



vasyakrg 12 ноября 2016 в 15:47

Установка PROXMOX 4.3 на Soft-RAID 10 GPT

Системное администрирование, Настройка Linux

[Из песочницы](#)

Добрый день, друзья. Сегодня я бы хотел поделиться своим личным опытом по настройке Proxmox на soft-Raid 10.

Что имеем:

- Сервер HP ProLiant DL120 G6 (10 GB ОЗУ)
- 4x1000Gb SATA винчестера – без физического RAID контроллера на борту
- Флешка с PROXMOX 4.3 (об этом ниже)

Что хотим:

- Получить инсталляцию PROXMOX 4.3 установленную полностью на S-RAID 10 GPT, что бы при отказе любого диска система продолжала работу.
- Получить уведомление об отказе сбойного диска на почту.

Что делаем – общий план действий:

- Устанавливаем PROXMOX 4.3
- Поднимаем и тестируем RAID10
- Настраиваем уведомления на почту

Под катом поэтапное прохождение квеста.

А теперь поэтапно.

Первый момент:

Подключил флешку – если вкратце — не найден установочный диск. Не могу смонтироваться.

```
[ 0.853310] ERST: Can not request [mem 0xbf7ff000-0xbf7fff
Proxmox startup
mounting proc filesystem
mounting sys filesystem
comandline: BOOT_IMAGE=/boot/linux26 ro ramdisk_size=16777216
loading drivers: shpchp pata_acpi 8250_fintek mac_hid i2c_i80
modprobe: ERROR: could not insert 'intel_powerclamp': No such
searching for cdrom
testing again in 5 seconds
[ 3.440470] sd 4:0:0:0: [sde] No Caching mode page found
[ 3.440794] sd 4:0:0:0: [sde] Assuming drive cache: write t
testing cdrom /dev/sde
umount: can't umount /mnt: Invalid argument
testing again in 5 seconds
testing cdrom /dev/sde
umount: can't umount /mnt: Invalid argument
testing again in 5 seconds
testing cdrom /dev/sde
umount: can't umount /mnt: Invalid argument
testing again in 5 seconds
testing cdrom /dev/sde
umount: can't umount /mnt: Invalid argument
no cdrom found
unable to continue (type exit or CTRL-D to reboot)
/ # _
```

Не стал разбираться что да как, да почему. Записал образ на CD-диск и подключил USB CDROM (благо он был рядом)

Второй момент:

Подключил к серверу CDROM и клавиатуру в передние порты сервера (их у него два) – первое что увидел, на первом приветственном скрине прогтох нельзя ничего нажать без мышки. То есть преекключение Tab-ом по управляющим кнопкам не происходит. Т.к. сервер был в стойке и залазить сзади было проблематично, начал по очереди втыкать клавишу и мышку. Мышкой щелкаю «далее», клавишей — ввожу данные.

Установка состоит из нескольких шагов:

- Согласится с их требованиями
- Выбрать винчестер, куда система установится.
- Выбрать страну и часовой пояс
- Указать имя сервера, адресацию
- И собственно немного подождать развертки образа на сервер.

PROXMOX установлен на первый диск, который он обозвал как /dev/sda. Подключаюсь со своего ноутбука на адрес, который указал при установке:

```
root@pvel:~#ssh root@192.168.1.3
```

Обновляю систему:

```
root@pvel:~#apt-get update
```

[На выходе вижу](#)

Это не дело. Покупать пока лицензию на поддержку не планирую. Меняю официальную подписку на их «бесплатный» репозиторий.

```
root@pvel:~#nano /etc/apt/sources.list.d/pve-enterprise.list
```

Там вижу:

```
deb https://enterprise.proxmox.com/debian jessie pve-enterprise
```

Меняю на:

```
deb http://download.proxmox.com/debian jessie pve-no-subscription
```

И снова обновляюсь и ставлю обновки системы:

```
root@pvel:~#apt-get update && apt-get upgrade
```

Теперь всё обновилось без запинки и система в новейшем состоянии. Ставлю пакеты для работы с рейдом:

```
root@pvel:~#apt-get install -y mdadm initramfs-tools parted
```

Теперь определим точный размер первого диска, он нам пригодится в дальнейшем:

```
root@pvel:~#parted /dev/sda print
```

```
Model: ATA MB1000EBNCF (scsi)
Disk /dev/sda: 1000GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number   Start    End      Size    File system  Name      Flags
  1       1049kB   10.5MB   9437kB              primary    bios_grub
  2       10.5MB   1000MB   990MB    ext4          primary
  3       1000MB   1000GB   999GB              primary
```

Видим что ровно 1000GB – запомним. Размечаем остальные разделы под наш массив. Первым делом очищаем таблицу разделов на трех пустых дисках и размечаем диски под GPT:

```
root@pvel:~#dd if=/dev/zero of=/dev/sb[bcd] bs=512 count=1
```

```
1+0 records in
1+0 records out
```

```
512 bytes (512 B) copied, 7.8537e-05 s, 6.5 MB/s
```

Размечаем:

Второй:

```
root@pvel:~#parted /dev/sdb mklabel gpt
```

```
Warning: The existing disk label on /dev/sdb will be destroyed and all data on this disk will be lost. Do you want to continue ?
Yes/No? yes
Information: You may need to update /etc/fstab.
```

Третий:

```
root@pvel:~#parted /dev/sdc mklabel gpt
```

```
Warning: The existing disk label on /dev/sdc will be destroyed and all data on this disk will be lost. Do you want to continue ?
Yes/No? yes
Information: You may need to update /etc/fstab.
```

Четвертый:

```
root@pvel:~#parted /dev/sdd mklabel gpt
```

```
Warning: The existing disk label on /dev/sdd will be destroyed and all data on this disk will be lost. Do you want to continue ?
Yes/No? yes
Information: You may need to update /etc/fstab.
```

Теперь воссоздаем разделы так же как на оригинальном первом диске:

1.

```
root@pvel:~#parted /dev/sdb mkpart primary 1M 10M
```

```
Information: You may need to update /etc/fstab.
```

2.

```
root@pvel:~#parted /dev/sdb set 1 bios_grub on
```

```
Information: You may need to update /etc/fstab.
```

3.

```
root@pvel:~#parted /dev/sdb mkpart primary 10M 1G
```

```
Information: You may need to update /etc/fstab.
```

Вот тут нам пригодится знание размера оригинального первого диска.

