T. HOEFLER
WITH S. DI GIROLAMO, K. TARANOV, D. DE SENSI, L. BENINI, R. E. GRANT, R. BRIGHTWELL, A. KURTH, M. SCHAFFNER, T. SCHNEIDER, J. BERÁNEK, M. BESTA, L. BENINI, D. ROWETH

General in-network processing – time is ripe!

Keynote at the High-Performance Interconnects Forum with HPC China 2020

The Development of High-Performance Networking Interfaces

Scalable Coherent Interface | Myrinet GM+MX | Virtual Interface Architecture | OFED | libfabric

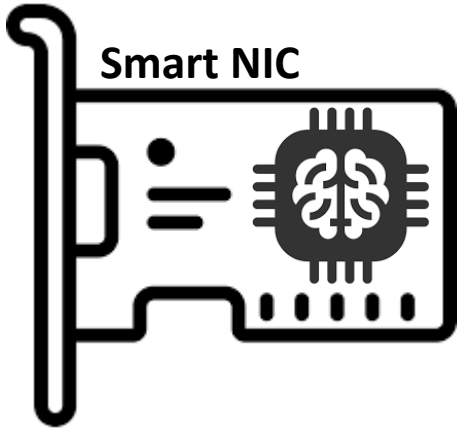
Ethernet+TCP/IP | Fast Messages | Quadrics QsNet | Cray Gemini | IB Verbs | Portals 4

Quadrics
A Finmeccanica Company

INFINIBAND TRADE ASSOCIATION

OPENFABRICS ALLIANCE

portals

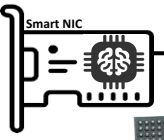


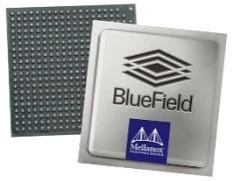
1980 | 1990 | 2000 | 2010 | 2020

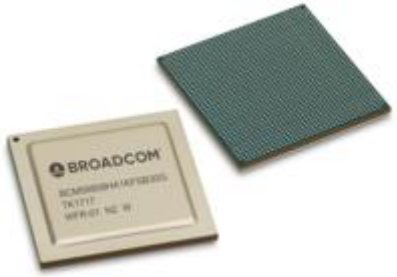
sockets | (active) message based | protocol offload | remote direct memory access (RDMA)

coherent memory access | OS bypass | zero copy | triggered operations

?







ARM cores
(with full OS, outside packet pipe)

Flow Processors
(limited flexibility, P4)

NETRONOME

BAREFOOT NETWORKS | an Intel company

Microsoft Catapult

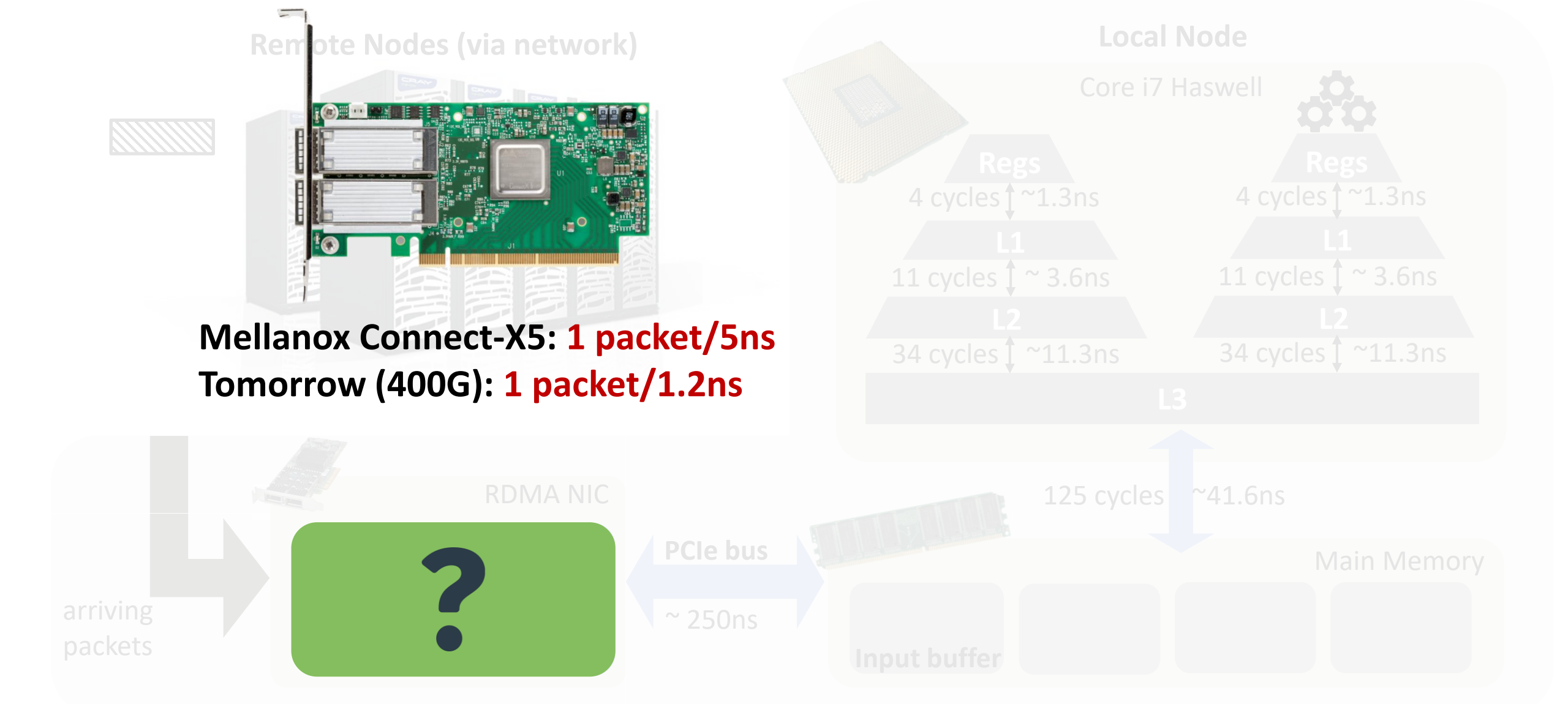
XILINX

intel

FPGAs
(limited productivity, silicon efficiency)

?

