Supporting iWARP Compatibility and Features for Regular Network Adapters

P. Balaji   H. –W. Jin   K. Vaidyanathan   D. K. Panda
Network Based Computing Laboratory (NBCL)
Ohio State University
Ethernet Overview

- Ethernet is the most widely used network infrastructure today
- Traditionally Ethernet has been notorious for performance issues
  - Near an order-of-magnitude performance gap compared to other networks
    - Cost conscious architecture
    - Most Ethernet adapters were regular (layer 2) adapters
    - Relied on host-based TCP/IP for network and transport layer support
    - Compatibility with existing infrastructure (switch buffering, MTU)
  - Used by 42.4% of the Top500 supercomputers
  - Key: Reasonable performance at low cost
    - TCP/IP over Gigabit Ethernet (GigE) can nearly saturate the link for current systems
    - Several local stores give out GigE cards free of cost!
- 10-Gigabit Ethernet (10GigE) recently introduced
  - 10-fold (theoretical) increase in performance while retaining existing features
Ethernet: Technology Trends

• Broken into three levels of technologies
  – Regular Ethernet adapters \([feng03:hoti, feng03:sc, balaji04:rait]\)
    • Layer-2 adapters
    • Rely on host-based TCP/IP to provide network/transport functionality
    • Could achieve a high performance with optimizations
  – TCP Offload Engines (TOEs) \([balaji05:hoti, balaji05:cluster]\)
    • Layer-4 adapters
    • Have the entire TCP/IP stack offloaded on to hardware
    • Sockets layer retained in the host space
  – iWARP-aware adapters \([jin05:hpide, wyckoff05:rait]\)
    • Layer-4 adapters
    • Entire TCP/IP stack offloaded on to hardware
    • Support more features than TCP Offload Engines
      – No sockets! Richer iWARP interface!
      – E.g., Out-of-order placement of data, RDMA semantics
Current Usage of Ethernet

System Area Network or Cluster Environment

Regular Ethernet

TOE

iWARP

Wide Area Network

Regular Ethernet Cluster

TOE Cluster

Distributed Cluster Environment

iWARP Cluster
Problem Statement

- Regular Ethernet adapters and TOEs are completely compatible
  - Network level compatibility (Ethernet + IP + TCP + application payload)
  - Interface level compatibility (both expose the sockets interface)
- With the advent of iWARP, this compatibility is disturbed
  - Both ends of a connection need to be iWARP compliant
    - Intermediate nodes do not need to understand iWARP
  - The interface exposed is no longer sockets
    - iWARP exposes a much richer and newer API
    - Zero-copy, asynchronous and one-sided communication primitives
    - Not very good for existing applications
- Two primary requirements for a wide-spread acceptance of iWARP
  - Software Compatibility for Regular Ethernet with iWARP capable adapters
  - A common interface which is similar to sockets and has the features of iWARP
Presentation Overview

Introduction and Motivation

TCP Offload Engines and iWARP

Overview of the Proposed Software Stack

Performance Evaluation

Conclusions and Future Work
What is a TCP Offload Engine (TOE)?
iWARP Protocol Suite

Middle Box Fragmentation

RDPD ULP
RDMAP ULP

RDMAP

RDPD

MPA
SCTP

TCP
IP

Feature Rich Interface
In-order Delivery and Out-of-order Placement

Courtesy iWARP Specification

More details provided in the paper or in the iWARP Specification
Presentation Overview

- Introduction and Motivation
- TCP Offload Engines and iWARP
- Overview of the Proposed Software Stack
- Performance Evaluation
- Conclusions and Future Work
Proposed Software Stack

- The Proposed Software stack is broken into two layers
  - Software iWARP implementation
    - Provides wire compatibility with iWARP-compliant adapters
    - Exposes the iWARP feature set to the upper layers
    - Two implementations provided: User-level iWARP and Kernel-level iWARP
  - Extended Sockets Interface
    - Extends the sockets interface to encompass the iWARP features
    - Maps a single file descriptor to both the iWARP as well as the normal TCP connection
    - Standard sockets applications can run WITHOUT any modifications
    - Minor modifications to applications required to utilize the richer feature set
Software iWARP and Extended Sockets Interface
Designing the Software Stack

