2
#/dev/hdg1       /data3        ext3    defaults        1 2
```

[править] Монтирование файловой системы нового RAID-массива

Для того чтобы получить доступ к файловой системе, расположенной на новом RAID-массиве, её нужно смонтировать. Монтирование выполняется с помощью команды **mount**.

Если новая файловая система добавлена в файл `/etc/fstab`, можно смонтировать её командой **mount** `-a` (смонтируются все файловые системы, которые должны монтироваться при загрузке, но сейчас не смонтированы).

```
%# mount -a
```

Можно смонтировать только нужный нам раздел (при условии, что он указан в `/etc/fstab`).

```
%# mount /raid
```

Если раздел в `/etc/fstab` не указан, то при монтировании мы должны задавать как минимум два параметра — точку монтирования и монтируемое устройство:

```
%# mount /dev/md0 /raid
```

[править] Проверка состояния RAID-массива

Информация о состоянии RAID-массива находится в файле `/proc/mdstat`.

```
%# raidstart /dev/md0
%# cat /proc/mdstat
Personalities : [raid5]
read_ahead 1024 sectors
md0 : active raid5 hdg1[2] hde1[1] hdf2[0]
    4120448 blocks level 5, 32k chunk, algorithm 3 [3/3] [UUU]

unused devices: <none>
```

Если в файле информация постоянно изменяется, например, идёт пересборка массива, то постоянно изменяющийся файл удобно просматривать при помощи программы **watch**:

```
%$ watch cat /proc/mdstat
```

Как выполнить проверку целостности программного RAID-массива md0:

```
echo 'check' >/sys/block/md0/md/sync_action
```

Как посмотреть нашлись ли какие-то ошибки в процессе проверки программного RAID-массива по команде check или repair:

```
cat /sys/block/md0/md/mismatch_cnt
```

[править] Проблема загрузки на многодисковых системах

В некоторых руководствах по **mdadm** после первоначальной сборки массивов рекомендуется добавлять в файл */etc/mdadm/mdadm.conf* вывод команды "mdadm --detail --scan --verbose":

```
ARRAY /dev/md/1 level=raid1 num-devices=2 metadata=1.2 name=linuxWork:1
UUID=147c5847:dabfe069:79d27a05:96ea160b
  devices=/dev/sda1
ARRAY /dev/md/2 level=raid1 num-devices=2 metadata=1.2 name=linuxWork:2
UUID=68a95a22:de7f7cab:ee2f13a9:19db7dad
  devices=/dev/sda2
```

, в котором жёстко прописаны имена разделов (**/dev/sda1, /dev/sda2** в приведённом примере).

Если после этого обновить образ начальной загрузки (в Debian вызвать 'update-initramfs -u' или 'dpkg-reconfigure mdadm'), имена разделов запишутся в файл mdadm.conf образа начальной загрузки и вы не сможете загрузиться с массива, если конфигурация жёстких дисков изменится (нужные разделы получат другие имена). Для этого не обязательно добавлять или убирать жёсткие диски: в многодисковых системах их имена могут меняться от загрузки к загрузке.

Решение: записывать в /etc/mdadm/mdadm.conf вывод команды "mdadm --detail --scan" (без --verbose).

При этом в файле mdadm.conf будут присутствовать UUID разделов, составляющих каждый RAID-массив. При загрузке системы mdadm находит нужные разделы независимо от их символических имён по UUID.

mdadm.conf, извлечённый из образа начальной загрузки Debian:

```
DEVICE partitions
HOMEHOST <system>
ARRAY /dev/md/1 metadata=1.2 UUID=147c5847:dabfe069:79d27a05:96ea160b
name=linuxWork:1
ARRAY /dev/md/2 metadata=1.2 UUID=68a95a22:de7f7cab:ee2f13a9:19db7dad
name=linuxWork:2
```

Результат исследования раздела командой 'mdadm --examine'"

```
/dev/sda1:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x0
     Array UUID : 147c5847:dabfe069:79d27a05:96ea160b
           Name : linuxWork:1
  Creation Time : Thu May 23 09:17:01 2013
     Raid Level : raid1
   Raid Devices : 2
```

Раздел с UUID **147c5847:dabfe069:79d27a05:96ea160b** войдёт в состав массива, даже если станет /dev/sdb1 при очередной загрузке системы.

Вообще, существует 2 файла mdadm.conf, влияющих на автоматическую сборку массивов:

- один при загрузке системы, записывется в образ начальной загрузки при его обновлении;
- другой находится в каталоге /etc/mdadm/ и влияет на автосборку массивов внутри работающей системы.

Соответственно, вы можете иметь информацию:

1) в образе начальной загрузки (ОНЗ) и в /etc/mdadm/mdadm.conf;

2) только в ОНЗ (попадает туда при его создании обновлении);

3) только в /etc/mdadm/mdadm.conf;

4) нигде.

В том месте, где есть mdadm.conf, сборка происходит по правилам; где нет - непредсказуемо.

Примечание: если вы не обновили ОНЗ после создания RAID-массивов, их конфигурация всё равно в него попадёт - при обновлении образа другой программой / при обновлении системы (но вы не будете об этом знать со всеми вытекающими).

[править] Дальнейшая работа с массивом

[править] Пометка диска как сбойного

Диск в массиве можно условно сделать сбойным, ключ `--fail` (`-f`):

