

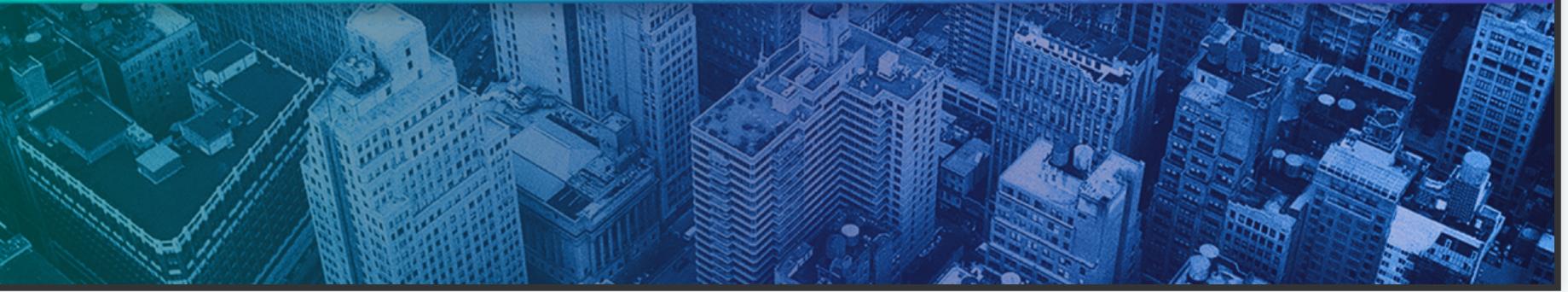


Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы



НЕ ЗАБЫТЬ ВКЛЮЧИТЬ
ЗАПИСЬ!!!

Дисковая подсистема Linux

О себе

Test Automation Engineer

- более 10 лет опыта работы в среде Linux
- тестирую Lustre



Преподаватель

- в OTUS
- и другие (курс “Основы администрирования Linux”)

Викентий Лапа

Правила вебинара

- Активно участвуем: выполняем задания, отвечаем на вопросы
- Если возникли сложности задаем вопрос в чат
- На вопросы постараюсь отвечать сразу, но возможны паузы

Маршрут вебинара

- Дисковая система Linux
 - Имена дисков
 - Разделы на дисках
- Массив дисков
 - Типы RAID
- Примеры обслуживания RAID
 - Создаем RAID 1 уровня - зеркало.
 - Остановка и старт массива
 - Старт когда один диск полностью поврежден
 - Создаем RAID10, RAID5
- Дополнительные настройки дисковой системы
 - Драйвера устройств
 - Планировщики ввода/вывода (I/O Schedulers)

После занятия вы сможете

1. Найти диски в системе Linux и определить их параметры
2. Разбить диск на разделы вручную или скриптом
3. Выбрать тип RAID и создать программный RAID
4. Поменять настройки дисков

Зачем вам это уметь

ВАШ ВАРИАНТ?

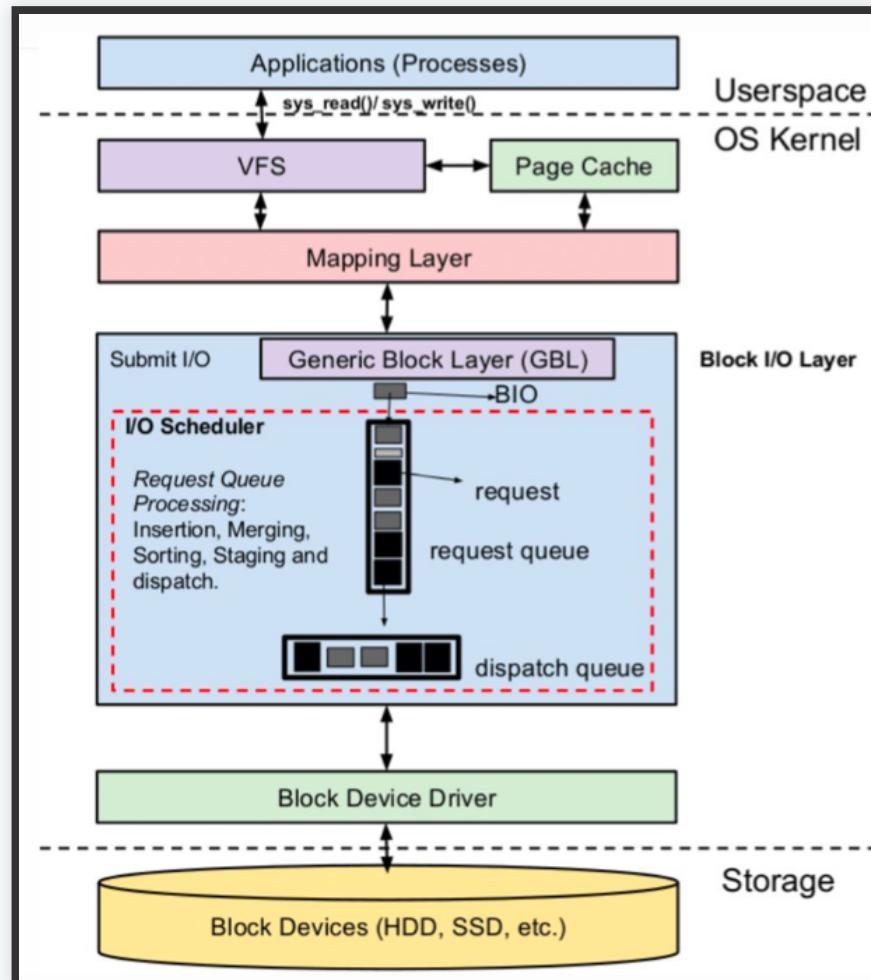
Зачем вам это уметь

МОЙ ВАРИАНТ

1. Сможете быстрее разобраться в конфигурации на сервере от другого админа
2. Повысите отказоустойчивость дисковой подсистемы
3. Меньше волнения при замене вылетевшего диска

Дисковая система Linux

Дисковая подсистема



Файлы устройств

Два типа:

- блочные (**block** special files)
- символьные (**character** special files).

```
# ls -l /dev/sda1  
brw-rw----. 1 root disk 8, 1 Jan 28 20:03 /dev/sda1  
# cat /proc/devices  
Block devices:  
8 sd  
9 md
```

b - класс устройства, **8** - старший номер устройства **1** - младший номер устройства.