4.

```
root@pvel:~#parted /dev/sdb mkpart primary 1G 1000GB
```

```
Information: You may need to update /etc/fstab.
```

Все эти четыре шага проделываем для всех наших дисков: sdb, sdc, sdd. Вот что у меня получилось:

Это оригинальный:

```
root@pvel:~#parted /dev/sda print
```

```
Model: ATA MB1000EBNCF (scsi)
Disk /dev/sda: 1000GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start   End     Size    File system  Name  Flags
  1      17.4kB  1049kB  1031kB                bios_grub
  2      1049kB  134MB   133MB   fat32          boot, esp
  3      134MB   1000GB  1000GB                lvm
```

А это второй, третий и четвертый (с разницей в букве диска).

```
root@pvel:~#parted /dev/sdb print
```

```
Model: ATA MB1000EBNCF (scsi)
Disk /dev/sdd: 1000GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start   End     Size    File system  Name      Flags
  1      1049kB  10.5MB   9437kB                primary  bios_grub
  2      10.5MB  1000MB   990MB                primary
  3      1000MB  1000GB   999GB                primary
```

Далее надо уточнить – если вы первый раз играете с этим кейсом и до этого на сервере, а главное на винчестерах, не было даже понятия RAID – можно пропустить этот пункт. Если же что-то не получилось, значит RAID уже возможно был установлен и на винчестерах есть суперблоки которые нужно удалять.

Проверить так:

```
root@pvel:~#mdadm --misc --examine /dev/sda
```

```
/dev/sda:
  MBR Magic : aa55
Partition[0] : 1953525167 sectors at 1 (type ee)
```

Проверить нужно все четыре диска.

Теперь настроим mdadm

Создаем конфиг на основе примера:

```
root@pvel:~#cp /etc/mdadm/mdadm.conf /etc/mdadm/mdadm.conf.orig
```

Опустошаем:

```
root@pvel:~#echo "" > /etc/mdadm/mdadm.conf
```

Открываем:

```
root@pvel:~#nano /etc/mdadm/mdadm.conf
```

Вводим и сохраняем:

```
CREATE owner=root group=disk mode=0660 auto=yes
MAILADDR user@mail.domain
```

Почту пока оставим как есть, потом к ней еще вернемся.

Теперь поднимаем наши RAID в режиме деградации (пропуская первый рабочий винчестер).

- В /dev/md0 – у меня будет /boot
- В /dev/md1 – VML раздел с системой

```
root@pvel:~#mdadm --create /dev/md0 --metadata=0.90 --level=10 --chunk=2048 --raid-devices=4 missing /dev/sd[bcd]2
```

```
mdadm: array /dev/md0 started.
```

И второй:

```
root@pvel:~#mdadm --create /dev/md1 --metadata=0.90 --level=10 --chunk=2048 --raid-devices=4 missing /dev/sd[bcd]3
```

```
mdadm: array /dev/md1 started.
```

Тут надо пояснить по ключам:

- --level=10 – говорит что наш RAID будет именно 10
- --chunk=2048 – размер кластера на разделе
- --raid-devices=4 – в рейде будут принимать участие четыре устройства
- missing /dev/sd[bcd]2 – первый рабочий раздел пока помечаем отсутствующим, остальные три добавляем в рейд

UDP. После массы комментариев я вышел на один важный момент.

В процессе создания я сознательно задавал chunk размер в 2048, вместо того что бы пропустить этот флаг и оставить его по умолчанию. Данный

флаг существенно снижает производительность. Особенно это даже визуально заметно на виртуалках с Windows.

То есть верная команда на создание должна выглядеть вот так:

```
root@pvel:~#mdadm --create /dev/md0 --metadata=0.90 --level=10 --raid-devices=4 missing /dev/sd[bcd] 2
```

и

```
root@pvel:~#mdadm --create /dev/md1 --metadata=0.90 --level=10 --raid-devices=4 missing /dev/sd[bcd] 3
```

Сохраняем конфигурацию:

```
root@pvel:~#mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Проверяем содержание:

```
root@pvel:~# cat /etc/mdadm/mdadm.conf
```

```
CREATE owner=root group=disk mode=0660 auto=yes
MAILADDR user@mail.domain
ARRAY /dev/md0 metadata=0.90 UUID=4df20dfa:4480524a:f7703943:85f444d5
ARRAY /dev/md1 metadata=0.90 UUID=432e3654:e288eae2:f7703943:85f444d5
```

Теперь нам нужен действующий LVM массив перенести на три пустых диска. Для начала создаем в рейде md1 — LVM-раздел:

```
root@pvel:~#pvcreate /dev/md1 -ff
```

```
Physical volume "/dev/md1" successfully created
```

И добавляем его в группу pve:

```
root@pvel:~#vgextend pve /dev/md1
```

```
Volume group "pve" successfully extended
```

Теперь переносим данные с оригинального LVM на новосозданный:

```
root@pvel:~#pvmove /dev/sda3 /dev/md1
```

```
/dev/sda3: Moved: 0.0%
```

Процесс долгий. У меня занял порядка 10 часов. Интересно, что запустил я его по привычке будучи подключенным по SSH и на 1,3% понял что сидеть столько времени с ноутом на работе как минимум не удобно. Отменил операцию через CTRL+C, подошел к физическому серверу и попробовал запустить команду переноса там, но умная железяка отписалась, что процесс уже идет и команда второй раз выполняться не будет, и начала рисовать проценты переноса на реальном экране. Как минимум спасибо :)

Процесс завершился два раза написав 100%. Убираем из LVM первый диск:


```
root@pvel:~#vgreduce pve /dev/sda3
```

```
Removed "/dev/sda3" from volume group "pve"
```

Переносим загрузочный /boot в наш новый рейд /md0, но для начала форматируем и монтируем сам рейд.

```
root@pvel:~#mkfs.ext4 /dev/md0
```

```
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 482304 4k blocks and 120720 inodes
Filesystem UUID: 6b75c86a-0501-447c-8ef5-386224e48538
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

Создаем директорию и монтируем туда рейд:

```
root@pvel:~#mkdir /mnt/md0
```

```
root@pvel:~#mount /dev/md0 /mnt/md0
```

Копируем содержимое живого /boot:

```
root@pvel:~#cp -ax /boot/* /mnt/md0
```

Отмонтируем рейд и удаляем временную директорию:

```
root@pvel:~#umount /mnt/md0
```

```
root@pvel:~#rmdir /mnt/md0
```

Определим UUID раздела рейда, где хранится /boot – это нужно, что бы правильно записать его в таблицу /etc/fstab:

```
root@pvel:~#blkid |grep md0
```

/dev/md0: UUID=«6b75c86a-0501-447c-8ef5-386224e48538» TYPE=«ext4»

Откроем таблицу и пропишем в ее конец данные загрузки /boot:

```
root@pvel:~#nano /etc/fstab
```

Прописываем и сохраняем:


```
UUID="6b75c86a-0501-447c-8ef5-386224e48538" /boot ext4 defaults 0 1
```

Теперь монтируем /boot:

```
root@pvel:~#mount /boot
```

Разрешим ОС загружаться, даже если состояние BOOT_DEGRADED (то есть рейд деградирован по причине выхода из строя дисков):

```
root@pvel:~#echo "BOOT_DEGRADED=true" > /etc/initramfs-tools/conf.d/mdadm
```

