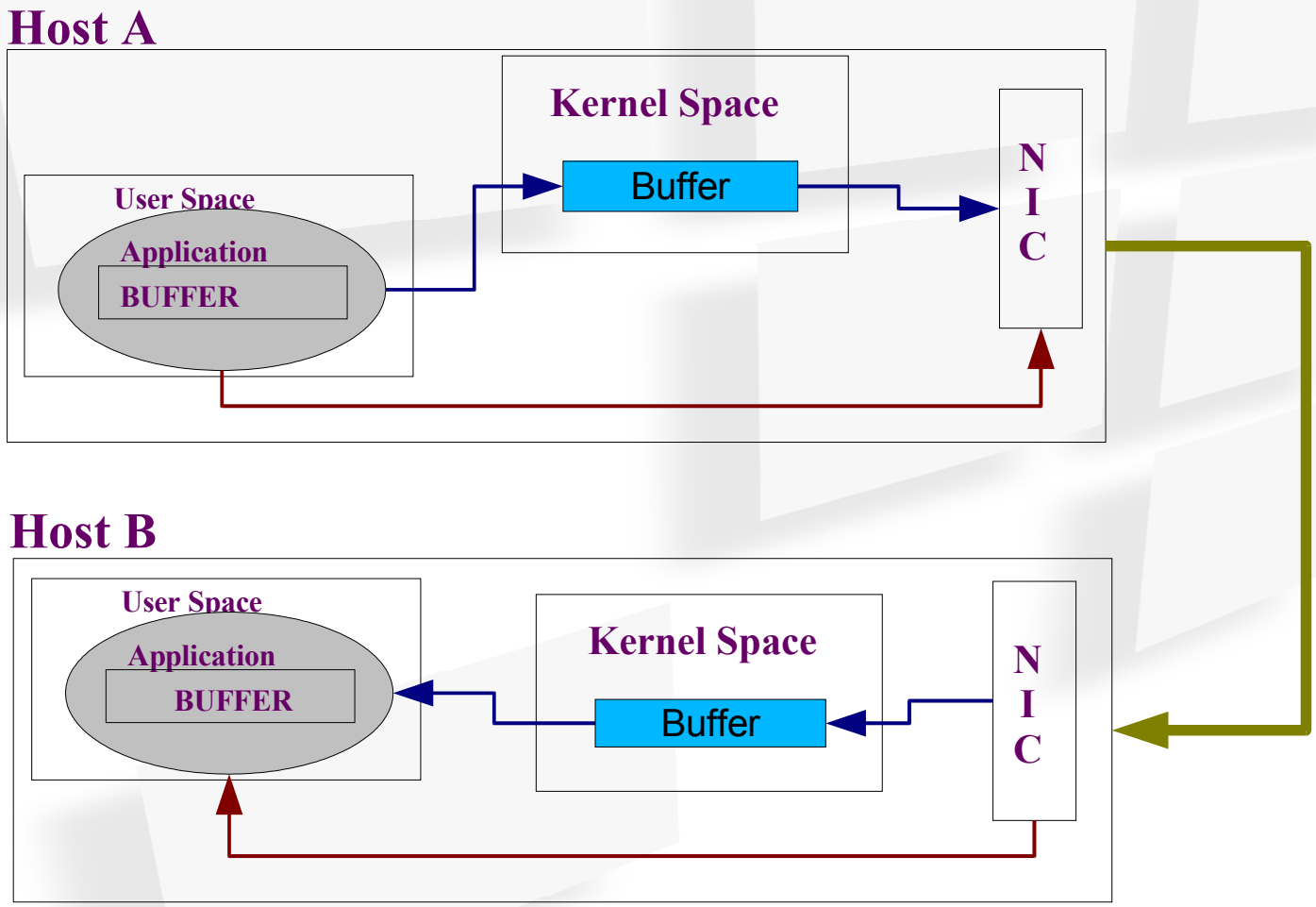
Design and Implementation of the iWarp Protocol in Software

Dennis Dalessandro, Ananth Devulapalli, Pete Wyckoff
Ohio Supercomputer Center

What is iWarp?

