An Accelerated Implementation of Portals on the Cray SeaStar

Ron Brightwell    Kevin Pedretti    Keith Underwood
Sandia National Laboratories
Center for Computation, Computers, Information, and Mathematics

Trammell Hudson
OS Research

Cray Users’ Group Annual Technical Conference
May 11, 2006
Outline

• SeaStar
• Portals
• CORPSE
• Performance
• Future Work
Cray SeaStar NIC/Router

- 16 1.6 Gb/s HyperTransport to Opteron
- 500 MHz embedded PowerPC 440
- 384 KB on-board scratch RAM
- Seven-port router
- Six 12-channel 3.2 Gb/s high-speed serial links
SeaStar Block Diagram
Portals 3.3 for SeaStar

- Cray started with Sandia reference implementation
- Needed single version of NIC firmware that supports all combinations of
  - User-level and kernel-level API
  - NIC-space and kernel-space library
- Cray added bridge layer to reference implementation to allow NAL to interface multiple API NALs and multiple library NALs
  - qkbridge for Catamount applications
  - ukbridge for Linux user-level applications
  - kbridge for Linux kernel-level applications
SeaStar NAL

- Portals processing in kernel-space
  - Interrupt-driven
  - “generic” mode
- Portals processing in NIC-space
  - No interrupts
  - “accelerated” mode
Prototype NIC-based Network Stack

• Allowed characterization of
  – Impact of interrupts on latency
  – Impact on throughput (Messages/second)
  – NIC vs. host CPU matching speed
  – Penalty of having multiple NIC mailboxes
  – NIC resource requirements of each CPU core
Addressing
Steps for Address Translation

After node and process selection, passing of ACL, and selecting correct Portals table entry.

Start

- next match entry

match? no

- MD exists and accepts?

  yes

  MD has EQ and START events are enabled?

    yes

    record START event into EQ

    no

  perform operation

    yes

    unlink ME

    yes

    record END event into EQ

    no

  unlink MD

    yes

  record END event into EQ

    no

  unlink ME

    yes

  unlink MD

    yes

  increment drop count

  no

discard message

End
Put Operation

Initiator

1. Put Request

Target

Access Control List

Portals Table

ME

ME

ME

MDs

MD

2. Data

3. (optional) Acknowledgment
Get Operation

1. Get Request
2. Data

Initiator

Target

Access Control List
Portals Table
ME
ME
ME
MDs

MD
Host-Based Portals Implementation

Application

Portals

DMA Commands

QK

DMA Commands

SeaStar Interrupt

Trap
NIC-Based Portals Implementation

Application

QK

Portals

DMA Commands

SeaStar

DMA Commands
Portals Application Mailbox

• Untrusted mailbox between application and firmware
• Initialization is via kernel mailbox
  – Maps the processes address space
  – Physically contiguous so whole mapping is done
  – Only works with Catamount
• Application mailbox
  – All other Portals command are delivered directly to the SeaStar
  – Trusted header
  – Sending from SeaStar memory is prohibitive
Flow Control

• CAM Overflow Remediation Protocol SystEm (CORPSE)
  – Sandia’s protocol that runs entirely on the SeaStar
• CAM Overflow Protocol
  – Cray’s protocol that runs entirely on the Opteron
Changes to Portals/MPI for Accelerated

• MPI receive posting was slow
  – Posting a receive involves a round trip to the SeaStar (1 us) relative a kernel trap (65 ns)
  – Combine PtIMEAAttach(), PtIMDAttach(), PtIMDUpdate int PtIMEMDPost()

• Reduce HT transfers on the send side
  – Move send-side MD creation out of fast path
  – Create three MDs that cover all of data, stack, and heap
  – Improves message rate and fixes the amount of resources on the send-side
Benefits of Accelerated

- OS does not run on message arrival
- No context switch overhead
- Portals address translation done by SeaStar
Latency
Bandwidth
Streaming Bandwidth
Bi-Directional Streaming Bandwidth
Barrier
Allreduce
Allgather
Where the Time Goes
Impact of the Length of the Posted Receive Queue

• Most CPU intensive part of Portals is traversing the list of match entries
• Opteron can do this at 2 GHz
• PowerPC can do this at 500 MHz
Overhead Comparison of Flow Control

• Is there an advantage for doing NIC-based flow control versus host-based flow-control?
Latency
Bandwidth
Reduce
Allreduce
Allgather
Ongoing Work

- Accelerated Portals (AP) being integrated into Cray’s development tree for a 1.5 release
- More extensive measurements of the impact of AP
- Portals collective library
  - Collective operations built on top of Portals
- Non-blocking collective functions
  - Collective operations integrated into Portals
  - SeaStar can support offloading collective operations
  - Barrier proof-of-concept is done and working
Impact of Linux on Latency
Impact of Linux on Latency