Прописываем загрузку ramfs:

```
root@pvel:~#mkinitramfs -o /boot/initrd.img-`uname -r`
```

Графический режим загрузчика отключаем:

```
root@pvel:~#echo "GRUB_TERMINAL=console" >> /etc/default/grub
```

Инсталируем загрузчик на все три диска:

```
root@pvel:~#grub-install /dev/sdb
```

```
Installing for i386-pc platform.
Installation finished. No error reported.
```

```
root@pvel:~#grub-install /dev/sdc>
```

```
Installing for i386-pc platform.
Installation finished. No error reported.
```

```
root@pvel:~#grub-install /dev/sdd
```

```
Installing for i386-pc platform.
Installation finished. No error reported.
```

Теперь очень важный момент. Мы берем за основу второй диск /dev/sdb, на котором система, загрузчик и grub и переносим всё это на первый диск /dev/sda, что бы в последствии сделать его так же частью нашего рейда. Для этого рассматриваем первый диск как чистый и размечаем так же, как другие в начале этой статьи

Занулим и пометим как GPT:

```
root@pvel:~#dd if=/dev/zero of=/dev/sda bs=512 count=1
```

```
1+0 records in
```

```
1+0 records out
512 bytes (512 B) copied, 0.0157829 s, 32.4 kB/s
```

```
root@pvel:~#parted /dev/sda mklabel gpt
```

```
Information: You may need to update /etc/fstab.
```

Разбиваем его по разделам в точности как другие три:

```
root@pvel:~#parted /dev/sda mkpart primary 1M 10M
```

```
Information: You may need to update /etc/fstab.
```

```
root@pvel:~#parted /dev/sda set 1 bios_grub on
```

```
Information: You may need to update /etc/fstab.
```

```
root@pvel:~#parted /dev/sda mkpart primary 10M 1G
```

```
Information: You may need to update /etc/fstab.
```

Тут нам снова понадобится точное знание размера диска. Напомню, получили мы его командой, которую в данном случае надо применять к диску /dev/sdb:

```
root@pvel:~#parted /dev/sdb print
```

Так как диски у нас одинаковые, то размер не изменился – **1000Gb**. Размечаем основной раздел:

```
root@pvel:~#parted /dev/sda mkpart primary 1G 1000Gb
```

```
Information: You may need to update /etc/fstab.
```

Должно получится так:

```
root@pvel:~#parted /dev/sda print
```

```
Model: ATA MB1000EBNCF (scsi)
Disk /dev/sda: 1000GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number   Start    End      Size    File system  Name      Flags
  1       1049kB   10.5MB   9437kB  fat32        primary  bios_grub
```

2	10.5MB	1000MB	990MB	primary
3	1000MB	1000GB	999GB	primary

Осталось добавить этот диск в общий массив. Второй раздел соответственно в /md0, а третий в /md1:

```
root@pvel:~#mdadm --add /dev/md0 /dev/sda2
```

```
mdadm: added /dev/sda2
```

```
root@pvel:~#mdadm --add /dev/md1 /dev/sda3
```

```
mdadm: added /dev/sda3
```

Ждем синхронизации...

```
root@pvel:~#watch cat /proc/mdstat
```

Данная команда в реальном времени показывает процесс синхронизации:

```
Every 2.0s: cat /proc/mdstat                               Fri Nov 11 10:09:18
2016

Personalities : [raid10]
md1 : active raid10 sda3[4] sdd3[3] sdc3[2] sdb3[1]
      1951567872 blocks 2048K chunks 2 near-copies [4/3] [_UUU]
      [>.....] recovery = 0.5% (5080064/975783936) finish=284.8min speed=56796K/sec
      bitmap: 15/15 pages [60KB], 65536KB chunk

md0 : active raid10 sda2[0] sdd2[3] sdc2[2] sdb2[1]
      1929216 blocks 2048K chunks 2 near-copies [4/4] [UUUU]
```

И если первый рейд с /boot синхронизировался сразу, то для синхронизации второго понадобилось терпение (в районе 5 часов).

Осталось установить загрузчик на добавленный диск (тут нужно понимать, что делать это нужно только после того, как диски полностью синхронизировались).

```
root@pvel:~#dpkg-reconfigure grub-pc
```

Пару раз нажимаем Enter ничего не меняя и на последнем шаге отмечаем галками все 4 диска md0/md1 не трогаем!

Осталось перезагрузить систему и проверить, что все в порядке:

```
root@pvel:~#shutdown -r now
```

Система загрузилась нормально (я даже несколько раз менял в BIOS порядок загрузки винтов — грузится одинаково правильно).

Проверяем массивы:

```
<source lang="vim">root@pvel:~#cat /proc/mdstat
```

```
Personalities : [raid10]
md1 : active raid10 sda3[0] sdd3[3] sdc3[2] sdb3[1]
      1951567872 blocks 2048K chunks 2 near-copies [4/4] [UUUU]
      bitmap: 2/15 pages [8KB], 65536KB chunk

md0 : active raid10 sda2[0] sdd2[3] sdc2[2] sdb2[1]
      1929216 blocks 2048K chunks 2 near-copies [4/4] [UUUU]
```

По четыре подковы в каждом рейде говорят о том, что все четыре диска в работе. Смотрим информацию по массивам (на примере первого, точнее нулевого).

```
root@pvel:~#mdadm --detail /dev/md0
```

```
/dev/md0:
   Version : 0.90
  Creation Time : Thu Nov 10 15:12:21 2016
    Raid Level : raid10
   Array Size : 1929216 (1884.32 MiB 1975.52 MB)
  Used Dev Size : 964608 (942.16 MiB 987.76 MB)
   Raid Devices : 4
  Total Devices : 4
Preferred Minor : 0
   Persistence : Superblock is persistent

   Update Time : Fri Nov 11 10:07:47 2016
     State : active
   Active Devices : 4
  Working Devices : 4
 Failed Devices : 0
   Spare Devices : 0


   Layout : near=2
  Chunk Size : 2048K


   UUID : 4df20dfa:4480524a:f7703943:85f444d5 (local to host pvel)
   Events : 0.27
```

Number	Major	Minor	RaidDevice	State	
0	8	2	0	active sync set-A	/dev/sda2
1	8	18	1	active sync set-B	/dev/sdb2
2	8	34	2	active sync set-A	/dev/sdc2
3	8	50	3	active sync set-B	/dev/sdd2

Видим, что массив типа RAID10, все диски на месте, активные и синхронизированы.

Теперь можно было бы поиграться с отключением дисков, изменении диска-загрузчика в BIOS, но перед этим давайте настроим уведомление администратора при сбоях в работе дисков, а значит и самого рейда. Без уведомления рейд будет умирать медленно и мучительно, а никто не будет об этом знать.