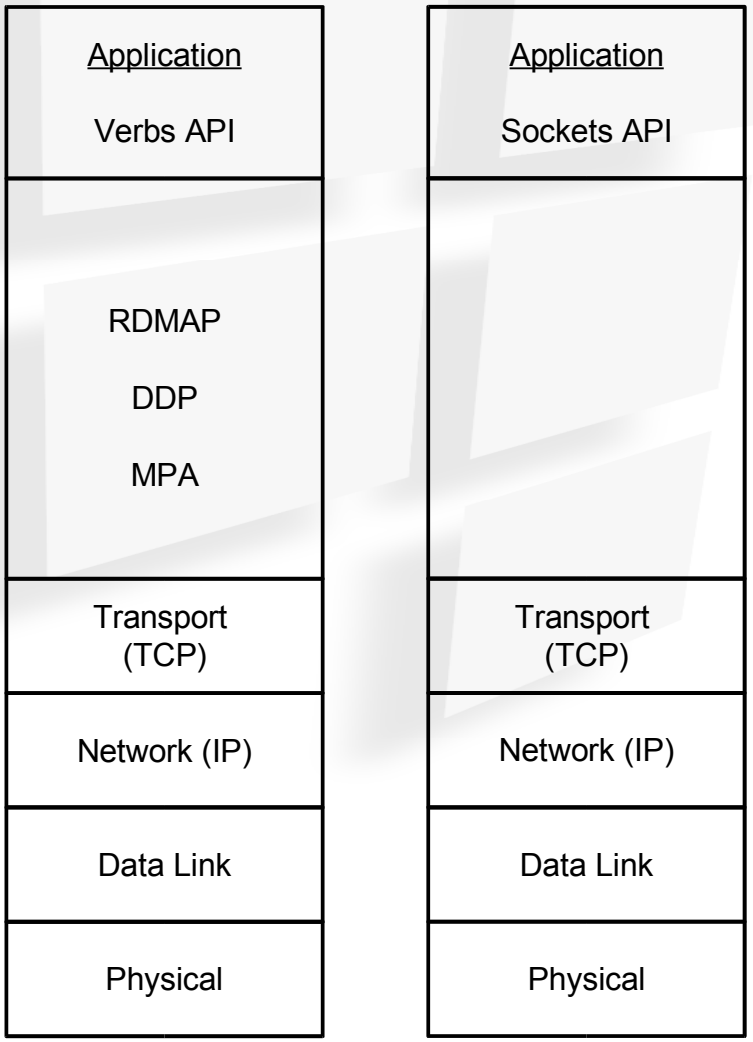
- RDMA over Ethernet.
  - Provides Zero-Copy mechanism to move data.
  - Facilitated by protocol offload.
- RDMA is what makes Infiniband so special.
  - But Infiniband does not work over Ethernet.
- TOE cards provide protocol offload.
  - But TOEs do not provide Zero-Copy.

What is RDMA?


Host A
Kernel Space
Buffer
User Space
Application
BUFFER
N
I
C
Host B
User Space
Application
BUFFER
Kernel Space
Buffer
N
I
C
TCP
iWarp

What is iWarp cont...

- IETF RFCs:
  - RDMAP.
    - RDMA Protocol.
  - DDP.
    - Direct Data Placement protocol.
  - MPA.
    - Marker PDU aligned framing layer.
- Upper level API
  - Verbs
  - MPI
  - DAPL



What is the catch?

- Clearly iWarp is a good thing but....
- Downside:

  Requires special hardware on both ends to take advantage.
- Upside:

  Based on commodity technology that any computer has....Ethernet.
- This is where software iWarp comes in!

