

+



Southbridge

Тема №3: Работайте как Southbridge. Технический регламент

Тема №3: Работайте как Southbridge. Технический регламент

- ➊ Основы настройки выделенных серверов и тюнинг для максимальной производительности
- ➋ Основы работы с OpenVZ/KVM. Установка, настройка, эксплуатация
- ➌ Работа с аппаратными RAID-контроллерами



+



Southbridge

Основы настройки выделенных серверов

Почему первичная настройка важна? И зачем трогать то, что работает?



Преимущества которые дает первичная настройка:

- ➊ Установка проверенного ПО
- ➋ Мониторинг
- ➌ Проверка на вредоносное ПО
- ➍ Применение стандартов
- ➎ Ускорение процессов
- ➏ Оптимизация сервера



Почему первичная настройка важна? И зачем трогать то, что работает?



Как производится первичная настройка:

1. Временный сервер
2. Перенос проекта на временный сервер.
3. Настройка нового сервера
4. Миграция проекта на новый сервер



Настройка сервера



Требования

- Проверить корректность сборки RAID и разбивки пространства

```
/boot 400M ext2
/
20G ext4
swap XG swap
/vz ALL ext4
```
- Настроить уведомления от хостинг провайдера
- Создать задачу по настройке сервере в системе управления проектами
- Прикрепить к задаче чек-лист по настройке
- PTR запись
- Hostname (dsXX.example.com / vsXX.example.com / mysql-a.example.com / kube-master.example.com)
- Регистрация сервера в системе управления конфигурациями
- Настройка мониторинга
- Обновление ОС до последней стабильной версии, установка репозиториев epel и
<http://rpms.southbridge.ru>
- Инициализация smartmontools

```
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sda && \
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sdb && \
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sdc && \
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sdd
```
- Устанавливаем параметры разделов

```
/dev/md2 / ext4 defaults,noatime 0 0
/dev/md3 /vz/private ext4 defaults,noatime 0 0
```
- Настройка резервного копирования

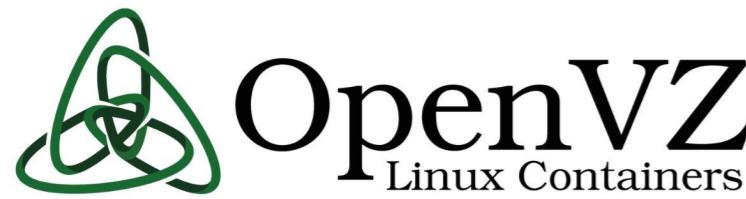


+



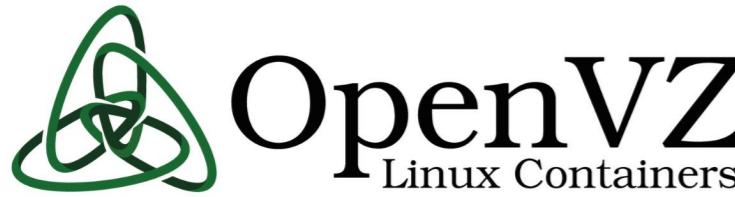
Southbridge

Основы работы с OpenVZ/ KVM. Установка, настройка, эксплуатация



- **Виртуализация и изоляция**
Файлы, пользователи и группы, дерево процессов, сеть, устройства и IPC
- **Управление ресурсами**
- **Двухуровневая дисковая квота**
- **Честный планировщик процессора (cpunits, cpulimit)**
- **Масштабируемость**
- **Плотность**





Создание контейнера

Скачиваем шаблон

```
cd /tmp && wget http://url (нужного темплейта)
tar --numeric-owner -xvf ./vz-rhel7-php56-maria102.tgz
rm vz-rhel7-php56-maria102.tgz
```

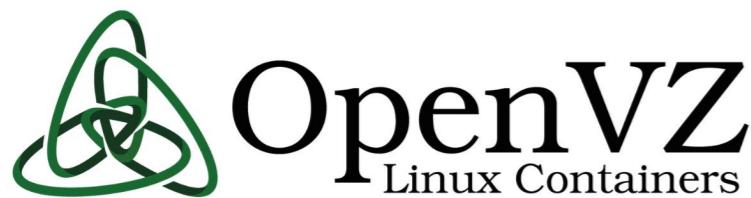
Перемещаем файлы шаблона на свои места

```
mv /tmp/vz/private/117 /vz/private/<уникальный id>
mv /tmp/etc/vz/conf/117.conf /etc/vz/conf/<уникальный id>.conf
```

Создаем root директорию контейнера и запускаем

```
mkdir /vz/root/<уникальный id>
vzctl start <уникальный id>
```





Эксплуатация

vzctl

```
[host-node]# vzctl create 101 --ostemplate centos-7-x86_64-minimal  
[host-node]# vzctl set 101 --ipadd 10.1.2.3 --save  
[host-node]# vzctl set 101 --nameserver 10.0.2.1 --save  
[host-node]# vzctl start 101
```