В Proxmox по умолчанию уже стоит postfix, удалять его я не стал, хоть и сознательно понимаю что другие MTA было бы проще настроить.

Ставим SASL библиотеку (мне она нужна, что бы работать с нашим внешним почтовым сервером):

```
root@pvel:/etc#apt-get install libsasl2-modules
```

Создаем файл с данными от которых будем авторизовываться на нашем удаленном почтовом сервере:

```
root@pvel:~#touch /etc/postfix/sasl_passwd
```

Там прописываем строчку:

```
[mail.domain.ru] pvel@domain.ru:password
```

Теперь создаем транспортный файл:

```
root@pvel:~#touch /etc/postfix/transport
```

Туда пишем:

```
domain.ru smtp:[mail.domain.ru]
```

Создаем generic_map:

```
root@pvel:~#touch /etc/postfix/generic
```

Тут пишем (обозначаем от кого будет отправляться почта):

```
root pvel@domain.ru
```

Создаем sender_relay (по сути, маршрут до внешнего сервера):

```
root@pvel:~#touch /etc/postfix/sender_relay
```

И пишем туда:

```
pvel@domain.ru smtp.domain.ru
```

Хешируем файлы:

```
root@pvel:~#postmap transport
```

```
root@pvel:~#postmap sasl_passwd
```

```
root@pvel:~#postmap geniric
```

```
root@pvel:~#postmap sender_relay
```

В файле /etc/postfix/main.cf у меня получилась вот такая рабочая конфигурация:

[main.cf](#)

Перезагружаем postfix:

```
root@pve1:~# /etc/init.d/postfix restart
```

Теперь нужно вернуться в файл настроек рейда и немного его поправить. Указываем кому получать письма счастья и от кого они будут приходить.

```
root@pve1:~# nano /etc/dmadm/mdadm.conf
```

У меня вот так:

```
CREATE owner=root group=disk mode=0660 auto=yes
MAILADDR info@domain.ru
MAILFROM pve1@dpmain.ru

ARRAY /dev/md0 metadata=0.90 UUID=4df20dfa:4480524a:f7703943:85f444d5
ARRAY /dev/md1 metadata=0.90 UUID=432e3654:e288eae2:f7703943:85f444d5
```

Перезапускаем mdadm, что бы перечитать настройки:

```
root@pve1:~# /etc/init.d/mdadm restart
```

Проверяем через консоль тестирование рейда и отправку письма:

```
root@pve1:~# mdadm --monitor --scan -1 --test --oneshot
```

У меня пришло два письма с информацией по обоим созданным мною рейдам. Осталось добавить задачу тестирования в крон и убрать ключ `--test`. Чтобы письма приходили только тогда, когда что-то произошло:

```
root@pve1:~# crontab -e
```

Добавляем задачу (не забудьте после строки нажать на Enter и перевести курсор вниз, что бы появилась пустая строка):

```
0 5 * * * mdadm --monitor --scan -1 --oneshot
```

Каждое утро в 5 утра будет производится тестирование и если возникнут проблемы, произойдет отправка почты.

На этом всё. Возможно перемудрил с конфигом postfix – пока пытался добиться нормальной отправки через наш внешний сервер, много чего надо добавлял. Буду признателен, если поправите (упростите).

В следующей статье я хочу поделиться опытом переезда виртуальных машин с нашего гипервизора Esxi-6 на этот новый Proxmox. Думаю будет интересно.

UPD.

Стоит отдельно отменить момент с физическим местом на разделе /dev/data – это основной раздел созданный как LVM-Thin. Когда ставился Proxmox, он автоматически разметил /dev/sda с тем учетом, что на /root раздел где хранится система, ISO, дампы и темплеи контейнеров, он выделил 10% емкости от раздела, а именно 100Gb. На оставшемся месте он создал LVM-Thin раздел, который по сути никуда не монтируется (это еще одна тонкость версии >4.2, после перевода дисков в GPT). И как вы понимаете этот раздел стал размером 900Gb. Когда мы подняли RAID10 из 4х дисков по 1Tb – мы получили емкость (с учетом резерва RAID1+0) – 2Tb. Но когда копировали LVM в рейд – копировали его как контейнер, с его размером в 900Gb.

При первом заходе в админку Proxmox внимательный зритель может заметить, что тыкая на раздел local-lvm(pve1) – мы и наблюдаем эти с копейками 800Gb.

Так вот что бы расширить LVM-Thin на весь размер в 1,9TB нам потребуется выполнить все одну команду:

```
lvextend /dev/pve/data -l +100%FREE
```

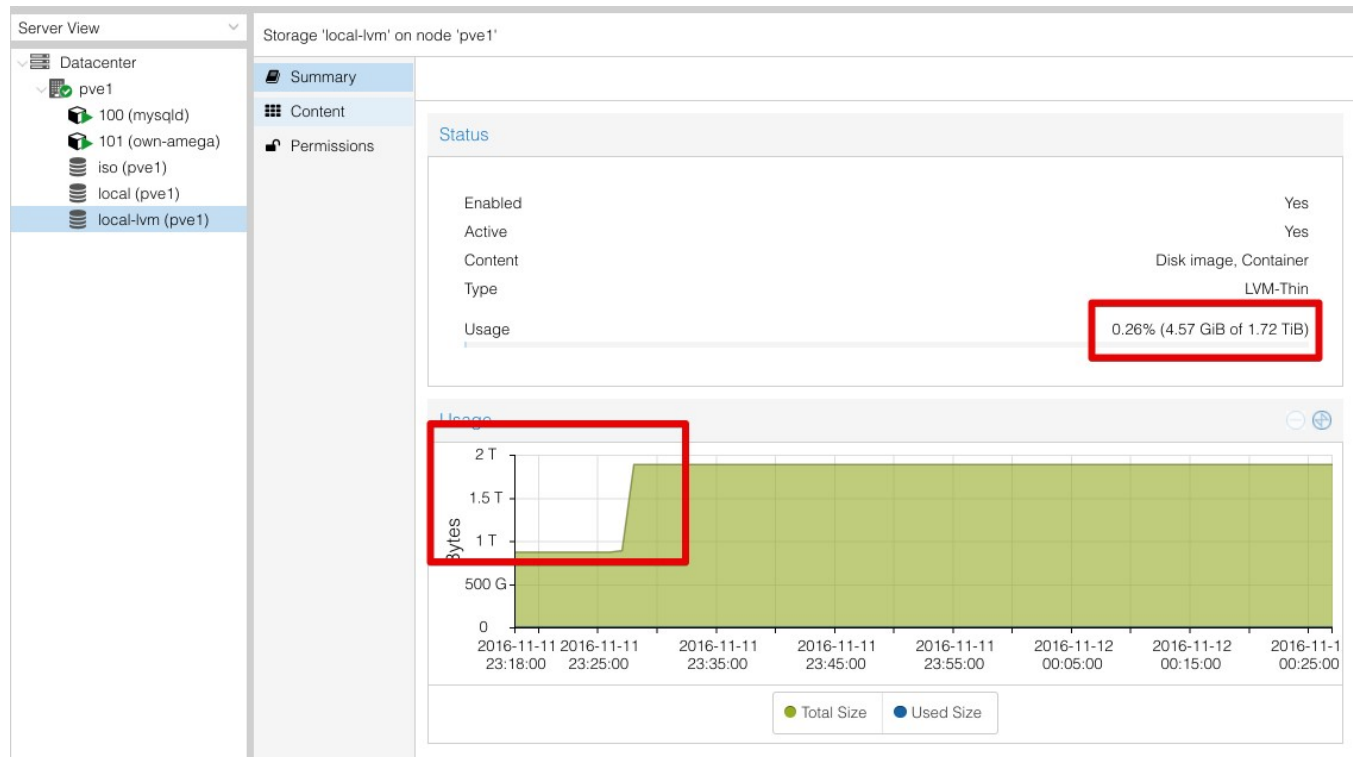
После этого систему не нужно даже перезапускать.

