Software FCoE - A Case Study

Sudheer Nair (Presenter)
Manu Gupta (Presenter)

Vipul Swali
Jayjit Lobhe
Ramesh Veluswamy

Patni Americas Inc.
Discussion Topics

- Introduction to FCoE
- Software & Hardware FCoE solutions
  - Comparison
  - Advantages & Disadvantages
- Introduction to OpenFCoE (Software FCoE solution)
- OpenFCoE discussion
  - High level Code flow
- OpenSolaris FCoE
  - High level Architecture
  - Task Management framework
- Windows Software FCoE
- Configure OpenFCoE initiator and target
Fibre Channel over Ethernet – An Introduction

FCoE - Concepts
- Replace FC-1 & FC-0 with MAC & PHY
- Encapsulate FC frames into Ethernet frames
- Enhancements in Ethernet – e.g. Priority Based Flow Control

FCoE - Benefits
- Reduce server’s n/w interfaces, cables & switches
- Reduce heat dissipation
- Reduce power required for IO and data center cooling
- Preserves SAN security & SAN mgmt infrastructure
- Hybrid deployment – FC infrastructure can be used

FCoE - Protocol
- Uses Ethernet MAC Addresses
- FPMA & SPMA
- FIP Discovery – Solicitation & Advertisement
- Virtual Link Initialization – FLOGI & FDISC
Developments in FCoE

- FC-BB-5 approved – June 2009
- FC-BB-6 under development – April 2011
  - Point-to-Point topology (OpenFCoE does not support true PTP)
  - Improvements for high BER Ethernet media (e.g. 10GBASE-T)
  - FCoE Data Forwarders (FDF)

In FC-BB-5 –
- Non-optimal forwarding
- FCF becomes performance bottleneck
- Multiple FCFs need multiple domain IDs – practical limitation

FDFs –
- Optimal forwarding
- No performance bottleneck
- Share domain ID with controlling FCF
- Lightweight; provide routing, zoning info
Hardware & Software FCoE

- Implemented in hardware
- E.g. Converged Network Adapter (CNA)

- Implemented in software (device driver)
- Works with 10 Gigabit Ethernet 10GbE NICs
- E.g. OpenFCoE & OpenSolaris FCoE

- No performance issues, uses controller on card
- Cost is high

- Can have performance issues, uses host CPU cycles
- Cost lower than hardware solution
OpenFCoE – An Introduction

- Open source project started by Intel on Linux
- Aim – develop native FCoE code for Linux - leverage Ethernet networks to connect to FC SAN
- Initiator available in kernel 2.6.29, target needs optimization
- Target uses SCSI Target infrastructure – SCST
- libfc module implements FC functionality
- fcoe module implements Encapsulation & De-encapsulation, Virtual Link establishment
OpenFCoE: Link Establishment

FCoE Module

Initialization

Linkup/down Handling

Encapsulation/Decapsulation

Net Device

Send Packet

Rcv Packet

Obtain VN_PortID and Fabric ID

Initialization

Fabric Login

Frame Send

Frame Receive

libfc
**FIP Discovery (Solicitation/Adv)**

- **fcoe_ctlr_solicit( )**
  - Multicast FIP Solicit
  - DA=ALL_FCF_MACS
  - SA=ENODE MAC
  - FPMA / SPMA

- **fcoe_ctlr_recv_adv( )**
  - Unicast FIP Advrt
  - DA=ENODE MAC
  - SA=FCF MAC

**FCF List**
- Entry 1: Priority 1, FP/SP, Switch WWN, VLAN 1001
- Entry 2: Priority 10, FP/SP, Switch WWN, VLAN 1002
FIP (FLOGI / ACC)

```
fc_lport_enter_flogi()
```

**ENODE**

```
fclport_enter_flogi ()
```

```
DA=FCF1 MAC
SA=ENODE MAC
```

**Ethernet Switch**

```
DA=ENODE MAC
SA=FCF 1 MAC
```

```
DA=ENODE MAC
SA=FCF 2 MAC
```

```
DA=FCF2 MAC
SA=ENODE MAC
```

**FCF 1**

**FCF 2**
Open FCOE (FCOE_INIT Flow)

- `fcoe_device_setup()`
  - Registers for state change notification

- `fcoe_device_notification()`
  - Handles various NETDEV events.

