Exceptional service in the national interest

Sandia National Laboratories

Challenges and Opportunities for HPC Interconnects and MPI

CCR Center for Computing Research

Ron Brightwell, R&D Manager

Scalable System Software Department

U.S. DEPARTMENT OF ENERGY | National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Outline

Sandia National Laboratories

- Prior analysis of challenges facing HPC interconnects
  - Seven years later
- Exascale interconnect hardware trends
  - Plus application and usage model trends
- Near-term research on extensions and optimizations for MPI
  - Matching optimizations
  - Finepoints – alternative proposal to Endpoints
- New portable programming model for network offloading

CCR Center for Computing Research

Sandia National Laboratories

Previous Predictions

CCR Center for Computing Research

Sandia National Laboratories

Challenges for High-Performance Networking for Exascale Computing

(Invited Paper)

Ron Brightwell, Brian Barrett, Scott Hemmert
Sandia National Laboratories
Albuquerque, NM, USA

Keith Underwood
Intel
Hillsborough, OR, USA

International Conference on Computer Communications and Networks

August 2, 2010

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Potential Exascale System Architecture Targets

System attributes	2010	"2015"		"2018"	
System peak	2 Peta	200 Petaflop/sec		1 Exaflop/sec	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32-64 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/sec	1 TB/sec	0.4 TB/sec	4 TB/sec
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	50,000	5,000	1,000,000	100,000
Total Node Interconnect BW	1.5 GB/s	20 GB/sec		200 GB/sec	
MTTI	days	O(1day)		O(1 day)	

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

MPI Will Likely Persist Into Exascale Era

- Number of network endpoints will increase significantly (5-50x)
- Memory and power will be dominant resources to manage
  - Networks must be power and energy efficient
  - Data movement must be carefully managed
  - Memory copies will be very expensive
- Impact of unexpected messages must be minimized
  - Eager protocol for short messages leads to receive-side buffering
  - Need strategies for managing host buffer resources
  - Flow control will be critical
  - N-to-1 communication patterns will (continue to be) disastrous
- Must preserve key network performance characteristics
  - Latency
  - Bandwidth
  - Message rate (throughput)

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Current Flow Control Strategies Not Sufficient

- Credit-based
  - Limit number of outstanding send operations
  - Used credits are replenished implicitly or explicitly
  - Effectiveness limited to N-to-1 scenario
  - Potential performance penalty for well-behaved applications
- Acknowledgment-based
  - Receiver explicitly confirms receipt of every message
  - Significant per-message performance penalty
    - Round trip acknowledgment doubles latency
  - Performance penalty for well-behaved applications
- Local copying (bounce buffer) mitigates latency penalty
- Both strategies limit message rate and effective bandwidth
- Flow control implemented at user-level inside MPI library
- Network transport usually has its own flow control mechanism
  - No mechanism for back pressure from host resources to network

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Applications Must Become More Asynchronous

- Applications cannot continue to be bulk synchronous
  - Overhead of synchronization will limit scaling
  - Synchronization increases susceptibility to noise
- Network API must provide asynchronous operations and progress
  - Data movement must be independent of host activity
- Active Messages
  - Polling is fundamental to all AM
  - Progress only when nothing else to do
  - Polling memory for message reception is inefficient
  - Needs hardware support to integrate message arrival with thread invocation
- Run-time systems will also need to communicate
  - Need to communicate evolving state of the system
  - Need a common portable API
  - Using TCP OOB connection will be infeasible

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Resiliency Will Impact Network API

- Network will need to expose errors to enable recovery
- Applications and system components will have different resiliency requirements
  - Reachability errors must be handled by run-time services
  - Graceful degradation may be appropriate for some applications
- May need OOB mechanism for recognizing network failures
  - AM or event-driven API would be ideal
  - Hardware support for network-level protection
    - RAS system invoking OS via network messages

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Network Interface Controller

- Power will be number one constraint for exascale systems
- Current systems waste energy
  - Using host cores to process messages is inefficient
  - Only move data when necessary
  - Move data to final destination
    - No intermediate copying due to network
- Specialized network hardware
  - Atomic operations
  - Match processing
- Addressing and address translation
  - Virtual address translation
    - Avoid registration cache
  - Logical node translation
    - Rank translation on a million nodes
- Hardware support for thread activation on message arrival

NNSA

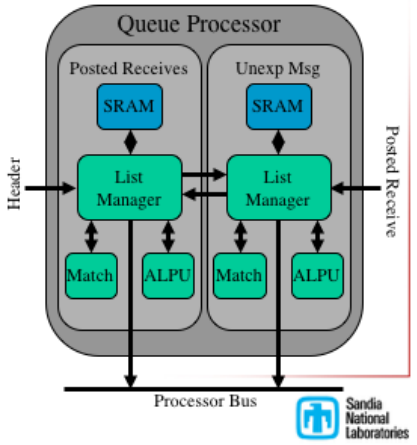
Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

High Message Throughput Challenges

- 20M messages per second implies a new message every 50ns
- Significant constraints created by MPI semantics
- On-load approach
  - Roadblocks
    - Host general purpose processors are inefficient for list management
    - Caching (a cache miss is 70-120ns latency)
      - Microbenchmarks are cache friendly, real life is not
  - Benefits
    - Easier & cheaper
- Off-load approach
  - Roadblocks
    - Storage requirements on NIC
    - NIC embedded processor is worse at list management (than the host processor)
  - Benefits
    - Opportunity to create dedicated hardware
    - Macroscale pipelining



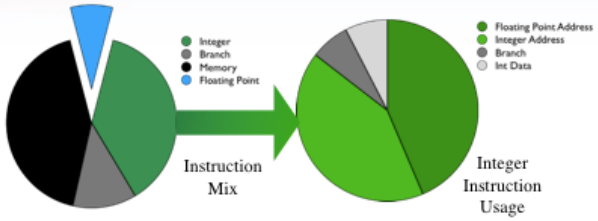
NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Minimizing Memory Bandwidth Usage in Network Stack



Memory Usage in Sandia Applications

- Memory bandwidth is most often the limiting factor for on node performance
- We must minimize the use of host memory bandwidth in the network stack

- Bounce buffers (or any other copying) incur a 2x memory bandwidth penalty
- A fast off-load approach can minimize host memory bandwidth utilization
  - Allows the NIC to determine where received messages need to be put in host memory and DMA the data directly there, eliminating the need for bounce buffers
  - High message rate can reduce the need for buffering of non-contiguous data

NNSA

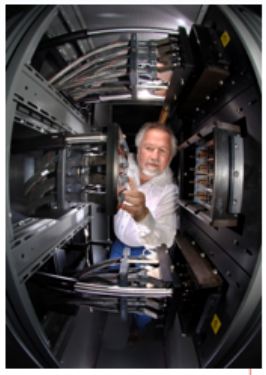
Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Topology

- No single network topology is best for all applications
- Meshes
  - Advantages: high local bandwidth, low wiring complexity, ability to easily add nodes (no distinct steps in expandability curve)
  - Disadvantages: high maximum latency, low global bandwidth
  - Works well for physical simulations which tend to talk nearest neighbor
- Trees
  - Advantages: high global bandwidth, low global latency
  - Disadvantages: high wiring complexity, lower local bandwidth, discrete steps in network topology (limiting expandability)
  - Works well for random accesses patterns and apps with lots of global communication



Red Storm
Red/Black Switching

- Hierarchical/Hybrid networks
  - Tend to inherent the strengths and weaknesses of the building blocks
- Network routers must be designed to allow different topologies with the same silicon

NNSA

Sandia National Laboratories

CCR Center for Computing Research

Sandia National Laboratories

Exascale Interconnects Hardware Trends

CCR Center for Computing Research

Many-Core Processors

Sandia National Laboratories

- Throughput-optimized core (TOC) versus latency-optimized core (LOC)
- TOC
  - Reduces single thread performance for network stack processing
    - May be good for progress thread for networks without hardware matching
  - Communication libraries are not currently highly threaded
    - Integrating threading and MPI
    - MPI Forum proposals – Endpoints and Finepoints
  - Overhead of MPI tag matching
    - Avoid tag matching with MPI RMA (or other one-sided alternatives)
  - Trend to offload more processing
    - Remote read/write capability reduces core interference but may increase memory interference
    - Increases potential for overlap of computation and communication
- Future processors likely a hybrid mixture of TOCs and LOCs

CCR Center for Computing Research

Many-Core Processors (cont'd)

Sandia National Laboratories

- Diminishing memory footprint per core
- Diminishing memory bandwidth per core
  - Increases importance of reducing intra-node memory-to-memory copies
  - Move away from single-copy models (MPI,RMA) to load/store (shared memory)
  - Motivates a hybrid programming model
    - Network potentially servicing many upper layer protocols
- Increasing number of network endpoints per node
  - Increasing numbers of threads and/or processes
  - Locality issues
    - Origin/destination of the data being read/written
  - More network contexts to avoid serialization of network requests
    - More network contexts may increase memory usage
    - Need to reduce amount of MPI state per context
  - Need more efficient ways of virtualizing network endpoints