Не нужно делать `resize2fs` – и это скорее даже невозможно, потому как система начнет ругаться на

```
root@pve1:~# resize2fs /dev/mapper/pve-data
resize2fs 1.42.12 (29-Aug-2014)
resize2fs: MMP: invalid magic number while trying to open /dev/mapper/pve-data
Couldn't find valid filesystem superblock.
```

И правильно начнет – этот раздел у нас не подмонтирован через `fstab`

В общем пока я пытался понять, как расширить диск и читал форум Proxmox – система тем временем уже во всю показывала новый размер, как в таблице, так и на шкале.



Метки: proxmox 4, raid10, postfix

↑ +17 ↓ 95 15,6k 64



5,0

Карма

0,0

Рейтинг

4

Подписчики

Yegorov Vassiliy @vasyakrg

ИТ Администратор

Поделиться публикацией

ПОХОЖИЕ ПУБЛИКАЦИИ

27 декабря 2016 в 11:53

Сeph в ProxMox на ZFS

↑ +10 10,3k 67 10

15 ноября 2016 в 14:53

Немного о дисковой производительности Windows VM в Proxmox VE. Результаты бенчмарков ZFS и MDADM+LVM

↑ +3 👁 6,7k 📖 17 💬 23

19 июля 2016 в 11:06

Автоответчик в postfix

↑ +2 👁 2,9k 📖 38 💬 2

AdBlock похитил этот баннер, но баннеры не зубы — отрастут

Подробнее

Реклама

Комментарии 64

 **Meklon** 12.11.16 в 16:41 🗨 📖 ↑ 0 ↓

Очень подробно, спасибо)

 **vasyakrg** 12.11.16 в 17:58 🗨 📖 📌 🔄 ↑ 0 ↓

я сейчас продолжаю настраивать это сервер.
настроил поддержку VLAN для виртуальных машин — оказалось, что по этому вопросу очень мало достоверной информации и она очень противоречива. если будет время, попробую выпустить отдельной заметкой.
так же думаю описать, как у меня с горем пополам получилось переконвертировать несколько разнотипных виртуалок с Esxi6 — думаю тоже кому то, да пригодится.

 **icCE** 12.11.16 в 19:02 🗨 📖 📌 🔄 ↑ 0 ↓

а что там такого сложного с vlan?
Уже лет 8 использую bonding,vlan,bridge.
Основная сложность была в момент перехода прохтох на openvswitch, так как в debian свое видение конфигов.
Ну и народ еще забывает при использование HA, что кластер общается по мультикасту :)

 **vasyakrg**  12.11.16 в 19:08 🗨 📖 📌 🔄 ↑ 0 ↓

основная сложность была изменить представление настройки, после панели Esxi6. :)
особенно постоянно перезагружать сервер, даже после добавления комментария к интерфейсу.

пробовал по трем разным статьям. ничего не получалось.
пробовал точь в точь с wiki прохтох — не завелось.
в итоге дошел сам.
[вот то получилось:](#)

кстати, как я понимаю, теперь mvbr0 — можно вообще удалить, т.к. management интерфейс нужен в VLAN2...

 **icCE** 12.11.16 в 19:11 🗨 📖 📌 🔄 ↑ 0 ↓

я вот тут писал — <http://unixforum.org/index.php?showtopic=136947>
Там в конце примеры конфигов, правда они и сейчас наверно требует правок и дополнений.
Просто я еще использую объединение интерфейсов :)

 **vasyakrg** 12.11.16 в 19:16 🗨 📖 📌 🔄 ↑ 0 ↓

а объединение в данном случае для чего?
транковые группы?




 **icCE** 12.11.16 в 19:21 🗨 📖 📌 🔄 ↑ 0 ↓

bonding используется для отказоустойчивости + увеличение производительности сети (условно)

https://en.wikipedia.org/wiki/Link_aggregation






Пример, есть 4 сетевые карты eth0,1,2,3, делается единый интерфейс bond0, на него уже вешается vlan0,vlan1,vlan2 + n и уже vlan могут быть сбриджеванны с вирт машинами.

Надо понимать только про некий overhead по пакетам и насколько у вас стоят задачи так делать и насколько вам будет так удобно и нужно ли вообще. Я обычно делаю на автомате, если конечно позволяют мощности сети. Везде нужен здравый смысл.

 **MagicGTS** 12.11.16 в 17:33  






 +4 

Я бы сказал, что статья из серии: как перенести установленную linux на soft-raid при наличие свободных дисков и, что она изначально на LVM (масса очепятков в статье). Таких статей много, новизны информации нет. В данной статье, к сожалению, многое описано как «черная магия».

 **vasyakrg** 12.11.16 в 17:56    





 0 

я был бы очень благодарен, если бы вы подсказали, где именно очепятки.
тогда я бы оперативно их исправил.
раз уж не удивил вас новизной, так хоть уберу банальные ошибки...

 **MagicGTS** 12.11.16 в 18:26    

 0 

Вариации букв LVM, это основное.

 **moropsk**  12.11.16 в 17:45  






 0 

вопрос вот такому моменту:

Планируется установить Proxmox на отдельный диск + 2-а массива raid 1 по 4 тб созданные на программном рейде десктопной материнской платы до установки Proxmox

Установленная виртуалка на Proxmox Debian 8 увидит эти массивы?






Их потом можно примонтировать?

