RDMA Fundamentals

Dror Goldenberg

May 2017
Agenda

- Motivation
- Background
- Enabling Kernel Bypass
- Transport Offload
- Remote DMA
- Transport Objects
- Verbs
- ULPs
- In Network Operations
- Summary
Motivation

Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs

In Network Operations
Summary
Seeking Efficient Way to Communicate

The Original Ethernet Invention
Bob Metcalfe 1972 – Xerox PARC
**TCP Sockets**

**RFC 793 – Transmission Control Protocol (TCP)**

`SocketFD = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);`

```c
if (SocketFD == -1) {
    perror("cannot create socket");
    exit(EXIT_FAILURE);
}
```

```c
memset(&sa, 0, sizeof sa);

sa.sin_family = AF_INET;
sa.sin_port = htons(1100);
res = inet_pton(AF_INET, "192.168.1.3", &sa.sin_addr);
```

```c
if (connect(SocketFD, (struct sockaddr *)&sa, sizeof sa) == -1) {
    perror("connect failed");
    close(SocketFD);
    exit(EXIT_FAILURE);
}
```

```c
n = write(SocketFD,buffer,strlen(buffer));
```

```c
if (n < 0)
    error("ERROR writing to socket");
```

```c
n = read(SocketFD,buffer,MAX_BUFFER_SIZE);
```

```c
if (n < 0)
    error("ERROR reading from socket");
```

```c
shutdown(SocketFD, SHUT_RDWR);
```

```c
close(SocketFD);
```

**BSD/POSIX Sockets**
RDMA Protocol for Better Efficiency
Enable the Most Resilient Cloud Infrastructure

Interconnect That Enables Unlimited Cloud Scalability
“To make storage cheaper we use lots more network!
How do we make Azure Storage scale? RoCE (RDMA over Converged Ethernet) enabled at 40GbE for Windows Azure Storage, achieving massive COGS savings”
The InfiniBand Architecture

- Industry standard defined by the InfiniBand Trade Association
- Defines System Area Network architecture
  - Comprehensive specification - from physical to applications
Layered Solution – The OSI Model

Compute Node
- Application
- Presentation
- Session
- Transport
- Network
- Link
- Physical

Switch
- Packet relay
  - PHY
  - PHY

Router
- Packet relay
  - Link
  - PHY
  - PHY

Compute Node
- Application
- Presentation
- Session
- Transport
- Network
- Link
- Physical

Packet Format
- MAC Header (L2)
- Network Header (L3)
- Transport Header (L4)
- Payload
- CRC

Applications:
- Browser
- HTML
- TCP/UDP
- IP
- Ethernet
InfiniBand Protocol Layers
RoCE – RDMA over Converged Ethernet

- InfiniBand transport over Ethernet
- API Compatible
- Efficient, light-weight transport, layered directly over Ethernet
- Takes advantage of DCB Ethernet
  - PFC, ETS, and QCN

---

**InfiniBand**
- LRH (L2 Hdr)
- GRH (L3 Hdr)
- BTH+ (L4 Hdr)
- IB Payload
- ICRC
- VCRC

**RoCEv2**
- MAC
- ET
- IP
- IP Proto UDP
- UDP Port=RoCE
- BTH+
- IB Payload
- ICRC
- FCS
Measuring Latency and Bandwidth

Latency = \( \frac{T_{\text{roundtrip}}}{2} \)

Lower is Better

Bandwidth / Message Rate

Higher is Better
### RDMA Operations

- Send/Receive
- RDMA Read
- RDMA Write
- Atomic

RDMA Write (no CPU intervention)
The Three RDMA Goodies

- Transport offload
- Kernel bypass
- RDMA and Atomic operations
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
Traditional Hardware Access

- **Kernel role**
  - Application interface
  - Protocol stack
  - Resource arbitration across apps
  - Memory management (pinning, DMA)

- **Device driver role**
  - Hardware abstraction
Kernel Bypass

- **Hardware roles**
  - Application interface
  - Protocol stack
  - Resource arbitration across apps
  - Memory management (pinning, DMA)
Kernel Bypass

- **Separation of Control and Data paths**
  - **Data path**
    - Send
    - Receive
    - RDMA
    - Completion Retrieval
    - Request event
  - **Control path**
    - Resource setup
    - Memory management
Memory Registration & Memory Regions

- Protection
  - Byte level range
  - Permission (R/W)
- Memory Pinning
- Translation
  - Page level
- On demand paging option
Application Interface

- Dedicated queues per application
  - QP – send and receive queue
  - CQ – completion queue
Protection Domains

- Protection domains enable isolation
  - Memory Regions
  - Queue Pairs
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
Transport (Layer 4)

- Enable data transfer between applications on hosts
- Reliability: Reliable / Unreliable
- Connectivity: Connection / Connectionless
- Message: Message / Stream

Examples
- TCP: Connection – Reliable
  - Similar to RDMA – RC (Reliable Connected) transport
- UDP: Connectionless – Unreliable
  - Similar to RDMA – UD (UnReliable Connected) transport