CCR Center for Computing Research

Integrated Network Interfaces

Sandia National Laboratories

- Network interface is coherent with processor cores
- Lowers latency by avoiding I/O bus (PCI, QPI, etc.)
- Network atomic operations coherent with core atomic operations
- Lightweight mechanisms for thread sleep/wakeup on memory events
  - Better support for “active message” functionality

CCR Center for Computing Research

Heterogeneity

Sandia National Laboratories

- Cores – accelerators
  - Initiating communication from accelerators
  - Trend to on-package coherence for accelerators
- Memory – multilevel
  - Coherent transfers to/from accelerator memory
  - Impacts locality of network data structures and target memory spaces
  - Performance variability based on target/source memory space

CCR Center for Computing Research

Fabric

Sandia National Laboratories

- Network topologies
  - Increasing switch port counts
  - Lower diameter networks with near constant relative global bandwidth as node count increases
  - Reduced average latency across the network
  - Flexibility to trade off local versus global bandwidth
  - More dependence on adaptive routing techniques
  - Trend towards better congestion management protocols to reduce average latency
  - Ordering becomes an issue
    - Packets/messages between identical source and destination pair can take different paths
  - Inter-job interference will continue to be an issue for some topologies
  - Bandwidth tapering strategies that offer more local bandwidth
- Offloading collective operations to switches
    - Should lower latencies for certain collectives
- Quality of service
  - More service levels to give performance guarantees to different types of traffic

CCR Center for Computing Research

Cross-Cutting

Sandia National Laboratories

- Non-volatile memory
  - Trend towards network attached storage (burst buffer, memory pools)
- Power/energy
  - Shutting down idle links
- Protection
  - Increasing support for inter-application communication
- Resilience
  - Multiple paths with adaptive routing can improve reliability

CCR Center for Computing Research

Convergence of NOC with NIC

Sandia National Laboratories

- Some say this means off-node will look a lot like on-node
  - Latency hiding will support global load/store
- On-node may start to look more like off-node
  - Easier to make off node capabilities work on node than vice versa
  - DMA engines for efficient on-node transfers
  - Matching hardware
  - Special-purpose cores for on-node data transformation
- Load/store model doesn't work on node
  - Locality
  - Performance portability
  - Resilience
  - Scalability

CCR Center for Computing Research

Future Challenges and Potential Solutions

Sandia National Laboratories

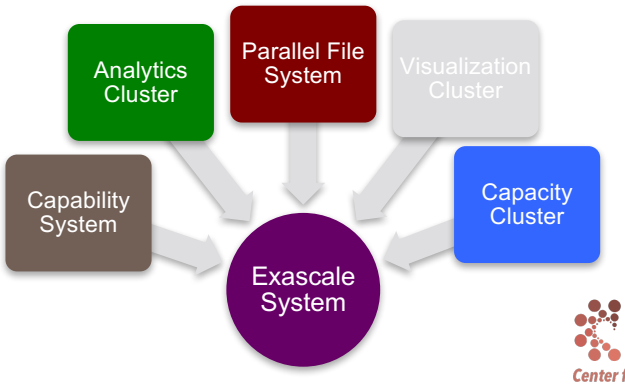
- Most pressing interconnect challenges
  - Parallelism in the network interface
  - Resiliency
  - Flow control
  - Active message semantics
  - New types of communication – runtime, workflows, analytics
  - Event-driven capability – for resilience and task-based programming models
- What hardware mechanisms can help?
  - More network contexts
  - Virtual addressing of endpoints
  - Flow control protocols that understand endpoint resources
  - QoS mechanisms for isolation
- What else can help?
  - Instrumentation to help understand application resource usage
  - Benchmarks that target specific desired capabilities

CCR Center for Computing Research

Systems Are Converging to Reduce Data Movement

Sandia National Laboratories

- External parallel file system is being subsumed
  - Near-term capability systems using NVRAM-based burst buffer
  - Future extreme-scale systems will continue to exploit persistent memory technologies
- In-situ and in-transit approaches for visualization and analysis
  - Can't afford to move data to separate systems for processing
  - GPUs and many-core processors are ideal for visualization and some analysis functions
- Less differentiation between advanced technology and commodity technology systems
  - On-chip integration of processing, memory, and network
  - Summit/Sierra using InfiniBand



CCR Center for Computing Research

Applications Workflows are Evolving

Sandia National Laboratories

- More compositional approach, where overall application is a composition of coupled simulation, analysis, and tool components
- Each component may have different OS and Runtime (OS/R) requirements, in general there is no “one-size-fits-all” solution
- Co-locating application components can be used to reduce data movement, but may introduce cross component performance interference
  - Need system software infrastructure for application composition
  - Need to maintain performance isolation
  - Need to provide cross-component data sharing capabilities
  - Need to fit into vendor’s production system software stack
- Remote message queue abstraction interfaces
- Interfaces for accessing remote persistent memory

CCR Center for Computing Research

Kathy Yelick – ASCAC Talk – What's Missing?

Sandia National Laboratories

Lowering Overhead for Smaller Messages

Send/Receive

two-sided message

message id | data payload

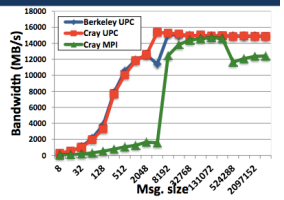
one-sided put message

address | data payload

network interface

cores

memory



The + in MPI+X

MPI+X today:

- **Communicate on one lightweight core**
- **Reverse offload to heavyweight core**

Want to allow all cores to communicate (but keep the protocol simple!)

Lightweight communication is more important with lightweight cores

Programing Models and Environments 23

Lightweight Communication for Lightweight Cores

- **DMA (Put/Get)**
  - Blocking and non-blocking (completion signaled on initiator)
  - Single word or Bulk
  - Strided (multi-dimensional), Index (sparse matrix)
- **Signaling Store**
  - All of the above, but with completion on receiver
  - What type of "signal"?
    - Set a bit (index into fixed set of bits ☹)
    - Set a bit (second address sent 😐)
    - Increment a counter (index into fixed set of counters ☹)
    - Increment a counter (second address for counter 😐)
    - Universal primitives: compare-and-swap (2nd address + value), fetch-and-add handy but not sufficient for multi/reader-writers ☺
- **Remote atomic (see above) – should allow for remote enqueue**
- **Remote invocation**
  - Requires resources to run: use dedicated set of threads?

DEGAS Overview 24

CCR Center for Computing Research

Active Messages

Sandia National Laboratories

- See Paul Hargrove's list of issues
  - https://sites.google.com/a/lbl.gov/gasnet-ex-collaboration/active-messages
- Handlers
  - Who allocates the resources?
  - What can be called?
  - Where do they run (context)?
  - When do they run relative to message delivery?
  - How long do they run?
  - Why?
    - One-sided messages decouple processor from network
    - Active messages tightly couple processor and network
      - Active messages aren't one-sided
      - Memory is the endpoint, not the cores
    - Lightweight mechanism for sleeping/waking thread on memory update
      - Why go through the network API for this?
- Scheduling lots of unexpected thread invocations leads to flow control issues

CCR Center for Computing Research

Sandia National Laboratories

Portable Programming for Network Offload

CCR Center for Computing Research

Motivation

Sandia National Laboratories

- Slower, more energy-efficient processors
- Remove the processor from the packet processing path (offload)
- RDMA only supports data transfer between virtual memory spaces
- Varying levels of steering the data at the endpoint
- Terabit per second networks are coming
- Modern processors require 15-20 ns to access L3 cache
  - Haswell – 34 cycles
  - Skylake - 44 cycles
- 400 Gib/s can deliver a 64-byte message every 1.2 ns
- Data is placed blindly into memory

CCR Center for Computing Research

Stream Processing in the Network (sPIN)

