THE LINUX SOFTROCE DRIVER

Liran Liss
Mellanox Technologies
March, 2017
AGENDA

- Introduction
- The RXE driver
- Development status and availability
- Configuring and using SoftRoCE
- Future work
**INTRODUCTION**

- **SoftRoCE is a software implementation of the IBTA RoCEv2 specification**
  - RDMA transport services over Ethernet network

- **Use cases**
  - Development
  - Testing
  - Asymmetric deployments
    - E.g., software clients connecting to hardware high-performance servers

**ROCEv2 Packet Format**

<table>
<thead>
<tr>
<th></th>
<th>Eth L2</th>
<th>IP</th>
<th>UDP</th>
<th>BTH+</th>
<th>Payload</th>
<th>iCRC</th>
<th>FCS</th>
</tr>
</thead>
</table>

**Network Protocols**

**Infiniband**

- Link layer
- IB Network Layer
- Infiniband Link Layer
- Infiniband Management

**Ethernet/IP**

- Link layer
- Ethernet Link Layer
- Ethernet/IP Management

**RDMA Transport**

- IB Network Layer
- Infiniband Link Layer
- Infiniband Management
**RXE DRIVER**

- **Standard RDMA provider**
  - Kernel rdma_rxe module
  - User-space librxe library

- **Kernel transport**
  - Supports both kernel ULPs and applications
  - Asynchronous progress
- **Kernel-allocated / user-mapped queues**
  - User-space prepares WQEs
  - System calls used only for send doorbells
  - No system calls for
    - Polling completions
    - Posting receive WQEs

- **Pinned memory regions**
  - Data copied in/out of buffers from any context

- **Per-QP tasklets**
  - Requester
  - Responder
  - Completer
Initial driver developed by System Fabrics Works

Restarted as Open-Source Github project in 2014

Driver submitted upstream and accepted in 2016
• Available from Kernel 4.8
• config RDMA_RXE

Library included in rdma-core repo

Bug fixes
• Logic and Semantics
  • Support 0-byte operations
  • Update work queue state before generating completion
  • Response PSN should increase monotonically
  • Correctly handle erroneous WQEcs
  • Correctly handle duplicate atomic requests
• Memory leaks and reference counts
  • Bad socket ref-count
  • QP ref-count while running tasklets
  • Freeing packets in some error flows
• Races
  • QP state update vs. sending out skbs
  • Packet processing vs. QP teardown

Performance improvements (ongoing)
STATUS

- **Features**
  - UD, RC, and UC transports
  - SRQ
  - Atomics support
  - Multicast
  - Fast registration, local/remote invalidate

- **ULPs and applications**
  - Low-level Verbs testing
  - RDMACM tests
  - Perftest
  - iSER initiator + target

- **Test configurations**
  - Bare-metal NIC
  - SRIOV VF

- **Interoperability**
  - Validated successfully with HW RoCE
    - ConnectX-3
    - ConnectX-4
AVAILABILITY

- **Linux distributions (planned)**
  - Redhat / CentOS 7.4
  - Ubuntu 17.10

- **Mellanox OFED**
  - Included in MOFED 4.0
  - Based on Kernel 4.9
  - Backports for Kernel 4.8
    - May be installed on Ubuntu 16.10
  - Forward-ports for Kernel 4.10
    - May be installed on Ubuntu 17.04
Once installed, RXE is managed via ‘rxe_cfg’ tool

- Start / stop RXE driver stack
- Add / remove / show configured devices
- Configured interfaces are persisted at /var/rxe/rxe

NAME

rxe_cfg - rxe configuration tool for RXE (Soft RoCE)

SYNOPSIS

rxe_cfg [status]
rxe_cfg start [-p proto]
rxe_cfg stop
rxe_cfg persistent
rxe_cfg add [-n] ethN
rxe_cfg remove [-n] ethN|rxeN
rxe_cfg crc enable|disable
rxe_cfg mtu [-f] [rxeN] mtu_size
dev-l-vrt-097:/kernel=> sudo rxe_cfg start
Name        Link  Driver     Speed  NMTU  IPv4_addr  RDEV  RMTU
eno2        yes   igb
enp17s0     yes   mlx5_core
eth0        no    mlx4_en

dev-l-vrt-097:/kernel=> #sudo rxe_cfg add enp17s0

dev-l-vrt-097:/kernel=> #sudo rxe_cfg status

dev-l-vrt-097:/kernel=> sudo rxe_cfg start
Name        Link  Driver     Speed  NMTU  IPv4_addr  RDEV  RMTU
eno2        yes   igb
enp17s0     yes   mlx5_core                          rxe0  1024  (3)
eth0        no    mlx4_en
dev-l-vrt-097:/kernel=> ip addr show dev enp17s0
8: enp17s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
     link/ether 7c:fe:90:75:3e:20 brd ff:ff:ff:ff:ff:ff
     inet 11.134.97.1/16 brd 11.134.255.255 scope global enp17s0
...
IT'S JUST ROCE...

```
local address: LID 0x0000, QPN 0x000011, PSN 0xc0ffee, GID ::ffff:11.134.146.5
remote address: LID 0x0000, QPN 0x00000ea, PSN 0xc0ffee, GID ::ffff:11.134.145.5
1024 bytes in 0.00 seconds = 26.60 Mbit/sec
1 iters in 0.00 seconds = 308.00 usec/iter
```
FUTURE WORK

Features

- **RDMA features**
  - Memory windows
  - On-demand-paging
  - SRQ resizing
  - XRC

- **Congestion control**
  - Flow fairness
  - Lossy networks

- **Container support**
  - Should come for free once RoCE network namespace support is completed
FUTURE WORK
Optimizations

- **0-copy for all transmitted packets**
  - Leverage the fact that MRs are pinned
  - Applies to
    - Send / RDMA-W packets
    - RDMA-R response packets
  - For UD, generate completions when skbs are destroyed
    - Rather than when enqueing copied skbs
  - Note: still need to go over all the data for CRC calculation

- **Efficient 1-copy for all received packets**
  - Remove *completely* any skb re-queuing
  - Scatter data to MR directly from NAPI context
    - Combine with CRC calculation
FUTURE WORK

Optimizations

- **Per-CPU execution context**
  - Replace existing per-QP tasklets for sender/responder/completer tasks

- **Transmit packet batching**
  - Output a batch of skbs as a list
  - Results in skb->xmit_more in netdev

- **Cache route for connected QPs**
  - Eliminates FIB lookup on each packet
  - Similar to TCP sockets

- **Use synchronous hash crypto driver for CRC calculations**
  - E.g., use crc32-pclmul driver on x86

- **Select different UDP source ports for each QP**
  - Utilize all NIC transmission queues
  - Leverage ECMP in the network
13th ANNUAL WORKSHOP 2017

THANK YOU

Liran Liss

Mellanox Technologies