В чем принципальное отличие?

Файлы блочных устройств

```
# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0   8G  0 disk 
└─sda1     8:1    0   1G  0 part /boot
└─sda2     8:2    0   7G  0 part
  ├─cl-root 253:0  0  6.2G 0 lvm   /
  └─cl-swap 253:1  0 820M 0 lvm   [SWAP]
sdb        8:16   0 250M 0 disk 
sr0       11:0    1 597M 0 rom  
nvme0n1   259:0   0 317.6M 0 disk 
└─nvme0n1p1 259:1  0 316.6M 0 part
nvme0n2   259:2   0 113.9M 0 disk
```

lsblk -o # покажет все параметры

lsblk -O # позволяет выбрать нужные

Файлы блочных устройств NVMe

```
/dev/nvme0n1  # контроллер 0 устройство 1
/dev/nvme0n1p1 # контроллер 0 устройство 1 раздел 1
```

Где еще искать файлы блочных устройств

```
ls /sys/block/  
dm-0  dm-1  nvme0n1  nvme0n2  sda  sdb  sr0  
  
ls /dev/disk/
```

Виртуальные диски

- Задачи:
 - получить из 3х дисков больший объем
 - из большого диска получить диски меньшего объема
- Варианты решения:
 - разделы,
 - RAID разделы,
 - тома LVM,
 - устройства device mapper
 - специальные файловые системы ZFS, BTRFS

Разбиваем диск на разделы

Зачем ?

Какие схемы разделки?

Директория Раздел

/ /dev/sda1

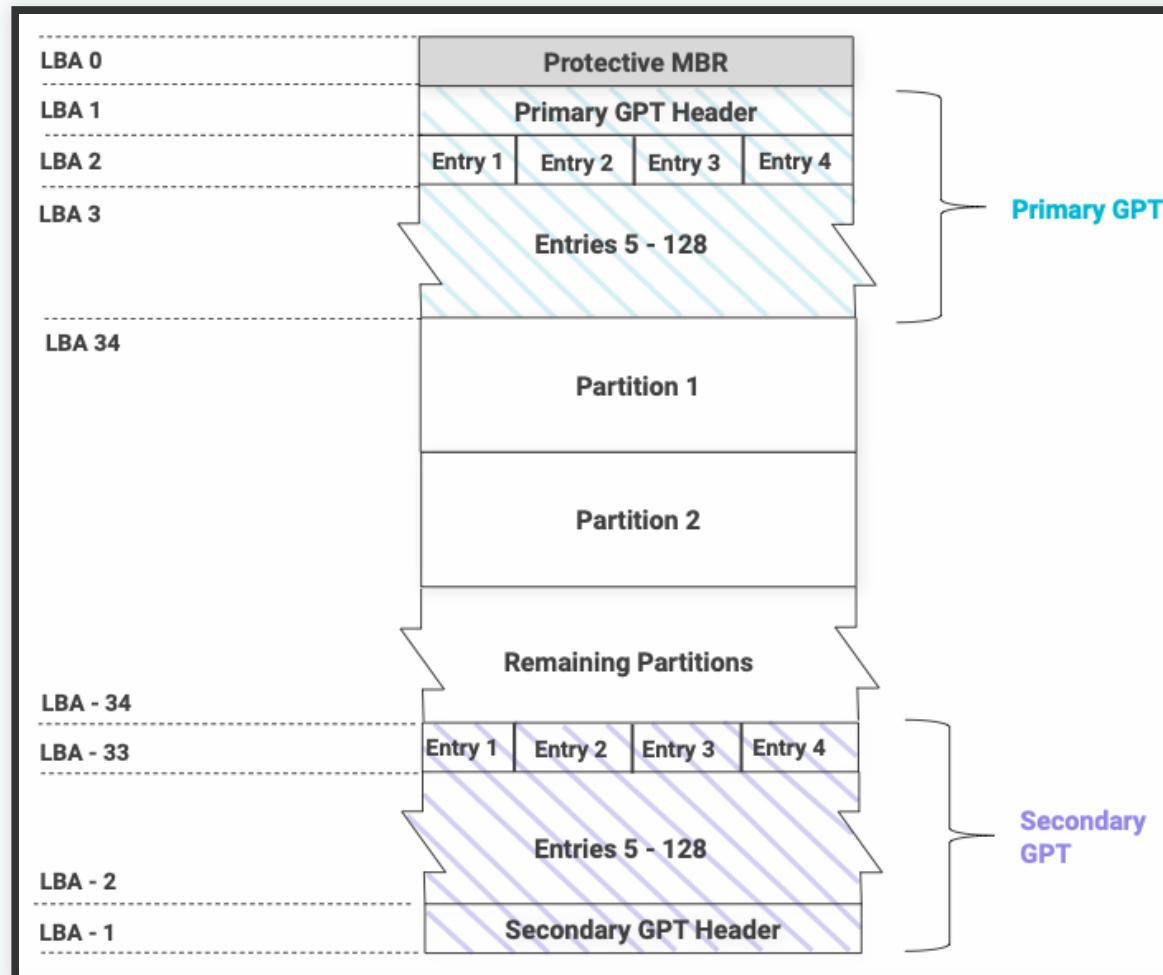
/boot /dev/sda2

/home /dev/sda3

Разделы MBR

- Преимущества MBR
 - совместима с большинством систем.
- Недостатки MBR
 - четыре раздела, дополнительные подразделы на основном
 - размер раздела два терабайта (512 байт сектор)
 - информация о разделе хранится только в одном месте

Разделы GPT



Разделы GPT

- Преимущества GPT
 - неограниченное количество разделов.
 - контрольные суммы CRC32 позволяют обнаружить ошибки и повреждения заголовка и таблицы разделов.
 - сохраняет заголовок резервной копии и таблицу разделов в конце диска.
 - доп информация о типе раздела и уникальный номер GUID раздела
- Недостатки GPT
 - Может быть несовместима со старыми системами.

