Portals 3.3 on the Sandia/Cray Red Storm System

Ron Brightwell Trammell Hudson Kevin Pedretti
Rolf Riesen Keith Underwood

Sandia National Laboratories

Scalable Computing Systems Department

Cray User Group Annual Technical Conference

May 18, 2005

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

Sandia National Laboratories

Outline

- Portals history
- Portals objects
- Portals implementation
- Portals for Red Storm
- Performance on Red Storm
- Conclusions

Sandia National Laboratories

Portals Timeline

- Portals 0.0 - 1991
  - SUNMOS (Sandia/UNM OS)
  - nCUBE, Intel Paragon
  - Direct access to network FIFOs
  - Message co-processor
- Portals 1.0 - 1993
  - Data structures in user-space
  - Kernel-managed and user-managed memory descriptors
  - Published but never implemented
- Portals 2.0 - 1994
  - Puma/Cougar
  - Message selection (match lists)
  - Four types of memory descriptors (three implemented)
- Portals 3.0 - 1998
  - Cplant/Linux
  - Functional API
  - Target intelligent/programmable network interfaces

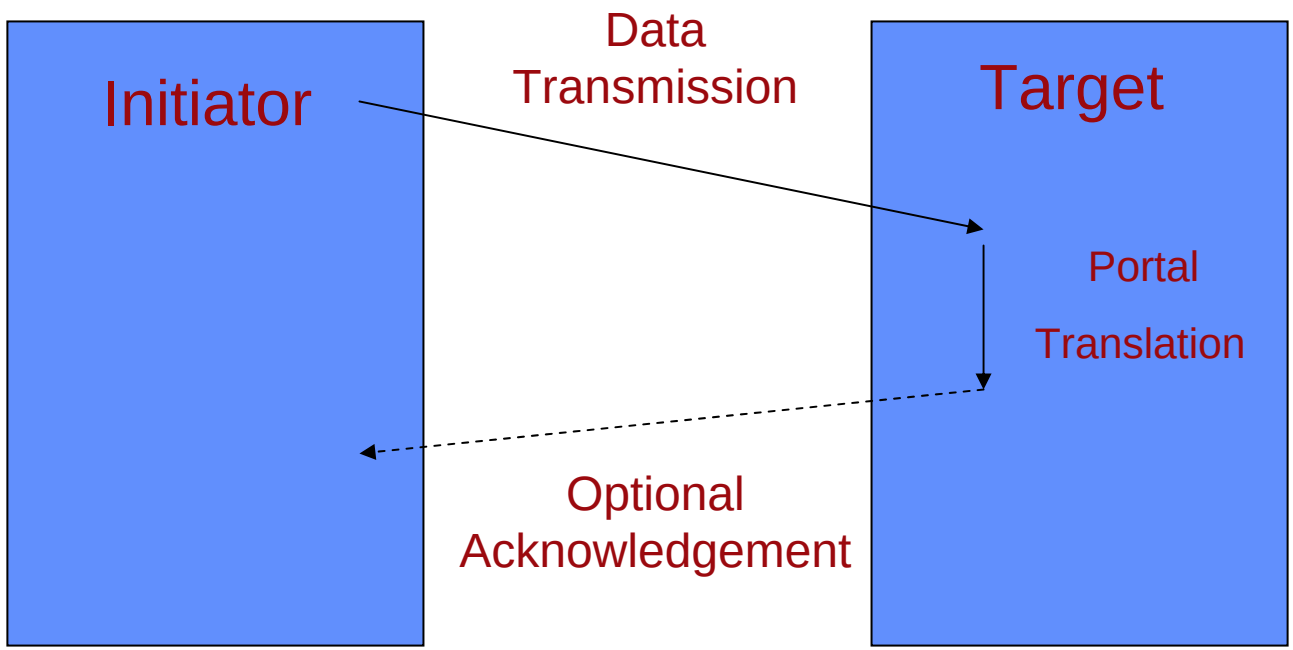
Sandia National Laboratories

Portals 3.3 Features

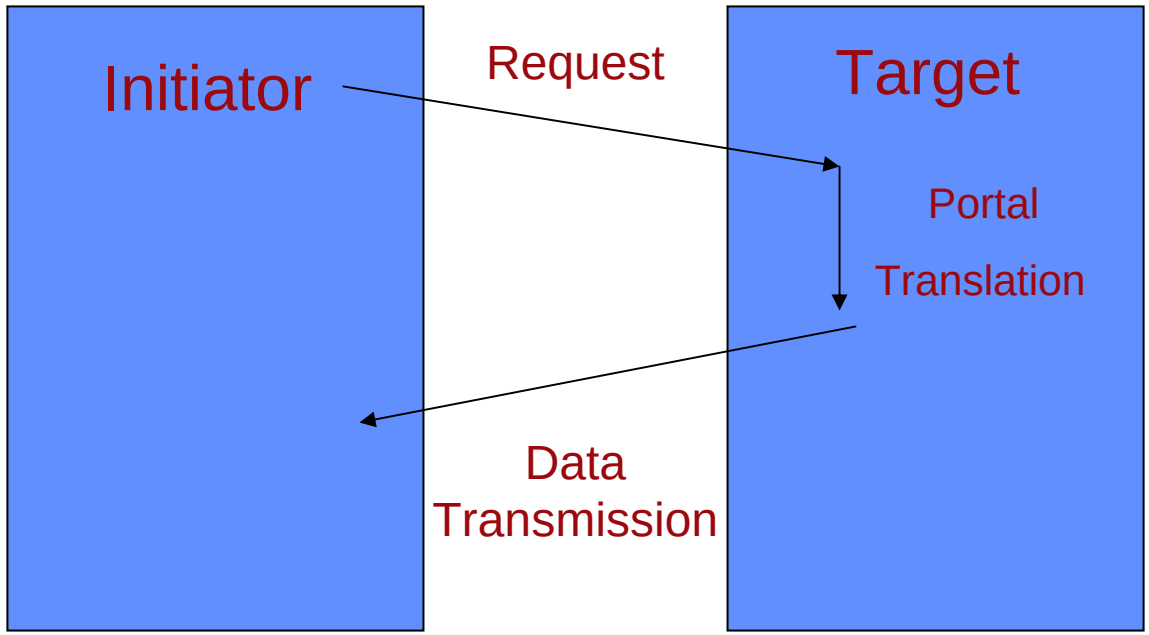
- Best effort, in-order delivery
- Well-defined transport failure semantics
- Based on expected messages
- One-sided operations
  - Put, Get, Atomic swap
- Zero-copy
- OS-bypass
- Application offload
  - No polling or threads to move data
  - No host CPU overhead
- Runtime system independent