Data Processing in modern RDMA networks



Mellanox Connect-X5: 1 packet/5ns
Tomorrow (400G): 1 packet/1.2ns

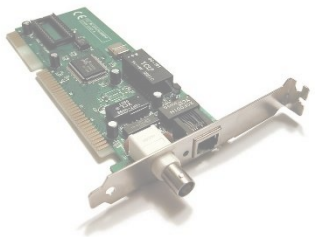
TH et al.: "sPIN: High-performance streaming Processing in the Network", SC17 best paper candidate

The future of High-Performance Networking Interfaces

sPIN
Streaming Processing
In the Network

Scalable Coherent Interface | Myrinet GM+MX | Virtual Interface Architecture | OFED | libfabric

Ethernet+TCP/IP | Fast Messages | Quadrics QsNet | Cray Gemini | IB Verbs | Portals 4



Quadrics
A Finmeccanica Company

INFINIBAND TRADE ASSOCIATION

OPENFABRICS ALLIANCE

portals

1980 | 1990 | 2000 | 2010 | 2020

sockets | (active) message based | protocol offload | remote direct memory access (RDMA)

coherent memory access | OS bypass | zero copy | triggered operations | ?

Established Principles for Compute Acceleration

OpenGL | Microsoft DirectX11

Generalization → Revolutionizes Acceleration

NVIDIA CUDA | OpenMP 4.0 | OpenCL

Where do we stand in Network Acceleration?

