An Introduction to Remote Direct Memory Access

Patrick MacArthur
pio3@wildcats.unh.edu

Department of Computer Science
University of New Hampshire

CS 980
Advanced Topics in Network Research
October 28, 2013

Outline

Traditional Networks

TCP

- Reliable, connected
- Byte stream-oriented protocol
- Buffered in kernel-space on both sides
- Flow control via window size
- Synchronous: data transfer operations block until they complete

IP/Ethernet

- Each switch and router completely independent, autonomous
- Datagram-oriented
- Lossy—relies on upper layers to retransmit lost packets

RDMA

- Message-oriented protocol
- “Zero-copy”: direct application virtual memory to application virtual memory transfers
- Kernel bypass: userspace application talks directly with the hardware to do data transfers
- Flow control via credits at link layer—no loss due to congestion
- Asynchronous: data transfer operations placed onto queue
- Message latencies on the order of microseconds

RDMA Technologies

InfiniBand [InfiniBand 2007]

- Most popular RDMA implementation
- Defines entire network stack from top to bottom
- Speeds: SDR (10 Gbps), DDR (20 Gbps), QDR (40 Gbps), FDR (56 Gbps)

iWARP (Internet Wide-Area RDMA Protocol) [RFC 5040, RFC 5041, RFC 5044]

- Implements RDMA on top of TCP with 3 layers
- Defined by IETF

RoCE (RDMA over Converged Ethernet) [InfiniBand 2010]

- Implements upper layers of InfiniBand on top of Ethernet
- Defined by InfiniBand Trade Association

Terminology

- A local-area RDMA network is usually referred to as a **fabric**
- A **channel adapter** is the hardware component that connects a system to the fabric
  - iWARP refers to it as an RNIC (RDMA Network Interface Card)

OpenFabrics Alliance

InfiniBand Verbs

InfiniBand defines a set of "verbs" to communicate with RDMA hardware, but not an API

- Semantic details but no specific function calls, data structures, ...

OFA Verbs

The OpenFabrics Alliance (OFA) created a vendor-independent C API to access the InfiniBand verbs

- Very low-level API, involving direct manipulation of data structures
- Lots of code needed to write a very simple data transfer

OFED

The OFA periodically releases OFED (OpenFabrics Enterprise Distribution) which contains drivers and userspace libraries/tools for RDMA

Outline

Memory Registration

Current RDMA hardware has specific requirements for memory used in data transfers:

- Memory must not be modified by application during data transfer
- Memory cannot be paged out by operating system—physical to virtual mapping must be fixed

Registration

To deal with the second requirement, applications must **register** memory to be used in data transfers

- Returns a local and remote key pointing to memory area
- Registration keys are supplied as part of data transfer request

Queues

Queue Pairs

- A **queue pair** is associated with each side of an RDMA transfer
- Assigned an integer identifier that is sent to the other communication endpoint(s)
- A queue pair consists of a send and a receive request queue
- Post Send Request and Post Recv Request verbs used to add **work requests** to the queue

Completion Queues

- A send and receive **completion queue** is associated with a queue pair (at creation time)
- May be same queue for both, or two separate queues
- Poll Completion Queue verb used to remove completion events from the queue

Data Transfer Operations

SEND/RECV

- Analogous to socket send/recv, but using direct transfers
- RECV **must** be ready prior to arrival of SEND
- Both sides of transfer receive notification of completion

RDMA_WRITE

- Push of local data to remote memory area
- Sender must supply virtual address and remote key of destination memory area
- Sender initiates and receives notification; receiver is completely **passive** and receives no notification

Data Transfer Operations (continued)

RDMA_READ

- Pull of remote memory area to local memory area
- Receiver must supply virtual address and remote key of destination memory area
- Receiver initiates and receives notification; sender is completely **passive** and receives no notification
- Inefficient since request moves in opposite direction from data

RDMA_WRITE_WITH_IMM

- Like RDMA_WRITE, but receiver receives notification as well
- Request includes four bytes of out-of-band **immediate data**
- Immediate data included in receive completion event
- Supported by InfiniBand and RoCE, but not current version of iWARP

Outline

Subnet Manager

- One master **subnet manager** (SM) is elected for the entire fabric
- Can have other SMs, but for failover only
- Controls switching tables for entire fabric
- Switching tables must be recomputed every time a node added/removed to/from fabric

Node Addressing

Global Identifier (GID)

- Every InfiniBand node has an assigned **EUI-64 identifier**
  - 24 bit company identifier assigned by IEEE
  - 40 bit extension identifier assigned by manufacturer
- **Global Identifier** (GID) is a 128-bit identifier built from **EUI-64 identifier** and 64-bit **subnet prefix** assigned by SM
- Used for routing between subnets