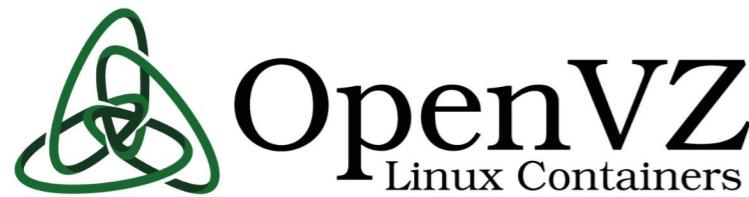
vzctl exec

```
[host-node]# vzctl exec CTID uptime  
19:05:37 up 431 days, 3:35, 0 users, load average: 0.75, 0.84, 0.77
```

vzctl enter

```
[host-node]# vzctl enter CTID  
entered into container CTID  
[container]#
```





Эксплуатация

vzctl stop

```
[host-node]# vzctl stop CTID
Stopping container ...
Container was stopped
Container is unmounted
```

vzctl destroy

```
[host-node]# vzctl destroy CTID
Destroying container private area: /vz/private/CTID
Container private area was destroyed
```

vzlist

```
[host-node]# vzlist (-a / -S)
CTID      NPROC STATUS      IP_ADDR          HOSTNAME
      1        515 running    178.0.0.0    vs01.example.com
      5        336 running    178.0.0.1    vs02.example.com
```





OpenVZ
Linux Containers

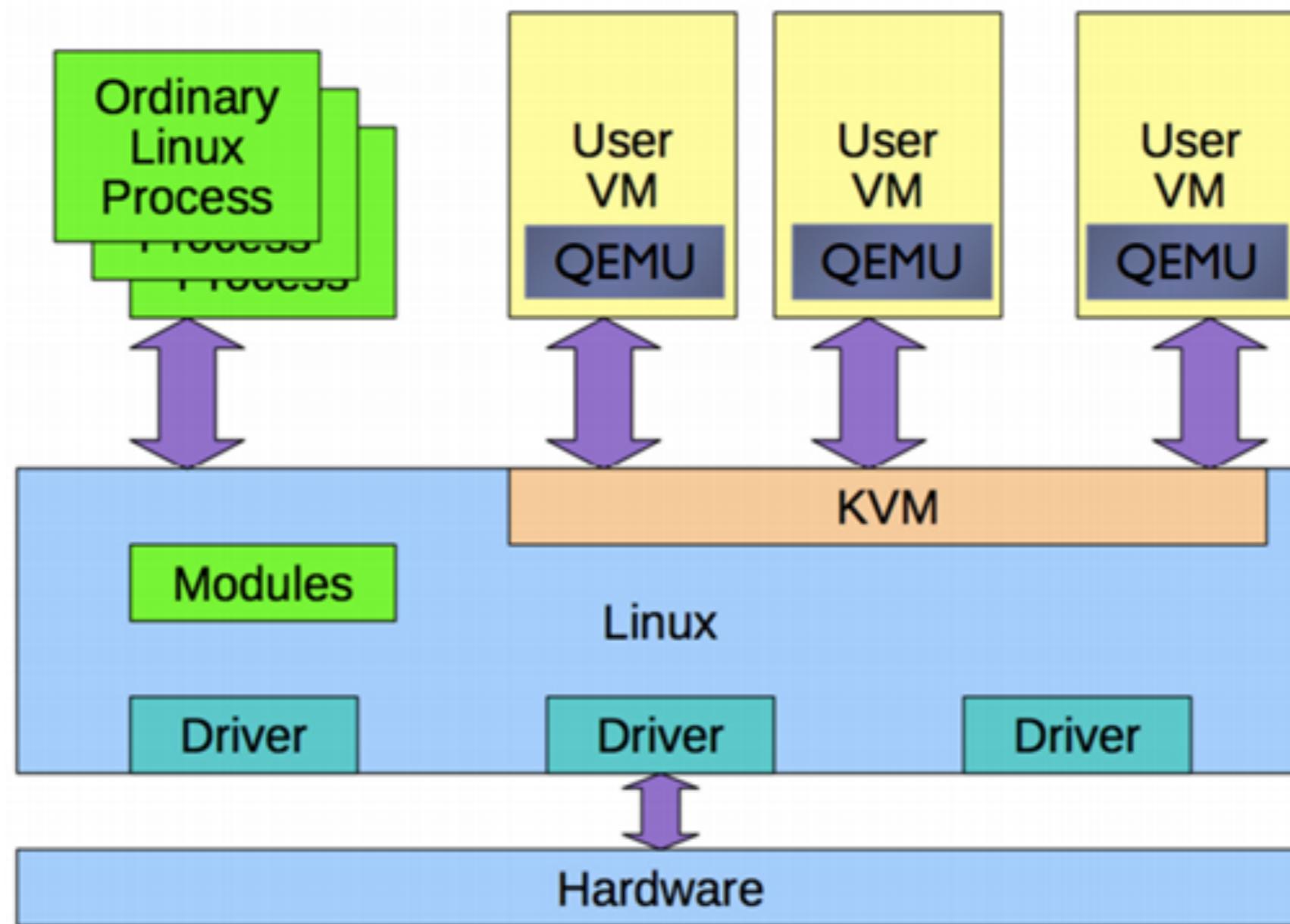
Эксплуатация

Дополнительные утилиты

- **vzmigrate**
- **vzcfgvalidate**
- **vzmemcheck, vzcpuchek, vzcalc**
- **vzspli**
- **vzpid**
- **vzquota**
- **vzubc**



iKVM





Virtual Machine Manager: <https://virt-manager.org>

virsh

Преимущества KVM:

- Независимо распределяемые ресурсы.
- Широкая поддержка гостевых ОС.
- Взаимодействие с ядром
- Наличие финансовой поддержки
- Простое администрирование

Недостатки KVM:

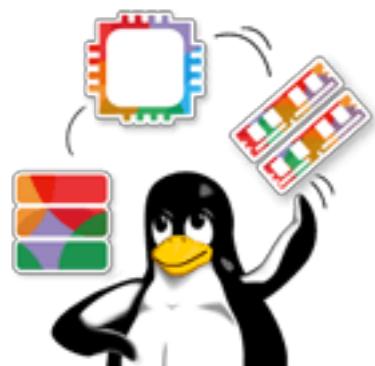
- Относительная молодость гипервизора
- Сложность настроек





ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ И СВОЙСТВА ГИПЕРВИЗОРА

- | Безопасность
- | Управление памятью
- | Хранение данных
- | Динамическая миграция
- | Производительность и масштабируемость





Развертывание гостевой системы

Схема разделов

```
● root (/) 8G;
● swap 32..64G;
● tmp (/tmp) 2G;
● home (/home) 8G;
● images (/var/lib/libvirt/images) 10G – если планируете
  использовать suspend to disk при выключении хоста, то
  размер этого раздела должен быть больше объёма RAM;
● logs (/var/log) 2G
```



Список Ansible ролей

```
roles:
  - [...]
  - bridge
  - nat
  - kvm
```





Развертывание гостевой системы

Настройка моста

bridge_devices:

```
- dev: "br0" # Имя устройства
  bootproto: "static" # Протокол поднятия адреса (static|dhcp|none)
  address: "192.0.2.12/24" # IP-адрес (CIDR)
  peer: "192.0.2.1" # scope peer (применяется при prefix /32 – например, в
Hetzner)
  onboot: "yes" # (default: yes) Поднимать ли интерфейс при загрузке (yes|no)
  stp: "on" # (default: off) Включать ли STP (on|off)
  delay: "1.0" # (default: 2.0) Время, в течение которого мост при поднятии
слушает трафик без форвардинга
  ports: [ eth0, eth1 ] # Интерфейсы, которые должны быть включены в мост*
  routes: # Описание добавляемого маршрута (маршрутов)
    - network: "default" # Сеть (напр., default, 0.0.0.0/0, 192.0.2.0/24)
      gateway: "192.0.2.1" # Шлюз, через который доступна эта сеть
      metric: "10" # Метрика для этого маршрута (при установке default сделать
больше, чем существующая!)
```





Развертывание гостевой системы

Конфигурация трансляции адресов

```
nat_network: "172.30.2.0/24"  
nat_interface: "br0"  
nat_ip_address: "192.0.2.4"
```

Конфигурация доступа

```
ansible_ssh_proxy_internal_address: "172.30.2.1"
```

Конфигурация пула хранения (LVM)

```
kvm_pool:  
  - kvm_pool_name: "main"  
    kvm_pool_type: "logical"  
    kvm_pool_path: "/dev/main"  
    kvm_pool_state: "active"
```

Конфигурация пула хранения (ФС)

```
kvm_pool:  
  - kvm_pool_name: "images"  
    kvm_pool_type: "dir"  
    kvm_pool_path: "/var/lib/libvirt/images"  
    kvm_pool_state: "active"
```





Развертывание гостевой системы

Конфигурация виртуальных машин

```
kvm_vm:  
  - kvm_vm_name: "guestname"  
    kvm_vm_state: "running"  
    kvm_vm_cpu_count: 1  
    kvm_vm_memory_mb: 1024  
    kvm_vm_nic_bridge: "br1"  
    kvm_vm_pool_name: "main"  
    kvm_vm_template_name: "vm_template_c7_sb.qcow2"  
    kvm_vm_disk_gb: 10  
    kvm_vm_disk_format: "raw"  
    kvm_vm_sysprep_ifcfg:  
      - sysprep_if_name: "eth0"  
        sysprep_if_bootproto: "static"  
        sysprep_if_address: "172.30.2.10/24"  
        sysprep_if_gateway: "172.30.2.1"  
  - kvm_vm_name: "otherguestname"  
    kvm_vm_state: "running"  
    [...]
```





Ручное управление ВМ (virsh)

