

---

# Современные коммутаторы для ЦОД на 25 и 100G

## Содержание

1. Эволюция ОС для коммутаторов .....	2
1.1. Традиционный вариант .....	2
1.2. Современный вариант .....	3
1.3. Что всё это значит? .....	6

# 1. Эволюция ОС для коммутаторов

Много лет коммутаторы были полностью закрытыми продуктами с управлением либо через CLI/web, либо через специальное API от производителя.

С появлением Cumulus Linux ситуация изменилась, коммутатор стал подобен Linux-серверу с управлением через стандартные утилиты и средства мониторинга. Однако, Cumulus Linux нацелен на конкретный рынок, ориентирован на L3 сети и поддерживает ограниченный набор протоколов.

Что происходит с традиционными ОС, использующими CLI?

Они становятся всё более открытыми. Рассмотрим на примере ICOS, пакета от Broadcom, который служит основой для сетевой ОС на матрицах их производства.

Коммутатор, в отличие от сервера, не позволяет просто поставить ОС с драйвером матрицы. Для полноценной работы необходима интеграция с системой управления индикацией, картой портов и прочими необходимыми моментами. Естественно, что такую интеграцию можно проводить разными способами.

## 1.1. Традиционный вариант

Общепринятым было попадание прямо в интерфейс CLI, так реализовано на коммутаторе Aurora 220 (48 x 1G Base-T + 8 x 10G SFP+):

```
FASTPATH Startup -- Main Menu

1 - Start FASTPATH Application
2 - Display Utility Menu
Select (1, 2):
BOOT_PART = /dev/mtd8
Copying /dev/mtd8 to RAM...done.
Extracting FASTPATH .....done
Loading FASTPATH ...../mnt/application
done
Loading hardware monitor
Uncompressing apps.lzma
SyncDB Running...
DMA pool size: 8388608
AXI unit 0: Dev 0xb344, Rev 0x01, Chip BCM56344_A0, Driver BCM56340_A0
SOC unit 0 attached to PCI device BCM56344_A0

Jun  3 10:24:51 0.0.0.0 General[fp_main_task]: bootos.c(199) 3 %%
Event(0xaaaaaaaaa) started!

(Unit 1)>

Applying Global configuration, please wait ...

Applying Interface configuration, please wait ...

User:admin
```

```
Password:  
(Aurora 220) >
```

Из CLI можно попасть в Linux shell командой `linuxsh`, но возможности, обычно, ограничены.

Например, вывод `lspci` выглядит так:

```
# lspci  
#
```

Или `ifconfig`:

```
# ifconfig  
eth0    Link encap:Ethernet  HWaddr 00:05:64:30:72:22  
        inet addr:192.168.0.62  Bcast:192.168.0.255  Mask:255.255.255.0  
        inet6 addr: fe80::205:64ff:fe30:7222/64 Scope:Link  
        UP BROADCAST RUNNING ALLMULTI MULTICAST  MTU:1500  Metric:1  
        RX packets:153 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:10 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:18963 (18.5 KiB)  TX bytes:1760 (1.7 KiB)  
        Interrupt:234 Base address:0x2000
```

```
lo      Link encap:Local Loopback  
        inet addr:127.0.0.1  Mask:255.0.0.0  
        inet6 addr: ::1/128 Scope:Host  
        UP LOOPBACK RUNNING  MTU:16436  Metric:1  
        RX packets:124 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:124 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:0  
        RX bytes:6723 (6.5 KiB)  TX bytes:6723 (6.5 KiB)
```

```
rt_vrf_0_drv Link encap:Ethernet  HWaddr AA:23:48:A8:6F:89  
        UP BROADCAST RUNNING MULTICAST  MTU:12288  Metric:1  
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
rt_vrf_0_stk Link encap:Ethernet  HWaddr 32:71:C7:DC:11:A9  
        UP BROADCAST RUNNING MULTICAST  MTU:12288  Metric:1  
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

## 1.2. Современный вариант

С коммутаторам на базе x86 процессоров ситуация поменялась. Стандартный загрузчик GRUB выбирает между установленной ОС (если есть) и средой ONIE

```
GNU GRUB  version 2.02~beta2+e4a1fe391
```



## Современные коммутаторы для ЦОД на 25 и 100G

---