Sandia National Laboratories

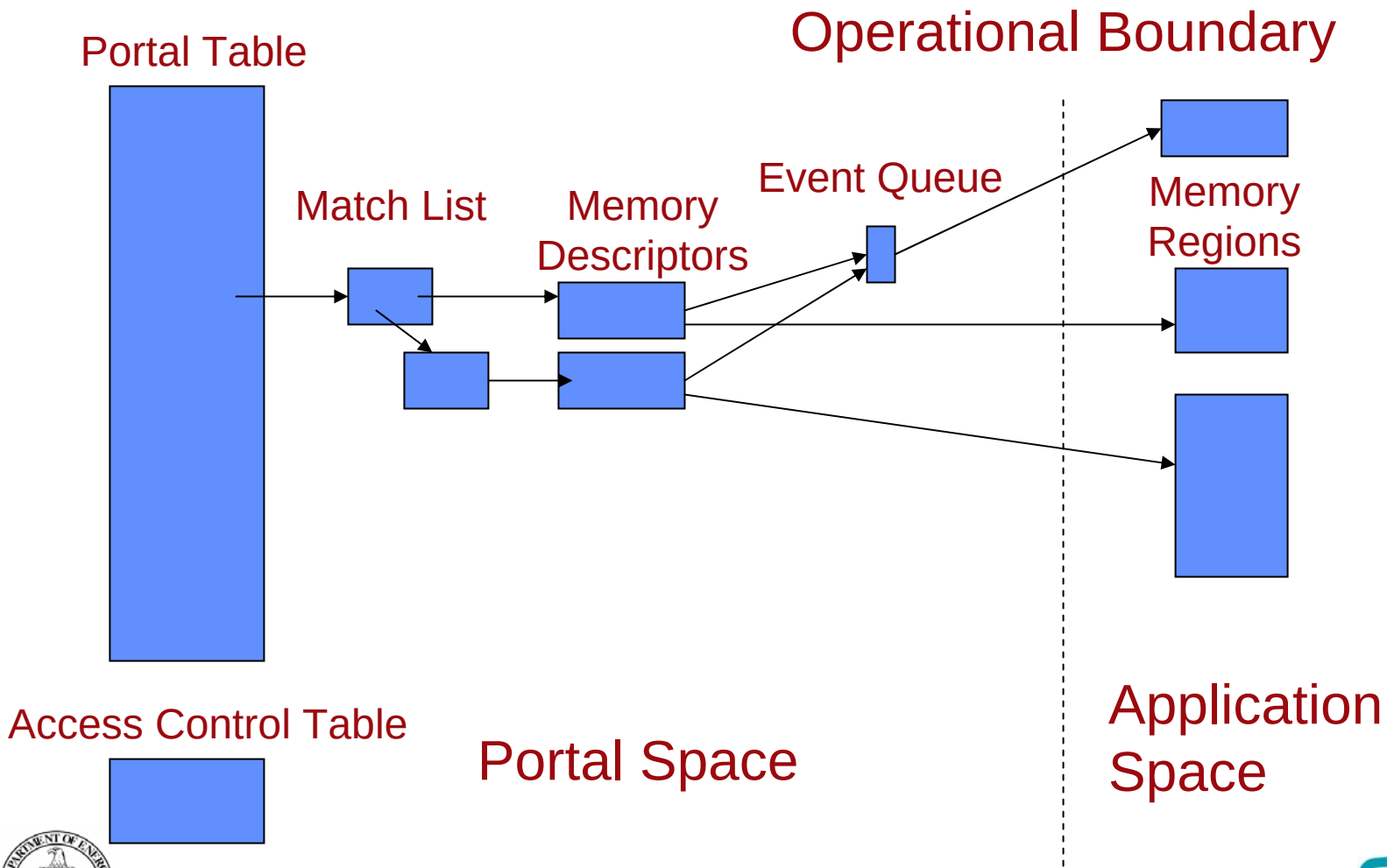
Portal Put



Portal Get



Portals Addressing



Match Entry Contents

- Source node id
- Source process id
- 64 match bits
- 64 ignore bits

Sandia National Laboratories

Memory Descriptor

- Start address
  - Optionally supports gather/scatter list
- Length in bytes
- Threshold
  - Number of operations allowed
- Max size
  - Low-water mark
- Options
  - Put/get
  - Receiver/sender managed offset
  - Truncate
  - Ack/no ack
  - Ignore start/end events
- 64 bits of user data
- Event queue handle
- Auto-unlink option