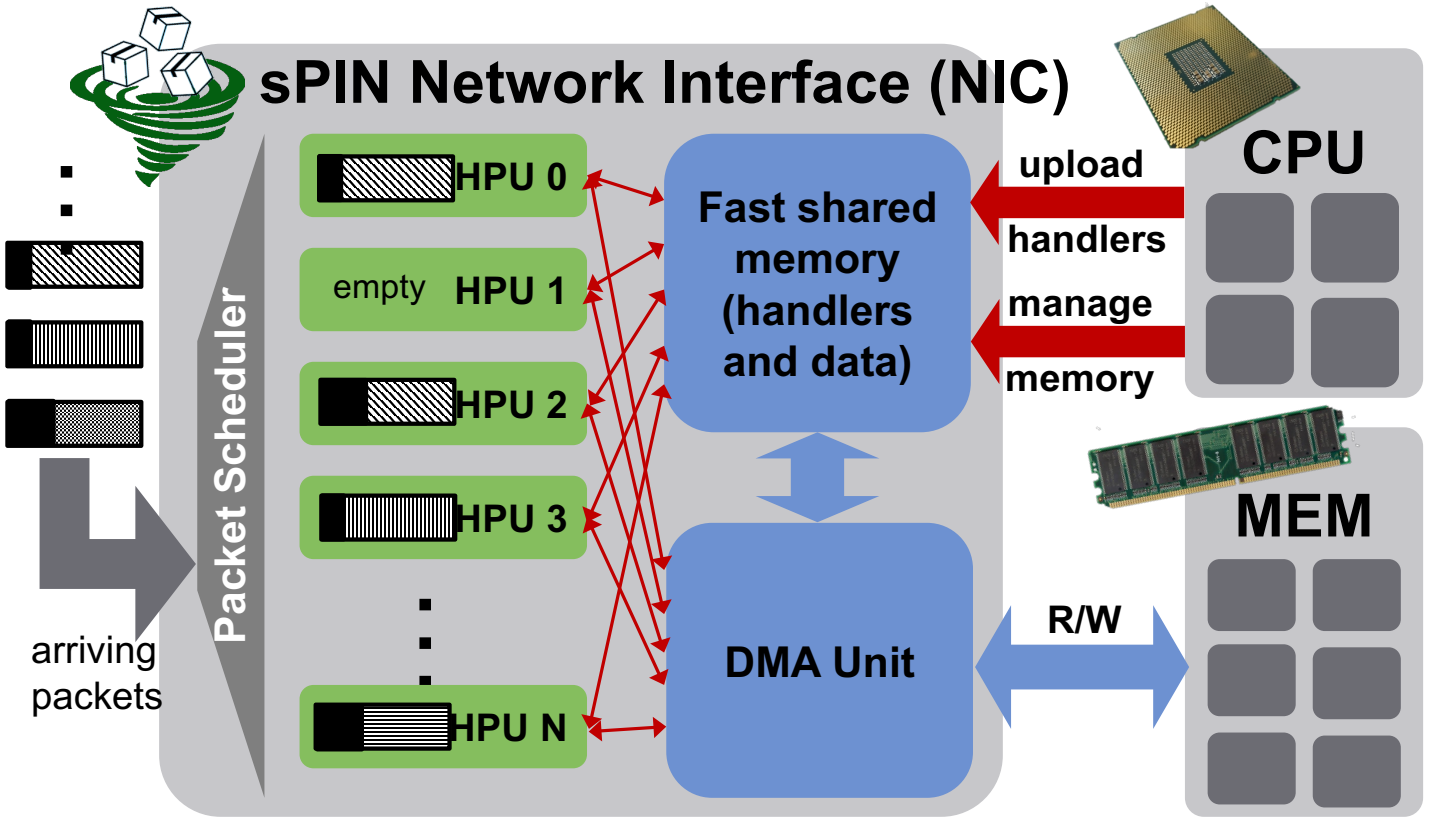
Sandia National Laboratories

- Augment RDMA and matching tasks with simple processing tasks
- Programming interface for specifying packet processing kernels
  - Similar to the way CUDA and OpenCL work for compute accelerators
- Tasks that can be performed on incoming packets
  - Limited state
  - Small fast memory on NIC
  - Execute short user-defined functions

- Handlers are compiled for the NIC and passed down at initialization time
- Vendor-independent interface would enable strong collaborative open-source environment similar to MPI

CCR Center for Computing Research

sPIN Architecture Overview



sPIN is not Active Messages (AM)

Sandia National Laboratories

- Tightly integrated NIC packet processing
- AMs are invoked on full messages
  - sPIN works on packets
  - Allows for pipelining packet processing
- AM uses host memory for buffering messages
  - sPIN stores packets in fast buffer memory on the NIC
  - Accesses to host memory are allowed but should be minimized
- AM messages are atomic
  - sPIN packets can be processed atomically

CCR Center for Computing Research

sPIN Approach

Sandia National Laboratories

- Handlers are executed on NIC Handler Processing Units (HPUs)
- Simple runtime manages HPUs
- Each handler owns shared memory that is persistent for the lifetime of a message
  - Handlers can use this memory to keep state and communicate
- NIC identifies all packets belonging to the same message
- Three handler types
  - Header handler – first packet in a message
  - Payload handler – all subsequent packets
  - Completion handler – after all payload handlers complete
- HPU memory is managed by the host OS
- Host compiles and offloads handler code to the HPU
- Handler code is only a few hundred instructions

CCR Center for Computing Research

sPIN Approach (cont'd)

Sandia National Laboratories

- Handlers are written in standard C/C++ code
- No system calls or complex libraries
- Handlers are compiled to the specific Network ISA
- Handler resources are accounted for on a per-application basis
  - Handlers that run too long may stall NIC or drop packets
- Programmers need to ensure handlers run at line rate
- Handlers can start executing within a cycle after packet arrival
  - Assuming an HPU is available
- Handlers execute in a sandbox relative to host memory
  - They can only access application's virtual address space
  - Access to host memory is via DMA

CCR Center for Computing Research

HPU Messages

Sandia National Laboratories

- Messages originating from HPU memory
  - Single packet, MTU sized
  - May block the HPU thread
- Messages originating from HPU memory
  - Enter the normal send queue as if from the host
  - Non-blocking

CCR Center for Computing Research

Portals Interconnect Programming Interface

Sandia National Laboratories

- Developed by Sandia, U. New Mexico, Intel
- Previous generations of Portals deployed on several production massively parallel systems
  - 1993: 1800-node Intel Paragon (SUNMOS)
  - 1997: 10,000-node Intel ASCI Red (Puma/Cougar)
  - 1999: 1800-node Cplant cluster (Linux)
  - 2005: 10,000-node Cray Sandia Red Storm (Catamount)
  - 2009: 18,688-node Cray XT5 – ORNL Jaguar (Linux)
- Focused on providing
  - Lightweight "connectionless" model for massively parallel systems
  - Low latency, high bandwidth
  - Independent progress
  - Overlap of computation and communication
  - Scalable buffering semantics
  - Protocol building blocks to support higher-level application protocols and libraries and system services
- At least three hardware implementations currently under development
- Portals influence can be seen in IB Verbs (Mellanox) and libfabric (Intel & others)

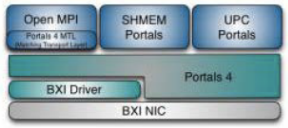
CCR Center for Computing Research

Portals Hardware and Co-Design Activities

Sandia National Laboratories

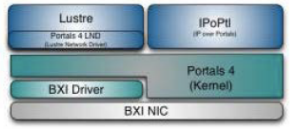
BXI application environment

BXI comes with a complete software stack to provide optimal performance and reliability to all traditional HPC components.



BXI Computing stack

- Parallel applications can take full advantage of the capabilities of the BXI network using MPI, SHMEM or UPC communication libraries.
- All components are implemented directly using the Portals 4 API.
- Kernel services are also implemented using the kernel Portals 4 implementation.
- A Portals 4 LND (Lustre Network Driver) provides the Lustre parallel filesystem with a direct / native access to Portals 4.



BXI Kernel services

- The IPoPtl (IP over Portals) component makes it possible to have large scale, efficient and robust IP communication for legacy software.

Please contact hpc@atos.net

gists. **Powering progress**

Atos

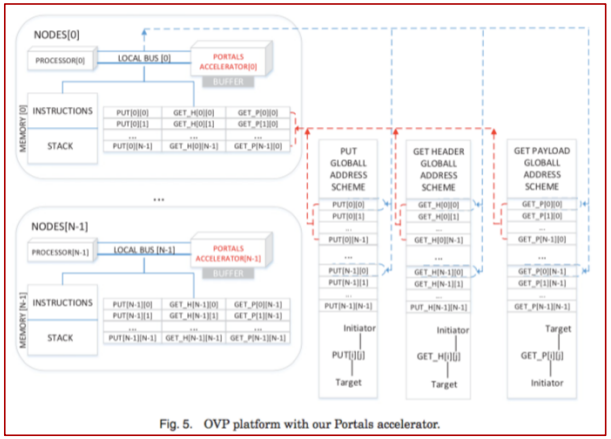
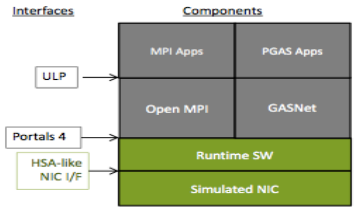


Fig. 5. OVP platform with our Portals accelerator.