 **vasyakrg** 12.11.16 в 17:54    

 0 






Proxmox — система на ОС debian. Программный и аппаратный рейд он скорее не распознает и будем думать что это один простой диск. Монтировать его в этом случае можно стандартными средствами Debian
mount /dev/sdX /mnt/disk2 и прописать в fstab что бы запускался с загрузкой системы. А в Промоксе добавить этот раздел как папку в разделе Storage.
другой вопрос: если вы на этом разделе будете хранить контейнеры — то лучше изначально создать его как LVM-thin, а потом в промаксе подключить. Можно будет на ходу менять размеры контейнеров.

вот кстати фокус с программным рейдом из под материнки у меня не вышел с Esxi6 — та просто видела их как отдельные диски. А на форуме у них так и написано — с программными не дружим и не собираемся. покупайте железу. Это одна из причин, почему начал пробовать заморачиваться с Промоксом.

 **vasyakrg** 12.11.16 в 18:03    




 0 

про видимость массива из виртуалки не понял. проще использовать рейд в самом промоксе, на нем для этой виртуалки «отрезать» с LVM нужный кусок...

 **MagicGTS** 12.11.16 в 18:33    

 +1 






RAID в большинстве десктопных материнок — это так называемый fake-raid. По русски — липовый. По сути это вариация на тему с soft-raid замаскированная под железную. В linux с ним может работать dmraid (используя подсистемму device mapper) и работает не со всеми контроллерами. Работает это обычно с чудесами в самые неожиданные моменты, и им крайне трудно управлять из ОС. Рекомендую перевести эти диски в обычный mdadm (soft-raid), и управляемость и стабильность выше.

 **icCE** 12.11.16 в 18:58  

 0 

Я обычно в таком случаи идут немного другим путем, а именно путем debotstrap.
Грузимся, размечаем как надо и ставим всю систему.






Сейчас вполне идут игры с ZFS и использования кеша L2 как на SSD так и на в памяти.
Профит велеколепный! Там же удобно делать raid, снпшоты и не городить огород.

 **vasyakrg** 12.11.16 в 19:01    

 0 

Вот про это я бы с удовольствием почитал.
Тем более что как раз лежит в резерве SSD64Gb.

попробовал при загрузке в инсталлере выбрать не раздел, а ZFS и собрать из 4х дисков.
ничего не вышло.
На сколько я понял, проблема была UEFI — который этот сервер не поддерживает.

 **icCE** 12.11.16 в 19:08    

 0 

можно начать вот тут

<https://habrahabr.ru/post/272249/>



vasyakrg 12.11.16 в 19:13



↑ 0 ↓

да, теперь убедился, что у меня проблема была именно в UEFI
сейчас закончу с этим сервером, перенесу на него виртуалки со второго.
а вот на втором, более свежем, как раз два диска — сделаю по вашей инструкции, а заодно поиграю с кэшем.



icCE 12.11.16 в 19:18



↑ 0 ↓

на всякий случай, инструкция не моя.
с UEFI ситуацию так же можно вполне разрулить, но я уже настолько привык все делать скриптами — что уже не использую стандартный установщик. Хотя тут наверно еще привычки с gentoo/archlinux, где по сути нет установщика.



vasyakrg 15.11.16 в 15:24



↑ 0 ↓

Возможно пригодится для размышления [статья про тестирование ZFS в сравнении с mdadm+vlm](#) от @kvaps



PinGniX 12.11.16 в 20:00



↑ 0 ↓

Процесс долгий. У меня занял порядка 10 часов. Интересно, что запустил я его по привычке будучи подключенным по SSH и на 1,3% понял что сидеть столько времени с ноутом на работе как минимум не удобно. Отменил операцию через CTRL+C

Во избежание подобных ситуаций рекомендую посмотреть в сторону tmux,screen и других менеджеров терминалов. Это первое, что я обычно запускаю после входа по ssh на свежий сервер.



vasyakrg 12.11.16 в 20:02



↑ 0 ↓

да, именно screen пользуюсь. но в данном случае сервер был в 3 шагах от меня.



winduzoid 12.11.16 в 20:07



↑ +1 ↓

Вот официальная документация по установке Proxmox на debian:

https://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Jessie

И проще, и быстрее.



vasyakrg 12.11.16 в 20:19



↑ 0 ↓

ну то есть, поднять debian, поднять на нем RAID10, разметить LVM, потом сверху установить proxmox?
а в чем разница по скорости?



winduzoid 12.11.16 в 20:26



↑ 0 ↓

Разница в том, что Debian можно сразу установить на Raid10.
По сути, не придется делать ничего из того, что у вас описано.

Плюс гибкость в установке. Все же это Debian.



vasyakrg 12.11.16 в 20:37



↑ 0 ↓

ничего не понял. как так сразу? в смысле создать через Raid10 в процессе установки debian через разметку дисков?

по поводу гибкости — в чем именно? Proxmox основан на дебиане, по сути...
при установке debian много вопросов задает в начале, потом может и проще.
Proxmox — косит под коробку — нажал 3 кнопки и готово. потом вот пришлось такую статью попутно писать, пока настраивал и разбирался.
возможно вы и правы :)



winduzoid 12.11.16 в 20:46








↑ +1 ↓

ничего не понял. как так сразу? в смысле создать через Raid10 в процессе установки debian через разметку дисков?

Да. В инсталляторе создать рейд, и поставиться на него. В итоге не будет шаманства с дисками, и шаманства с LVM.
Гибкость в том, что на этапе инсталляции вы вольны разбить диски как вам вздумается. Например, откусить немного места под систему, и поставиться хоть на Raid1 из 4 дисков (имеет смысл, кстати). А оставшееся место использовать под Raid10.

Поставили Debian, накатили Proxmox. Все.






Косит-то он под коробку, жаль, что толку от этого мало. Приходится выбирать между коробочностью и продуктивностью.

 icCE 12.11.16 в 21:01    

 +1 


в debian можно создать файл ответа и автоматизировать установку.

В итоге используя tftp, делалась установка по сети. День работы, стойка с прохтох готова без особого напряжения.

 vasyakrg 12.11.16 в 21:04    

 -1 

думал об этом. из этой статьи можно по сути, сделать файл ответов с несколькими входными данными типа кол-ва винтов, типа рейда и тд.






 merlin-vrm  12.11.16 в 21:07    

 +1 

ВООБЩЕ никаких танцев с бубном. Абсолютно. Просто ставишь дебиан как обычно, сверху проксмокс по инструкции.






Так просто проксмокс ставить удобнее, если скрипт автоматизации дебиана есть. Мы его так разворачиваем всегда, даже если аппаратный райд и в принципе пошло бы с проксмокс-ового диска — ибо есть нормально сделанный инсталлятор дебиана с сети, и так оно по часам на стене может и больше времени занимает, а если считать время, которое администратор реально копаётся с системой (а не она сама там скачивается и ставится) — принципиально меньше.

