

Содержание

Утилиты управления	3
<i>dpdkinfo</i>	3
<i>bpctl_util</i>	3
<i>driverctl</i>	3
<i>checklock</i>	4
<i>checkproto</i>	4
<i>checknat</i>	4

Утилиты управления

dpdkinfo

Получение диагностических данных с SFP-модулей.

Параметры:

- -h - подсказка
- module_eeprom - информация по оптической диагностике модуля SFP (если она поддерживается модулем).

bpctl_util

Ручное управление bypass.

DPI управляет bypass самостоятельно, но в случае необходимости ручное управление осуществляется данной утилитой.

Параметры:

- get_bypass - получить состояние bypass
- set_bypass on - активировать bypass
- set_bypass off - деактивировать bypass
- get_std_nic - диагностика
- set_std_nic off - установка карты в режим bypass (переключает режим в НЕстандартный, то есть с bypass режимом)
- set_dis_bypass off
- set_bypass_pwoff on
- set_bypass_pwup on
- get_bypass_change on
- get_tx on
- get_tpl off
- get_wait_at_pwup off
- get_hw_reset off
- get_disc off
- get_disc_change off
- get_dis_disc off
- get_disc_pwup off
- get_wd_exp_mode bypass
- get_wd_autoreset disable

driverctl

Управление DPDK.

Параметры:

- list-overrides - проверить список карт, находящихся в режиме DPDK

- `unset-override 0000:04:00.0` - вывести карту из режима DPDK
Предварительно необходимо остановить процесс fastDPI командой `service fastdpi stop!`
- `-v set-override 0000:04:00.0 vfio-pci` - вернуть карты обратно под управление DPDK после работ со штатным драйвером
При переводе карт в режим DPDK будьте внимательны и не переведите случайно управляющий интерфейс сервера в режим DPDK — связь с сервером сразу прервется!



Конфигурирование DPDK в Hyper-V подробно описано в соответствующем разделе.

checklock

Проверка вхождения адреса или порта в черный список.
Пример проверки порта:

```
checklock 41:00.0
```

checkproto

Проверка вхождения адреса или порта в custom протокол.
Пример проверки порта:

```
checkproto 41:00.0
```

checknat

Проверка распределения белых адресов. Показывает, как распределяется сеть для NAT между воркерами-процессами.
Формат записи:

```
nthr=x, algo=0|1|2, cidrs='list cidrs'
[,tcheck_correct_hash=0:1,gr_cidrs='list gray cidrs',dst_cidrs='list
destination cidrs']
```

Параметры:

- `nthr` - количество рабочих потоков на кластер
- `algo` - **0** - hashmask (по умолчанию), **1** - crc, **2** - rxdsp_2
- `cidrs` - список белых адресов `cidr`
- `check_correct_hash` - проверка хэш-функции
 - `gr_cidrs='список серых адресов cidr для проверки'`

- `dst_cidrs='список адресов назначения cidr для проверки'`

Примеры:

```
# Пример 1
nthr=16 algo=0 cidrs='16.35.120.0/24,91.210.24.128/26'
# Пример 2
nthr=16 algo=0 cidrs='16.35.120.0/24,91.210.24.128/26' check_correct_hash=1
gr_cidrs='10.0.0.0/24,192.168.4.0/28' dst_cidrs='30.0.0.0/24,50.0.0.0/24'
```