FASTFORWARD NIC SOFTWARE STACK

AMD

- Portals 4 API chosen for initial investigation
  - Supports multiple programming models: PGAS, MPI
- Implemented in thin software layer over hardware interface
- Leverage existing ULPs that have Portals 4 implementations
  - GASNet
  - Open MPI



EXPERIMENTAL FRAMEWORK

RESULTS

AMD

- All data collected in gem5[6]
  - System call emulation mode (no OS)
  - AMD GPU model[7]
  - Full Support for HSA
  - Tightly coupled system
- Portals 4-based NIC model[8]
  - Low-level RDMA network programming API currently supported by:
    - MPICH, Open MPI, GASNet, Berkeley UPC, GNU UPC, and others
  - XTQ implemented as an extension of the Portals 4 remote Put operation

CPU and Memory Configuration	
CPU Type	8-wide OOO, 4Ghz, 8 cores
I,D-Cache	64K, 2-way, 1 cycle
L2-Cache	2MB, 8-way, 4 cycles
L3-Cache	16MB, 16-way, 20 cycles
DRAM	DDR3, 4 Channels, 800MHz
GPU Configuration	
GPU Type	1 Ghz, 24 Compute Units
D-Cache	16kB, 64B line, 16-way, 4 cycles
I-Cache	32kB, 64B line, 8-way, 4 cycles
L2-Cache	768kB, 64B line, 16-way, 24 cycles
NIC Configuration	
Link Speed	100ns/ 100Gbps
Network API	Portals 4
Topology	Star

19 | EXTENDED TASK QUEUING: ACTIVE MESSAGES FOR HETEROGENEOUS SYSTEMS | AUGUST 10, 2017

[6] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," SIGARCH Comput. Archit. News, pp. 1–7, 2011.
[7] AMD. (2015) The AMD gem5 APU simulator: Modeling heterogeneous systems in gem5. http://gem5.org/GPU_Models.
[8] Sandia National Laboratories, "The Portals 4.0.2 network programming interface," http://www.cs.sandia.gov/Portals/portals402.pdf, 2014.

CCR Center for Computing Research

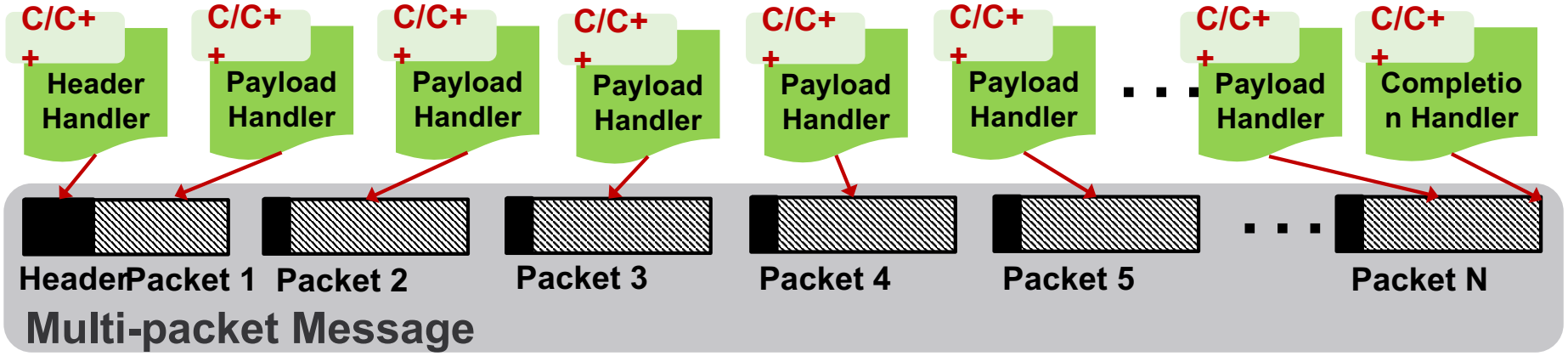
sPIN Interface for Portals4 (P4sPIN)

Sandia National Laboratories

- All packet handlers are associated with a Match Entry (ME)
- Extend PtlMEAppend() to enable handler registration
- HPU memory is allocated using PtlHPUAllocMem()
  - Allows user the use same HPU memory for multiple MEs
- If no HPU execution contexts are available NIC may trigger flow control
  - Similar to running out of host resources
  - Completion handler is invoked and notified via flow control flag

CCR Center for Computing Research

sPIN Message Handler Overview



Header Handler

Sandia National Laboratories

- Called once per message
- No other handler is started until it is completed
- Access only to header fields, potentially user-defined header info
- Pre-defined headers could be in special registers
- User-defined headers in HPU memory

CCR Center for Computing Research

Payload Handler

Sandia National Laboratories

- Called after the header handler completes
- Payload does not include user-header
- Multiple instances execute concurrently
- Share all HPC memory coherently
- PTL_NUM_HPUS is the maximum number that can be active
- Handlers do not migrate
- PTL_MY_HPU to emulate HPU-private data

CCR Center for Computing Research

Completion Handler

Sandia National Laboratories

- Called once per message after all payload handers have completed
- Before completion event is delivered to host memory
- Passed in the number of dropped bytes
- Flag indicating whether flow control was initiated

CCR Center for Computing Research

HPU Design

Sandia National Laboratories

- HPU should have single-cycle access to local memory and packet buffers
- HPU memory is not cached
- HPU instructions should be executed in a single cycle
  - Documentation needs to include instruction costs
- Handlers should be invoked immediately after packet arrival or after previous handler completes
- Handlers require no initialization, loading, or other boot activities
  - Context is pre-loaded and pre-initialized
- HPUs can be implemented using massive multithreading
  - Pre-emptible to deal with high latency DMAs
- Need enough HPU cores to process at full bandwidth

CCR Center for Computing Research

HPU Memory Size

Sandia National Laboratories

- Use Little's Law
- Handler executes between 10 and 500 instructions at 2.5 GHz and IPC=1
- Minimum delay of 200 ns per packet
- For 1 Tb/s
  - 1 Tb/s x 200 ns = 25 kB
- More space can be added to hide more latency, likely up to several microseconds

CCR Center for Computing Research

Simulation Environment

Sandia National Laboratories

- LogGOPSim
  - Network simulator
  - Drives the simulation by running trace-based discrete-event loop
  - Traced all Portals4 and MPI functions and invocations of handlers
  - Invokes gem5 for each handler execution and measures execution time
  - Communicates with gem5 through special memory mapped region through which an executing handler can issues simcalls from gem5 to LogGOPSim
- gem5
  - System-level and processor microarchitecture simulator
  - Simulates various CPU and HPU configurations

CCR Center for Computing Research

Simulation Settings

Sandia National Laboratories

- Network uses LogGP model
  - o = 65 ns (injection overhead)
  - 150 million msgs/sec
  - g = 6.7 ns (inter-message gap)
  - 400 Gib/s
  - G = 2.5ps (inter-byte gap)
  - 50 ns switch hop latency
  - 33.4 ns delay, wire length of 10m
- NIC configuration
  - 4 2.5 GHz ARM Cortex A15 out-of-order HPU cores using ARMv8-A 32-bit ISA
  - Cores are configured without cache using gem5's SimpleMemory module configured as a scratchpad that can be accessed in k cycles (k = 1)
  - Matching header packet takes 30 ns
  - Each packet takes 2 ns for CAM lookup
  - Network gap (g) can proceed in parallel
- Host CPU configuration
  - 8 2.5 GHz Intel Haswell cores
  - 8 MIB cache
  - 51 ns latency
  - 150 GiB/s bw

CCR Center for Computing Research

DMA and Memory Contention

Sandia National Laboratories

- HPU access to host memory via DMA
- Extended simulation by adding support to model contention for host memory
- DMA at each host also uses LogGP model
  - o = 0, g = 0 since these are already captured by gem5
  - Discrete NIC
    - L = 250 ns
    - G = 15.6 ps (64 Gib/s)
  - Integrated NIC
    - L = 50 ns
    - G = 6.7 ps (150 Gib/s)
  - DMA time is added to message transmission when NIC delivers data to memory