Именно вы, автор, избрали незыблемый путь, хотя рядом есть правильный, простой и надёжный.

 vasyakrg 12.11.16 в 21:18    

 0 







ну если вы прочитали статью до конца, но могли заметить, что статья была не только и не столько про установку промoxa как обычно. это скорее моя практика работы с LVM и уведомлениями после установки.

 DmitryPanteleev 14.11.16 в 14:41    

 0 

Сто раз плюсю. Сам так делал. Просто. Быстро. Надежно.






Никаких «плясок с бубном». Рейд собириается из визарда при инсталляции дебиана. Потом подключается дополнительная репа и apt update & install. И все! Просто ВСЕ! Ничего больше делать не надо!

 vasyakrg  14.11.16 в 14:42    

 0 




напомните, как в визарде задать размер chunk?

я вот сейчас хочу все это сделать на виртуалке с 4мя винтами.

 merlin-vrm 14.11.16 в 15:57    






 0 

Ну, да, это в гуи никак. В инсталляторе дебиана alt+f2 и делайте райды руками. Там можно настроить ЧТО УГОДНО.

 dmitry_ch 12.11.16 в 23:59  






 0 

Может тогда ставить именно Proxmox, но на ZFS, что он умеет «из коробки» (прямо кнопкой из диалога выбора диска)?

 Pilat 13.11.16 в 03:21    

 0 

Тут в данной конфигурации может ожидать неожиданность — ZFS требует много памяти, а на данном сервере её и так маловато






 icCE 13.11.16 в 04:00    

 0 

Я бы сказал по другому, у ZFS есть свои подводные камни.

Памяти он не особо требует много, если мы не начинаем заниматься с L2ARC итд.

Хотя если памяти очень много, я бы рекомендовал это сделать + SSD под еще один кеш.

 vasyakrg 13.11.16 в 06:53    

 0 






а можете привести сравнение по памяти?

к примеру, сервер с 32Gb ОЗУ и 6 винтов X 4Gb

памяти на работу с ZFS уходит 6Gb

что бы грубо представить.

понятное дело, что расходование сильно зависит от проводимых операций...

 icCE 13.11.16 в 17:09    

 0 

все зависит от настроек. Сжатие, дедупликация, пулы и предсказания.

минимально zfs требует 1Gb.






Если памяти мало то `vfs.zfs.prefetch_disable=1` — отключение режима prefetch

`vfs.zfs.arc_max` — сколько будет кешироваться данных в пуле.

Если все по default и почитать гайд http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide




Станет понятно, что `zfs` старается забрать всю память кроме 1 Gb, но когда она потребуется он ее освободить. Я обычно прибавляю это гвоздями и отдаю не больше 4gb, остальное кеш на `ssd`.

Кешировать без `SSD` только в память бессмысленно, при перезагрузки вам надо опять будет прогревать кеш.

 **vasyakrg** 13.11.16 в 06:51    






↑ -1 ↓

в моем случае не получилось. т.к. сервер не поддерживал `UEFI` система попросту не загрузилась после установки.

 **bARmaleyKA** 13.11.16 в 09:08  






↑ -2 ↓

Ну к чему всё это? В инсталляторе есть же нормальная оснастка для создания программного массива. `Jessy` поддерживает работу с `btfrfs`, поэтому можно просто создать программный `RAID` с этой файловой системой без танцев с `LVM`. Другой момент — эта система для продакшена или дома побаловаться? Если для дома, для души, то понятно. Иначе, почему не аппаратный контроллер и диски `SAS`. Чтобы там не фантазировали, но аппаратный контроллер будет быстрее программного, так ещё с удобствами диагностики и горячей заменой. Интерфейс `SAS` полнодуплексный в отличие от полудуплексного `SATA` и диски быстрее и надёжней.

 **merlin-vrn** 13.11.16 в 09:13    





↑ +1 ↓

Что касается удобства диагностики, то тут программному равных нет. Вон, [@amarao](#) любит про это. А горячая замена есть в `SATA` уже лет десять.

 **IcCE** 13.11.16 в 17:00    

↑ 0 ↓







про программный и аппаратный `RAID` уже сломанно много копий. Если задачи относительно простые, то лучше программный.

 **bARmaleyKA** 14.11.16 в 01:03    

↑ 0 ↓

А горячая замена есть в `SATA` уже лет десять.






В программном массиве? Речь до сих пор о нём идёт?

 **merlin-vrn**  14.11.16 в 09:05    

↑ 0 ↓

Да, про программный. Но вообще горячая смена дисков `SATA` не имеет прямого отношения к программному `RAID` — это просто способность `AHCI` свободно подключать и отключать устройства. Просто эта способность очень пригодилась именно в инсталляциях с программным `RAID`, и как сухой остаток — в программном массиве уже лет десять как можно менять винты на горячую (т.е., не останавливая `OS` и прозрачно для прикладных задач).

Например, если вы когда-нибудь использовали любую `Synology DS` (даже старшие модели, которые на самом деле представляют из себя стоечные серверы на `i3`) — там везде внутри именно `Linux Software RAID` в чистом виде и всё на горячую меняется. Уже давно.






 **vasyakrg** 13.11.16 в 09:16    

↑ 0 ↓

Оснастка есть. но мне нужен был именно `LVM`. Система для продакшена, но сервер будет выполнять скорее резервные функции, для экстренного переноса некоторых виртуалок с основного кластера, в случае проблем.

На счет аппаратного, скажу честно, сам не сравнивал, но тут на хабре есть ряд статей, которые, в целом, говорят о 6-8% прироста скорости по сравнению с программным. Опять же, у нас есть сервера разношерстные и, как минимум, узким местом становится сам железный контроллер. В программном, я могу в любом момент развернуть новый дебиан и восстановить массив. А в случае с железным — нужно держать `ЗИП` на каждый такой сервер.

да и не хотелось на этот сервер докупать что-то дополнительно.






 **IcCE** 13.11.16 в 17:00    

↑ 0 ↓

>А в случае с железным — нужно держать `ЗИП` на каждый такой сервер.





Мне кажется нужно держать `Васкуп`, не? :)

Хотя в общем мысль конечно ясно.

 **vasyakrg** 13.11.16 в 17:47    

↑ 0 ↓

Бакап конечно есть, но хотелось бы заменить винт и включить сервер, нежели переливать пару терров. В общем то я и задался такими вот экспериментами, чтобы бакапы отодвинуть как последнюю инстанцию

 **merlin-vrn** 14.11.16 в 09:10    

↑ 0 ↓

Нет, неверно. Бакап предназначен для резерва на случай человеческого фактора, и не предназначен для резервирования на случай аппаратных сбоев.