Local Identifier (LID)

- Local ID (LID) is a 16-bit identifier dynamically assigned by SM
- Used for switching within a subnet
- may change during switching table recalculation

Transport Protocols

InfiniBand specifies four transport protocols:

- Reliable Connected (analogous to TCP)
- Reliable Datagram
- Unreliable Connected
- Unreliable Datagram (analogous to UDP)

InfiniBand Packet Format [InfiniBand 2007]

Local Route Header
Base Transport Header
Extended Transport Headers *(if needed)*
Packet Payload
Invariant CRC
Variant CRC

This is the packet format for packets using InfiniBand transport protocols

Local Route Header

<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td>7</td><td>8</td><td></td><td></td><td></td><td></td><td></td><td></td><td>15</td></tr>
<tr><td colspan="4">virtual lane</td><td colspan="4">version</td><td colspan="4">service level</td><td colspan="2">reserved</td><td colspan="2">LNH</td></tr>
<tr><td colspan="16">destination LID</td></tr>
<tr><td colspan="5">reserved</td><td colspan="11">packet length</td></tr>
<tr><td colspan="16">source LID</td></tr>
</table>

- Virtual Lane and Service Level fields used in QoS
- LNH is Link Next Header–1 bit global/local, 1 bit IB/raw
- Packet Length is in 4-byte words

Base Transport Header

0 7	8			15	16 23 24 31
opcode	SE	M	PadCnt	Version	Partition Key (P_KEY)
reserved	destination QPN				
A reserved	packet sequence number				

- SE (Solicited Event) bit indicates that CQ event handler should fire
- M is related to Automatic Path Migration
- PadCnt is number of bytes of padding in the Packet Length
- Partitions are analogous to VLANs on Ethernet
- Note that source Queue Pair Number is not part of this header

RDMA Extended Transport Header

<table>
<tr><th>0 … 31</th><th>32 … 63</th></tr>
<tr><td colspan="2">Virtual Address</td></tr>
<tr><td>Remote Key (R_KEY)</td><td>DMA Length</td></tr>
</table>

ACK Extended Transport Header

0 7	8 15 16 23 24 31
Syndrome	Message Sequence Number

Outline

Simultaneous Operations

Due to the asynchronous nature of RDMA, a key part of using RDMA effectively is having as many simultaneously outstanding transfer operations as possible.

The usual strategy is:

- Post as many RECV operations as possible
- Post a new RECV as soon as a RECV completion event fires
- Post send operations as soon as the data is ready to send

More simultaneous operations are needed over distance

Completion Event Detection Strategies

Event Notification

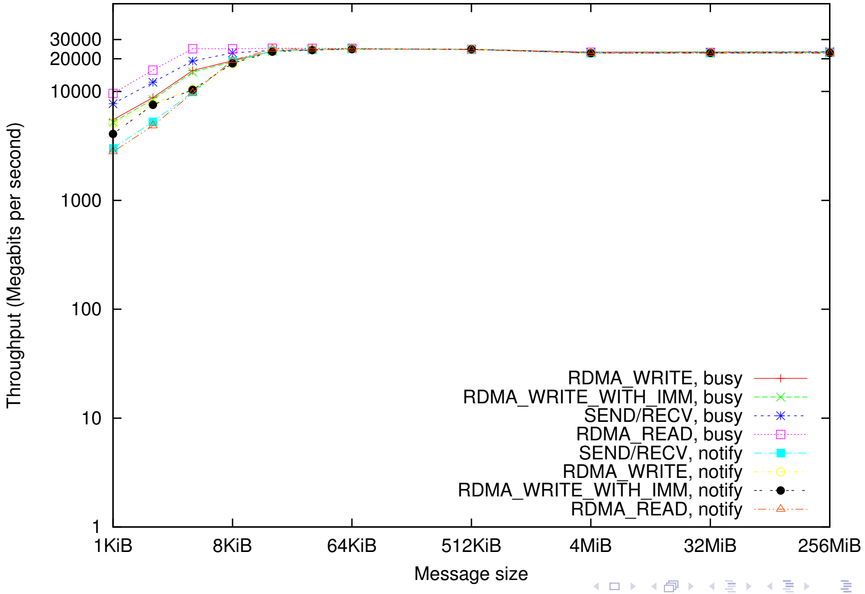
- User requests notification when a completion event arrives
- Requires kernel involvement to put thread/process to sleep and wake it up when event arrives
- Benefit: Very low CPU usage
- Cost: Lower throughput and higher latency