CCR Center for Computing Research

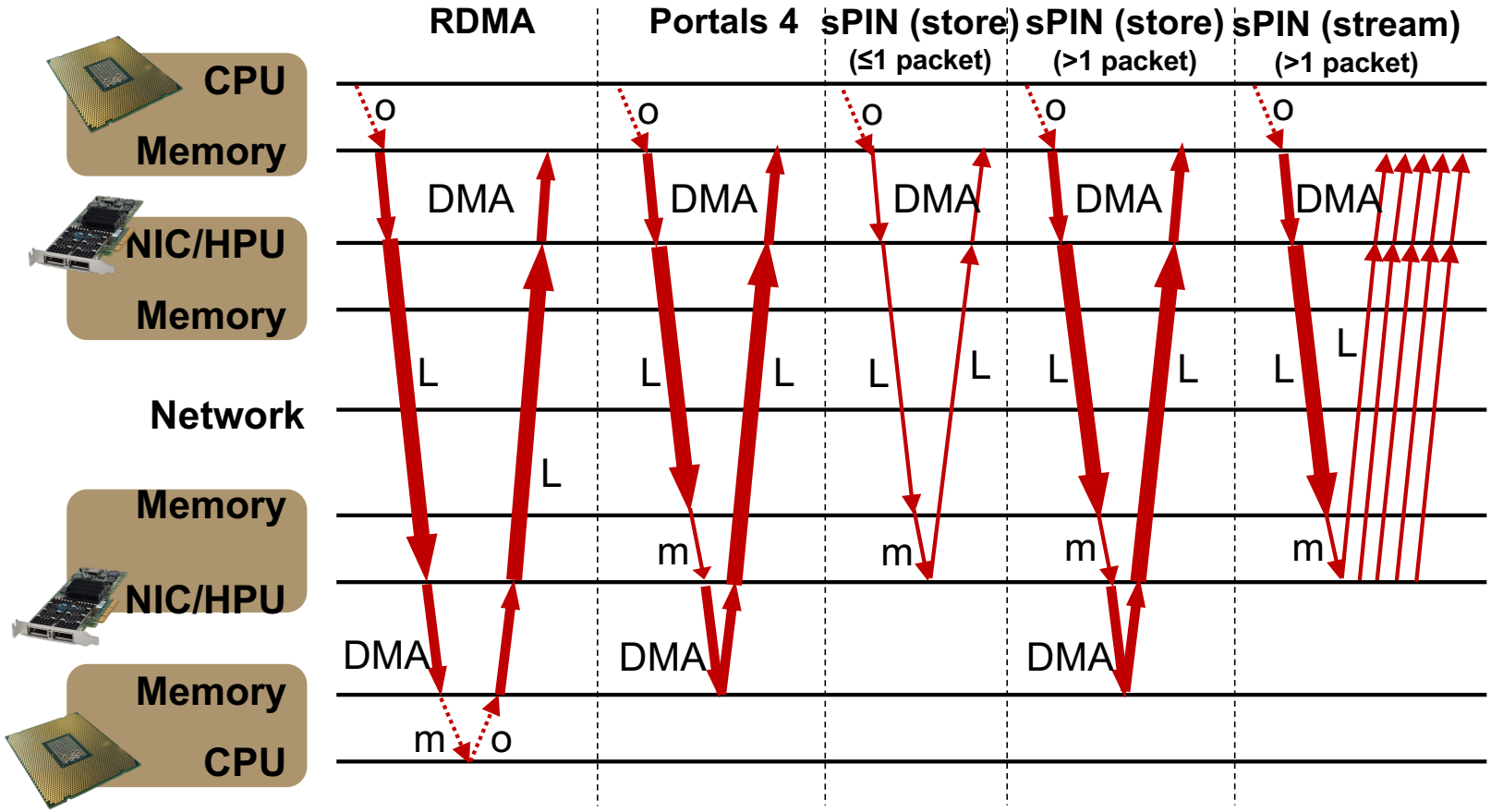
Micro-Benchmarks

Sandia National Laboratories

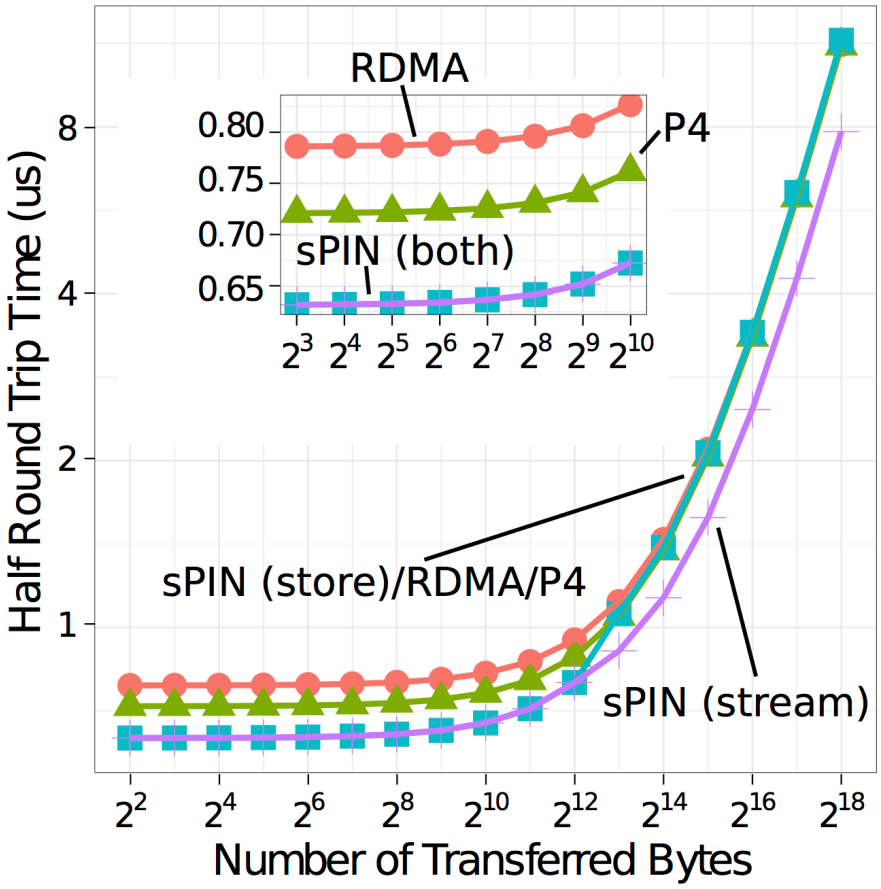
- Ping-pong latency
  - For RDMA, pong is sent by host CPU core
  - For sPIN, pong is pre-setup by destination and reply is triggered
  - Multiple options for generating the pong
    - Ping message is a single packet and pong can be issued with put from HPU memory
    - Ping message is larger than a packet and can be issued with put from host memory
    - Payload handler could generate pong message from HPU memory for each packet
- Accumulate
  - Array of double complex numbers to be multiplied with destination buffer
  - For RDMA, data is delivered to temporary buffer and accumulated into destination
  - For sPIN, packet handlers fetch data from host memory, apply operation, and write it back
  - Results are for coherent memory, which is worse than non-coherent memory