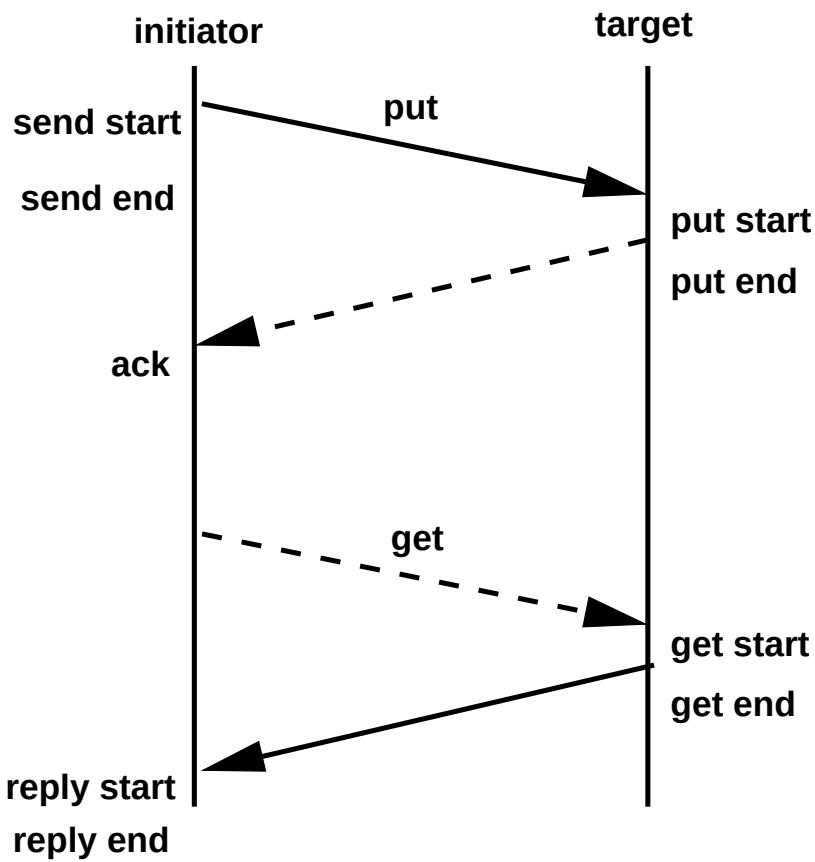
Sandia National Laboratories

Event Queue

- Circular queue that records operations on MDs
- Types of events
  - Get (PTL_EVENT_GET_{START,END})
    - MD has received a get request
  - Put (PTL_EVENT_PUT_{START,END})
    - MD has received a put request
  - Reply (PTL_EVENT_REPLY_{START,END})
    - MD has received a reply to a get request
  - Send (PTL_EVENT_SEND_{START,END})
    - Put request has been processed
  - Ack (PTL_EVENT_ACK)
    - MD has received an ack to a put request

Sandia National Laboratories

Event Scenarios



Sandia National Laboratories

Event Entry Contents

- Event type
- Initiator of event (nid,pid)
- Portal index
- Match bits
- Requested length
- Manipulated length
- Offset
- MD
- 64 bits of out-of-band data
- Link
- Sequence number

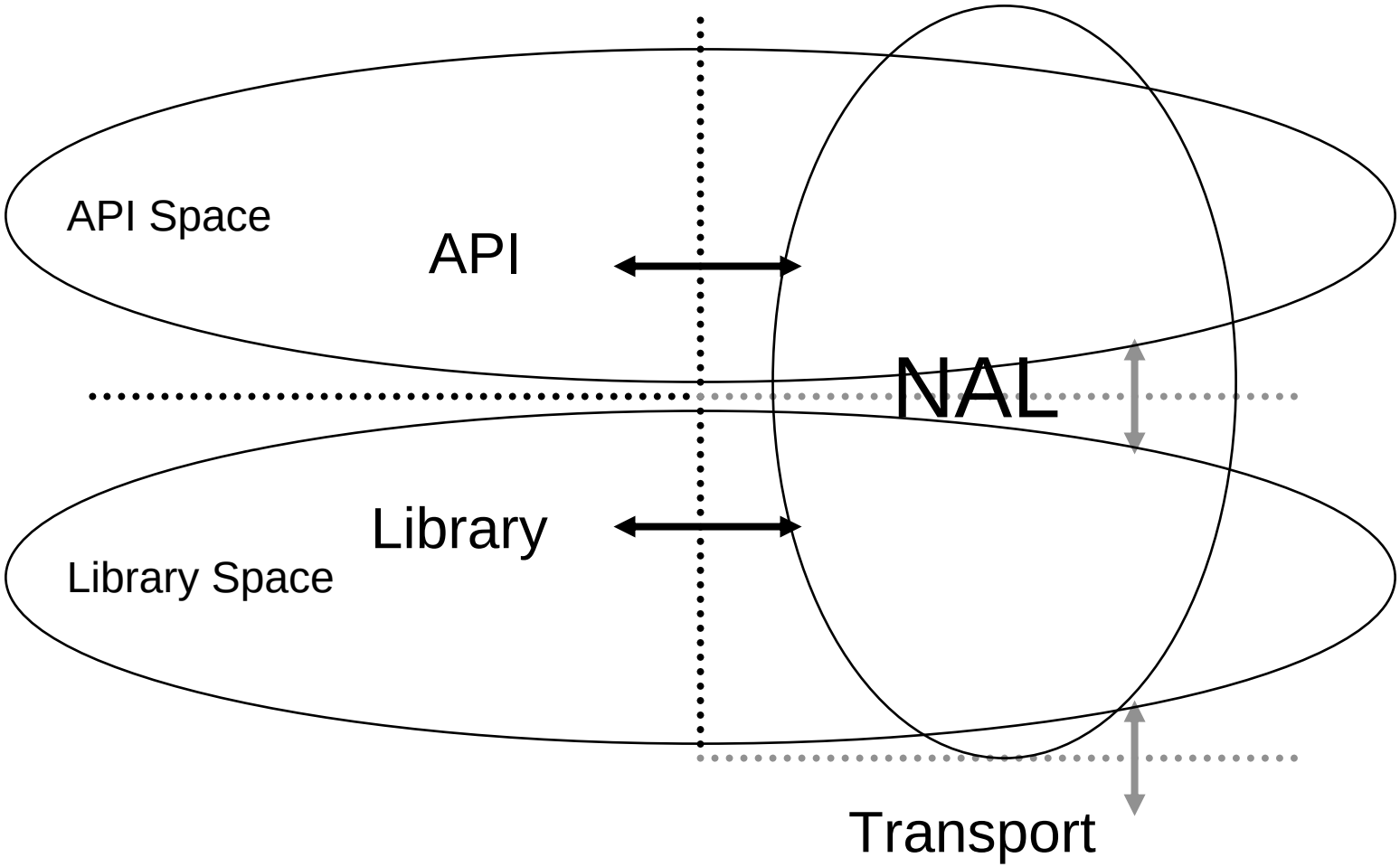
Sandia National Laboratories

What Makes Portals Different?

- Provides elementary building blocks for supporting higher-level protocols well
- Allows structures to be placed in user-space, kernel-space, or NIC-space
- Receiver-managed offset allows for efficient and scalable buffering of “unexpected” messages
- Supports multiple protocols within a process