  Enable a host to speak the iWarp protocol completely in software.

  iWarp RNIC can talk to a regular Ethernet NIC.

Why Software iWarp?

- Allows a server equipped with an RNIC to take advantage of it even if the other side does not.
- Likely only most crucial of servers will be outfitted with RNICs at first.

  Software iWarp running on clients allows for easy adoption of iWarp.
- No benefit realized on the client end running software iWarp though.

  Point is to benefit the server not the client.

  Thus server can handle many more connections
  - Which in reality benefits the clients.

Our Work

- Implemented RDMA, DDP, MPA, and a verbs API layer in software.
  - This work based on user space software.
  - Kernel space module recently completed!
  - Need both user and kernel space.
- Can successfully communicate with hardware iWarp RNICs from Ammasso.
- Utilize existing TCP stack.
  - Requires no system changes.
  - Regular user can install and run, which makes it possible to incorporate into applications.

Performance Overview

- Add very little overhead to TCP for small messages.
  - CRC has an effect on larger message sizes.
- CPU utilization has been reduced.
  - Sender from 35% to 5%!
  - Receiver from 90% to less than 5%!
  - Results in server being able to scale to many more clients.

Latency

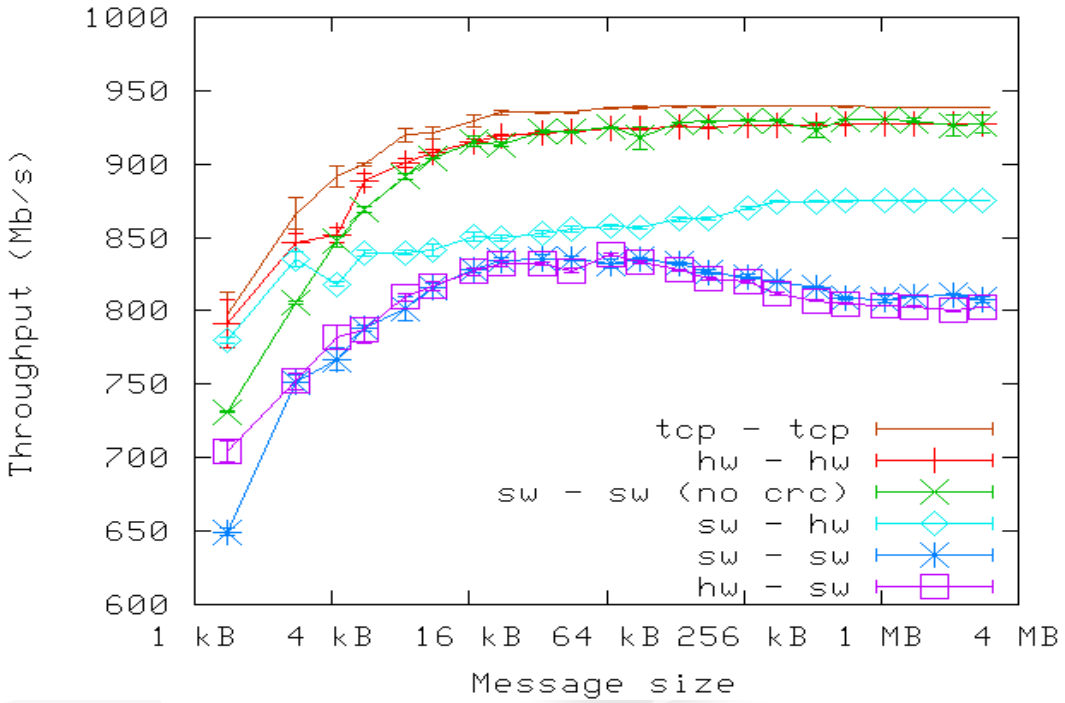
	4 Bytes	64 Kilobytes
hw to hw	15	609.7
sw to sw (no crc)	62.7	687.5
sw to sw (with crc)	62.2	1401.9
hw to sw	62.8	950.4
sw to hw	61.2	937.9
tcp to tcp	62.7	624.5

*times in microseconds

- Small message sizes add very little overhead to TCP.
  CRC shows effect on 64kB messages.
- SW and HW combinations show larger latency at 64kB
  This is due to CRC.
- Latency of HW-HW is very good for small messages.
- Latency is not the main benefit of iWarp. Lack of overhead is important though.