CCR Center for Computing Research

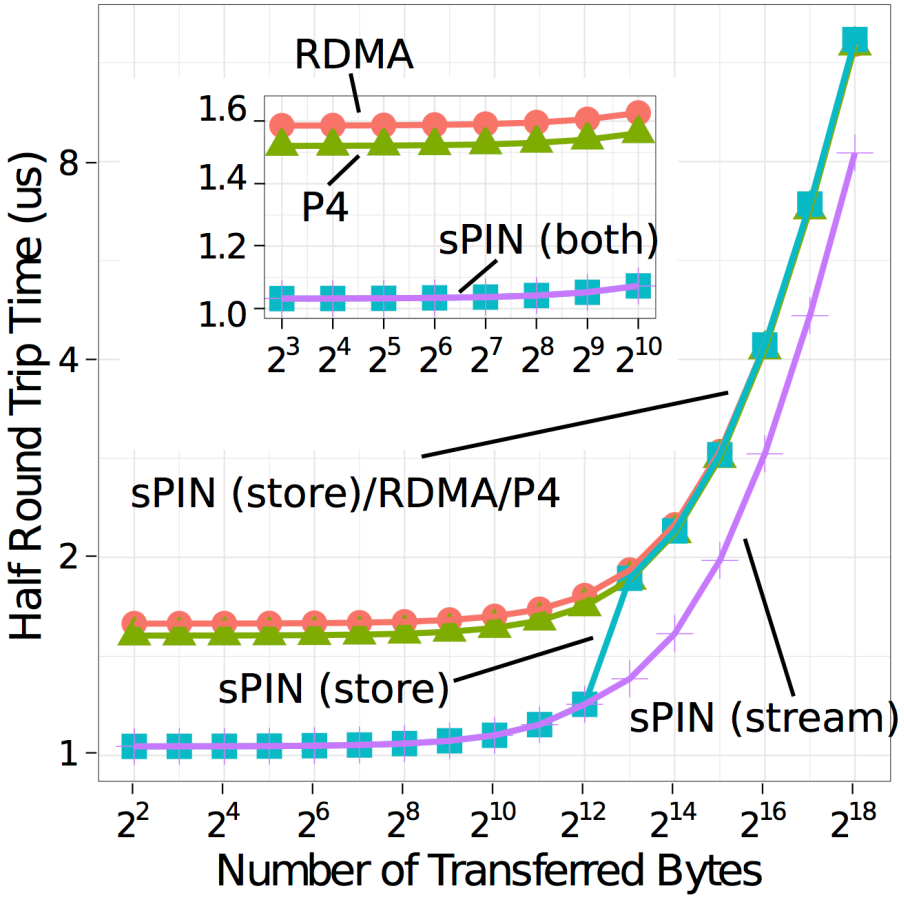
Breakdown of Ping-Pong



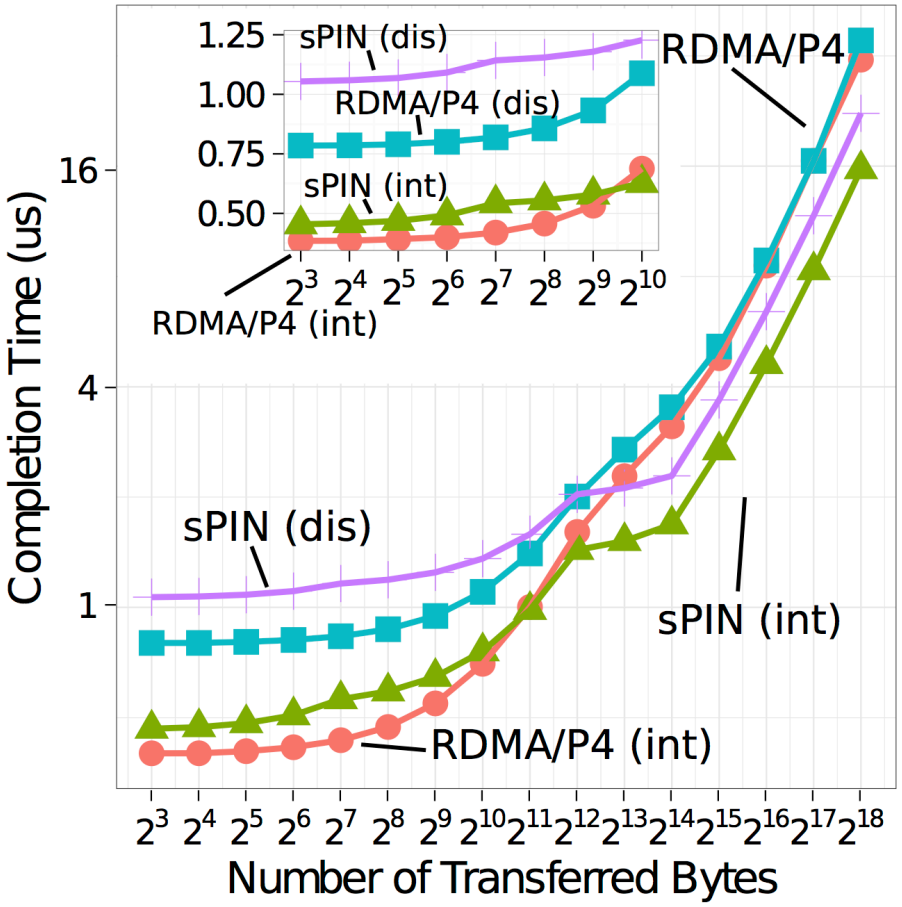
Ping-Pong (Integrated NIC)



Ping-Pong (Discrete NIC)



Accumulate (Both NIC Types)



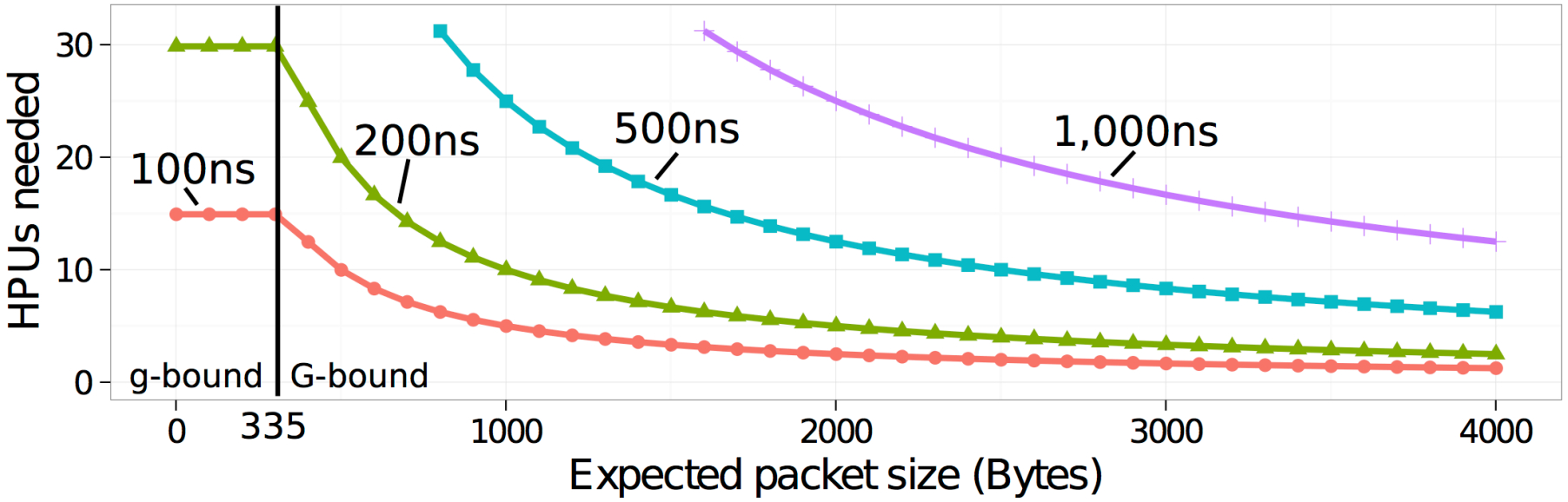
How Many HPUs Are Needed?

Sandia National Laboratories

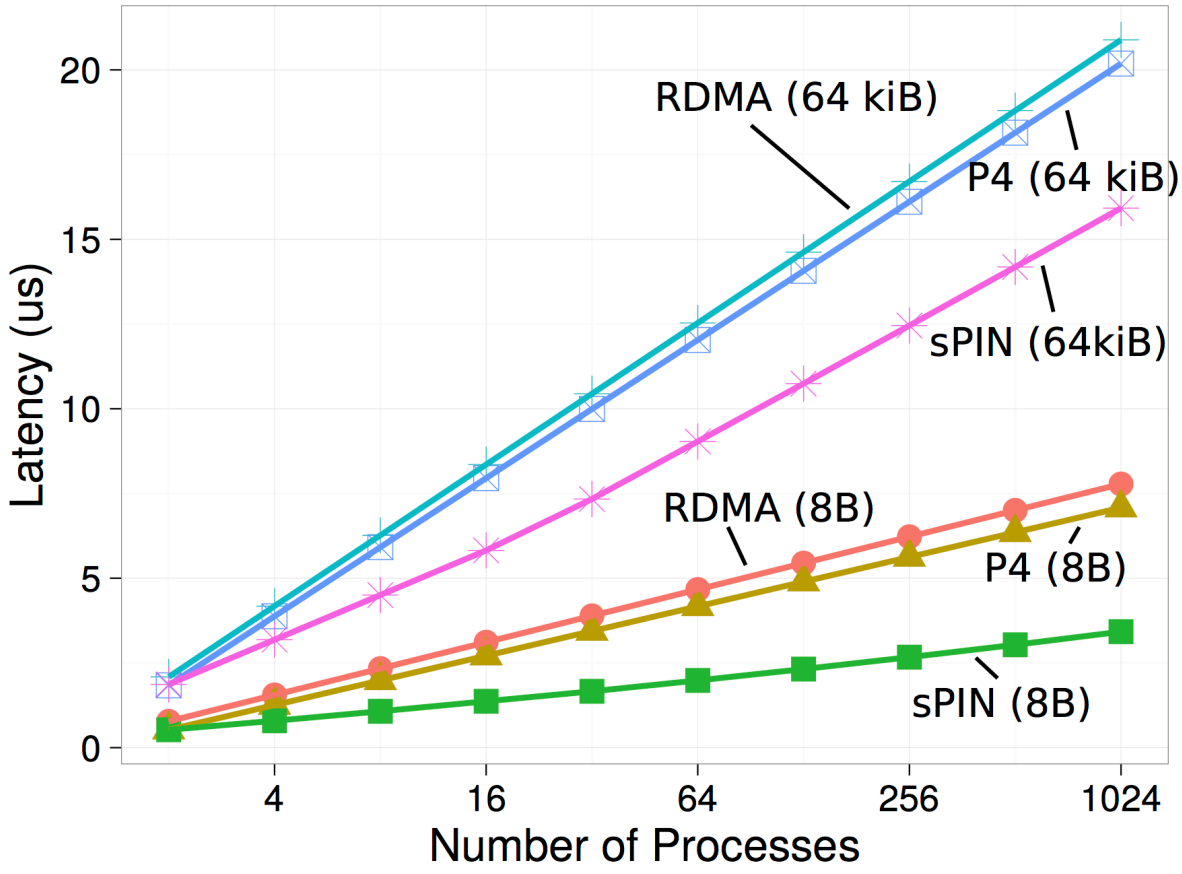
- Little's Law again
- Average execution time per packet of T
- Expected arrival rate of A
- Need T x A HPUs
- Fixed bandwidth (1/G)
- Packet size s
- Gap g
- A = min(1/g, 1/(G x s)
- For 8 HPUs supports any packet size if handler takes less than 53 ns
- For 4 KiB packets, handlers must execute in 650 ns