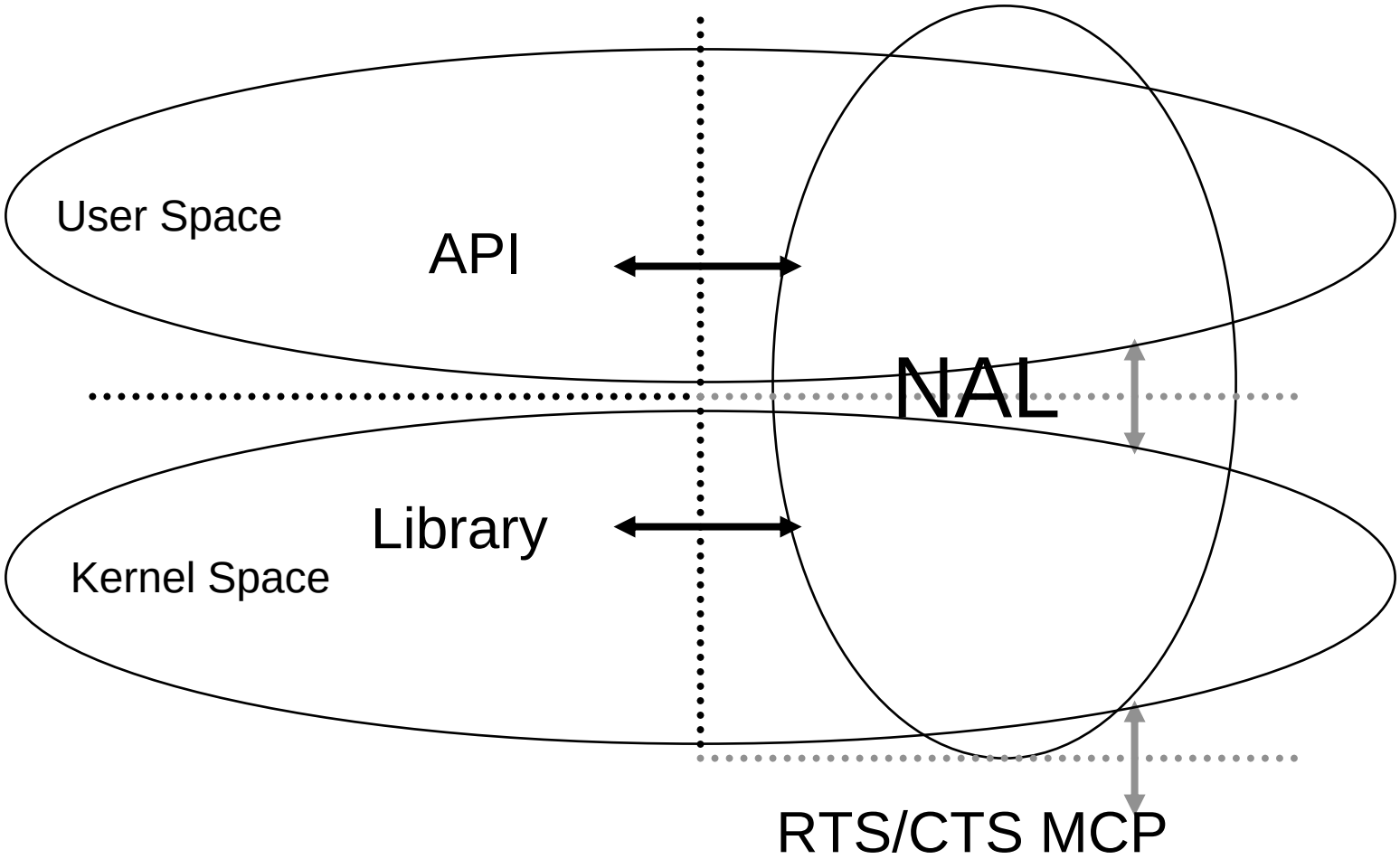
Sandia National Laboratories

Portals Reference Implementation Design



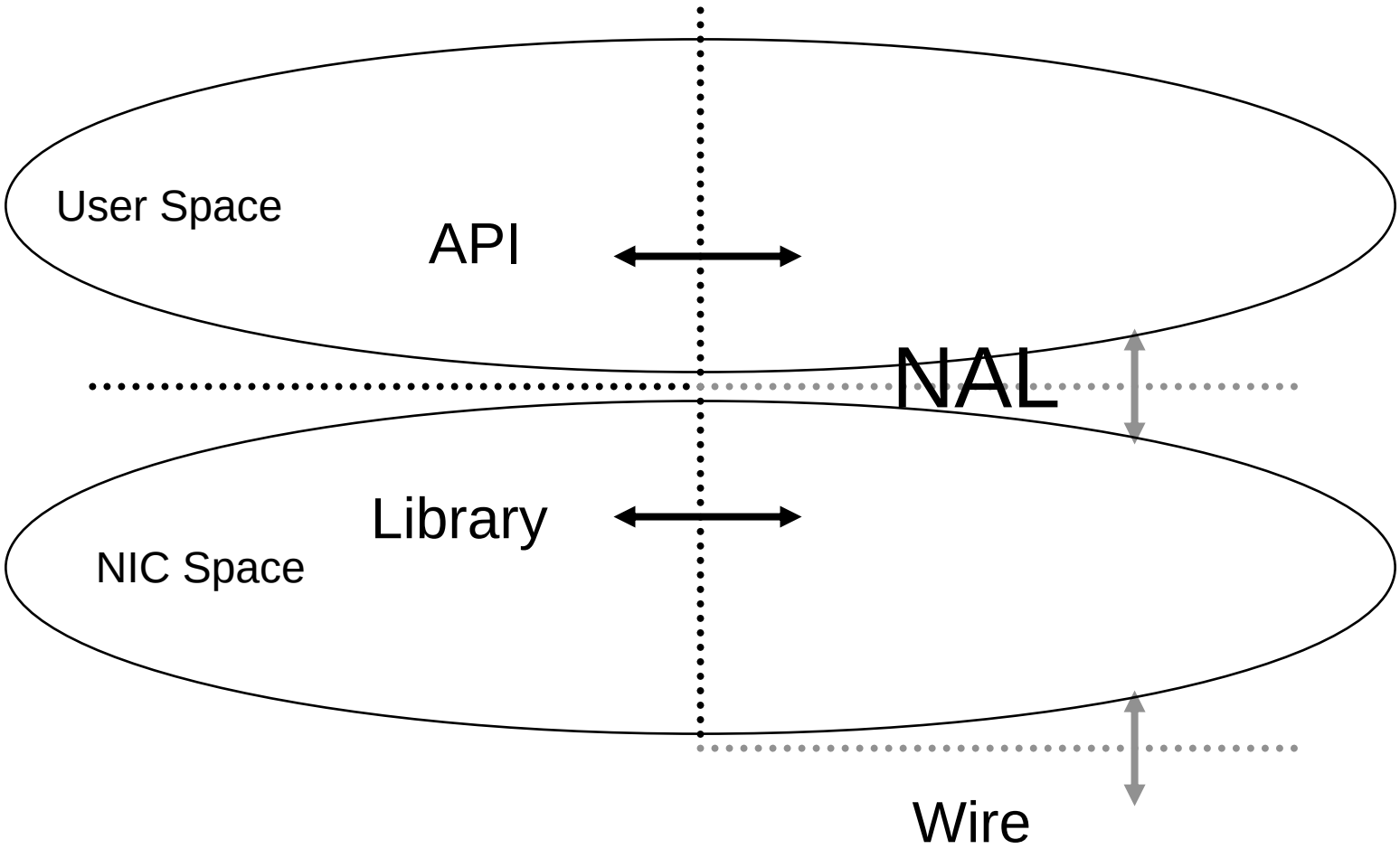
Sandia National Laboratories

Myrinet Kernel Implementation



Sandia National Laboratories

Myrinet MCP Implementation



Sandia National Laboratories

Cray Portals Bridge

- 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

Sandia National Laboratories

SeaStar NAL

- Kevin already told you this ☺
- Portals library currently in kernel-space
  - Interrupt-driven
  - “generic”
- Portals library moving to NIC-space
  - No interrupts
  - “accelerated”

Sandia National Laboratories