Implementing Reliable Transport Example
Transport - Implementing Reliability – Out of Sequence

Requester

Responder

Message 1
Message 2
Message 3
Nak 2
Message 2
Message 3
Ack 3
Transport - Implementing Reliability - Timeouts

Requester

Responder

Message 1

Ack 1

X

Message 1

Message 2

Ack 2

Timeout
IB/RoCE Transports

- **RC (~= TCP)**
  - Reliable, connection oriented, transport. Guarantees full, in-order, delivery of messages and RDMA

- **UD (~= UDP)**
  - Unreliable, connection-less, transport. Best effort to deliver messages. Optional multicast support

- **UC**
  - Unreliable, connection oriented, transport. Best effort, in-order, delivery of messages and partial RDMA support

- **RD**
  - Most scalable connection and connectionless non-existent protocol...

- **XRC**
  - eXtended Reliable Connected – improved RC for better scalability

- **DC**
  - Dynamically Connected – fully scalable transport – dynamically connect and ensure reliability
Queue Pair (QP) – Transport Endpoint

- Send and Receive Queue
- Operations
  - Send
  - Receive
  - RDMA Read
  - RDMA Write
  - Atomic
- Asynchronous operation
- Transport implemented by the HCA
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
DMA from I/O devices
- Memory Read/Write
- Scatter/Gather operations
Remote DMA

RDMA Write (no CPU intervention)
Remote DMA Supported Operations

- **Send (with immediate)**
  - The sender sends data
  - The receiver specifies where it will be written in his side

- **RDMA Write**

- **RDMA Write with immediate**

- **RDMA Read**

- **Atomic operations**
  - The sender sends read/write request and specifies the local and remote addresses it will access (read/write)
# Service Types and Allowed Operations

$X = \text{allowed operation}$

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Send Queue Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Send</td>
</tr>
<tr>
<td>Reliable Connected</td>
<td>$X$</td>
</tr>
<tr>
<td>Reliable Datagram</td>
<td>$X$</td>
</tr>
<tr>
<td>Unreliable Connected</td>
<td>$X$</td>
</tr>
<tr>
<td>Unreliable Datagram</td>
<td></td>
</tr>
<tr>
<td>Raw Datagram</td>
<td>$X$</td>
</tr>
</tbody>
</table>
Data Movement Techniques

- **Buffer Copy**

- **Zero Copy**
  - RDMA Read
  - RDMA Write
Typical Buffer Copy Flow

Data Source

App Buf
Proto Buf
Proto Buf
Proto Buf

Data Message (Send)

Data Sink

Proto Buf
Proto Buf
Proto Buf

App Buf
Typical Read Zero Copy Flow

Data Source

App Buf

Advertise Message (Send – Data=key,addr,len)

RDMA Read

Read Response

Completion Msg (Send)

Data Sink

App Buf
Typical Write Zero Copy Flow

Data Source

Advertise Message (Send Data=key, addr, len)

App Buf

RDMA Write

Completion Msg (Send)

Data Sink

App Buf
# BCopy & ZCopy Summary

<table>
<thead>
<tr>
<th></th>
<th>BCopy</th>
<th>Read ZCopy</th>
<th>Write ZCopy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Transfer</strong></td>
<td>Send</td>
<td>RDMA Read</td>
<td>RDMA Write</td>
</tr>
<tr>
<td><strong>Overhead</strong></td>
<td>Data copy</td>
<td>Pinning, registration and handshake</td>
<td></td>
</tr>
<tr>
<td><strong>Pipelines data</strong></td>
<td>Yes</td>
<td>Multiple connections or asynchronous IO</td>
<td></td>
</tr>
<tr>
<td><strong>Optimal message size</strong></td>
<td>Small messages</td>
<td>Large messages</td>
<td></td>
</tr>
<tr>
<td><strong>Use when</strong></td>
<td>Non blocking I/O or small messages</td>
<td>Sender is ahead</td>
<td>Receiver is ahead</td>
</tr>
</tbody>
</table>
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
The Verbs API

- Available both in kernel and user-space
- Verbs classes
  - Resource management (QPs, CQs, SRQs, etc)
  - WR processing (post send, poll CQ)
  - Memory registration
  - Address handles
- Operations
  - Device operations
  - Verb context operations
  - Protection domain operations
  - Queue pair bring up
  - Active queue pair operations
HCA Resources

- **HCA**
  - PD - Protection Domain
    - Associates: QPs, MRs, MWs, AHs
  - RDD* - Reliable Datagram Domain
    - Associates: EEC and QPs

- **Address**
  - AH – Address Handle
    - UD messages destination address

- **QP – Queue Pair**
  - Transport endpoint – asynchronous interface

- **EEC* - End to End Context**

- **CQ – Completion Queue**
  - Transport endpoint completions – asynchronous interface