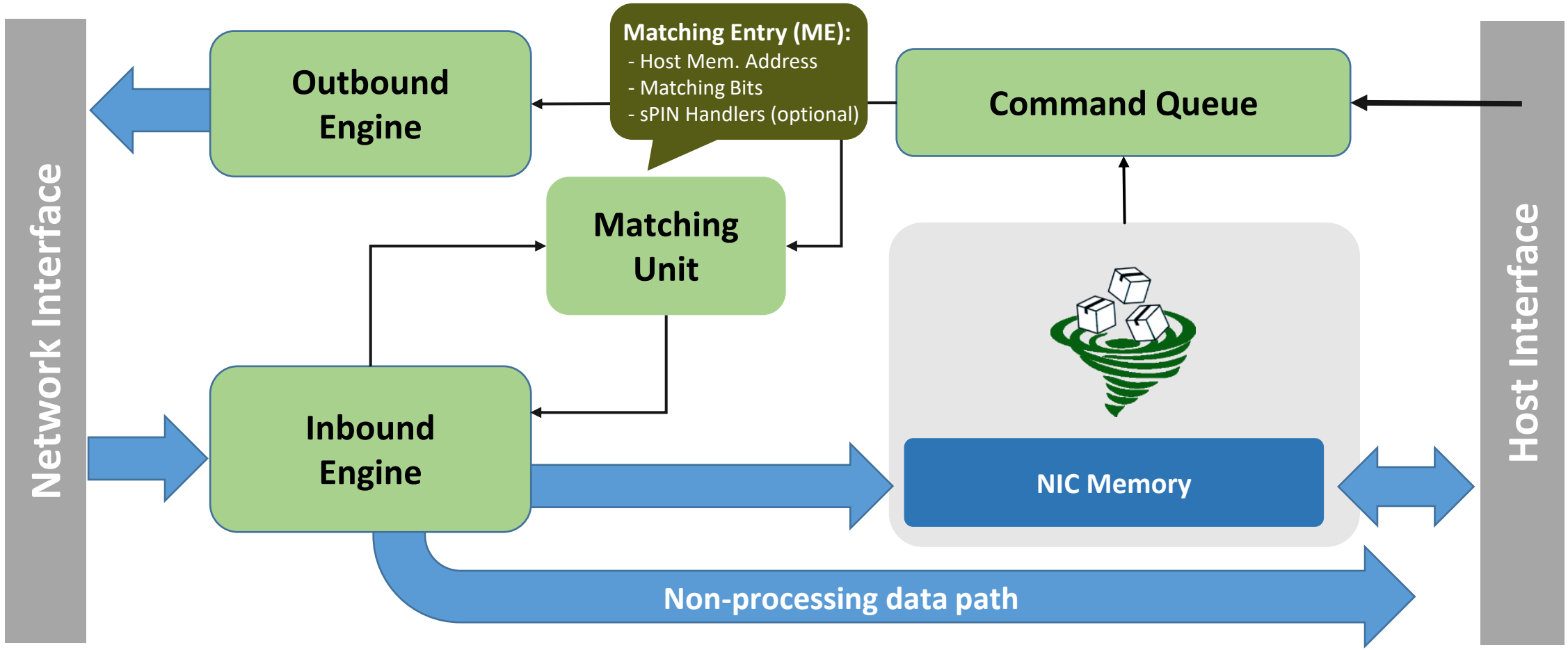
P4 | eBPF

Generalization → Data Acceleration

RISC-V

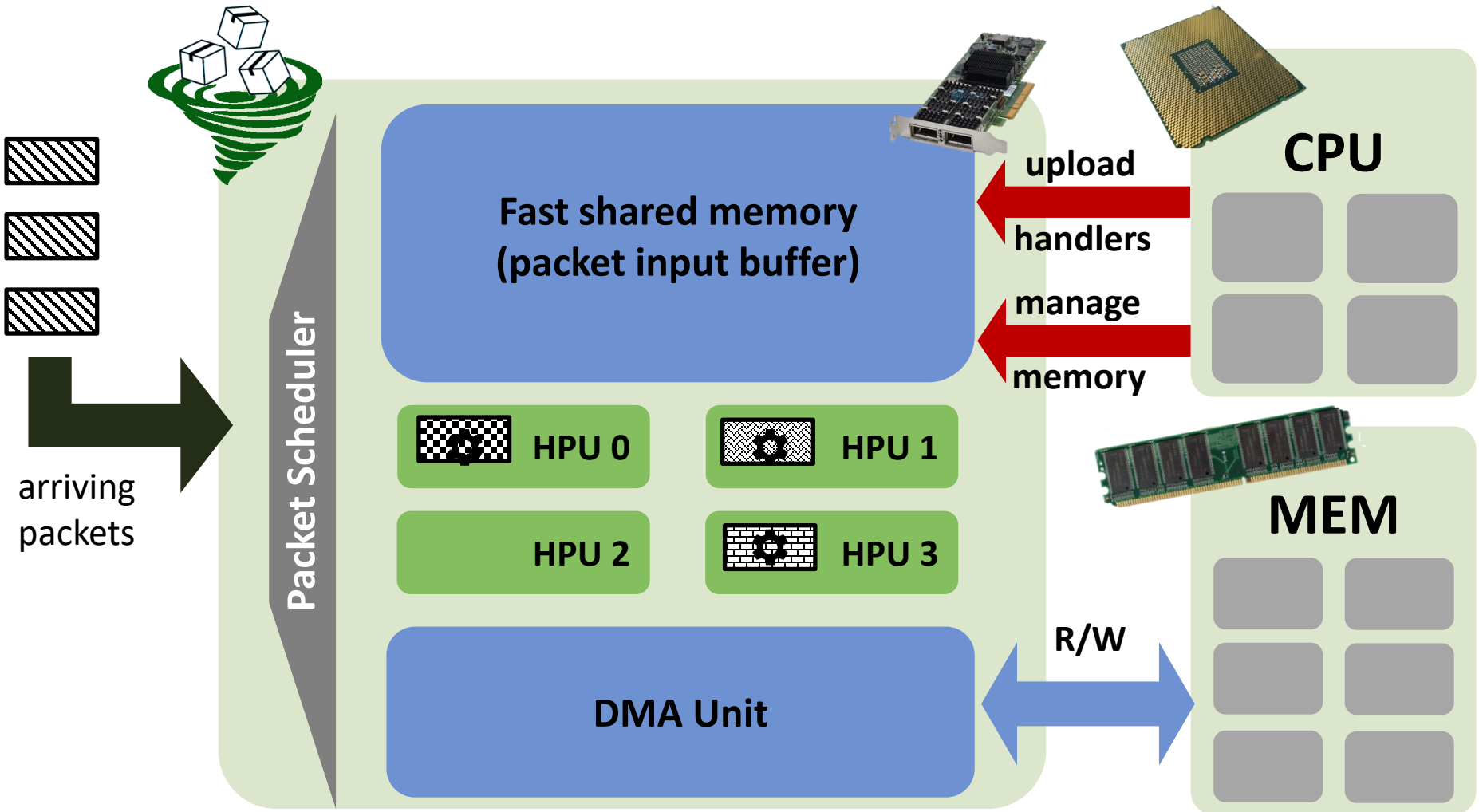
TH et al.: "sPIN: High-performance streaming Processing in the Network", SC17 best paper candidate

sPIN NIC – Architecture for fast Network Processing



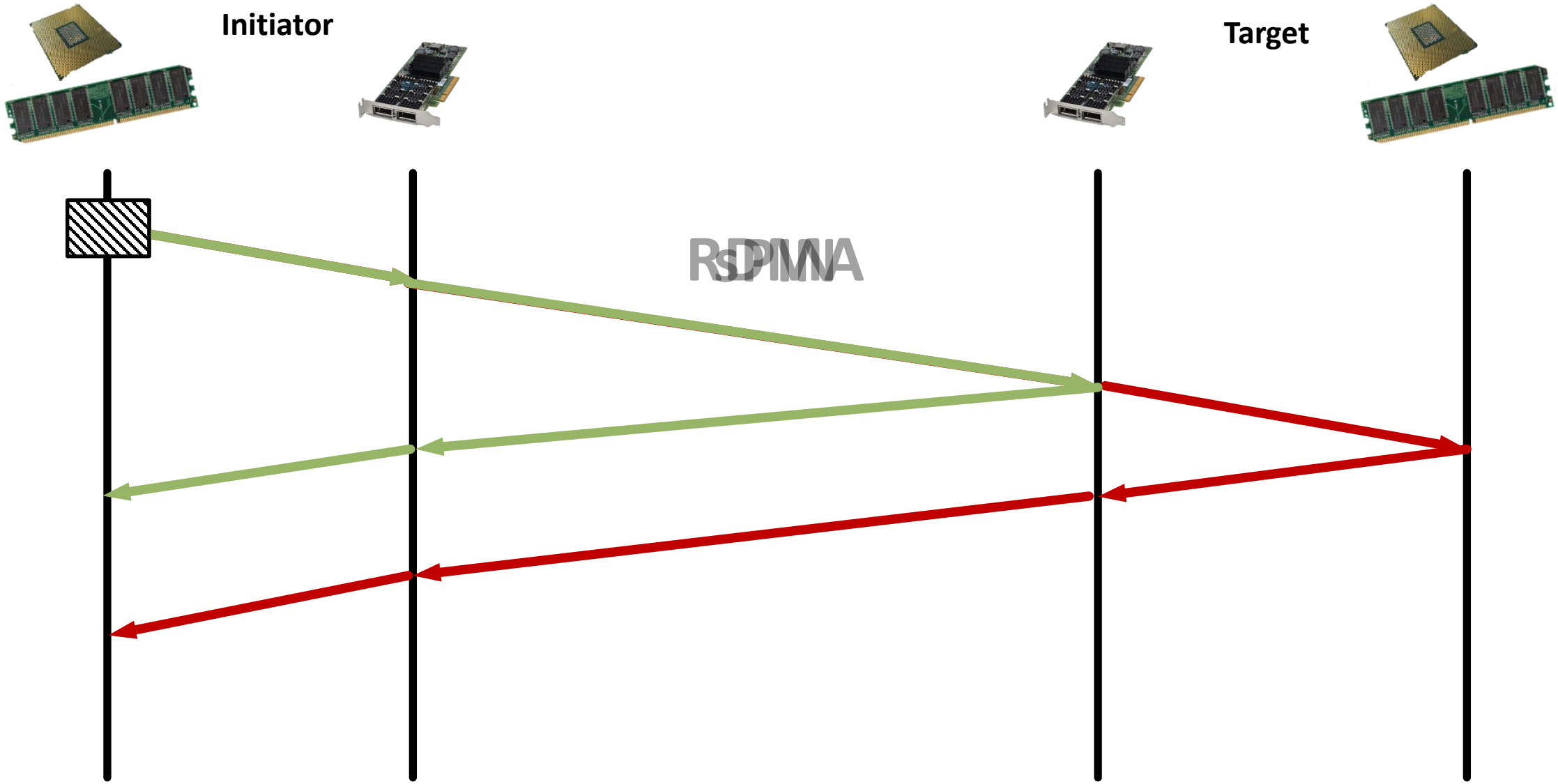
TH et al.: "sPIN: High-performance streaming Processing in the Network", SC17 best paper candidate