CCR Center for Computing Research

Number of HPUs Needed

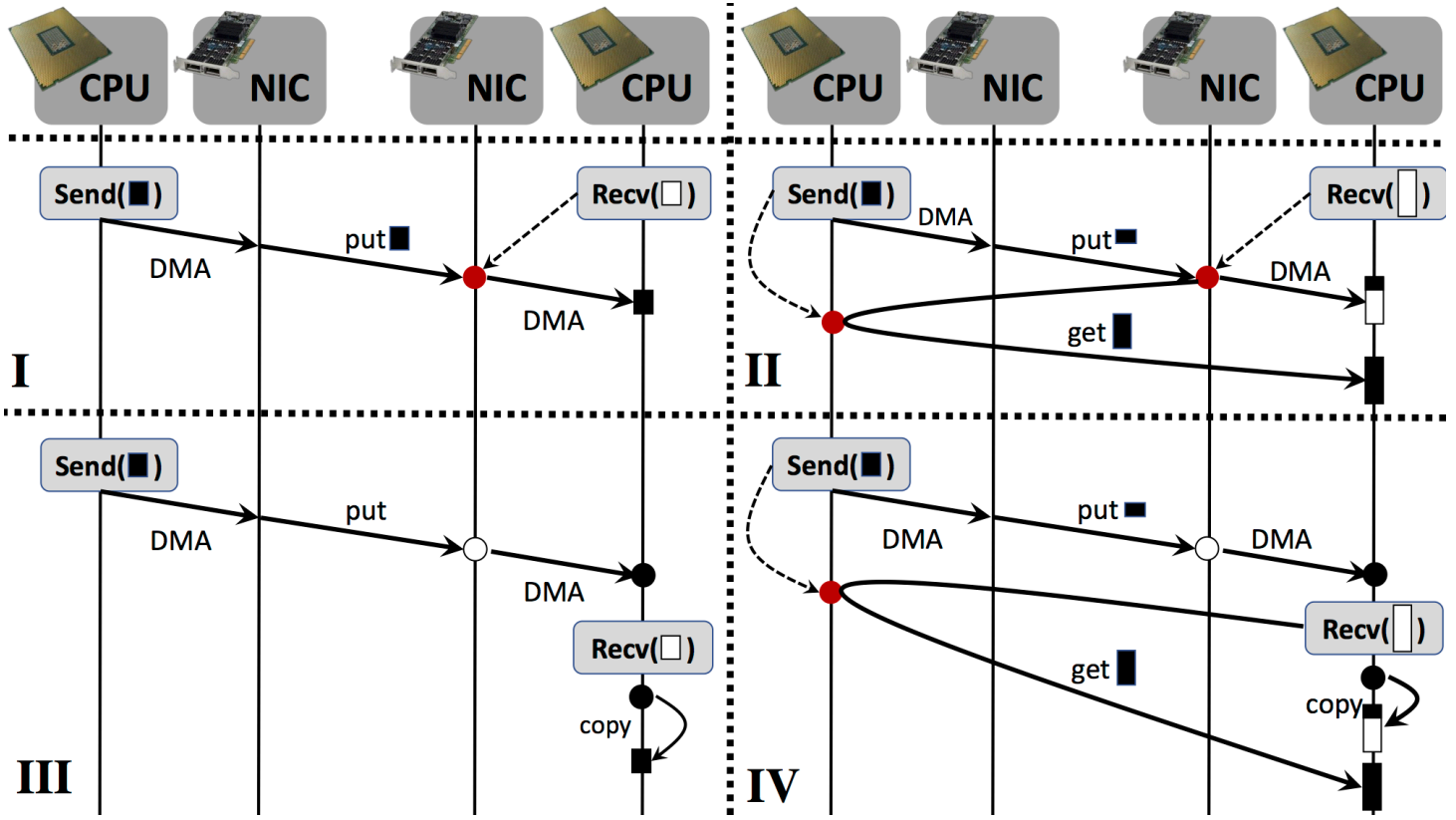


Broadcast on a Binomial Tree (Discrete NIC)



Matching Protocols

I Short/Exp
II Long/Exp
III Short/Un
IV Long/UN



Application Performance

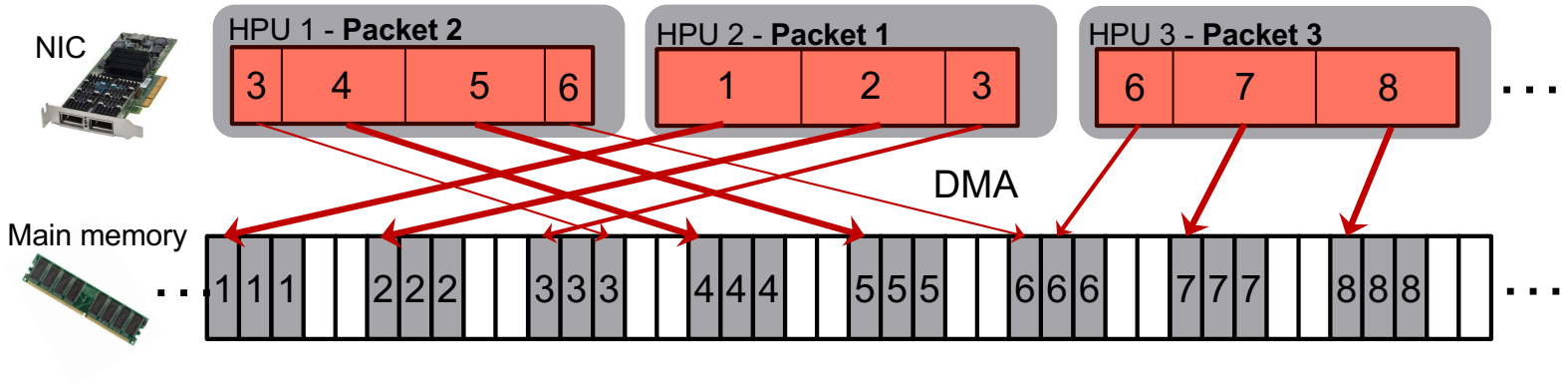
Sandia National Laboratories

program	p	msgs	ovhd	spdup
MILC	64	5.7M	5.5%	3.6%
POP	64	772M	3.1%	0.7%
coMD	72	5.3M	6.1%	3.7%
coMD	360	28.1M	6.5%	3.8%
Cloverleaf	72	2.7M	5.2%	2.8%
Cloverleaf	360	15.3M	5.6%	2.4%