```
%# mdadm /dev/md0 --fail /dev/hde1
%# mdadm /dev/md0 -f     /dev/hde1
```

[править] Удаление сбойного диска

Сбойный диск можно удалить с помощью ключа `--remove` (`-r`):

```
%# mdadm /dev/md0 --remove /dev/hde1
%# mdadm /dev/md0 -r       /dev/hde1
```

[править] Добавление нового диска

Добавить новый диск в массив можно с помощью ключей `--add` (`-a`) и `--re-add`:

```
%# mdadm /dev/md0 --add /dev/hde1
%# mdadm /dev/md0 -a    /dev/hde1
```

[править] Сборка существующего массива

Собрать существующий массив можно с помощью **mdadm** `--assemble`. Как дополнительный аргумент указывается, нужно ли выполнять сканирование устройств, и если нет, то какие устройства нужно собирать.

```
%# mdadm --assemble /dev/md0 /dev/hde1 /dev/hdf2 /dev/hdg1
%# mdadm --assemble --scan
```

[править] Расширение массива

Расширить массив можно с помощью ключа `--grow` (`-G`). Сначала добавляется диск, а потом массив расширяется:

```
%# mdadm /dev/md0 --add /dev/hdh2
```

Проверяем, что диск (раздел) добавился:

```
%# mdadm --detail /dev/md0
%# cat /proc/mdstat
```

Если раздел действительно добавился, мы можем расширить массив:

```
%# mdadm -G /dev/md0 --raid-devices=4
```

Опция --raid-devices указывает новое количество дисков используемое в массиве. Например, было 3 диска, а теперь расширяем до 4-х - указываем 4.

Рекомендуется задать файл бэкапа на случай прерывания перестроения массива, например добавить:

```
--backup-file=/var/backup
```

При необходимости, можно регулировать скорость процесса расширения массива, указав нужное значение в файлах

```
/proc/sys/dev/raid/speed_limit_min
/proc/sys/dev/raid/speed_limit_max
```

Убедитесь, что массив расширился:

```
%# cat /proc/mdstat
```

Нужно обновить конфигурационный файл с учётом сделанных изменений:

```
%# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
%# vi /etc/mdadm/mdadm.conf
```

[править] Возобновление отложенной синхронизации

Отложенная синхронизация:

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
[raid10]
md0 : active(auto-read-only) raid1 sda1[0] sdb1[1]
      78148096 blocks [2/2] [UU]
        resync=PENDING
```

Возобновить:

```
echo idle > /sys/block/md0/md/sync_action
```

P.S.: Если вы увидели «`active (auto-read-only)`» в файле `/proc/mdstat`, то возможно вы просто ничего не записывали в этот массив. К примеру, после монтирования раздела и любых изменений в примонтированном каталоге, статус автоматически меняется:

```
md0 : active raid1 sdc[0] sdd[1]
```

[править] Переименование массива

Для начала отмонтируйте и остановите массив:

```
%# umount /dev/md0
%# mdadm --stop /dev/md0
```

Затем необходимо пересобрать как md5 каждый из разделов sd[abcdefghijk]1

```
%# mdadm --assemble /dev/md5 /dev/sd[abcdefghijk]1 --update=name
```

или так

```
%# mdadm --assemble /dev/md5 /dev/sd[abcdefghijk]1 --update=super-minor
```

[править] Удаление массива

Для начала отмонтируйте и остановите массив:

```
%# umount /dev/md0
%# mdadm -S /dev/md0
```

Затем необходимо затереть superblock каждого из составляющих массива:

```
%# mdadm --zero-superblock /dev/hde1
%# mdadm --zero-superblock /dev/hdf2
```

Если действие выше не помогло, то затираем так:

```
%# dd if=/dev/zero of=/dev/hde1 bs=512 count=1
%# dd if=/dev/zero of=/dev/hdf2 bs=512 count=1
```

[править] Дополнительная информация

- man mdadm (англ.)
- man mdadm.conf (англ.)
- Linux Software RAID (англ.)

- Gentoo Install on Software RAID (англ.)
- HOWTO Migrate To RAID (англ.)
- Remote Conversion to Linux Software RAID-1 for Crazy Sysadmins HOWTO (англ.)
- Migrating To RAID1 Mirror on Sarge (англ.)
- Настройка программного RAID-1 на Debian Etch (рус.), а также обсуждение на ЛОРе
- Недокументированные фишки программного RAID в Linux (рус.)

[править] Производительность программных RAID-массивов

- Adventures With Linux RAID: Part 1, Part 2 (англ.)
- Параметры, влияющие на производительность программного RAID (рус.)

[править] Разные заметки, имеющие отношение к RAID

- BIO_RW_BARRIER - what it means for devices, filesystems, and dm/md. (англ.)