sPIN NIC - Abstract Machine Model for Packet Processing



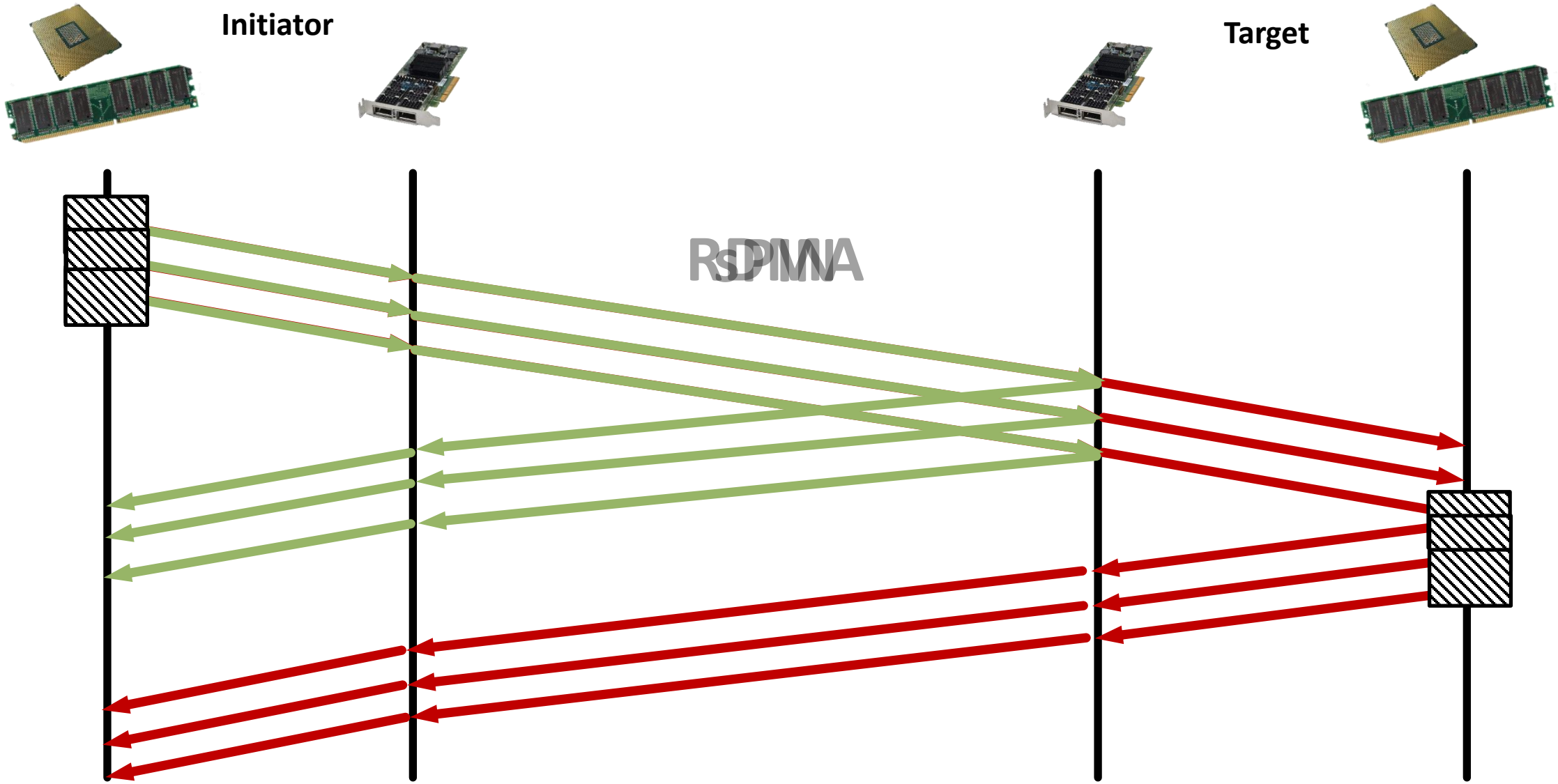
TH et al.: "sPIN: High-performance streaming Processing in the Network", SC17 best paper candidate