Утилиты для создания разделов

Интерактивные Для скриптов

fdisk

sfdisk

gdisk

sgdisk

parted

parted

Пример интерактивного создания разделов

- `gdisk /dev/nvme0n2`

```
Number  Start (sector)    End (sector)  Size            Code  Name
      1                  2048          104447   50.0 MiB       8300  Linux fi

Command (? for help): ?
o      create a new empty GUID partition table (GPT)
b      back up GPT data to a file
p      print the partition table
l      list known partition types
n      add a new partition
d      delete a partition
i      show detailed information on a partition
```

```
lsblk
```

Примеры работы с parted в ручном режиме

```
# создаем GPT разделы
parted $DEVICE
(parted) mklabel gpt
(parted) mkpart LVM ext4 2048s 5%
(parted) help mkpart
(parted) help unit
```

```
# Создаем загрузочный раздел EFI
EFI/GPT example
(parted) mkpart "EFI system part" fat32 53.5MB 70MB
(parted) set 2 esp on
(parted) print
```

Пример создания разделов скриптом

```
for i in {1..10} ; do
    sgdisk -n ${i}:0:+10M /dev/nvme0n1
done
lsblk
```

Копия разделов

```
sgdisk -R /dev/nvme0n3 /dev/nvme0n1
sgdisk -G /dev/nvme0n3
```

Некоторые опции совпадают с интерактивной версией

```
sgdisk --help
sgdisk -o /dev/nvme0n2 # очистка разделов
```

Пример создания разделов parted

Интерактивно

```
parted /dev/nvme0n3
```

Скрипт

```
parted --script /dev/nvme0n3 mklabel gpt\  
mkpart primary 2048s 10MiB\  
mkpart primary 10MiB 20MiB
```

https://www.thomas-krenn.com/en/wiki/Partition_Alignment_detailed_explan

Время вопросов

Массив дисков RAID

Типы RAID

Redundant Array of Inexpensive Disks

Redundant Array of Independent Disks

- аппаратный
 - internal
 - external
- программный

Аппаратный RAID

Преимущества

- + Аппаратное решение, не влияющее на производительность основной системы
- + Выделенный CPU
- + Выделенная память для Кэшей
- + Возможность использовать BBU (Battery Backup Unit)
- + Возможность подключения большого количества дисков
- + Прозрачность для загрузчиков (возможность грузиться с любого массива)

Недостатки

- Высокая стоимость
- Высокая сложность
- Разнообразность интерфейсов управления и драйверов
- Низкая «мобильность» /переносимость
- Привязка к железу
- Большой простой по времени при аварии
- Очень дорогой ремонт, необходимость закупать впрок контроллеры, которые потом могут прекратить выпускать

Программный RAID

Преимущества

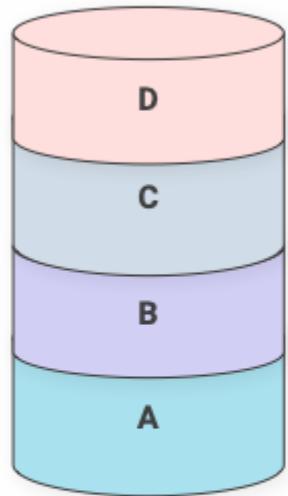
- + Бесплатно
- + Отсутствие привязки к конкретному железу
- + Прозрачность конфигурации
- + Примерно одинаковый интерфейс управления в любом linux.
- + Легкая переносимость между компьютерами.
- + Гибкость конфигурации

Недостатки

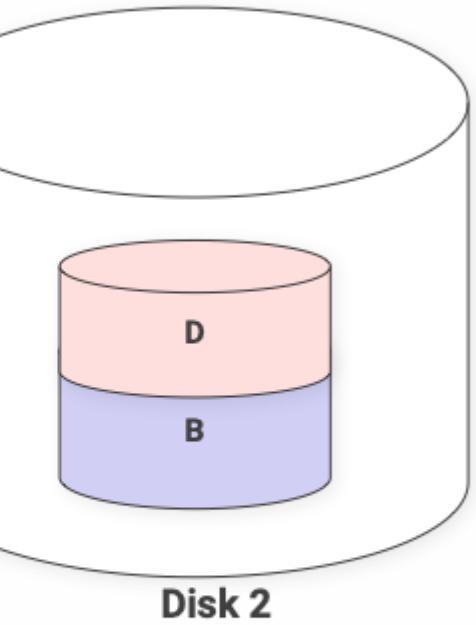
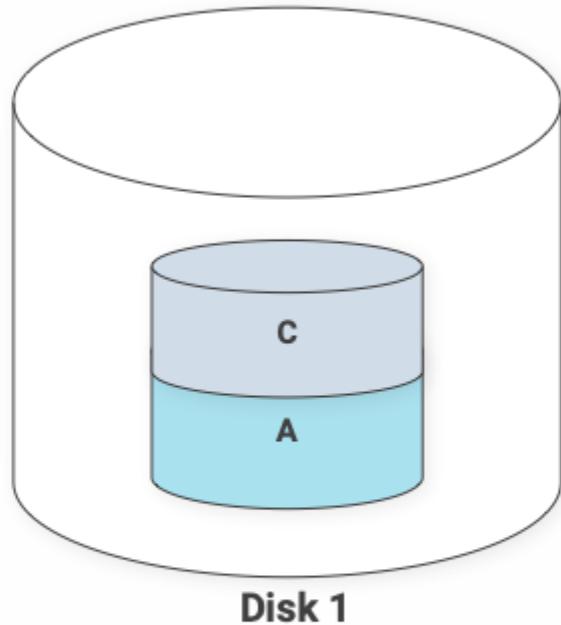
- Отсутствие BBU
- Отсутствие выделенного кэша
- Отсутствие службы поддержки :-)