- **Memory**
  - MR - Memory Region
  - MW - Memory Window
    - Lightweight object, associated with MRs (lightweight binding)

* Reliable Datagram currently not implemented by any HW
Transport – Host Channel Adapter (HCA) Model

- Asynchronous interface - Verbs
  - Consumer posts work requests
  - HCA processes
  - Consumer polls completions
- Transport executed by HCA
- I/O channel exposed to the application
  - Kernel bypass
- Polling and interrupt models supported
RDMA vs TCP (SOCK_STREAM)
Basic Model: Posting

Channel Interface

post Work Requests

Verbs

creating Work Queue Entries

Consumer

WR 1

WR 2

Channel Interface

Send or Receive Queue

Completion Queue

WQE 2

WQE 1

WQE 2

WQE 1

WR 1

WR 2
Basic Model: Processing

CI processes WQEs

Consumer

Verbs

creating Completion Queue Entries

Channel Interface

Send or Receive Queue  Completion Queue

WR 2

WR 1

CQE 1

CQE 2
Basic Model: Get Completions

retrieve Work Completions

Verbs

Channel Interface

Send or Receive Queue Completion Queue
Receive Queue: Posting

For each DS:
- L_Key
- virtual address
- length

Consumer

Verbs

Channel Interface

Receive Queue

Completion Queue

SGL
Data
Segments

DS 1

DS 2

DS 3

WR 1

WQE 1

Data
Segments

Consumer

For each DS:
- L_Key
- virtual address
- length
Receive Queue: Processing

- Consumer
  - WR 1
- Verbs
- Channel Interface
  - Receive Queue
  - Completion Queue
  - CQE 1
- DS 1
- DS 2
- DS 3

Incoming Data
Receive Queue: Get Completion

- Channel Interface
- Receive Queue
- Completion Queue

- Verbs
- WR 1
- WC 1

- Consumer

- DS 1
- DS 2
- DS 3
Send WR: Posting
Send WR: Processing

Consumer

Verbs

Channel Interface

Send Queue

Completion Queue

Send Data

WR 1

DS 1

DS 2

DS 3

CQE 1
Send WR: Get Completion

Send Queue

Completion Queue
RDMA Read: Remote End

- **Consumer**
  - Receive Queue
  - Completion Queue
- **Verbs**
  - Memory Tables
- **Channel Interface**
- **Memory Regions**
  - MR 1
  - MR 2

Lookup:
- R_Key
- Virtual address
- Length from incoming RDMA request
RDMA Read: Remote End

Memory Regions

Consumer

Verbs

Channel Interface

Read Data

Receive Queue

Completion Queue
RDMA Write: Remote End

- **MR 1**
- **MR 2**

**Channel Interface**
- **Receive Queue**
- **WQE 1**
- **Completion Queue**
- **Memory Tables**

**Verbs**
- **Consumer**

**Lookup**
- R_Key
- Virtual address
- Length

from incoming RDMA request
RDMA Write: Remote End

Consumer

Verbs

Channel Interface

Write Data

Receive Queue

Completion Queue

WQE 1

MR 1

MR 2
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
Upper Layer Protocols – Transparent Acceleration

- ULPs connect RDMA to common interfaces
- Supported on mainstream operating systems

- Clustering
  - MPI (Message Passing Interface)
  - RDS (Reliable Datagram Socket)
  - ...

- Network
  - IPoIB/EoIB (IP/Eth over InfiniBand)
  - SDP (Socket Direct Protocol)
  - VMA – usermode socket accelerator
  - ...

- Storage
  - NVMF (NVMe over Fabrics Protocol)
  - iSER (iSCSI Extensions for RDMA)
  - NFSoRDMA (NFS over RDMA)
  - ...
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
Four Process Barrier Example – Using Managed Queues – Rank 0

Step 1

Step 2

Managed QP
QP Proc 2
Send to 2
NULL

Managed QP
QP Proc 3
Send to 3
NULL

Regular QP
Master QP
Enable send 2 (1)
Wait on QC 2 (1)
Enable send 3 (1)
Wait on QC 3 (1)
Signal Completion
NULL

© 2017 Mellanox Technologies
Optimize Higher Level Operation – 2x Latency Improvement

- Data reduction operation
- Commonly used in
  - Scientific applications
  - Deep learning

### SHArP

**Scalable Hierarchical Aggregation Protocol**

#### 128 Node All Reduce Latency (μSec)

- **Host**
- **SHARP**
- **Improvement**

#### 128 Node Pipelined All Reduce Latency (μSec)

- **Host**
- **SHARP**
- **Improvement**
Motivation
Background
Enabling Kernel Bypass
Transport Offload
Remote DMA
Transport Objects
Verbs
ULPs
In Network Operations
Summary
Summary

- **RDMA Goodies**
  - Kernel Bypass
  - Transport Offload
  - Remote DMA

- **RDMA Interface**
  - Verbs
  - ULP for better software abstraction / transparent access

- **In network operation**
  - Endpoint compute
  - Switch compute
Thank You