Команда	Description
help	Краткая справка.
list	Просмотр всех виртуальных машин.
dumpxml	Вывести файл конфигурации XML для заданной виртуальной машины.
create	Создать виртуальную машину из файла конфигурации XML и ее запуск.
start	Запустить неактивную виртуальную машину.
destroy	Принудительно остановить работу виртуальной машины.
define	Определяет файл конфигурации XML для заданной виртуальной машины.
domid	Просмотр идентификатора виртуальной машины.
domuuid	Просмотр UUID виртуальной машины.
dominfo	Просмотр сведений о виртуальной машине.
domname	Просмотр имени виртуальной машины.
domstate	Просмотр состояния виртуальной машины.
quit	Закрыть интерактивный терминал.
reboot	Перезагрузить виртуальную машину.
restore	Восстановить сохраненную в файле виртуальную машину.
resume	Возобновить работу приостановленной виртуальной машины.
save	Сохранить состояние виртуальной машины в файл.
shutdown	Корректно завершить работу виртуальной машины.
suspend	Приостановить работу виртуальной машины.
undefine	Удалить все файлы виртуальной машины.
migrate	Перенести виртуальную машину на другой узел.



Ручное управление ВМ (virsh)

Подключение к гипервизору

Подключение к сессии гипервизора с помощью **virsh** :

```
# virsh connect {hostname OR URL}
```

Создание XML-файла конфигурации виртуальной машины

Выдаст файл конфигурации виртуальной машины:

```
# virsh dumpxml {domain-id, domain-name or domain-uuid}
```

Создание виртуальной машины на основе файла конфигурации

Виртуальный сервер может быть воссоздан из полученного ранее дампа конфигурации

```
# virsh create configuration_file.xml
```





Ручное управление ВМ (virsh)

Редактирование файла конфигурации виртуальной машины

```
# virsh edit softwaretesting
```

Откроется окно текстового редактора, заданного переменной оболочки **\$EDITOR** (по умолчанию используется **vi**).

Приостановка виртуальной машины

Команда приостановки виртуальной машины с помощью **virsh**:

```
# virsh suspend {domain-id, domain-name or domain-uuid}
```

Возобновление работы виртуальной машины

Возобновить работу приостановленной виртуальной машины можно с помощью параметра **resume** команды **virsh**:

```
# virsh resume {domain-id, domain-name or domain-uuid}
```





Ручное управление ВМ (virsh)

Сохранение виртуальной машины

Команда сохранения текущего состояния виртуальной машины:

```
# virsh save {domain-name, domain-id or domain-uuid}  
filename
```

Восстановление виртуальной машины

```
# virsh restore filename
```

Сохраненная машина будет восстановлена из файла и перезапущена, что может занять некоторое время. Имя и идентификатор UUID виртуальной машины останутся неизменными, но будет предоставлен новый идентификатор домена.

Завершение работы виртуальной машины

Команда завершения работы:

```
# virsh shutdown {domain-id, domain-name or domain-uuid}
```

Поведение выключаемого гостя можно контролировать с помощью параметра `on_shutdown` в его файле конфигурации.





Ручное управление ВМ (virsh)

Перезагрузка виртуальной машины

Команда перезагрузки:

```
#virsh reboot {domain-id, domain-name or domain-uuid}
```

Поведение перезагружаемого гостя можно контролировать с помощью параметра **on_reboot** в его файле конфигурации.

Принудительная остановка виртуальной машины

Команда принудительной остановки:

```
# virsh destroy {domain-id, domain-name or domain-uuid}
```

Определение идентификатора домена

Команда определения идентификатора домена виртуальной машины:

```
# virsh domid {domain-name or domain-uuid}
```





Ручное управление ВМ (virsh)

Получение информации о виртуальной машине

Команда для получения информации:

```
# virsh dominfo {domain-id, domain-name or domain-uuid}
```

Пример вывода **virsh dominfo**:

```
# virsh dominfo r5b2-mySQL01
id:          13
name:        r5b2-mysql01
uuid:        4a4c59a7-ee3f-c781-96e4-288f2862f011
os type:    linux
state:       blocked
cpu(s):      1
cpu time:   11.0s
max memory: 512000 kb
used memory: 512000 kb
```





Ручное управление ВМ (virsh)

Получение информации об узле

Команда получения информации об узле:

```
# virsh nodeinfo
```

Пример вывода **virsh nodeinfo**:

```
# virsh nodeinfo
CPU model                  x86_64
CPU (s)                    8
CPU frequency              2895 Mhz
CPU socket(s)              2
Core(s) per socket          2
Threads per core:          2
Numa cell(s)                1
Memory size:               1046528 kb
```

Вывод содержит информацию об узле и машинах, поддерживающих виртуализацию.





Ручное управление ВМ (virsh)

Просмотр списка виртуальных машин

Команда для просмотра списка виртуальных машин и их состояния:

```
# virsh list
```

Можно добавить аргументы:

--inactive покажет список неактивных доменов (неактивным считается тот домен, который был определен, но в настоящий момент не является активным).