RAIDO

Data Chunks



RAID-0



$$V_t = V_1 * N$$

RAIDO

Преимущества

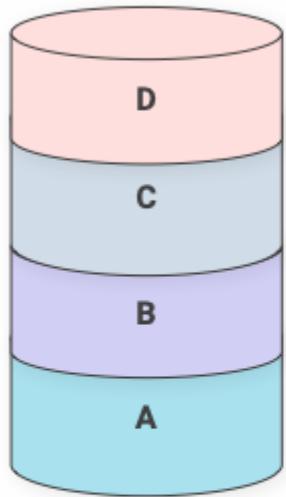
- + Самое быстрое чтение
- + Очень простой
- + Максимальная эффективность использования дискового пространства

Недостатки

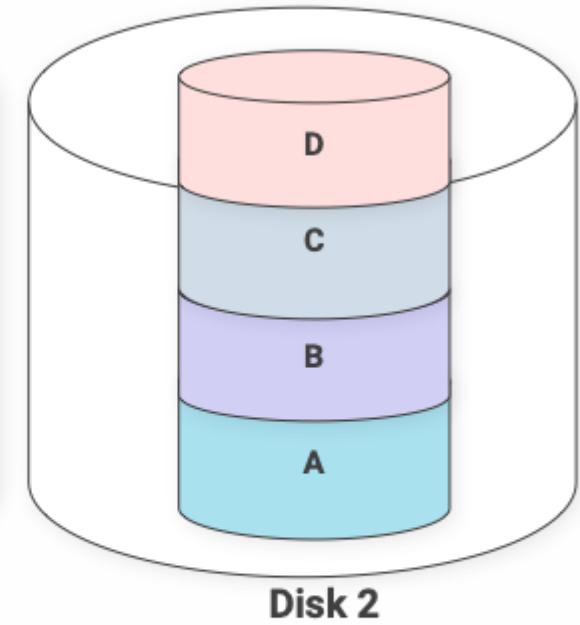
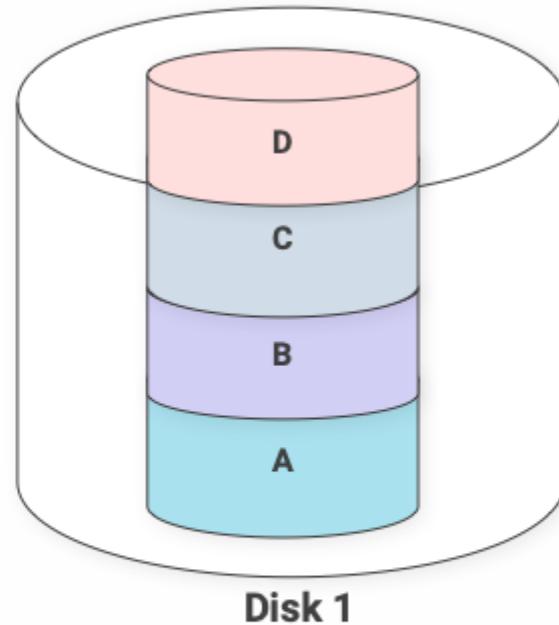
- Не «настоящий» RAID, нет отказоустойчивости: отказ одного диска влечет за собой потерю всех данных массива