Вот `ЗИП` (холодный резерв) — хорошая идея на этот случай. Лучше — только дублирующий сервер, который постоянно получает копию и может в

считанные мгновения завести виртуалки в случае сбоя (горячий резерв).



bARmaleyKA 14.11.16 в 01:56

↑ 0 ↓

В своё время также проникся софтовыми массивами на статьях с хабра. Хорошие и познавательные статьи. Только прирост 6-8% для дисковой подсистемы уже не мало. Но под хорошей нагрузкой программный сливает намного больше 8%. Просто задайте себе вопрос, что будет лучше работать специализированная железка или программная эмуляция? Вы бы что предпочли лично встроенную графическую карту или дискретную видеокарту для обработки «тяжёлого» видеопотока? Ведь по той логике софтовый/программный контроллер дисков, встроенная видеокарточка будет немногим хуже, где-то на 6-8%.

Про ЗИП из контроллеров совсем не понял. Получается, что это вроде расходника, который надо периодически менять? Обычно диски раньше помирают чем контроллер. Как это бы не блок питания в нем нет переходных процессов при выключении/включении. Греться там особо нечему, подвижных частей замечено не было. Довольно надёжный узел, понятное дело что речь не идёт о дешёвых fake-контроллерах за 500 рэ. Оно вон бывает помирает материнка на сервере, но ещё не встречал в прозапасе материнок к серверам.



kvazimoda24 13.11.16 в 19:54

↑ 0 ↓

А почему у RAID'a chunk всего 2 килобайта? Сейчас же новые диски идут с секторами по 4 килобайта, соответственно, с меньшим chunk'ом можно здорово потерять в скорости.



vasyakrg 13.11.16 в 19:55

↑ 0 ↓

а вот это попробую протестировать на лаб-сервере... отличный вопрос!



kvazimoda24 14.11.16 в 21:18

↑ +1 ↓

Ещё и разметка диска не выравнена по секторам. Первый раздел начинается с первого килобайта. В общем, какая-то дикая разметка диска...



vasyakrg 15.11.16 в 04:43

↑ 0 ↓

Подскажите, как сделать правильно? У меня второй сервер на очереди. За одно и тут поправлю.



lcCE 15.11.16 в 08:54

↑ 0 ↓

man parted

-a alignment-type, --align alignment-type

Set alignment for newly created partitions, valid alignment types are:

none Use the minimum alignment allowed by the disk type.

cylinder

Align partitions to cylinders.

minimal

Use minimum alignment as given by the disk topology information. This and the opt value will use layout information provided by the disk to align the logical partition table addresses to actual physical blocks on the disks. The min value is the minimum alignment needed to align the partition properly to physical blocks, which avoids performance degradation.

optimal

Use optimum alignment as given by the disk topology information. This aligns to a multiple of the physical block size in a way that guarantees optimal performance.



kvazimoda24 15.11.16 в 18:09

↑ +1 ↓

Первый раздел обычно начинает с 2048 сектора, это если сектора программа считает по 512 байт. Получается, что раздел начинается со второго мегабайта. Размер раздела должен быть кратен четырём килобайтам, следующие разделы так же должны быть сразу после предыдущего, либо пропуск должен быть кратен четырём килобайтам. Если короче, то каждый раздел должен начинаться с позиции на диске, кратной четырём килобайтам.

В самой файловой системе использовать размер кластера так же кратным четырём килобайтам. И размер чанка у рейда установить кратным четырём килобайтам. Обычно, его и ставят размером 4096 байт.



vasyakrg 15.11.16 в 19:19

↑ 0 ↓

Нашел неплохую [статью](#) (@dmbarsukov), по поводу смены чанка на действующей системе. И в этой же статье ссылка на отличную [таблицу](#), которая я думаю многим пригодится.

К сожалению на этом сервере уже не смогу поиграть. А вот на лабораторном думаю смогу поднять текущую конфигурацию и применить эту статью по подбору верного решения.



kvazimoda24 15.11.16 в 23:32

↑ 0 ↓

Ну, если учитывать, что по умолчанию mdadm использует размер чанка в 64K, а то и в 512K, то не совсем понимаю, зачем надо было

ставить такой маленький размер. Для какой цели? Зачем его вообще было вручную задавать? Какую цель преследовали?



vasyakrg 16.11.16 в 06:58

📖 🔄 🗨️

↑ 0 ↓

вот этот момент я упустил и с чего-то взял, что делает он по 1024, вот и задал по 2048.
считаю должным поправить этот момент в статье.



merlin-vrn 16.11.16 в 08:43

📖 🔄 🗨️

↑ 0 ↓

Да можно переделать, можно. У вас очень гибкая структура, которая переживёт такую переделку без останова сервисов (но, конечно, на период миграции диски будут притормаживать).



vasyakrg 16.11.16 в 09:44

📖 🔄 🗨️

↑ 0 ↓

сейчас на другом сервере два винта по 930Гб (vmware) собрал в рейд-1 стандартными средствами.
chunk при этом там равен 65536 KB
что я полагаю, тоже не совсем верно?



merlin-vrn 16.11.16 в 09:58

📖 🔄 🗨️

↑ 0 ↓

чанк raida 64 метра? Вы что-то неправильно поняли. Это скорее чанк VMFS5, а не массива



vasyakrg 16.11.16 в 19:42

📖 🔄 🗨️

↑ 0 ↓

да, возможно...

[вывод](#)



merlin-vrn 16.11.16 в 21:33

📖 🔄 🗨️

↑ 0 ↓

Это не чанк raida, а встроенного write-intent bitmap. Читайте ман, что там искать, я написал.

Размер, блока которыми оперируют RAID-ы, обычно называют полосой (stripe). Да и вообще, для RAID1 подобный параметр не имеет смысла, поэтому и не указывается в --detail и тому подобное.

Только [полноправные пользователи](#) могут оставлять комментарии. [Войдите](#), пожалуйста.

ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ



9 учебных проектов для бэкендера

↑ +9 👁 1,6k 📖 33 💬 7

Конкурс дешифрования в Аризонском Государственном Университете (интервью)

↑ +5 👁 599 📖 4 💬 5

Как построить сообщество. Перевод книги «Социальная архитектура»: Глава 1. Инструментарий

↑ +10 👁 1k 📖 41 💬 1

SecurityWeek 50: хактивист устал и мухожук, фальшивый криптокошелек для любителей панд, двуликий Янус под Android

↑ +5 👁 1,3k 📖 2 💬 0

Клятва Гиппократа или как защищать информацию в медицинских учреждениях

↑ +6 👁 1,1k 📖 7 💬 1

Аккаунт	Разделы	Информация	Услуги	Приложения
Войти	Публикации	О сайте	Реклама	<div><div> Загрузите в App Store</div><div> доступно в Google Play</div></div>
Регистрация	Хабы	Правила	Тарифы	
	Компании	Помощь	Контент	
	Пользователи	Соглашение	Семинары	
	Песочница	Конфиденциальность		
<div> © 2006 – 2017 «TM»</div>		Служба поддержки	Мобильная версия	<div></div>