Standard Red Storm/XT3 Performance Disclaimer

- Performance results are from a snapshot of a developer code base
- Sandia C firmware stack
- Some features that may impact performance are not implemented
  - End-to-end reliability protocol

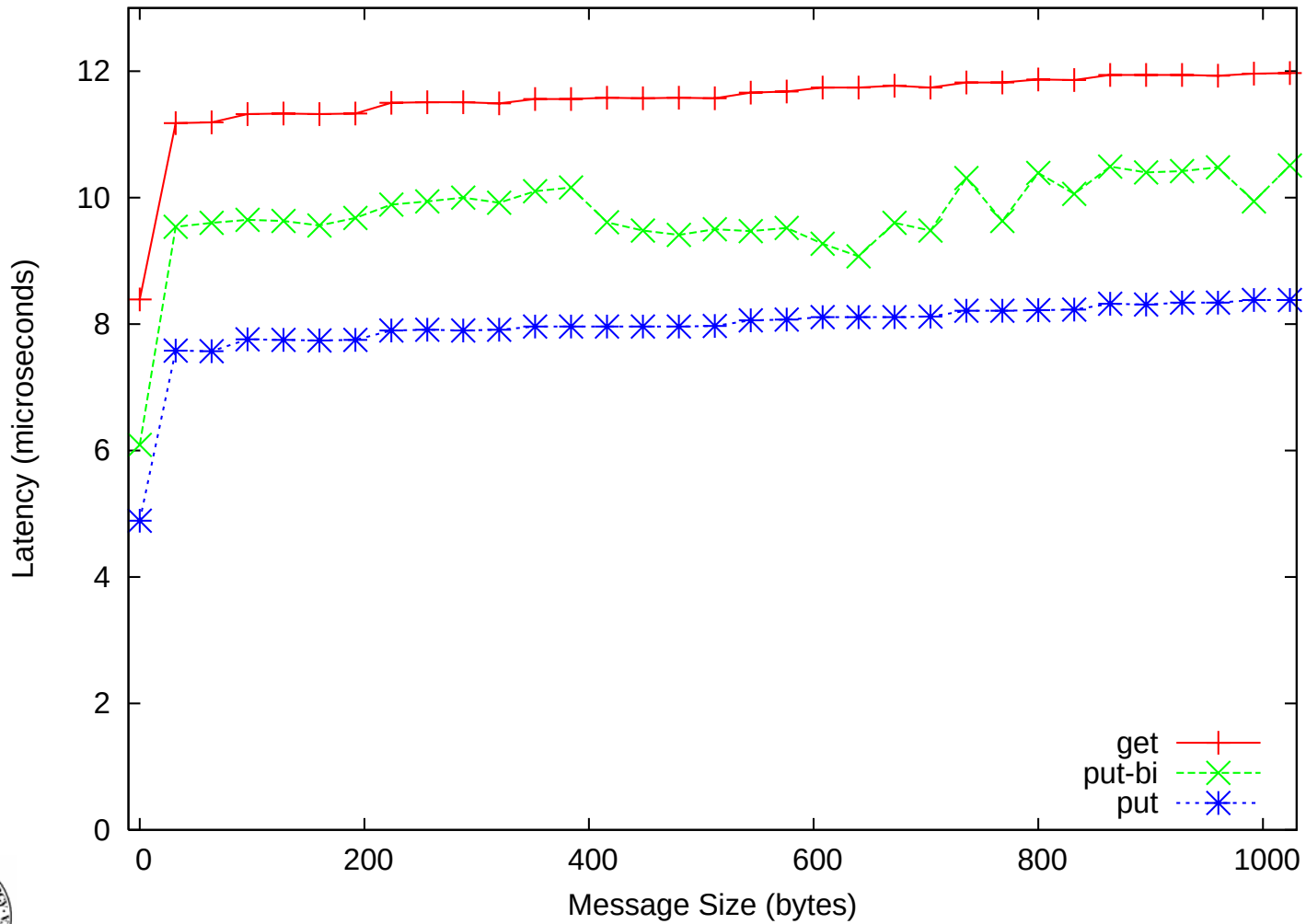
Sandia National Laboratories

Micro-Benchmarks

- **PtlPerf**
  - **Ping-pong latency and bandwidth (uni- and bi-directional)**
  - **Single, persistent ME, MD, EQ**
  - **Best-case performance for Portals**
- **NetPIPE 3.6.2**
  - **Ping-pong latency and bandwidth (uni- and bi-directional)**
  - **Streaming bandwidth**
  - **Implemented Portals module**