Busy Polling

- User polls in a tight loop for new completion events
- Get events as soon as they arrive with no kernel involvement
- Benefit: High throughput and low latency
- Cost: 100% CPU usage

[MacArthur and Russell 2012]

Completion Event Detection Strategies

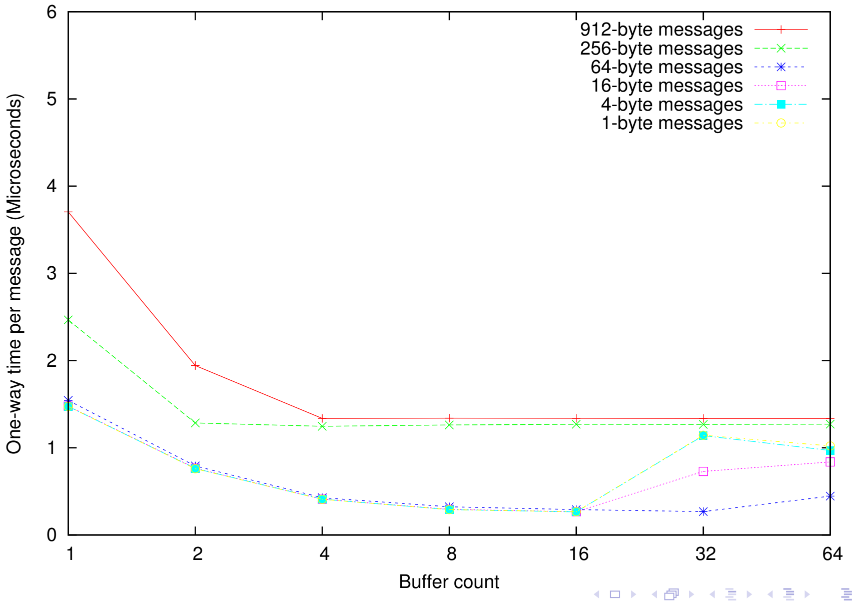


Inline Send Requests

- The **inline** feature causes an RDMA adapter to copy a small message into a buffer on the CA at the time that the work request is posted
- Subject to channel adapter limit
- Avoids need for memory registration on sending side
- Performance gain for very small messages, but hurts for larger messages

[MacArthur and Russell 2012]

One-way Time for Inline Send Requests



Outline

UNH EXS (Extended Sockets)

- Based on ES-API (Extended Sockets API) published by the Open Group [Interconnect 2005]
- Extensions to sockets API to provide asynchronous, zero-copy transfers
  - Memory registration (exs_mregister(), exs_mderegister())
  - Event queues for completion of asynchronous events (exs_qcreate(), exs_qdequeue(), exs_qdelete())
  - Asynchronous operations (exs_send(), exs_recv(), exs_accept(), exs_connect())
- UNH EXS supports SOCK_SEQPACKET (reliable message-oriented) and SOCK_STREAM (reliable stream-oriented) modes

Sample asynchronous send operation

```
exs_mhandle_t mh = exs_mregister(buf, bufsize, EXS_ACCESS_READ);
exs_qhandle_t qh = exs_qcreate(10);

if (exs_send(fd, buf, bufsize, 0, mh, 0, qh) < 0) {
        perror("Could not start send operation");
        /* bail out */
}

/* do work in parallel with data transfer */

exs_event_t ev;
if (exs_qdequeue(qh, &ev, 1, NULL) < 0) {
        perror("Could not get send completion event");
        /* bail out */
}
fprintf(stderr, "Send of %d bytes complete with errno=%d; actual byte count %d\n",
        bufsize, ev.exs_evt_errno,
        ev.exs_evt_union.exs_evt_xfer.exs_evt_length);
```

UNH EXS Protocol

For SOCK_SEQPACKET:

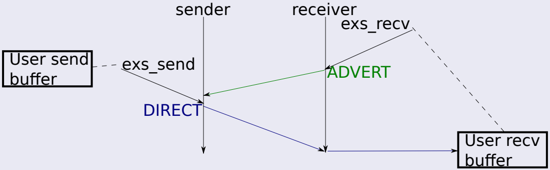
- Receiver sends ADVERT containing address, length, and remote key of receive buffer
- Sender uses RDMA_WRITE_WITH_IMM to write data directly into destination buffer

For SOCK_STREAM, same as above, except there is also an intermediate receive buffer that is used if no advertisements are available