- User-level iWARP implementation
  - Non-blocking Communication Operations
  - Asynchronous Communication Progress
- Kernel-level iWARP implementation
  - Zero-copy data transmission and single-copy data reception
  - Handling Out-of-order segments
- Extended Sockets Interface
  - Generic Design to work over any iWARP implementation
Non-Blocking and Asynchronous Communication

User-level iWARP is a multi-threaded implementation
Zero-copy Transmission in Kernel-level iWARP

- Memory map user buffers to kernel buffers
- Mapping needs to be in place till the reliability ACK is received
- Buffers are mapped during memory registration
  - Avoids mapping overhead during data transmission
Handling Out-of-order Segments

- Data is retained in the Socket buffer even after it is placed!
- This ensures that TCP/IP handles reliability and not the iWARP stack
Presentation Overview

- Introduction and Motivation
- TCP Offload Engines and iWARP
- Overview of the Proposed Software Stack
- Performance Evaluation
- Conclusions and Future Work
Experimental Test-bed

- Cluster of Four Node P-III 700MHz Quad-nodes
- 1GB 266MHz SDRAM
- Alteon Gigabit Ethernet Network Adapters
- Packet Engine 4-port Gigabit Ethernet switch
- Linux 2.4.18-smp
Ping-Pong Latency Test

![Graph of Ping-Pong Latency (Extended Interface)](image1)

![Graph of Ping-Pong Latency (Sockets Interface)](image2)
Uni-directional Stream Bandwidth Test

Bandwidth (Extended Interface)
- TCP/IP
- User-level iWARP
- Kernel-level iWARP

Bandwidth (Sockets Interface)
- TCP/IP
- User-level iWARP
- Kernel-level iWARP
Software Distribution

- Public Distribution of User-level and Kernel-level Implementations
  - User-level Library
  - Kernel module for 2.4 kernels
  - Kernel patch for 2.4.18 kernel
  - Extended Sockets Interface for software iWARP

- Contact Information
  - {panda, balaji}@cse.ohio-state.edu
  - http://nowlab.cse.ohio-state.edu
Presentation Overview

- Introduction and Motivation
- TCP Offload Engines and iWARP
- Overview of the Proposed Software Stack
- Performance Evaluation

- Conclusions and Future Work
Concluding Remarks

- Ethernet has been broken down into three technology levels
  - Regular Ethernet, TCP Offload Engines and iWARP-compliant adapters
  - Compatibility between these technologies is important
- Regular Ethernet and TOE are completely compatible
  - Both the wire protocol and the ULP interface are the same
  - iWARP does not share such compatibility
- Two primary requirements for a wide-spread acceptance of iWARP
  - Software Compatibility for Regular Ethernet with iWARP capable adapters
  - A common interface which is similar to sockets and has the features of iWARP
- We provided a software stack which meets these requirements
Continuing and Future Work

• The current Software iWARP is only built for Regular Ethernet
  – TCP Offload Engines provide more features than Regular Ethernet
  – Needs to be extended to all kinds of Ethernet networks
    • E.g., TCP Offload Engines, iWARP-compliant adapters, Myrinet 10G adapters

• Interoperability with Ammasso RNICs
  – Modularized approach to enable/disable components in the iWARP stack

• Simulated Framework for studying NIC architectures
  – NUMA Architectures on the NIC for iWARP Offload

• Flow Control/Buffer Management Features for Extended Sockets
Acknowledgments
Web Pointers

NBCL

Website: http://www.cse.ohio-state.edu/~balaji

Group Homepage: http://nowlab.cse.ohio-state.edu

Email: balaji@cse.ohio-state.edu