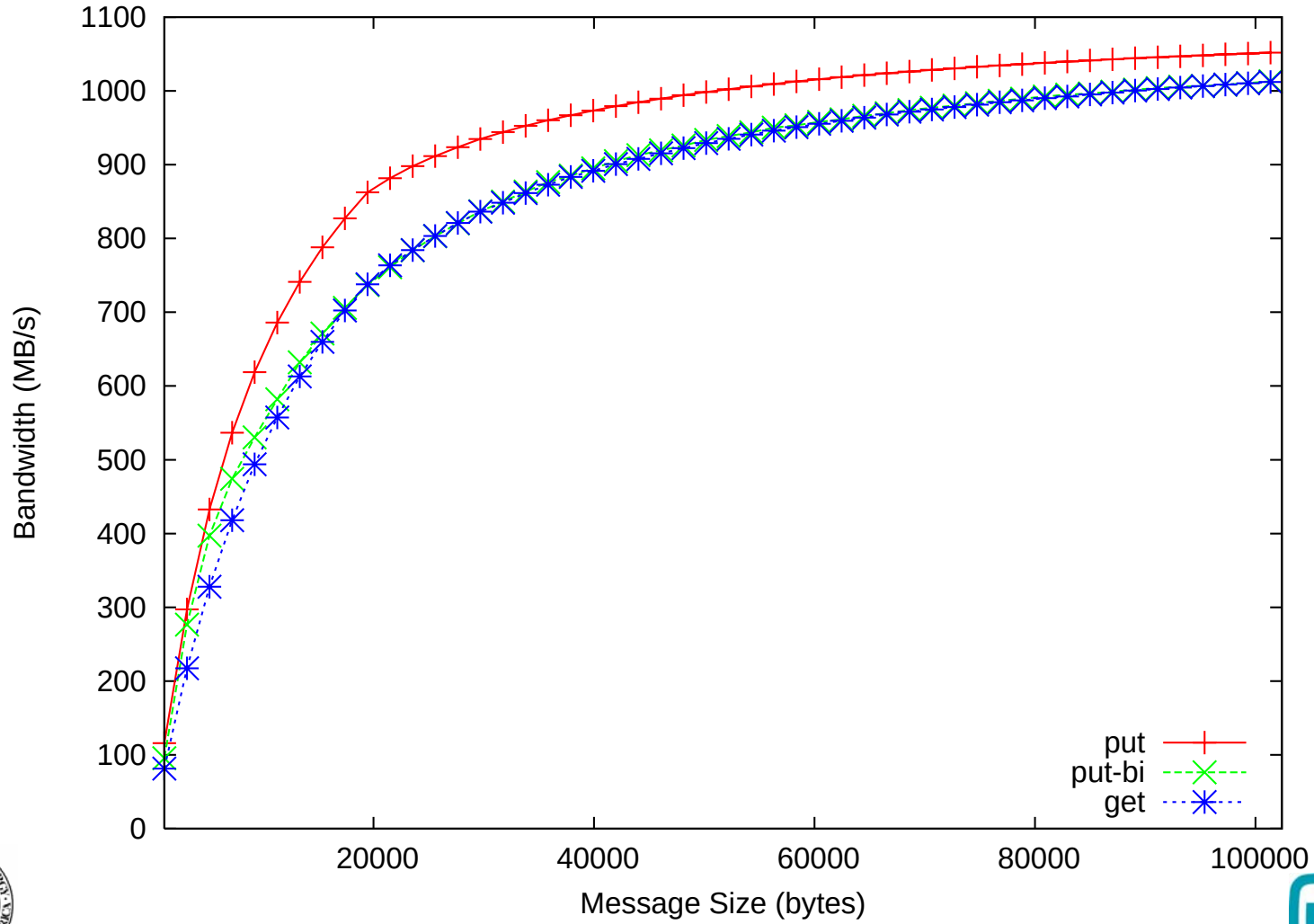
Sandia National Laboratories

PtlPerf Latency



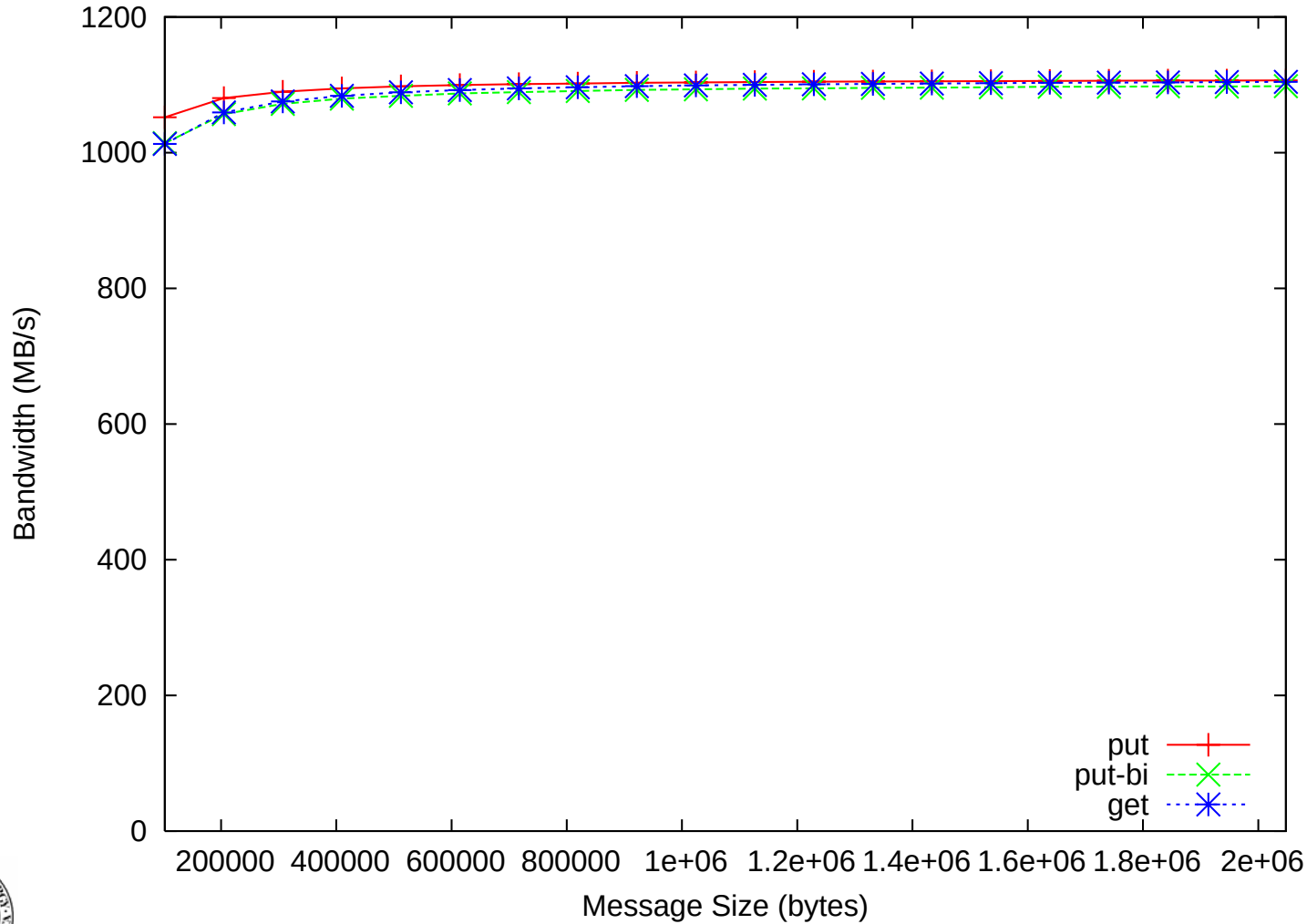
Sandia National Laboratories

PtlPerf Bandwidth (10KB-100KB)