RDMA vs. sPIN in action: Simple Ping Pong



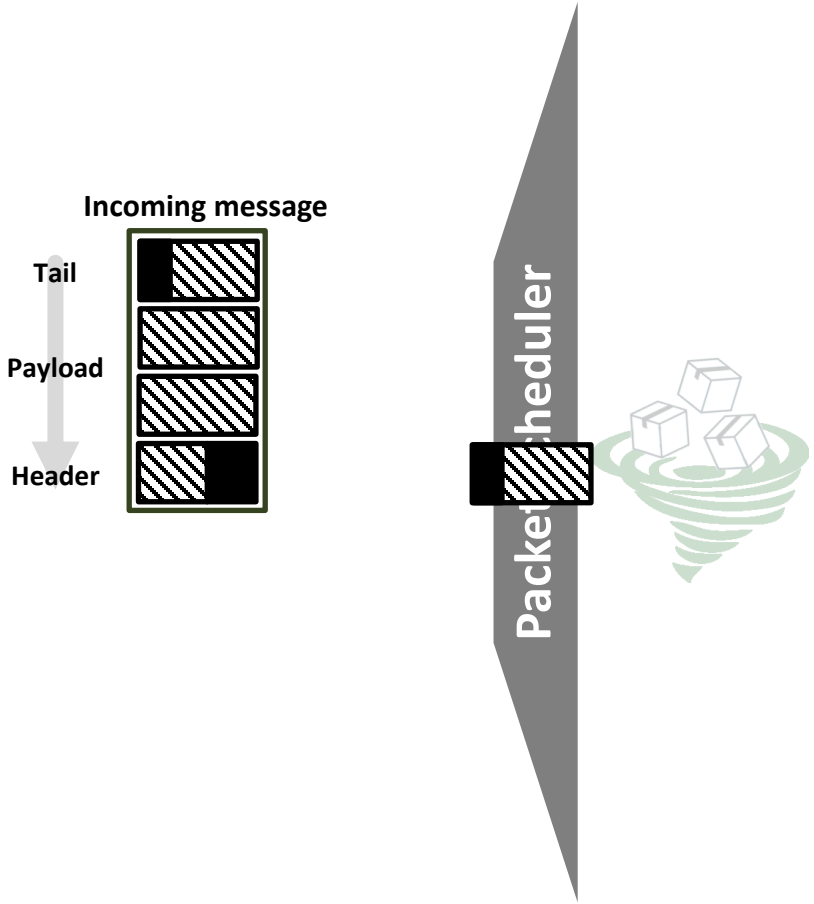
TH et al.: “sPIN: High-performance streaming Processing in the Network”, SC17 best paper candidate

RDMA vs. sPIN in action: Streaming Ping Pong



TH et al.: “sPIN: High-performance streaming Processing in the Network”, SC17 best paper candidate

sPIN – Programming Interface



Header handler

```
__handler int pp_header_handler(const ptl_header_t h, void *state) {
   pingpong_info_t *i = state;
   i->source = h.source_id;
   return PROCESS_DATA; // execute payload handler to put from device
}
```

Payload handler

```
__handler int pp_payload_handler(const ptl_payload_t p, void * state) {
   pingpong_info_t *i = state;
   PtlHandlerPutFromDevice(p.base, p.length, 1, 0, i->source, 10, 0, NULL, 0);
   return SUCCESS;
}
```

Completion handler

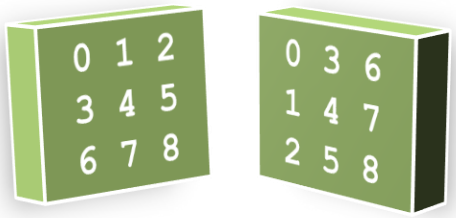
```
__handler int pp_completion_handler(int dropped_bytes,
                                    bool flow_control_triggered, void *state) {
   return SUCCESS;
}
```

```
connect(peer, /* … */, &pp_header_handler, &pp_payload_handler, &pp_completion_handler);
```

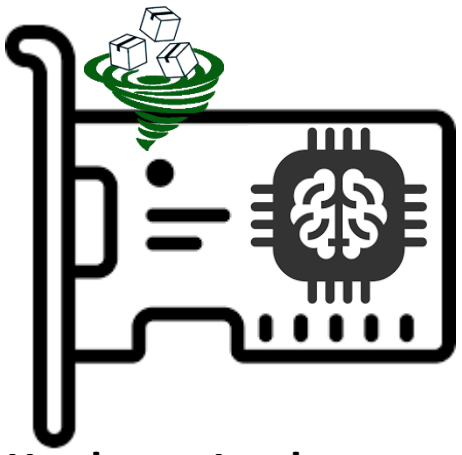
TH et al.: “sPIN: High-performance streaming Processing in the Network”, SC17 best paper candidate

Talk roadmap

Motivation and Overview

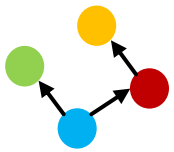


Data Layout Transformation



Hardware Implementation

further use cases



Network Group Communication

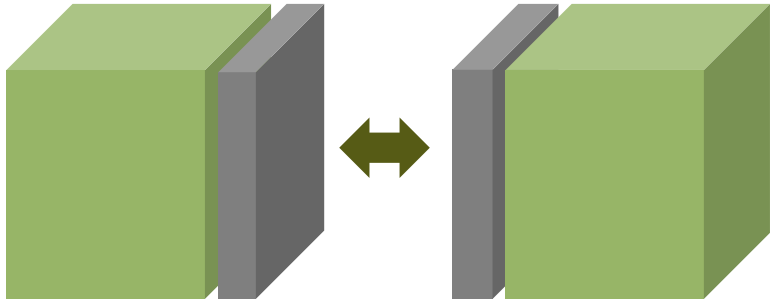
Distributed Data Management

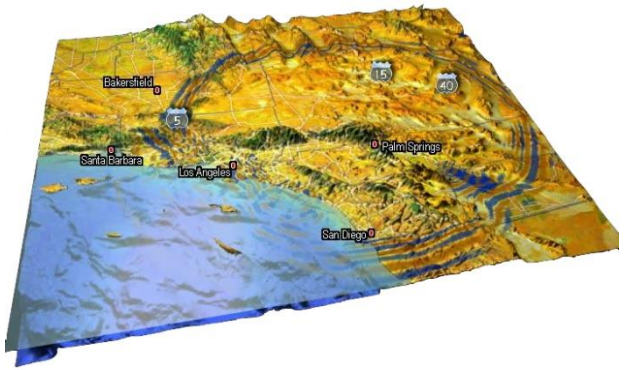
...

Application domain

Memory layout

Structured Exchange

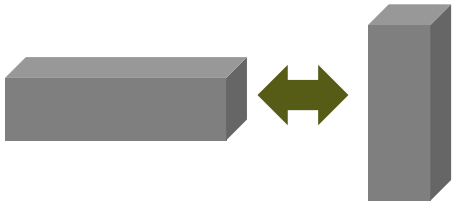




https://specfem3d.readthedocs.io/en/latest/

L. Carrington et al. High-frequency simulations of global seismic wave propagation using SPECFEM3D_GLOBE on 62K processors. SC 2008.