```
00:03.0 PCI bridge: Intel Corporation Atom processor C2000 PCIe Root Port 3 (rev 02)
00:04.0 PCI bridge: Intel Corporation Atom processor C2000 PCIe Root Port 4 (rev 02)
00:0b.0 Co-processor: Intel Corporation Atom processor C2000 nCPM (rev 02)
00:0e.0 Host bridge: Intel Corporation Atom processor C2000 RAS (rev 02)
00:0f.0 IOMMU: Intel Corporation Atom processor C2000 RCEC (rev 02)
00:13.0 System peripheral: Intel Corporation Atom processor C2000 SMBus 2.0 (rev 02)
00:14.0 Ethernet controller: Intel Corporation Ethernet Connection I354 (rev 03)
00:14.1 Ethernet controller: Intel Corporation Ethernet Connection I354 (rev 03)
00:14.2 Ethernet controller: Intel Corporation Ethernet Connection I354 (rev 03)
00:14.3 Ethernet controller: Intel Corporation Ethernet Connection I354 (rev 03)
00:16.0 USB controller: Intel Corporation Atom processor C2000 USB Enhanced Host Controller
00:17.0 SATA controller: Intel Corporation Atom processor C2000 AHCI SATA2 Controller
00:18.0 SATA controller: Intel Corporation Atom processor C2000 AHCI SATA3 Controller
00:1f.0 ISA bridge: Intel Corporation Atom processor C2000 PCU (rev 02)
00:1f.3 SMBus: Intel Corporation Atom processor C2000 PCU SMBus (rev 02)
01:00.0 Ethernet controller: Broadcom Corporation Device b960 (rev 11)
01:00.1 Ethernet controller: Broadcom Corporation Device b960 (rev 11)
```

### Вывод ifconfig

```
root@nba720:~# ifconfig
bond1      Link encap:Ethernet  HWaddr 00:05:64:2f:3c:7e
UP BROADCAST MASTER MULTICAST  MTU:1518  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
...
bond64     Link encap:Ethernet  HWaddr ca:0a:c6:54:d4:86
UP BROADCAST MASTER MULTICAST  MTU:1518  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth0      Link encap:Ethernet  HWaddr 00:05:64:2f:3c:7c
inet addr:192.168.0.74  Bcast:192.168.0.255  Mask:255.255.255.0
inet6 addr: fe80::205:64ff:fe2f:3c7c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:86 errors:0 dropped:0 overruns:0 frame:0
TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:9635 (9.6 KB)  TX bytes:4922 (4.9 KB)

fptil_0_2 Link encap:Ethernet  HWaddr 00:05:64:2f:3c:7e
UP BROADCAST MULTICAST  MTU:1518  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
...
fptil_0_36 Link encap:Ethernet  HWaddr 00:05:64:2f:3c:7e
```

```
UP BROADCAST SLAVE MULTICAST MTU:1518 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
lo          Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:240 errors:0 dropped:0 overruns:0 frame:0
TX packets:240 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:19104 (19.1 KB) TX bytes:19104 (19.1 KB)
```

```
rfc5549if  Link encap:Ethernet HWaddr a6:2e:4d:c2:f8:5d
inet addr:250.0.0.1 Bcast:250.0.0.3 Mask:255.255.255.252
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:0 (0.0 B) TX bytes:42 (42.0 B)
```