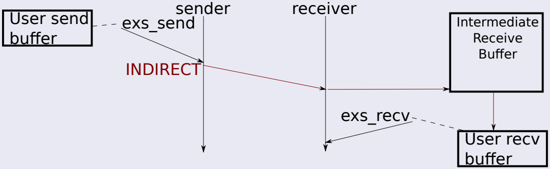
UNH EXS: Dynamic SOCK_STREAM Protocol

Key idea: allow sender to use direct or indirect transfer based on current conditions

Direct Transfer

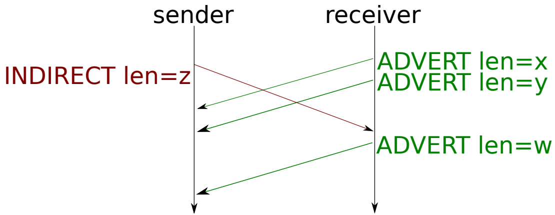


Indirect Transfer



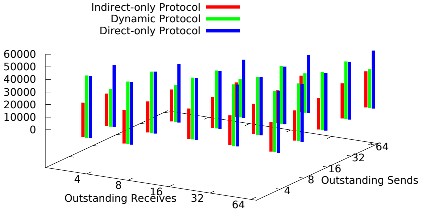
UNH EXS: Dynamic SOCK_STREAM Protocol

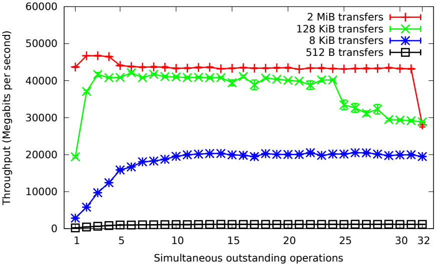
Key challenge: Advertisements may arrive late (i.e., after sender has already sent corresponding data), and sender must distinguish between "fresh" and "stale" advertisements

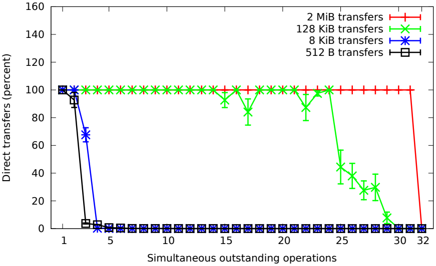


Here, the sender does not know how many exs_recv operations (corresponding to ADVERTs) were consumed by its INDIRECT send. The solution is to assign each advertisement a phase number and increment the phase number at the sender when an indirect transfer is sent.

UNH EXS Dynamic Stream Protocol Performance







Outline

Summary

- RDMA is used all over high-performance computing due to its high throughput and low latency
- Asynchronous, kernel bypass, and “zero-copy”
- Three standard protocol stacks: InfiniBand, iWARP, and RoCE

Open Issues

- OFED verbs library
  - Too complex for casual network programmers
  - Does not match semantics of consumers such as MPI
- Applications not written with RDMA in mind
  - Messages that are large for TCP/IP are **small** for RDMA
  - Applications often use only double buffering if they have any support at all for multiple outstanding transfers
- Subnet manager scalability
- Performance over distance
- Error handling

P. Cully, U. Elzur, R. Recio, S. Bailey, J. Carrier,
"Marker PDU Aligned Framing for TCP Specification,"
RFC 5044 (Standards Track), Oct. 2007. [Online]. Available:
http://www.ietf.org/rfc/rfc5044.txt

P. Grun,
Introduction to InfiniBand for End Users,
InfiniBand Trade Association, 2010.

InfiniBand Trade Association,
"InfiniBand Architecture Specification version 1.2.1,"
Nov. 2007.

—,
"Supplement to Infiniband Architecture Specification Volume 1, version 1.2.1: Annex A16: RDMA over Converged Ethernet (RoCE)",
Apr. 2010.

References II

- Interconnect Software Consortium in association with the Open Group,
  "Extended Sockets API (ES-API) Issue 1.0,"
  Jan. 2005.

- P. MacArthur, R. Russell,
  "A Performance Study to Guide RDMA Programming Decisions,"
  Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications, Jun. 2012.

- R. Recio, B. Metzler, P. Culley, J. Hilland, D. Garcia,
  "A Remote Direct Memory Access Protocol Specification,"
  RFC 5040 (Standards Track), Oct. 2007. [Online]. Available:
  `http://www.ietf.org/rfc/rfc5040.txt`

References III

H. Shah, J. Pinkerton, R. Recio, P. Culley,
"Direct Data Placement over Reliable Transports,"
RFC 5041 (Standards Track), Oct. 2007. [Online]. Available:
http://www.ietf.org/rfc/rfc5041.txt