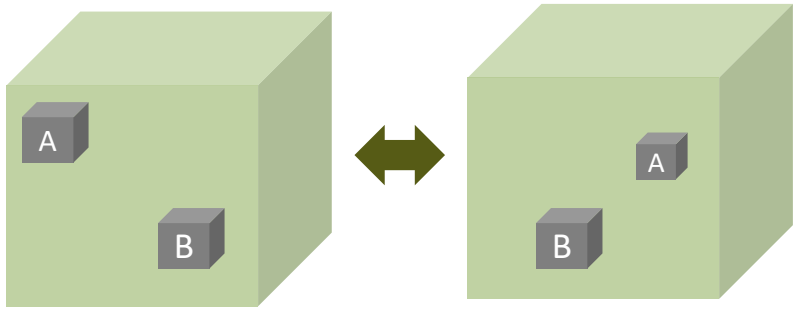
Reshaping

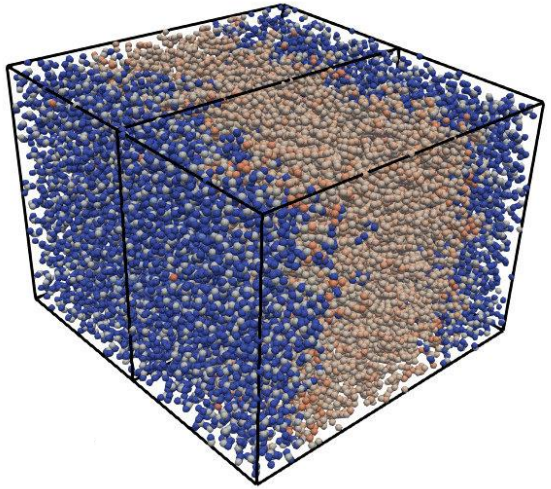


http://fourier.eng.hmc.edu/e161/lectures/fourier/node10.html

T. Hoefler et al. Parallel zero-copy algorithms for fast Fourier transform and conjugate gradient using MPI datatypes. EuroMPI 2010.

Unstructured Exchange





W. Usher et al. libIS: a lightweight library for flexible in transit visualization. ISAV 2018.

Programming Support for Non-Contiguous Transfers

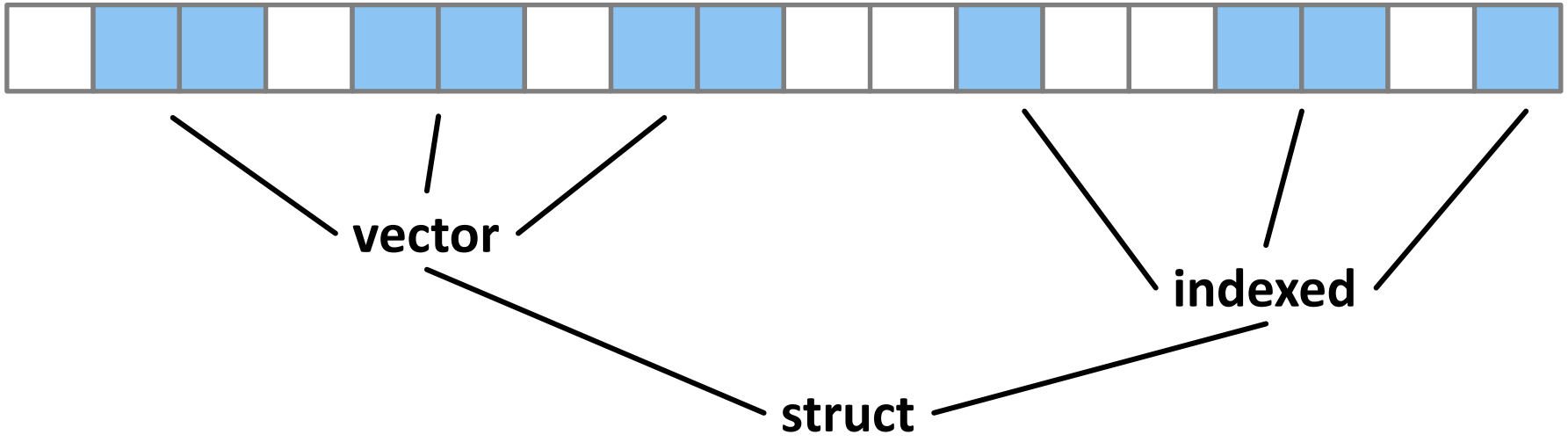
ARMCI **CAF** **Chapel** **Portals 4**
SHMEM **UPC** **X10**

I/O Vectors Strided transfers Compiler-Assisted Aggregation

Support for multiple strides (e.g., 3D faces)