RAID1

Data Chunks



RAID-1



$$V_t = V_1$$

RAID1

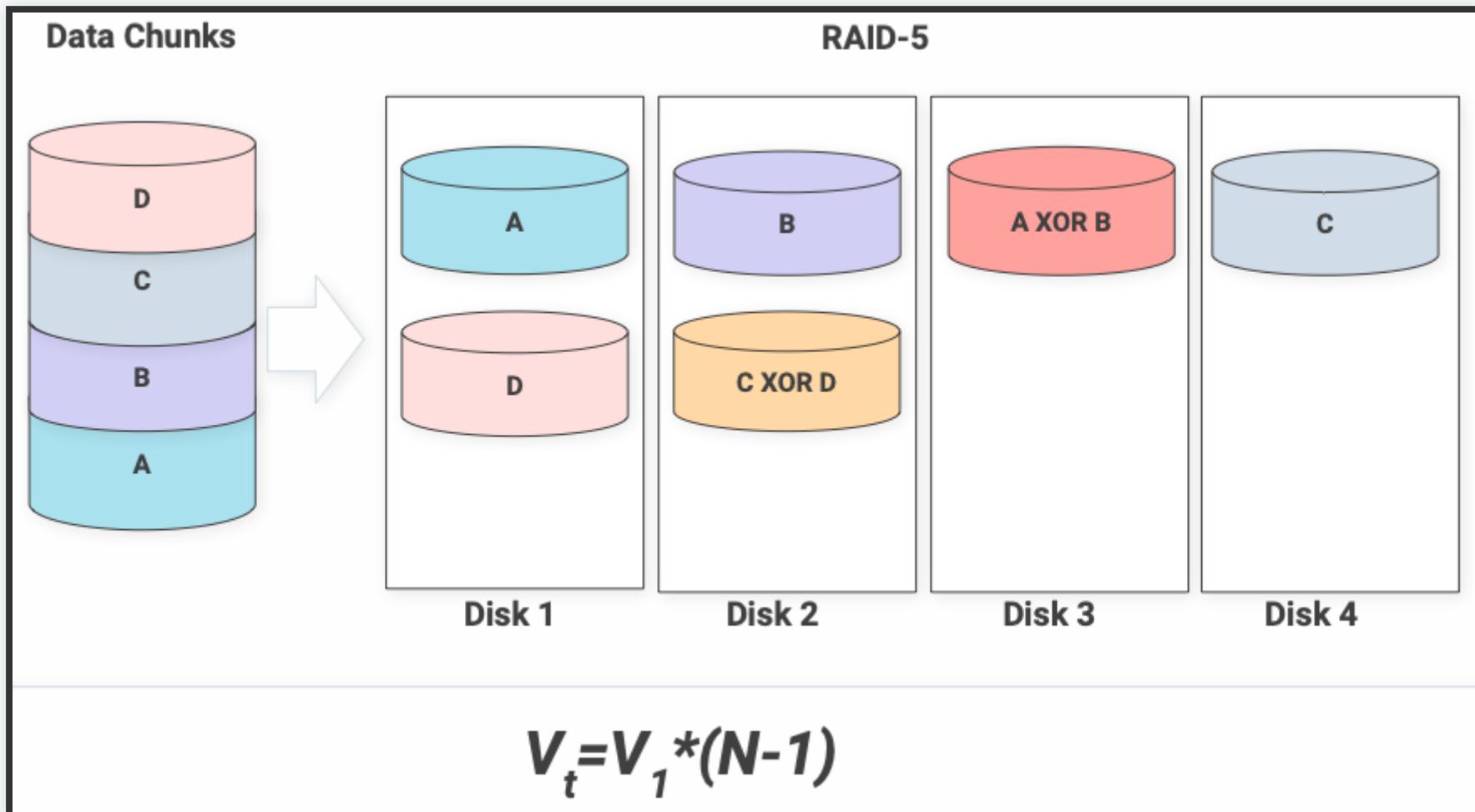
Преимущества

- + Простота реализации
- + Простота восстановления:
перекопировать все данные с
«выжившего» диска
- + Высокая скорость на чтение