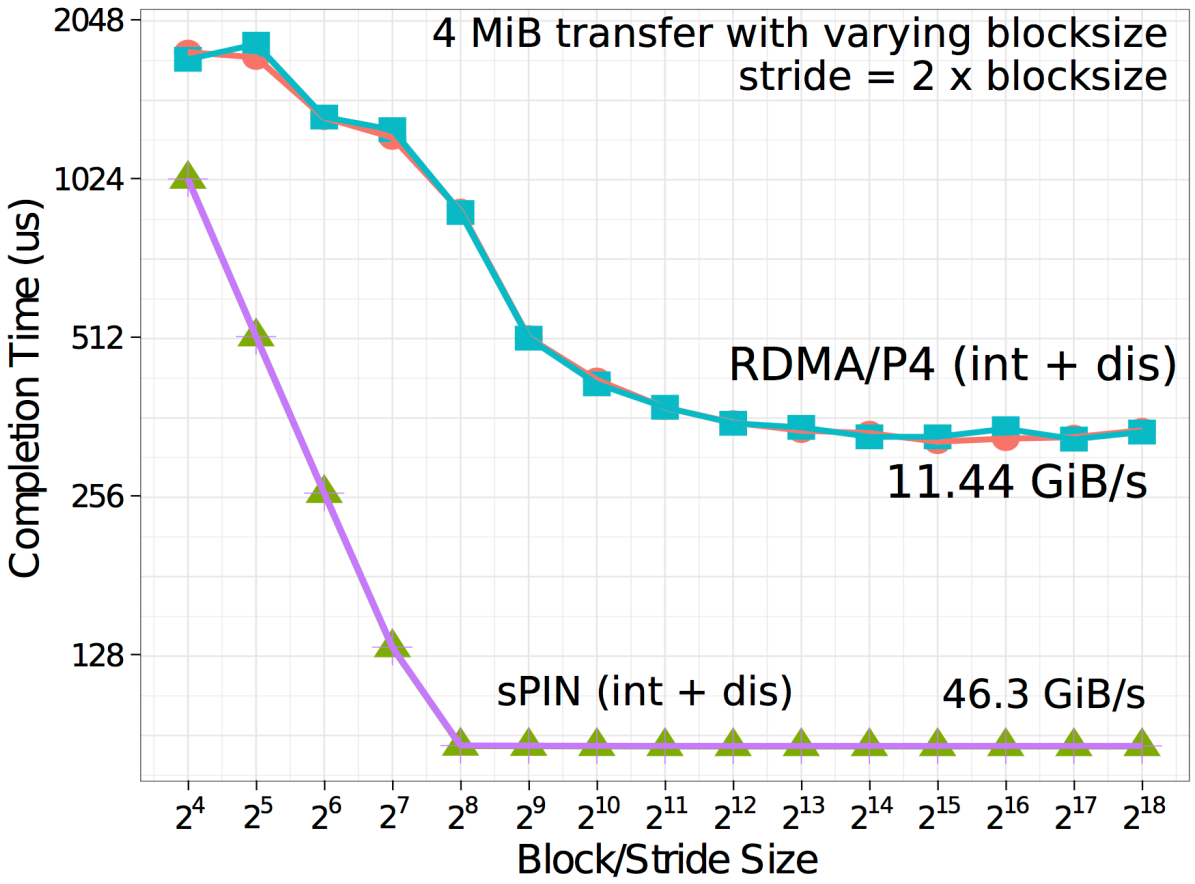
CCR Center for Computing Research

Processing Vector Datatypes in Payload Handlers

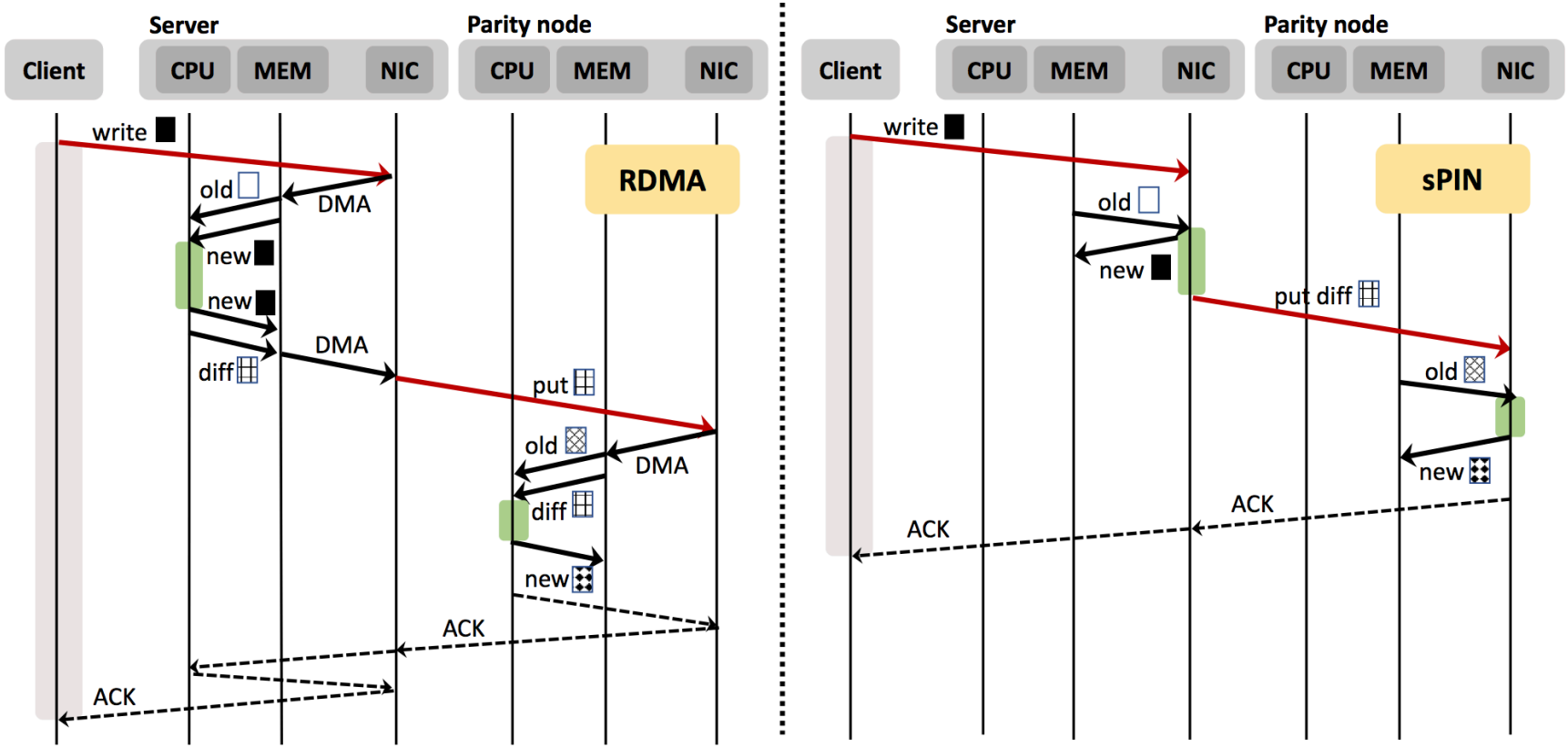
- Stride = 2.5 KiB
- Blocksize = 1.5 KiB
- Count = 8



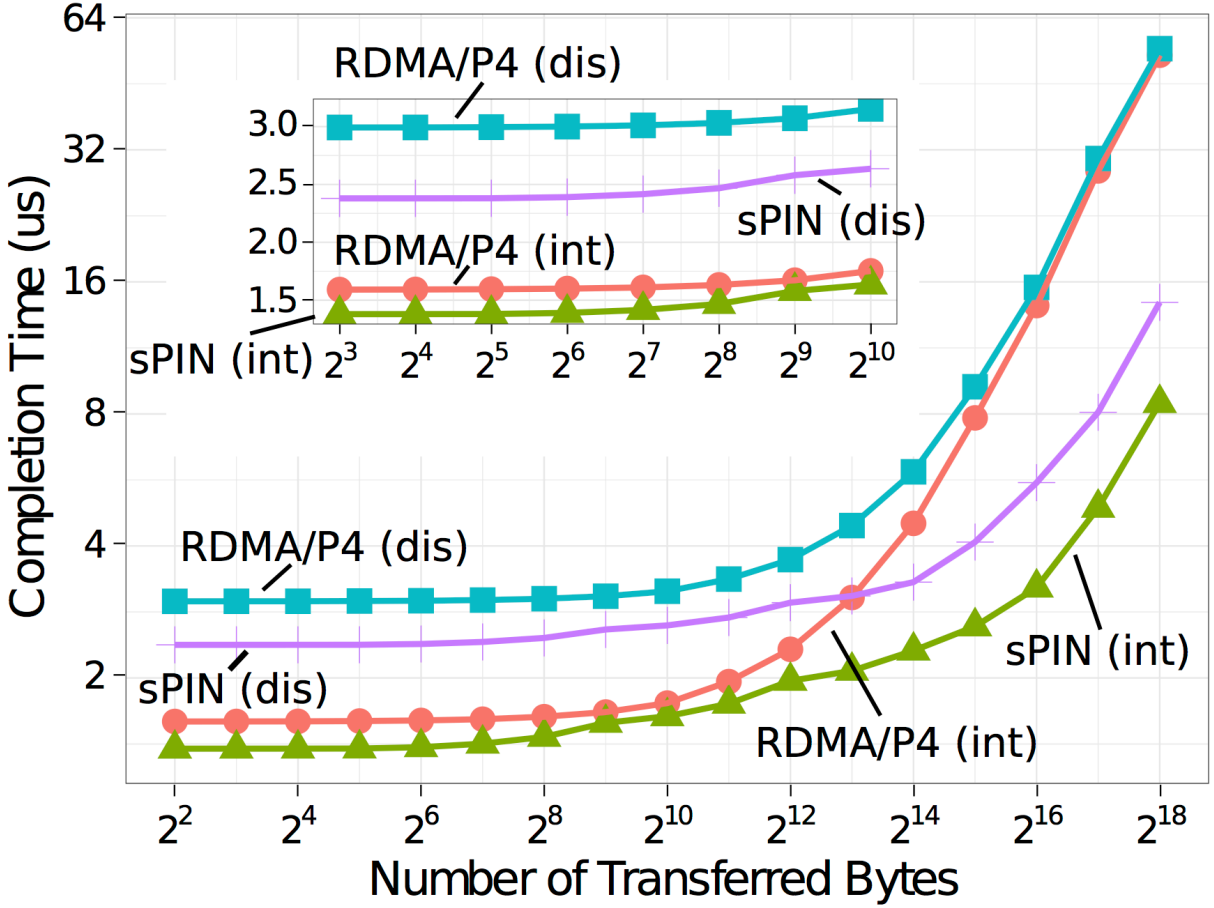
Strided Receive with Varying Blocksize



Distributed RAID Using RDMA and sPIN



Update Time in a Distributed RAID-5 System



Acknowledgments

- Sandia
  - Ryan Grant
  - Scott Hemmert
  - Kevin Pedretti
  - Mike Levenhagen
- ETH
  - Torsten Hoefler
  - Salvatore Di Girolamo
  - Konstantin Taranov
- LBL
  - Paul Hargrove