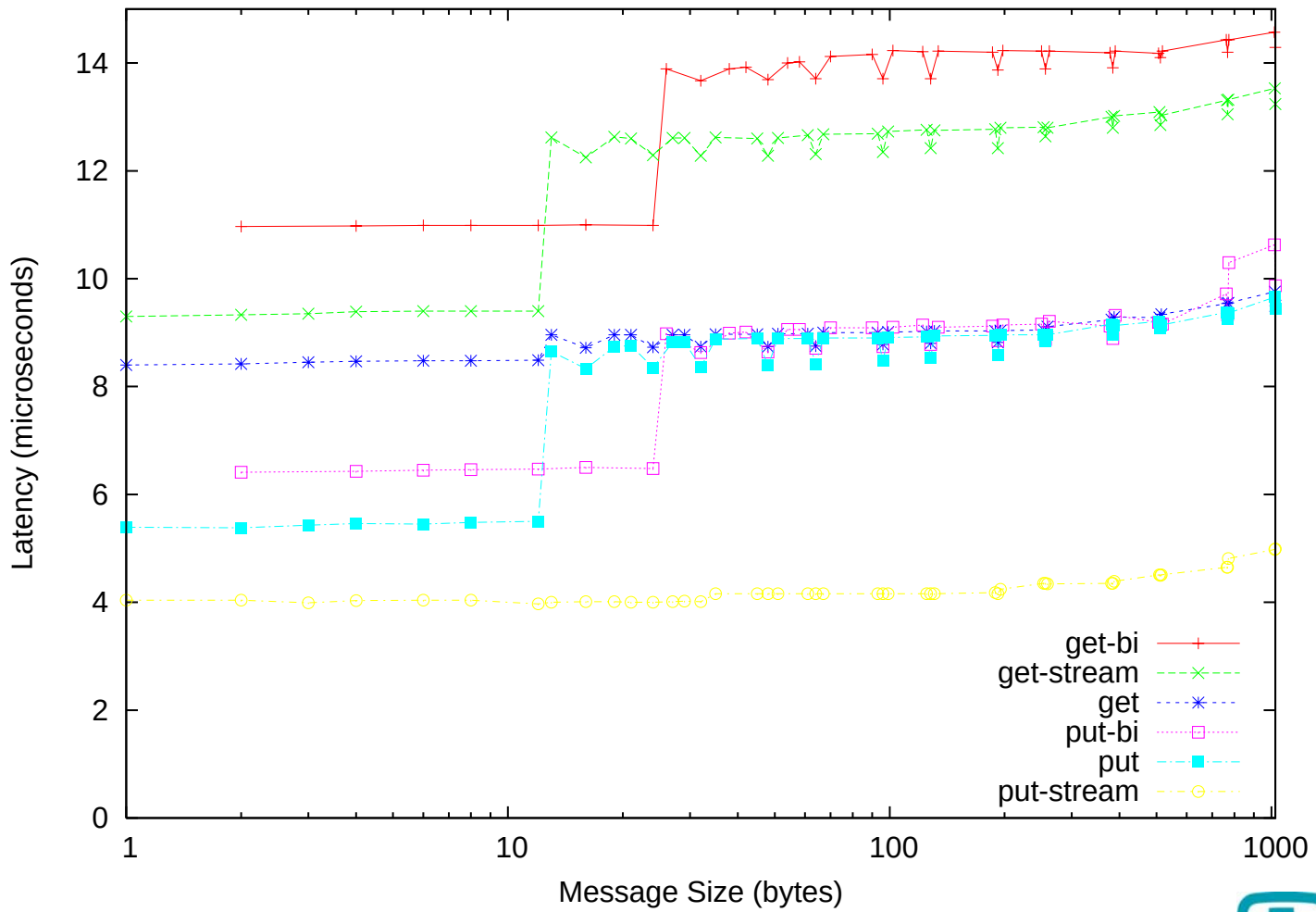
Sandia National Laboratories

PtlPerf Bandwidth (100KB-2MB)