Недостатки

- Высокая стоимость на единицу объема: 100% избыточность

RAID5



RAID5

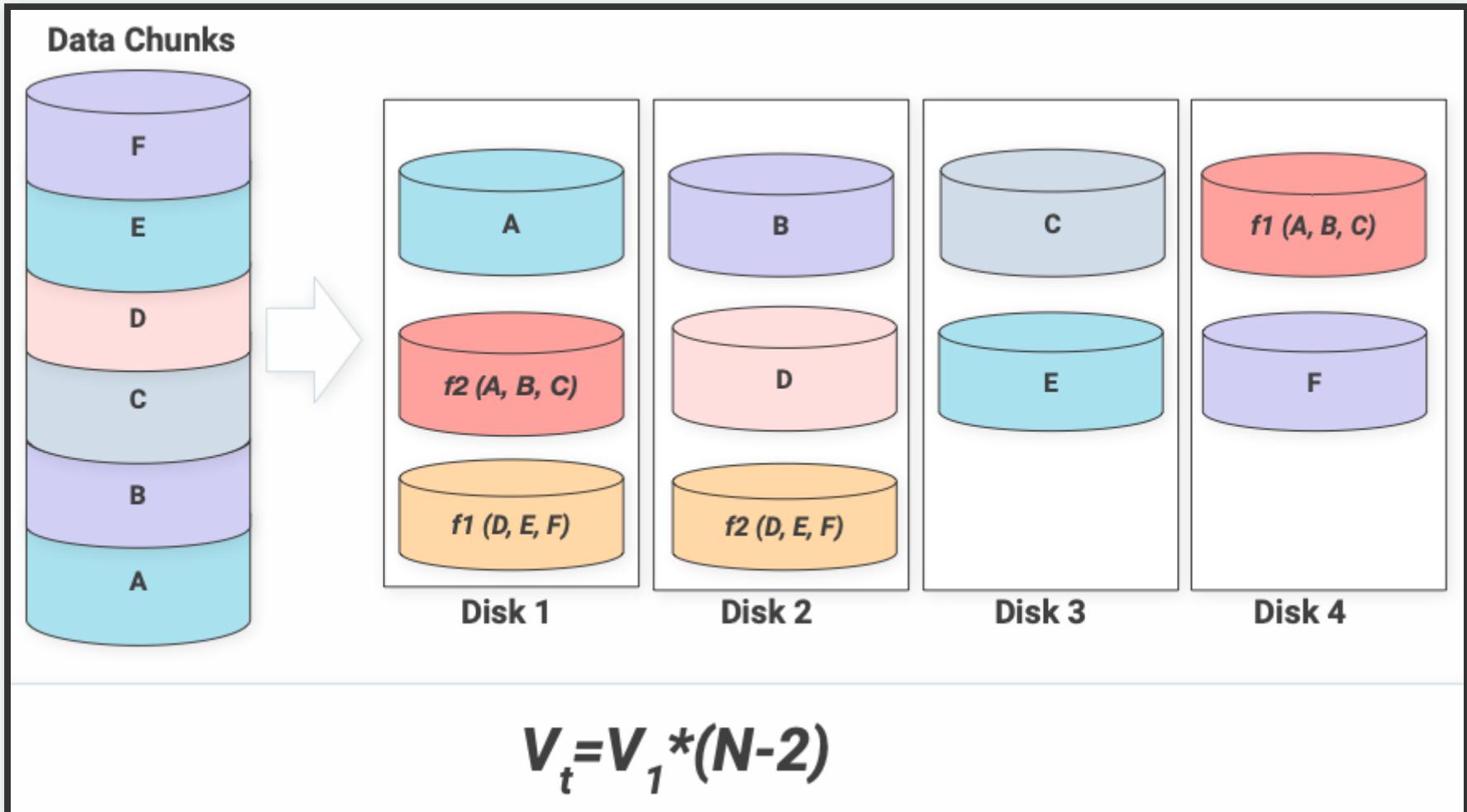
Преимущества

- + Высокая скорость записи данных
- + Достаточно высокая скорость чтения данных
- + Высокая производительность при большой интенсивности запросов чтения/записи данных
- + Малые накладные расходы для реализации избыточности

Недостатки

- Низкая скорость чтения/записи данных малого объема при единичных запросах
- Достаточно сложная реализация
- Сложное восстановление данных

RAID6



RAID6

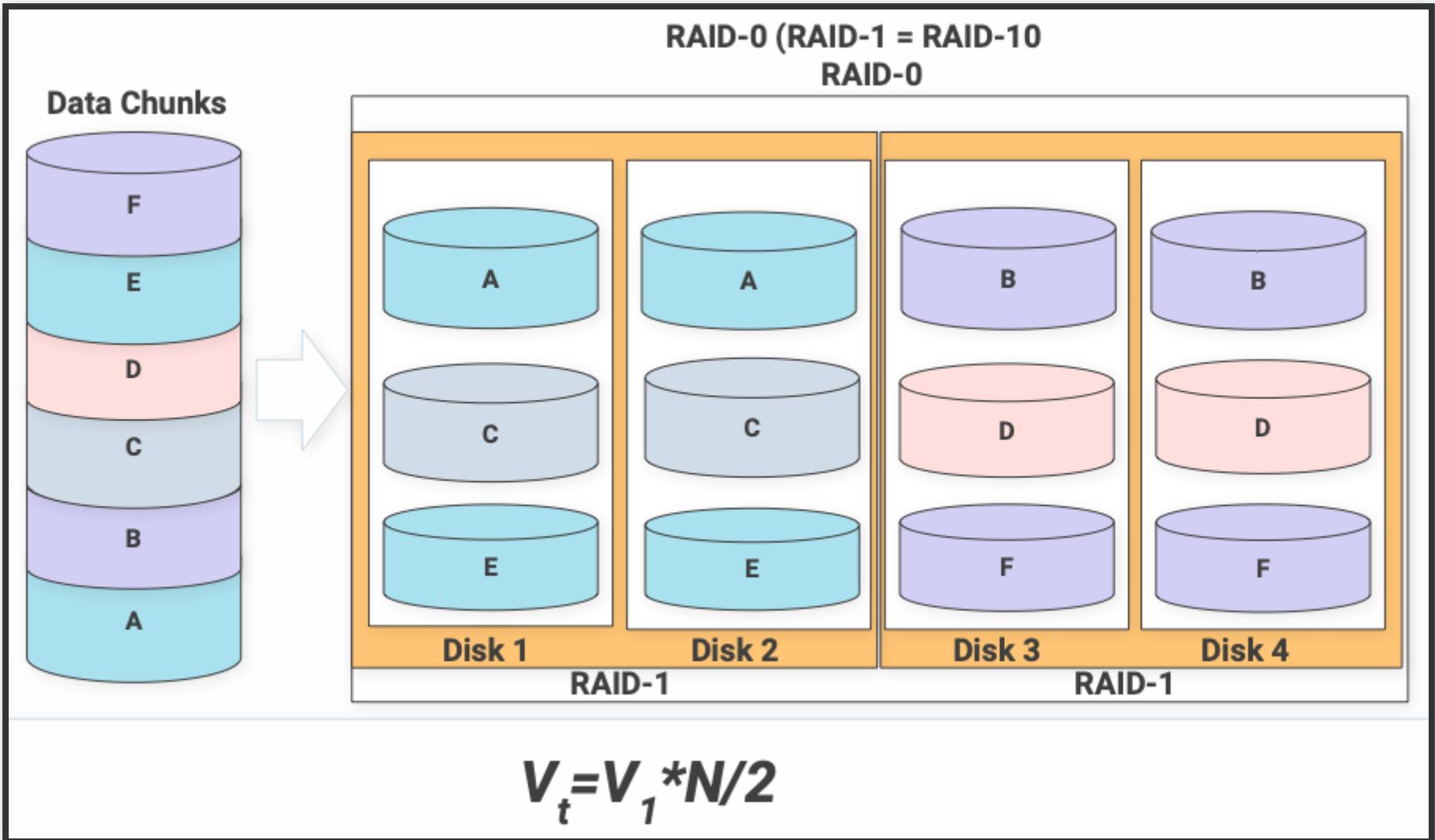
Преимущества

- + Высокая отказоустойчивость
- + Достаточно высокая скорость обработки запросов
- + Относительно малые накладные расходы для реализации избыточности

Недостатки

- Очень сложная реализация
- Сложное восстановление данных
- Очень низкая скорость записи данных

RAID10



RAID10

Преимущества

- + Самая высокая отказоустойчивость
- + Самая высокая производительность
- + Сочетает в себе преимущества R0 и R1

Недостатки

- Двойная стоимость пространства

Практика выбора RAID

Есть 8 дисков по 2 ТВ какой объем будет доступен в RAID6?

Практика выбора RAID

Есть 8 дисков по 2 ТВ какой объем будет доступен в RAID6?