```
rt_vrf_0_drv Link encap:Ethernet HWaddr 3e:f7:83:10:a8:a1
UP BROADCAST RUNNING MULTICAST MTU:9416 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

```
rt_vrf_0_stk Link encap:Ethernet HWaddr 06:8e:c0:35:1a:98
UP BROADCAST RUNNING MULTICAST MTU:9416 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Порты коммутатора появляются в системе как устройства `fpt1_0_<n>`, где `<n>` - номер порта на передней панели, `fpt1_0_36` соответствует 36 порту. В нашем случае нумерация начинается с 2 и заканчивается 36, так как первые два порта переведены в режим 4x25G. `bond<n>` - LAG интерфейс. `rt1_0_<n>` - интерфейс маршрутизации.

### 1.3. Что всё это значит?

Это значит, что для управления портами можно использовать стандартные средства Linux!

`ethtool`, `ifconfig`, `route`, `ip` теперь управляют коммутатором также, как управляли портами сервера. Можно установить Quagga и управлять маршрутизацией, конфигурация также хранится в текстовом формате в `/mnt/fastpath`, файл незатейливо называется `startup-config`.

Первопроходцами использования стандартных Linux для управления коммутатором стала компания Cumulus Networks, до этого традиционный подход с CLI и специальными API был

Современные коммутаторы для ЦОД на 25 и 100G

единственно доступным способом общения с сетевым оборудованием. Однако, индустрия не стоит на месте и новые возможности добавлены в ICOS.

Возможность	x86 платформа	Традиционный коммутатор
Linux	Ubuntu 14.04 LTS с пакетами для поддержки платформы.	Встроенный Linux.
Open Source приложения для Linux	Любые пакеты, доступные для Ubuntu.	Нет.
Zero-Touch Provisioning	Загрузка скриптов через DHCP option 239. Команды скрипта позволяют устанавливать дополнительные пакеты и конфигурировать коммутатор.	Возможна загрузка файла конфигурации и обновление образа ОС.
ONIE Code Installation	Через ONIE происходит первичная установка образа.	Среда ONIE опциональна.
Двойной образ ICOS.	Не поддерживается. У коммутатора есть одна файловая система с ICOS и ещё одна с ONIE. Если ОС повреждена, то восстановление происходит через ONIE.	Поддерживается, смена образа происходит автоматически в случае повреждения.
Конфигурация с помощью Puppet.	Puppet предустановлен. Пользователю нужно только настроить.	Поддержка возможна через специальные пакеты от производителя. Загрузка Puppet и других приложений происходит через процедуру Zero-Touch Provisioning и возможна только при использовании ONIE.
Управление файлами и образами	Любые файлы, включая конфигурацию, скрипты, образ ОС, передаются с использованием стандартного инструментария Linux или систем типа Puppet. Команда <i>copy</i> из ICOS CLI не поддерживается.	Управление через командную строку ICOS. Команда <i>copy</i> поддерживает TFTP, FTP, SCP и SFTP.
Сжатие файлов конфигурации	Не производится, все файлы доступны для просмотра и редактирования.	Для экономии NVRAM, файлы конфигурации и скрипты сжимаются. Несжатые файлы получаются с помощью команды <i>copy</i> на внешний сервер.
Управление пользователями	Рекомендуются использовать инструменты Linux для доступа пользователей, которые смогут получить доступ к оболочке ICOS с помощью команд ICOS-console и ICOS-cli. Методы аутентификации ICOS также доступны и могут быть активированы.	Контролируется ICOS. ICOS поддерживает статически заданные сведения и авторизацию через RADIUS и TACACS+.
Клиент Domain	Используется клиент Linux, который настраивается администратором из	Клиент DNS встроен в ICOS и конфигурируется из ICOS CLI.

Современные коммутаторы для ЦоД на 25 и 100G

Возможность	x86 платформа	Традиционный коммутатор
Name Server (DNS)	командной строки Linux. Клиент DNS не включен в ICOS, используется библиотека libresolv.	
Время	Используется Linux NTP для настройки времени и часового пояса. Внутренний клиент SNTP в ICOS не используется.	ICOS полагается на встроенный клиент SNTP, который управляется через CLI.
Syslog	На платформе x86 события отправляются в Linux System Logger. Рекомендуется настраивать клиент Linux syslog вне оболочки ICOS, чтобы все приложения на коммутаторе пользовались одним журналом событий. Встроенный в ICOS клиент syslog также доступен и может быть использован вместо Linux syslog.	Доступен только встроенный в ICOS клиент syslog.
Порт управления Ethernet	Порт управления контролируется Linux, из оболочки ICOS доступен только просмотр данных.	ICOS использует команды CLI для настройки порта управления.
Название хоста	ICOS читает название из Linux. Команда ICOS CLI для назначения имени поддерживается, но не изменяет файла конфигурации, а вызывает sethostname().	ICOS использует команды CLI для назначения имени и записи в файл конфигурации.
Шаблоны Switch Database Management (SDM) – настройка масштабирования платформы	Шаблоны SDM могут быть назначены через команду ICOS CLI и через файл конфигурации, расположенный в <code>/mnt/fastpath/sdm-templatetext.cfg</code> . Прямые изменения файла позволяют избежать дополнительной перезагрузки при работе скриптов ZTP до загрузки ICOS.	Шаблон SDM задаётся командами ICOS CLI и коммутатор должен быть перезагружен для применения изменений.
Управление портами в Linux	Возможно управление и мониторинг портов, LAG групп и интерфейсов маршрутизации с помощью таких инструментов, как ethtool и iproute2. Эти же операции доступны через ICOS CLI.	Не поддерживается. Всё управление происходит через CLI.
Приём/передача пакетов в Linux	Приложения могут использовать интерфейсы в Linux для передачи данных. Это позволяет использовать инструменты наподобие tcpdump/Wireshark или протоколы типа LLDPD прямо на коммутаторе.	Не поддерживается.



Современные коммутаторы для ЦОД на 25 и 100G

Возможность	x86 платформа	Традиционный коммутатор
Управление таблицей маршрутизации в Linux	Возможно использование сторонних приложений для управление маршрутизацией IPv4 и IPv6, используя стандартные утилиты, например iproute2, либо напрямую через сокет NETLINK. Это даёт возможность запускать такие пакеты, как Quagga, напрямую на коммутаторе. На данный момент не все возможности маршрутизации поддерживаются, например VRF не работает.	Не поддерживается.

Теперь, с новыми возможностями ICOS, два мира сблизились и доступны в одной ОС с развитым стеком сетевых протоколов.