- `fcoe_ctlr_link_up()`
  - Depending on the state sends link up to the FC layer and also calls the solicit func

- `fcoe_if_init()`
- `fc_attach_transport()`
  - Attaches to the FC transport
FC – SCSI interaction in OpenFCoE

Fibre Channel subsystem device tree attached to SCSI (using scsi_transport_fcp)
OpenFCoE - Transmit Flow

Application Layer (dd or cp or any I/O)

SCSI Upper and Mid Layer (sd_init_command and scsi_dispatch_cmd)

libfc layer (fc_queue_command and fc_fcp_cmd_send)

fcoe layer (fcoe_xmit)

Ethernet layer (dev_queue_xmit)

Application performs I/O on scsi devices

I/O converted into SCSI command & transferred to libfc layer

FC frame created on top of SCSI command

- Copy FC frame into Ethernet Buffer
- Set Ether Type as FCoE
- Fill up Ethernet structure with source & destination MAC address

Queue a buffer for transmission over Network device
OpenFCoE - Receive Flow

1. **Ethernet layer (netif_receive_skb)**
   - Receive Ethernet Packets
   - Call receive function of Ethernet Packet

2. **Ethernet layer (deliver_skb)**
   - FC frame is extracted from Ethernet frame
   - Obtaining SCSI Command from FC Frame

3. **fcoe layer (fcoe_recv_frame)**
   - Call Response Handler for SCSI

4. **libfc layer (fc_io_compl)**
   - Validate the response and send it to the Application Layer

5. **SCSI Layer (scsi_finish_command)**

6. **SCSI layer (scsi_io_completion)**
OpenSolaris FCoE – An Introduction

- Uses existing framework in OpenSolaris
  - COMSTAR (Common SCSI Target framework)
  - Leadville (SCSI Initiator Stack)
- Uses 3 new drivers
  - fcoei (uses services provided by fcoe)
  - fcoet (uses services provided by fcoe)
  - fcoe (common to both initiator & target)
- Part of source code in fcoet & fcoei has been ported from OpenFCoE
- OpenSolaris FCoE has both initiator and target stacks implemented
FCoE Task Management

OpenFCoE (Proposed Implementation)

openfctgt layer
openfct_rcv_cmd()

openfctgt layer
openfct_process_scsi_cmd()

SCST layer
scst_rx_cmd()

SCST layer
scst_rx_mgmt_fn()

Check for both Data & Task Management commands

Process Data commands

Process Task Management commands
## Software FCOE in Windows

<table>
<thead>
<tr>
<th>NDIS Protocol Driver</th>
<th>Virtual Bus Driver</th>
<th>Virtual Storport Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handles FC / FCoE protocol stuff.</td>
<td>Acts as virtual bus driver for virtual storport miniport.</td>
<td>Used when there is no physical device to talk to. For e.g. Ram Disk.</td>
</tr>
<tr>
<td>Interacts with NDIS Ethernet miniport driver</td>
<td>Bridge between storport and protocol driver</td>
<td>Exposes Ethernet ports as Scsiport to storage stack</td>
</tr>
<tr>
<td>First driver to start in stack</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Pros
- Useful in testing environments
- Uses 10G Ethernet card

### Cons
- Complex implementation involving multiple drivers
- Performance
Observation 1:
Execution of "fcconf eth0 enable" command hangs, call trace logged in syslog.

Resolution:
Perform Target setup followed by Initiator setup. i.e., Execute "fcconf ethx enable" command in FCoE Target, before executing "echo "ethX" > /sys/module/fcoe/parameters/create" Command in FCoE Initiator.

Observation 2:
Execution of "fcconf eth0 enable" command fails with following error -

```
[r...@fcoe ~]# fcconf eth0 enable
fcconf: fcc_fcoe_config: FCoE create of eth0 failed
fcconf: fcc_fcoe_config: error 95 Operation not supported
fcconf: fcc_fcoe_config: exiting at fcc_fcoe.c:142
```

Resolution:
Transmission and Reception of PAUSE are not set to ON.
It can be set to ON by using command "ethtool -A tx on rx on"
Observation 3:
During build of SCST module, error message related to "page member not in scatterlist structure" is logged in syslog, when kernel 2.6.29 is used.

Resolution:
For FCoE Target setup, use 2.6.23 kernel. Using higher kernels leads to such problem. Page member has been removed from Scatterlist structure in 2.6.29 kernel.

Observation 4:
Loading of module scst_disk fails with following error message-
[...@fcoe ~]# modprobe scst_disk FATAL: Error inserting scst_disk (/lib/modules/2.6.23/extra/dev_handlers/scst_disk.ko): Invalid argument

Resolution:
i) cd <path-to-scst>/trunk/scst/src
ii) Open Makefile and check CONFIG_SCST STRICT SERIALIZING flag is defined. If it is not, define it. (EXTRA_CFLAGS += DCONFIG_SCST STRICT SERIALIZING)
iii) Build SCST module.
iv) Build openfctgt and fcoe modules.
Observation 5:
Loading of openfctgt.ko module fails with following error message –
[r...@fcoe ~]# insmod /home/FCoE_Target/open-fcoe-target/openfctgt.ko
insmod: error inserting '/home/FCoE_Target/open-fcoe-target/openfctgt.ko': -1 Invalid module format

Resolution:
Check whether openfc module is loaded by using lsmod command.
If yes, remove the module by using "rmmod openfc" command.
Load openfctgt.ko module.
References

- [http://open-fcoe.org](http://open-fcoe.org)
- [http://open-fcoe.org/rwlove/fc_sysfs.jpg](http://open-fcoe.org/rwlove/fc_sysfs.jpg)
Q & A