<http://www.raid-calculator.com/>

https://en.wikipedia.org/wiki/Standard_RAID_levels#Co

- 2 диска
- 3 - 8 дисков
- больше 8 дисков

Создание программного RAID

На блочных устройствах

- тома LVM
- разделы (позволяют выравнивать диски разных размеров)
- диски

Метаданные (superblock)

- 0.9, 1.0 - конец устройства (необходимо для загрузки в некоторых случаях)
- 1.1 - начало
- 1.2 - 4К от начала устройства

Создание программного RAID

- Подготовка к созданию, “занулить суперблок”
 - `mdadm --zero-superblock $dev_list`
- Создание массива
 - `mdadm --create $raiddev -l $level -n $numdev $dev_list`
- Остановка массива
 - `mdadm --stop $raiddev`

Опции утилиты mdadm

- Информация о массиве
 - `mdadm --detail $raiddev`
- Информация о массиве
 - `cat /proc/mdstat`
- Генерация данных для конфигурационного файла
 - `mdadm --examine --scan`
 - `mdadm --detail --scan`

Время вопросов

Дополнительные настройки дисковой системы

Планировщик системы ввода/ вывода

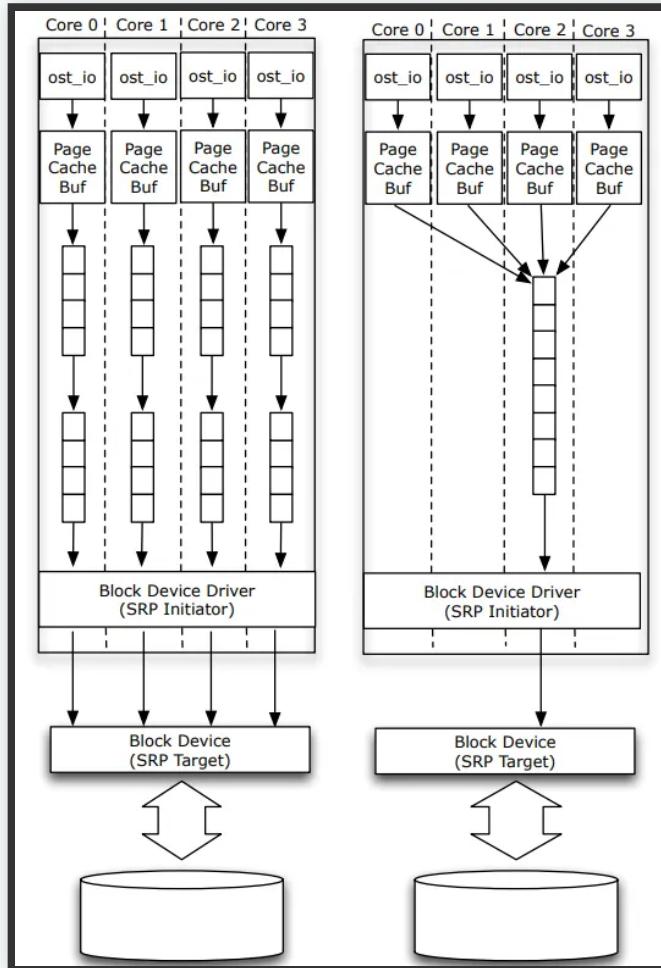
I/O Scheduling – очередью операций ввода-
вывода к жесткому диску

Увеличение производительности дисков

Как это работает:

- слияние запросов
- установка приоритета выполнения

Планировщик multi queue vs single queue



Планировщики системы ввода/вывода

Non-multiqueue I/O schedulers

noop

deadline

cfq

Multiqueue

mq-deadline

kyber

bfq

Узнаем какие поддерживаются

```
# cat /sys/block/sda/queue/scheduler  
[mq-deadline] kyber bfq none
```

Пример смены планировщика

```
# echo none >/sys/block/sda/queue/scheduler  
# cat /sys/block/sda/queue/scheduler  
[none] mq-deadline kyber bfq
```

Рекомендации по выбору. Их нет. Тестируйте.

- NVMe
 - любой из Multiqueue
 - none для уменьшения нагрузки на процессор.
- HDD
 - сервер - mq-deadline
 - desktop - bfq

Пример правил системы udev

Как создаются эти имена файлов?

```
find /dev/disk/  
/dev/disk/by-label/CentOS-8-1-1911-x86_64-dvd
```

```
cat /usr/lib/udev/rules.d/40-elevator.rules  
cat /usr/lib/udev/rules.d/60-persistent-storage.rules
```

- правила синтаксис man 7 udev
 - There are two kinds of keys: **match** and **assignment** or "actions"
- файлы с описанием правил:
 - system rules directory /usr/lib/udev/rules.d
 - volatile runtime directory /run/udev/rules.d
 - local administration directory /etc/udev/rules.d

Пример добавления правил

Создадим еще одно имя для диска sda

```
udevadm info -n /dev/sda  
udevadm info -n /dev/sda | grep ID_SERIAL_SHORT
```

Создаем файл с правилом

```
# cat /etc/udev/rules.d/69-disk.rules  
ACTION=="add", KERNEL=="sd[a-z]", \  
ENV{ID_SERIAL_SHORT}=="VBca5e42b9-32e027c5", SYMLINK+="my_virtual,
```

Тестируем правило, применяем правило

```
udevadm test $(udevadm info \  
--query=path --name=/dev/sda)  
sudo udevadm trigger --action=add
```