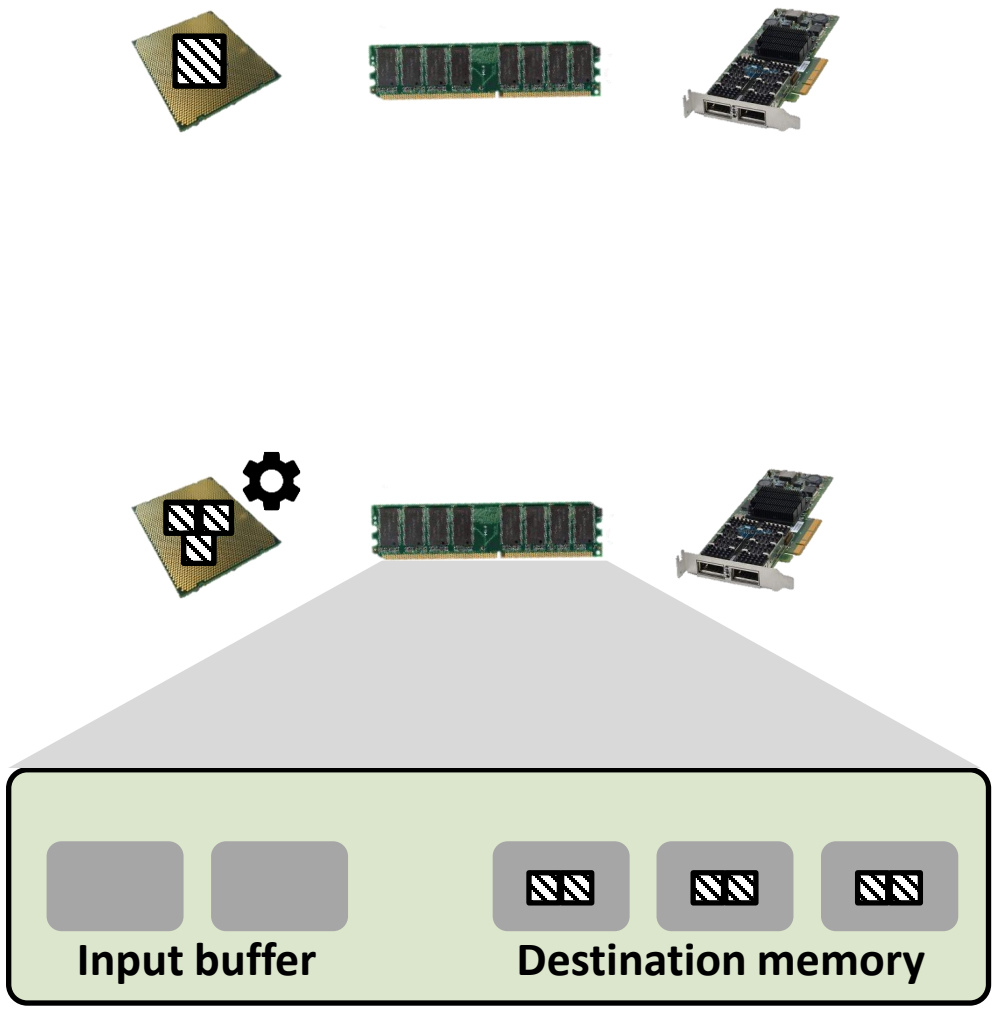
MPI

Derived Datatypes



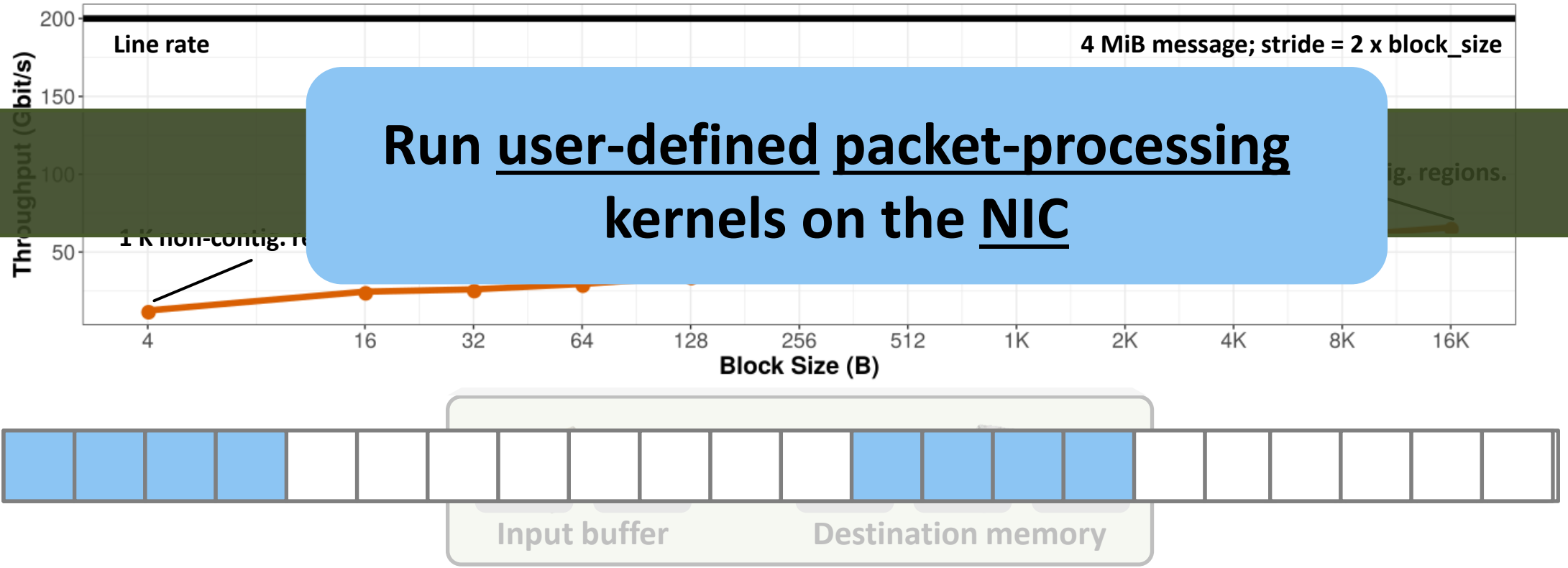
Salvatore Di Girolamo et al.: "Network-Accelerated Non-Contiguous Memory Transfers", IEEE/ACM SC19

State of the Art in MPI Datatypes Processing



Gropp, W., et al., March. Improving the performance of MPI derived datatypes. *MPIDC'99*
Torsten Hoefler, Salvatore Di Girolamo, Konstantin Taranov, Ryan E. Grant, and Ron Brightwell. 2017. sPIN: High-performance streaming Processing In the Network. SC'17