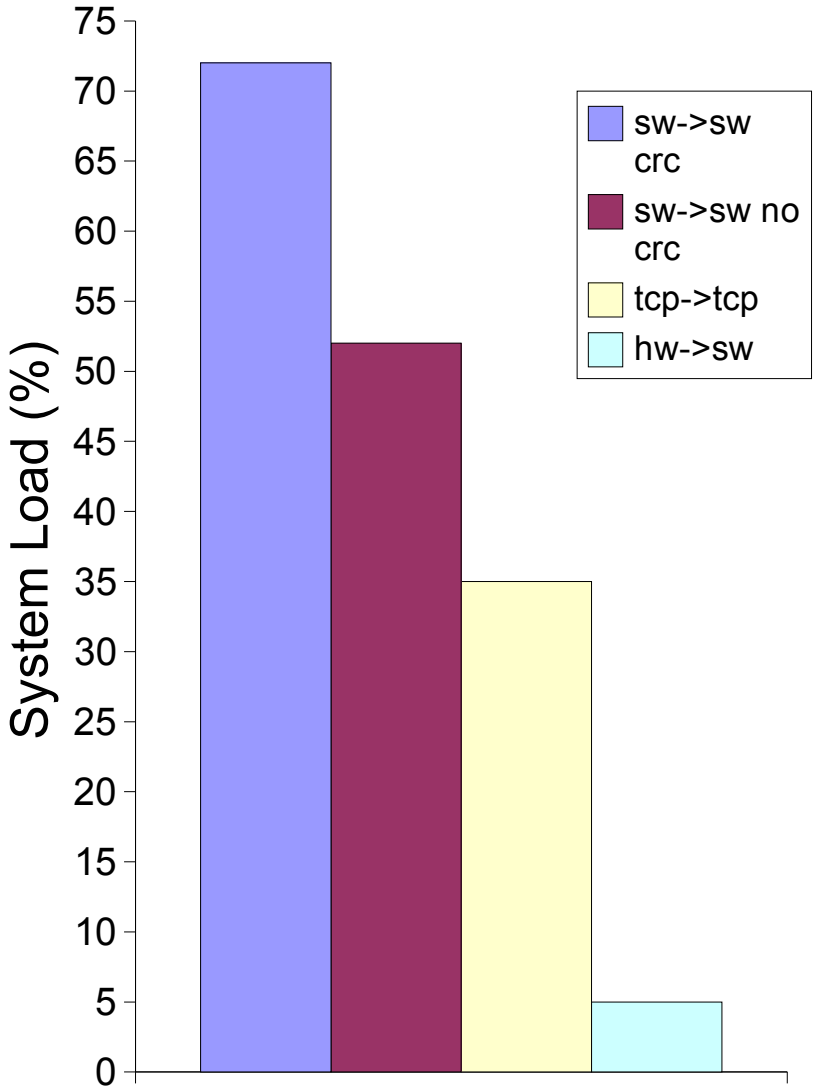
Throughput

- Key point:
  - Everything bounded by TCP.
- CRC Effect:
  - SW-SW without CRC rivals TCP, and HW.
  - With CRC noticeably lower throughput.
- SW-HW higher throughput than HW-SW
  - Because HW can respond faster.