Пример правила для установки планировщика

```
cat /sys/block/sdd/queue/scheduler  
udevadm info --name=/dev/sdd | grep -E '(WWN|SERIAL)'  
E: ID_WWN=0x5002538d00000000  
E: ID_SERIAL=Generic- SD MMC_20120501030900000-0:0  
E: ID_SERIAL_SHORT=20120501030900000
```

```
# file /etc/udev/rules.d/99-scheduler.rules  
ACTION=="add|change", SUBSYSTEM=="block", \  
ENV{ID_SERIAL}=="VBOX_HARDDISK_disk3_0000000000000000",  
\ ATTR{queue/scheduler}="bfq"
```

```
udevadm control --reload-rules  
udevadm trigger --type=devices --action=change  
cat /sys/block/device/queue/scheduler
```

Система udev

- правила именования устройств
- постоянно закрепленные за устройствами имена, которые не зависят от того, какое положение они занимают в дереве устройств
- уведомление внешних по отношению к ядру программ, если устройство было заменено

Найдем контроллеры дисков

Команда `lspci`

```
0000:00:1f.2  
00:1f.2
```

B/D/F (bus/device/function)
(шина / устройство / функции).

```
# Опции команды  
lspci -v -vv -vvv # больше деталей  
lspci -tv # дерево соответствует физической структуре  
lspci -D # показывает домен  
lspci -nnA # показывает ID производителя  
lspci -v -s 00:1f.2 # показать только одно устройство
```

Пример использования утилиты hdparm и udev

```
# Тест производительности  
hdparm -Tt --direct /dev/nvme0n1  
# Отключаем энергосбережение  
/usr/bin/hdparm -B 254 -S 0 /dev/sda
```

Применяем после перезагрузки

```
ACTION=="add", SUBSYSTEM=="block", KERNEL=="sda",  
RUN+="/usr/bin/hdparm -B 254 -S 0 /dev/sda"
```

Пример использования smartctl для конфигурации диска

- Две стратегии поведения диска при обнаружении ошибки:
 - **standalone/desktop** – пытаться прочитать до последнего
 - **RAID** – не ждать помечать диск как сбойный
- Разные производители называют по разному
 - **SCT ERC** (SMART Command Transport Error Recovery Control), **TLER** (Time-Limited Error Recovery), **CCTL** (Command Completion Time Limit)
 - **Error Recovery Control (ERC)** - количество времени которое контроллеру диска разрешено потратить на восстановление от ошибки чтения или записи.

Пример использования утилиты smartctl

Узнать значение ERC

```
smartctl -l scterc /dev/sda
SCT Error Recovery Control:
    Read: Disabled
    Write: Disabled
```

Установить значение ERC

```
smartctl -l scterc,150,150 /dev/sda
SCT Error Recovery Control:
    Read:      150 (15.0 seconds)
    Write:     150 (15.0 seconds)
```

Дополнительная информация о дисках

- Утилиты
 - hdparam
 - sdparm (SCSI)
 - smartctl

Утилита hdparm

hdparm - позволяет получить/установить параметры устройств SATA/IDE и тестировать производительность

- кэши дисков (drive caches)
- режим сна (sleep mode)
- управлять питанием (power management)
- уровнем шума (acoustic management)
- настройки DMA (Direct Memory Access)

Утилита smartctl

управление и мониторинг дисков использующих
технологию SMART (Self-Monitoring, Analysis and
Reporting Technology System)

- поддерживает ATA/SATA, SCSI/SAS and NVMe
- работает в системах Linux, FreeBSD, NetBSD, OpenBSD,
Darwin (macOS), Solaris, Windows, Cygwin, OS/2,
eComStation or QNX

Примеры использования

```
sudo smartctl -a /dev/sda
```

https://en.wikipedia.org/wiki/S.M.A.R.T.#ATA_S.M.A.R.T

Примеры использования для определения размера сектора

Advanced Sector Format of Block Devices

```
sudo hdparm -I /dev/sda | grep Sector  
Logical/Physical Sector size: 512 bytes
```

```
sudo smartctl -a /dev/sda | grep Sector  
Sector Size: 512 bytes logical/physical
```

Навязчивое повторение

КОНТРОЛЬНЫЕ ВОПРОСЫ?

- можно ли использовать диски разных размеров в программном RAID?
- Как определить количество, тип дисков?

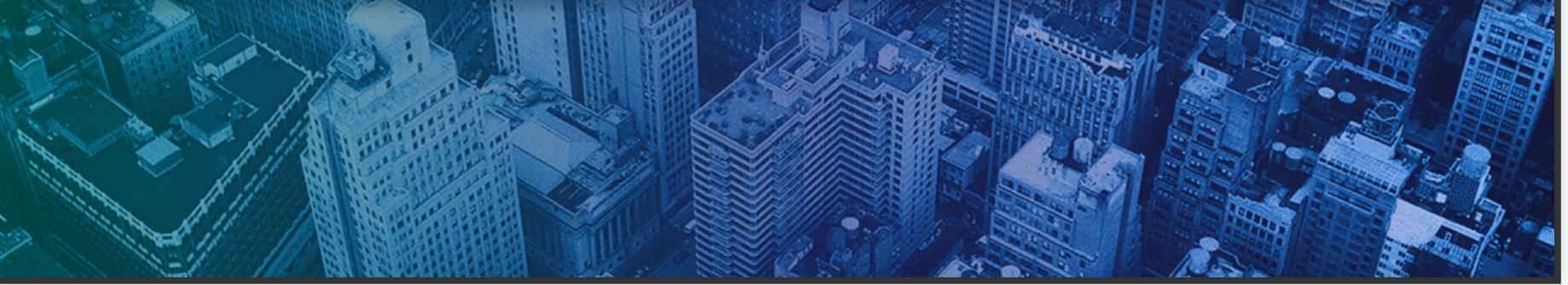
Рефлексия



Отметьте 3 пункта, которые вам запомнились с вебинара



Что вы будете применять в работе из сегодняшнего вебинара?



Заполните, пожалуйста,
опрос о занятии по ссылке в чате