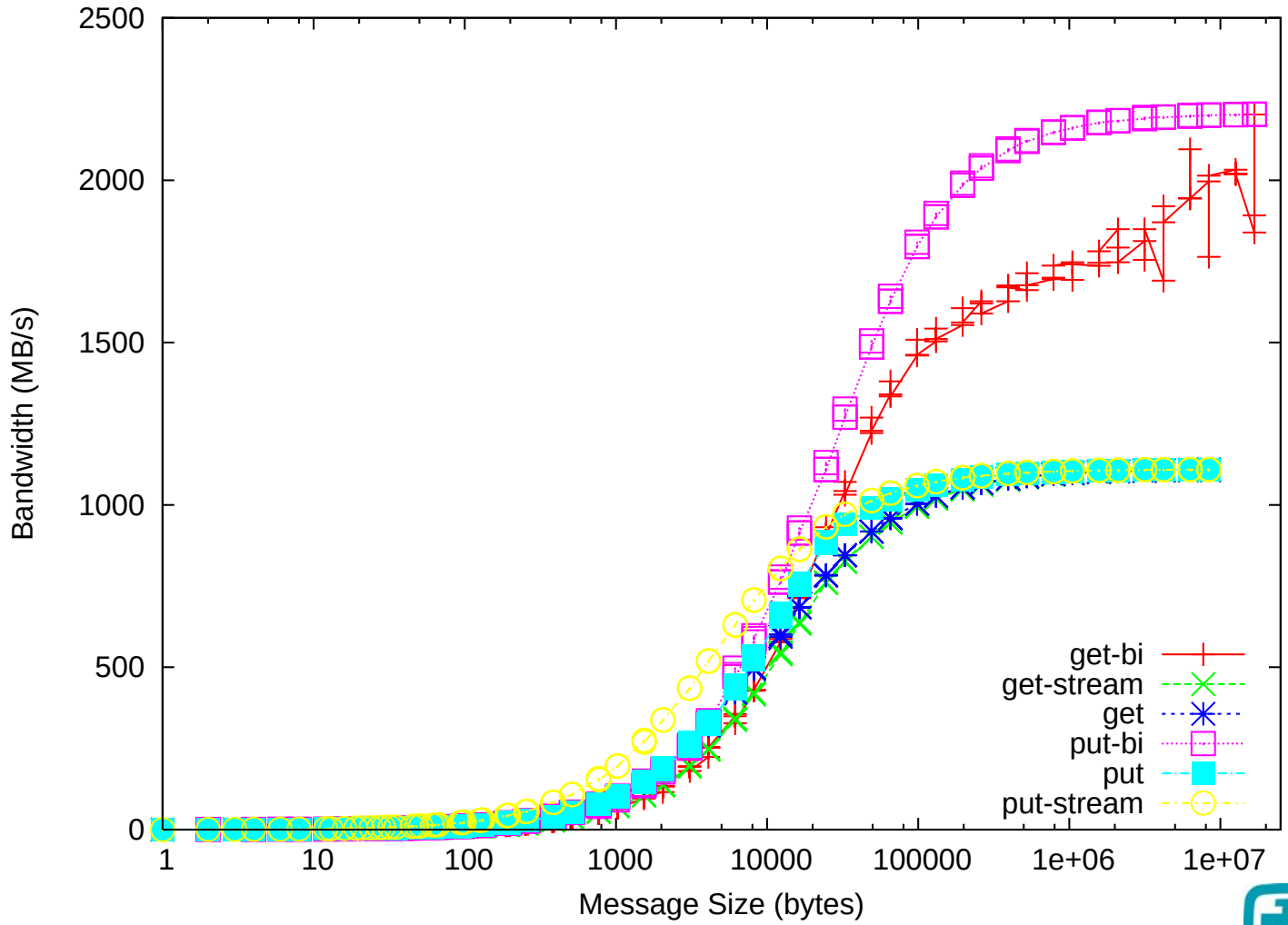
Sandia National Laboratories

NetPIPE Latency



Sandia National Laboratories

NetPIPE Bandwidth



Sandia National Laboratories

Conclusions

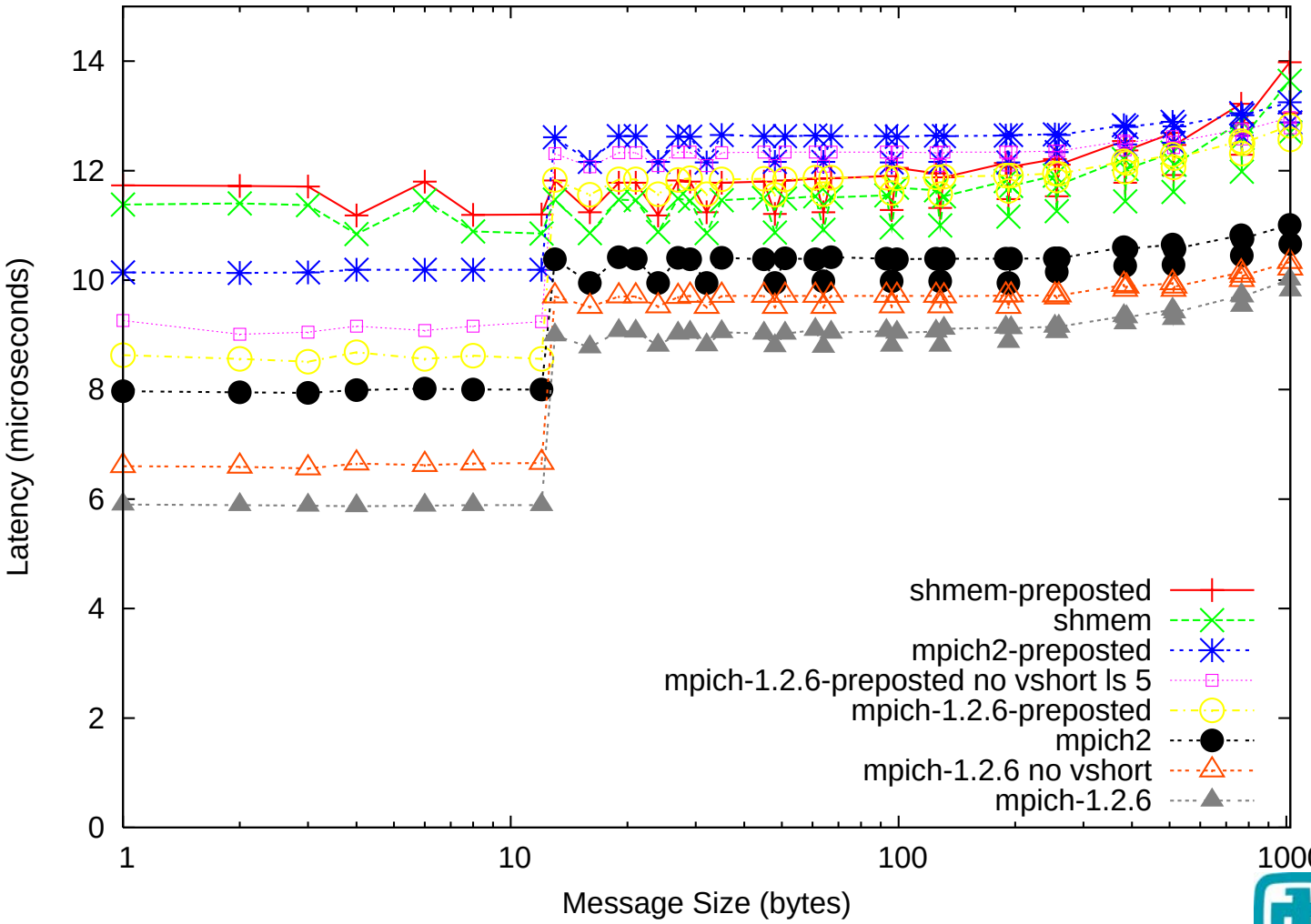
- Portals 3.3 is the lowest-level network programming interface on Red Storm
- Cray bridge abstraction allows single instance of firmware to support multiple API and Library paths
- Interrupt-driven kernel-space Library implementation achieves ~4.8 µs
- Expect NIC-space Library implementation to do better

Sandia National Laboratories

Questions?

Sandia National Laboratories

MPI Latency



Sandia National Laboratories