Система

- Bonding
- GVT-g
- Nvidia драйвер для хоста
- Trim
- Автоматическое увеличение системного пула
- Включение rc.local
- Изменение swap раздела
- Смена имени

Bonding

• Round-robin (balance-rr): Передача сетевых пакетов в последовательном порядке от первого доступного связанного сетевого интерфейса (NIC) до последнего. Этот режим обеспечивает балансировку нагрузки и отказоустойчивость.

• Active-backup (active-backup): Только один из связанных в bond сетевых интерфейсов активен. Другой связанный интерфейс становится активным, если и только если активный интерфейс выходит из строя. MAC-адрес одного логического связанного интерфейса виден извне только на одном сетевом адаптере (порту), чтобы избежать искажений в сетевом коммутаторе. Этот режим обеспечивает отказоустойчивость.

• XOR (balance-xor): Передача сетевых пакетов на основе [(source MAC address XOR’d with destination MAC address) modulo NIC slave count]. Это выбирает тот же ведомый сетевой адаптер для каждого MAC-адреса назначения. Этот режим обеспечивает балансировку нагрузки и отказоустойчивость.

• Broadcast (broadcast): Передает сетевые пакеты по всем связанным сетевым интерфейсам. Этот режим обеспечивает отказоустойчивость.

• IEEE 802.3ad Dynamic link aggregation (802.3ad)(LACP): Создает группы агрегации с одинаковыми настройками скорости и дуплекса. Использует все связанные сетевые интерфейсы в активной группе агрегаторов в соответствии со спецификацией 802.3 ad.

• Adaptive transmit load balancing (balance-tlb): Linux bonding driver mode, который не требует специальной поддержки сетевых коммутаторов. Исходящий сетевой пакетный трафик распределяется в соответствии с текущей нагрузкой (вычисленной относительно скорости) на каждом ведомом устройстве сетевого интерфейса. Входящий трафик получен одним в настоящее время назначенным ведомым сетевым интерфейсом. Если этот принимающий интерфейс отказывает, другой связанный интерфейс берет на себя MAC-адрес отказавшего принимающего интерфейса.

• Adaptive load balancing (balance-alb): Включает balance-tlb plus receive load balancing (rlb) для трафика IPV4 и не требует специальной поддержки сетевых коммутаторов. Балансировка нагрузки на прием достигается путем согласования ARP. Драйвер связи перехватывает ответы ARP, отправленные локальной системой на их выходе, и перезаписывает исходный аппаратный адрес с уникальным аппаратным адресом одного из подчиненных сетевых карт в едином логическом связанном интерфейсе так, чтобы различные сетевые узлы использовали различные MAC-адреса для своего сетевого пакетного трафика.

GVT-g

Загружаем модули

```
echo "vfio" >> /etc/modules-load.d/modules.conf
echo "vfio_iommu_type1" >> /etc/modules-load.d/modules.conf
echo "vfio_pci" >> /etc/modules-load.d/modules.conf
echo "vfio_virqfd" >> /etc/modules-load.d/modules.conf
echo "kvmgt" >> /etc/modules-load.d/modules.conf
```

Вносим изменения в загрузчик

```
nano /etc/default/grub
```

> GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on kvm.ignore_msrs=1 i915.enable_gvt=1 i915.enable_execlists=0 drm.debug=0"

Обновляем конфиги и модули

```
update-grub && update-initramfs -u -k all
```

Перезапускаем сервер

В свойствах виртуальной машины (предварительно ее отключив) раздел процессора приводим к виду

> cpu: host,hidden=1

Теперь добавляем PCI устройство с видеокартой и отключаем виртуальную

Nvidia драйвер для хоста

Добавить дополнительные репозитории в /etc/apt/sources.list

```
deb http://ftp.us.debian.org/debian bullseye main contrib non-free
deb http://ftp.us.debian.org/debian bullseye-updates main contrib non-free
deb http://security.debian.org bullseye-security main contrib
```

Обновить пакеты и систему

```
apt update
apt upgrade
```

Установить заголовки ядра

```
apt install pve-headers
```

Установить драйвер и необходимые пакеты

```
apt install libnvidia-cfg1 nvidia-kernel-source nvidia-kernel-common nvidia-driver
```

Перезагрузить систему

```
reboot
```

Проверить установку драйвера

```
nvidia-smi
```

Trim

Для автоматизации операции trim для физических дисков, а также для lxc контейнеров выполняем ряд операций ниже.

Создаем скрипт:

```
nano /opt/pve_trim.sh
```

Вносим в него:

```
#!/bin/bash

FSTRIM=/sbin/fstrim

for i in $(pct list | awk '/^[0-9]/ {print $1}'); do
  pct fstrim "$i" 2>&1 | logger -t "pct fstrim [$$]"
done

$FSTRIM -av 2>&1 | logger -t "fstrim [$$]"
```

После создаем задание:

```
nano /etc/cron.d/pve_trim
```

Вносим в него следующее:

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

0 6 * * 1 /opt/pve_trim.sh
```

Все. Теперь каждый понедельник в 6 утра будет запускаться сначала трим по всем контейнерам, после чего трим физических дисков сервера.

P.S. Не ставить задание чаще, чем раз в неделю!

Автоматическое увеличение системного пула

Поменять значение **thin_pool_autoextend_threshold**:

```
sed -i 's/# thin_pool_autoextend_threshold = 70/thin_pool_autoextend_threshold = 70/'
/etc/lvm/lvm.conf
```

Перезагрузить сервер:

```
reboot
```

Включение rc.local

```
nano /etc/systemd/system/rc-local.service
```

> [Unit]
> Description=/etc/rc.local Compatibility
> ConditionPathExists=/etc/rc.local
>
> [Service]
> Type=forking
> ExecStart=/etc/rc.local start
> TimeoutSec=0
> StandardOutput=tty
> RemainAfterExit=yes
> SysVStartPriority=99
>
> [Install]
> WantedBy=multi-user.target

```
nano /etc/rc.local
```

> #!/bin/bash
> exit 0

```
chmod +x /etc/rc.local
systemctl enable --now rc-local
```

Изменение swap раздела

Отключить подкачку

```
swapoff -v /dev/pve/swap
```

Увеличить размер раздела

```
lvm lvresize /dev/pve/swap -L +8G
```

Отметить раздел как swap

```
mkswap /dev/pve/swap
```

Включить swap

```
swapon -va
```

Смена имени

Отредактировать файлы

> /etc/hostname
> /etc/hosts
> /etc/postfix/main.cf
> /etc/mailname (if you're using this)
> /etc/pve/storage.cfg
> /var/lib/rrdcached/db/pve2-storage
> /var/lib/rrdcached/db/pve2-node
> /etc/pve/nodes/

Перезапустить сервер

```
reboot
```