--all покажет все виртуальные машины независимо от их состояния. Пример:

```
# virsh list --all
```

ID	Name	State
0	Domain-0	running
1	Domain202	paused
2	Domain010	inactive
3	Domain9600	crashed



+



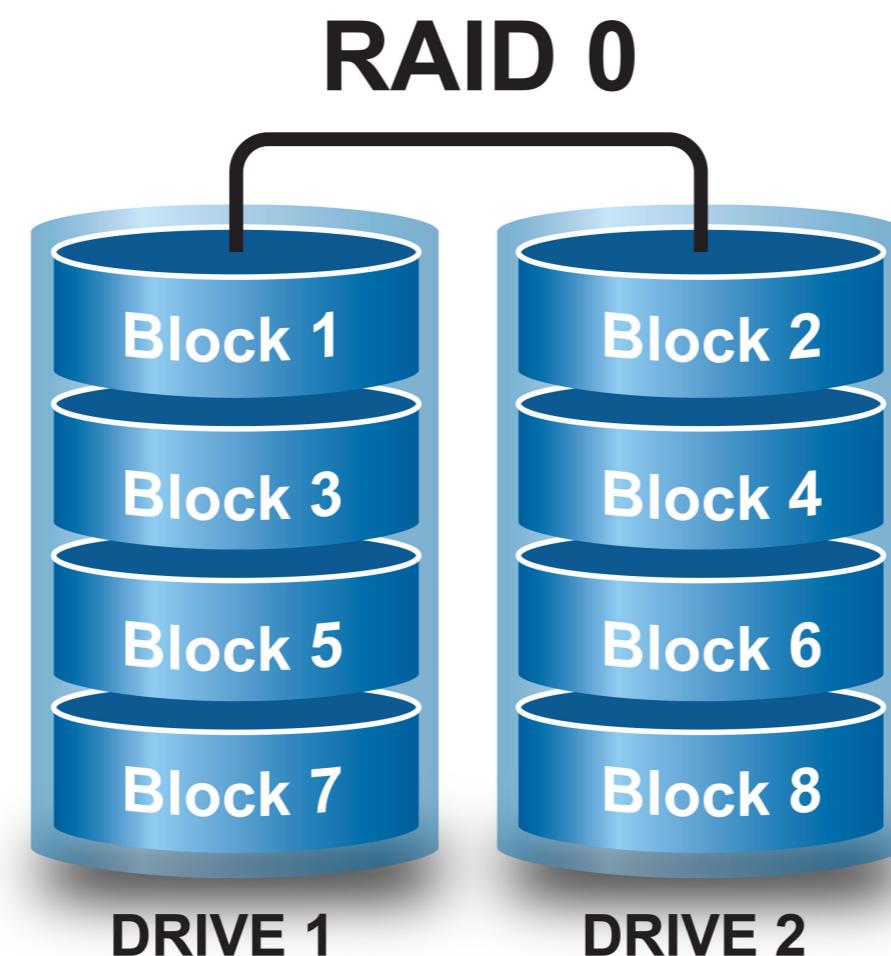
Southbridge

Работа с аппаратными RAID- контроллерами

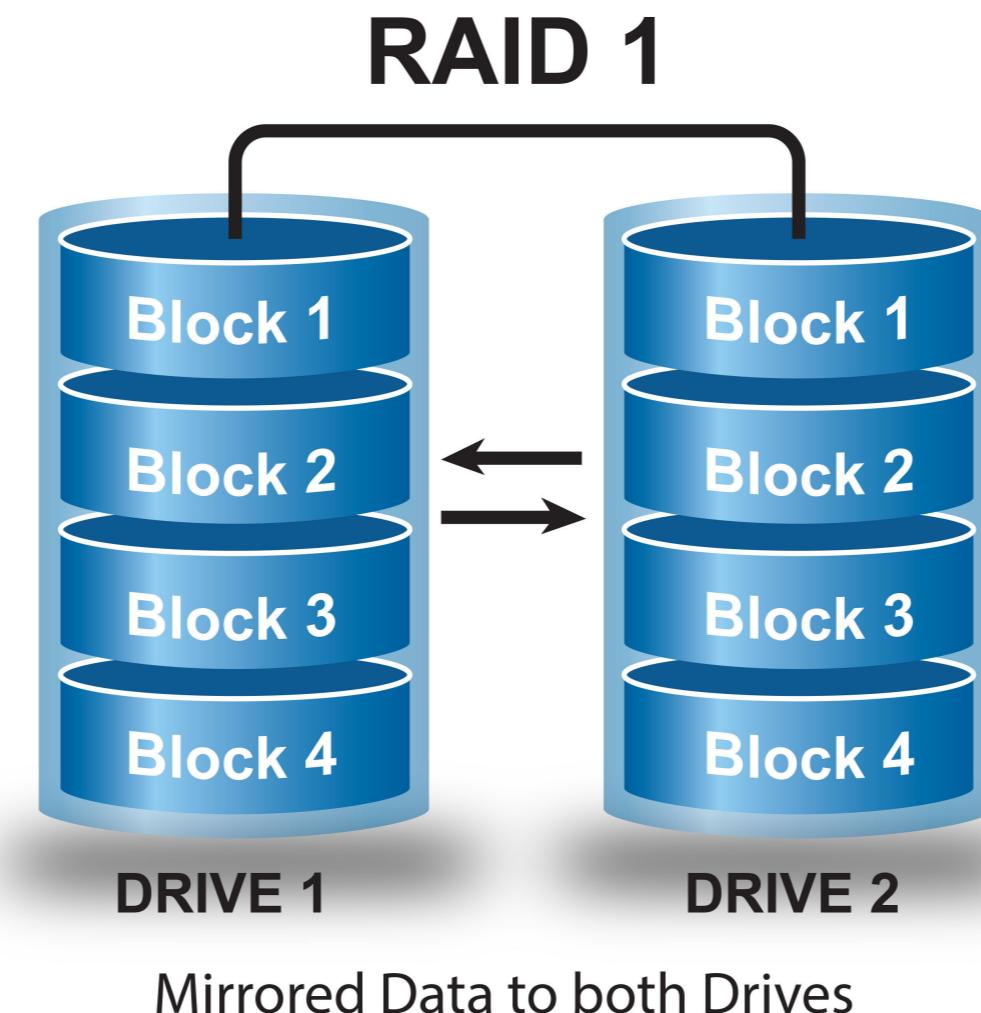
RAID



RAID

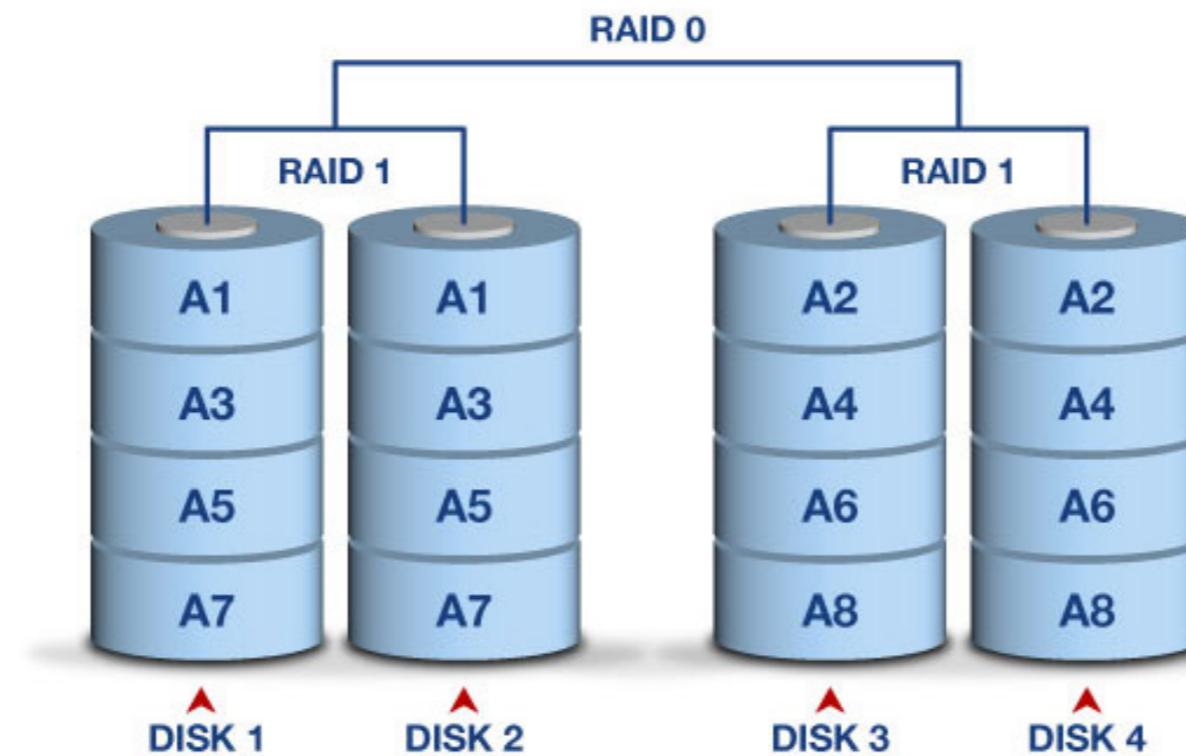


RAID



RAID

RAID 10



■ Hardware RAID controller





Устанавливаем утилиту arrconf

```
cd /tmp
yum install -y compat-libstdc++-33 unzip
wget http://download.adaptec.com/raid/
storage_manager/arcconf_v1_7_21229.zip
unzip -j arcconf_v1_7_21229.zip linux_x64/cmdline/
arcconf -d /usr/bin/
chmod +x /usr/bin/arcconf
```

Включаем smart

```
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sg0
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sg1
smartctl --smart=on --offlineauto=on --saveauto=on /dev/sg2
...
```

Фиксируем в конфиге /etcSMARTD.conf

```
/dev/sg0 -H -m root
/dev/sg1 -H -m root
/dev/sg2 -H -m root
...
```