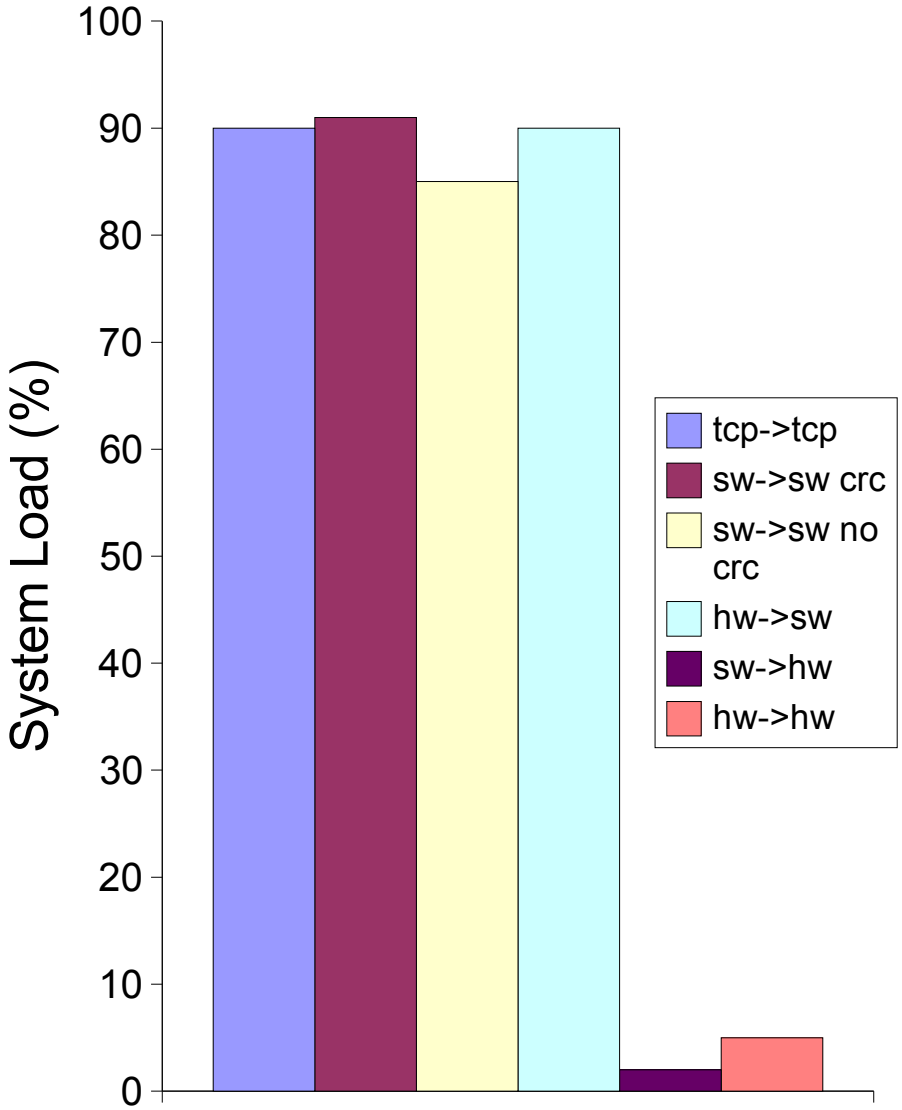
Sending Side System Load

- CRC Effect
  - 23% extra load
- Overhead ontop of TCP/IP
  - CRC – 35%
  - No CRC – 17%
- Improvement with HW->SW
  - 30% Decrease!



Receiver Side System Load

- Receiving is costly
  - Nearly 90% in all tcp and sw cases.
- If receiver is hardware
  - Makes no difference if sender is hardware or software!
  - Exactly what we want!
- Why hw->hw more load than sw->hw?
  - hw->hw gets more done.
  - In sw->hw, the hw has to wait for slow sw to catch up.



Thanks!

- Software available soon.
  - Both user and kernel space implementations.
  - Email for more info.
- Any Questions?

http://www.osc.edu/~dennis/iwarp

Dennis Dalessandro
dennis@osc.edu

OSC SPRINGFIELD

ASC™