Работа с утилитой arrconf

Состояние RAID

```
[root@ds01.example.com ~]# arccconf getconfig 1 ld
Controllers found: 1
Logical device number 0
  Logical device name : clienthdd
  Block Size of member drives : 512 Bytes
  RAID level : 1
  Unique Identifier : E6D2072A
  Status of logical device : Optimal
  Size : 3814390 MB
  Parity space : 3814400 MB
  Read-cache setting : Enabled
  Read-cache status : On
  Write-cache setting : Enabled
  Write-cache status : On
  Partitioned : Yes
  Protected by Hot-Spare : No
  Bootable : Yes
  Failed stripes : No
  Power settings : Disabled
```

Segment 0: Present (Controller:1, Enclosure:0, Slot:28)

PN1334PEHNAZZS

Segment 1: Present (Controller:1, Enclosure:0, Slot:29)

PN1334PEHN54AS





Работа с утилитой arrconf

Состояние жесткого диска

```
[root@ds01.example.com ~]# arrconf getconfig 1 pd
Controllers found: 1
Device #0
  Device is a Hard drive
  State : Online
  Block Size : 512 Bytes
  Supported : Yes
  Transfer Speed : SATA 6.0 Gb/s
  Reported Channel,Device(T:L) : 0,0(0:0)
  Reported Location : Enclosure 0, Slot 31
  Reported ESD(T:L) : 2,0(0:0)
  Vendor : ATA
  Model : INTEL SSDSC2BB48
  Firmware : G2010140
  Serial number : BTWA544406SE480FGN
  Reserved Size : 138008 KB
  Used Size : 457728 MB
  Unused Size : 64 KB
  Total Size : 457862 MB
  Write Cache : Enabled (write-back)
  FRU : None
  S.M.A.R.T. : No
  S.M.A.R.T. warnings : 0
  Power State : Full rpm
  Supported Power States : Full power,Powered off
  SSD : Yes
  NCQ status : Enabled
```





Работа с утилитой arrconf

Создание RAID 1

```
arcconf CREATE 1 LOGICALDRIVE METHOD QUICK MAX 1 0 2 0 3 noprompt
```

Расшифровка команды

```
arcconf CREATE 1 LOGICALDRIVE [опции] <объём> <тип raid массива> <список Т:L дисков>  
noprompt
```

LSI MegaRAID

Установка megacli

```
rpm -Uhv http://quattor.web.lal.in2p3.fr/packages/site/Lib_Utils-1.00-09.noarch.rpm
rpm -Uhv http://quattor.web.lal.in2p3.fr/packages/site/MegaCli-8.02.16-1.i386.rpm
cp /opt/MegaRAID/MegaCli/MegaCli64 /usr/bin/megacli
chmod +x /usr/bin/megacli
```

Проверяем состояние RAID

```
[root@ds02.example.com ~]# megacli -LDInfo -Lall -Aall
Adapter 0 -- Virtual Drive Information:
Virtual Drive: 0 (Target Id: 0)
Name          :
RAID Level    : Primary-1, Secondary-0, RAID Level Qualifier-0
Size          : 3.637 TB
Mirror Data   : 3.637 TB
State         : Optimal
Strip Size    : 256 KB
Number Of Drives per span:2
Span Depth    : 2
Default Cache Policy: WriteBack, ReadAhead, Direct, Write Cache OK if Bad BBU
Current Cache Policy: WriteBack, ReadAhead, Direct, Write Cache OK if Bad BBU
Default Access Policy: Read/Write
Current Access Policy: Read/Write
Disk Cache Policy : Disk's Default
Encryption Type   : None
Bad Blocks Exist: No
Is VD Cached: No
```



LSI MegaRAID

Удалить логические устройства

```
megacli -CfgLdDel -Lall -aAll
```

Очистка конфигурации

```
megacli -CfgClr -aAll
```

Создание RAID 10

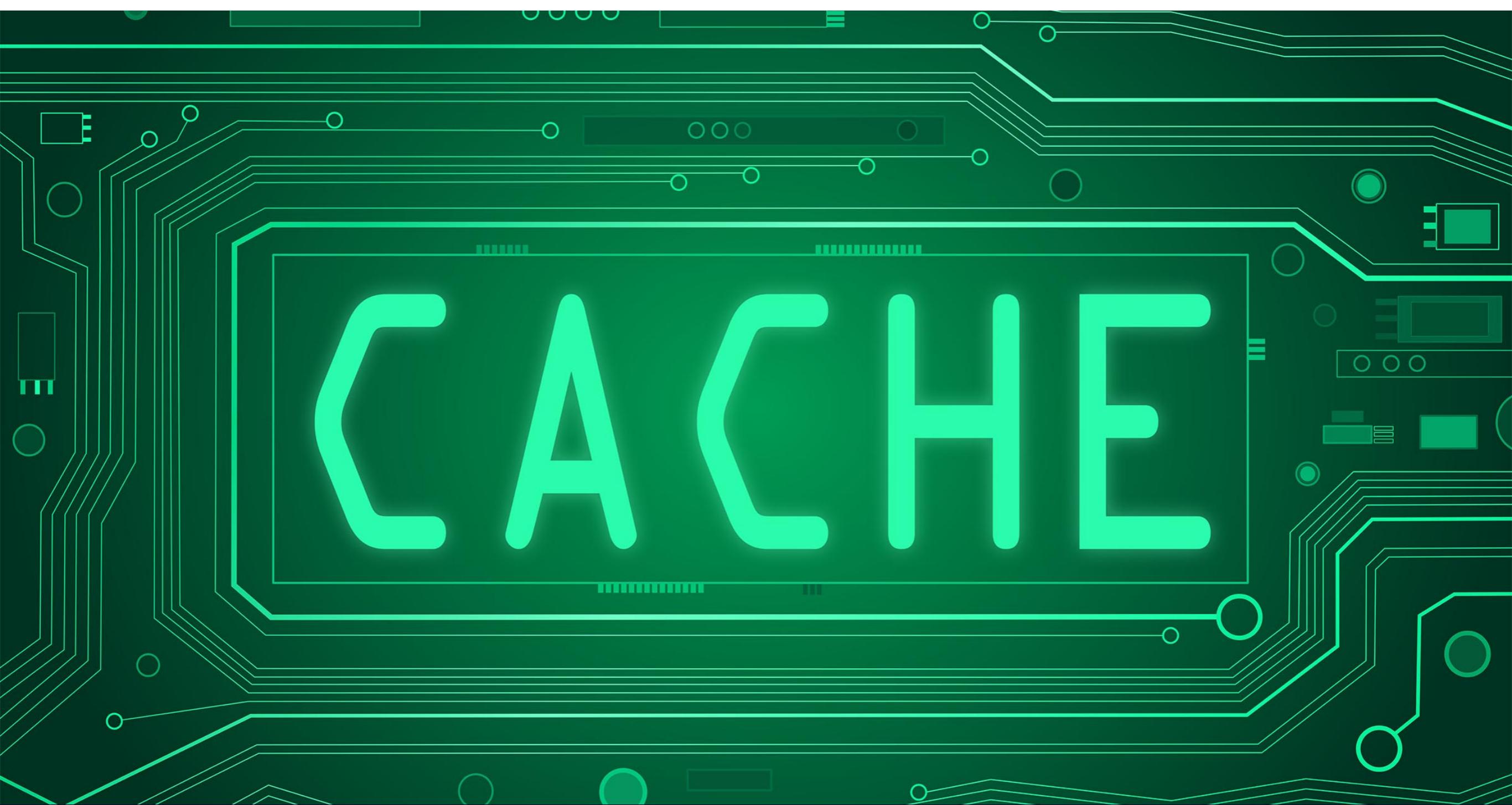
```
megacli -CfgSpanAdd -r10 -Array0[252:0,252:1] Array1[252:2,252:3] WB RA Direct CachedBadBBU  
-sz100GB -a0
```

Включение SMART

```
smartctl --smart=on --offlineauto=on --saveauto=on -d megaraid,4 /dev/sda  
smartctl --smart=on --offlineauto=on --saveauto=on -d megaraid,5 /dev/sda  
smartctl --smart=on --offlineauto=on --saveauto=on -d megaraid,6 /dev/sda  
...
```



Cache



Cache LSI MegaRAID

Проверить состояние RAID

```
megacli -LDInfo -Lall -Aall
```

Проверить наличие батареи и её состояние

```
megacli -AdpBbuCmd -aAll  
megacli -AdpBbuCmd -GetBbuStatus -a0 | grep -e '^isSOHGood' -e '^Charger Status' -e  
'^Remaining Capacity'
```

Проверить наличие батареи и её состояние

```
megacli -LDSetProp NoCachedBadBBU -LALL -aALL (выключаем)  
megacli -LDSetProp CachedBadBBU -LALL -aALL (включаем)
```

Включение ReadCache

```
megacli -LDSetProp -Cached -LAll -aAll (включаем)
```

Включение DiskCache

```
megacli -LDSetProp EnDskCache -LAll -aAll (включаем)  
megacli -LDSetProp DisDskCache -LAll -aAll (выключаем)
```



Cache LSI MegaRAID

Результат состояния RAID после включения кэша

```
Adapter 0 -- Virtual Drive Information:  
Virtual Drive: 0 (Target Id: 0)  
Name :  
RAID Level : Primary-1, Secondary-0, RAID Level Qualifier-0  
Size : 5.457 TB  
Mirror Data : 5.457 TB  
State : Optimal  
Strip Size : 256 KB  
Number Of Drives per span:2  
Span Depth : 2  
Default Cache Policy: WriteBack, ReadAhead, Cached, Write Cache OK if Bad BBU  
Current Cache Policy: WriteBack, ReadAhead, Cached, Write Cache OK if Bad BBU  
Default Access Policy: Read/Write  
Current Access Policy: Read/Write  
Disk Cache Policy : Enabled  
Encryption Type : None  
Bad Blocks Exist: No  
Is VD Cached: No
```



Вопросы?