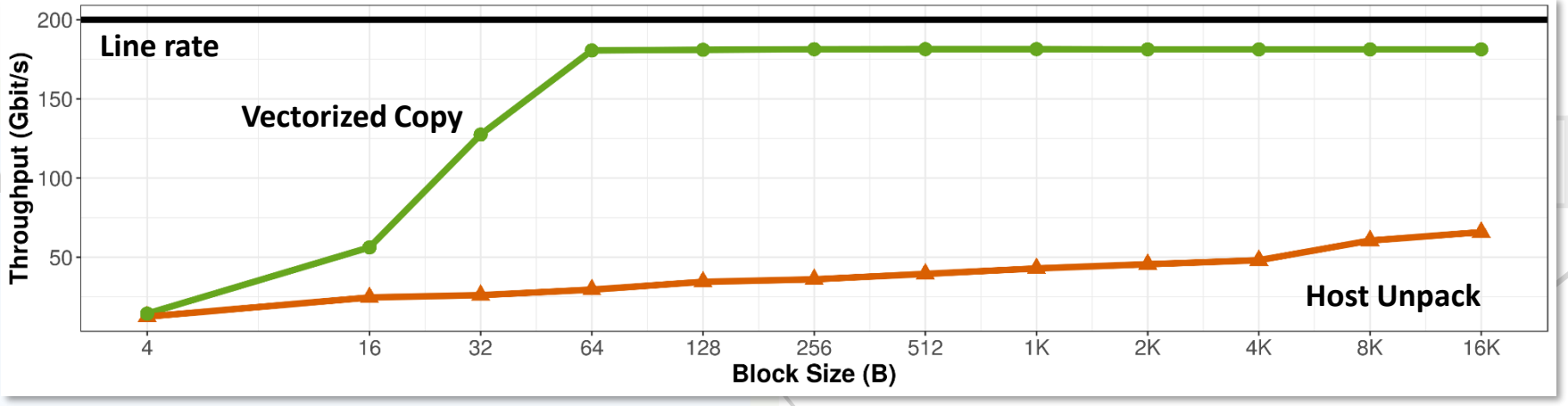
State of the Art in MPI Datatypes Processing

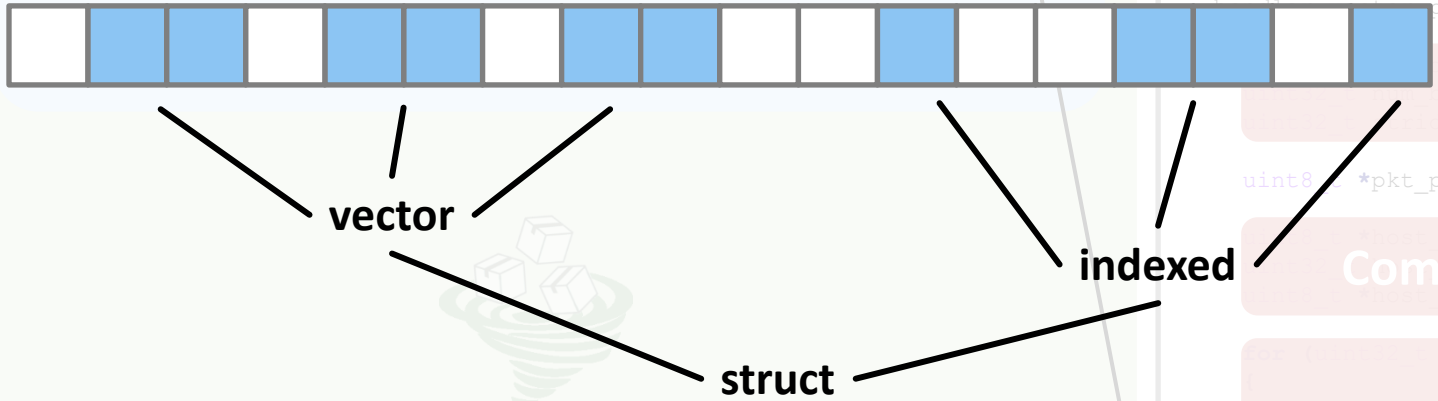


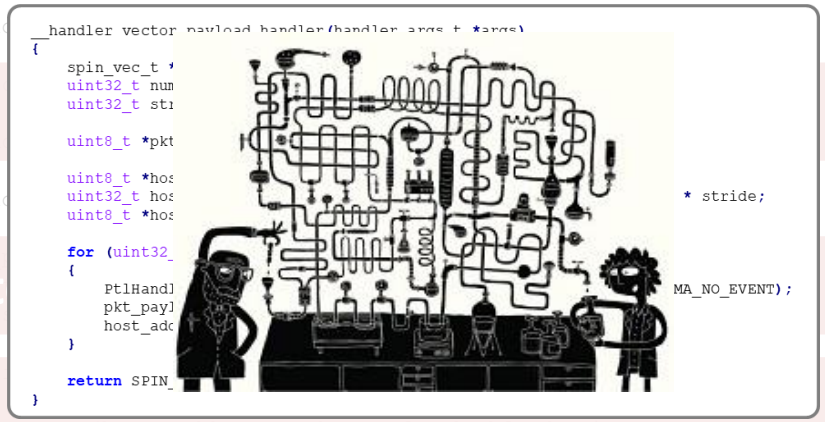
Run <u>user-defined</u> <u>packet-processing</u> kernels on the <u>NIC</u>

Gropp, W., et al., March. Improving the performance of MPI derived datatypes. *MPIDC'99*

A simple vectorize scatter datatype







Can we define a general handler to process arbitrary datatypes?

Salvatore Di Girolamo et al.: "Network-Accelerated Non-Contiguous Memory Transfers", IEEE/ACM SC19

Porting the MPI Types Library [1] to sPIN

NIC Memory

Index{ #blocks: 2, blocklen: 1, offsets: {0, x}, basetype: * }

↓

Vector{ #blocks: 3, blocklen: 2, stride: 3, basetype: ■ }

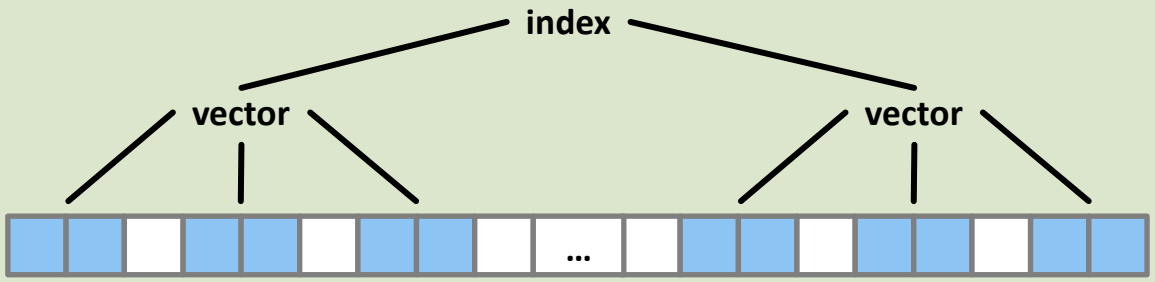
Dataloops

Index:

Vector:

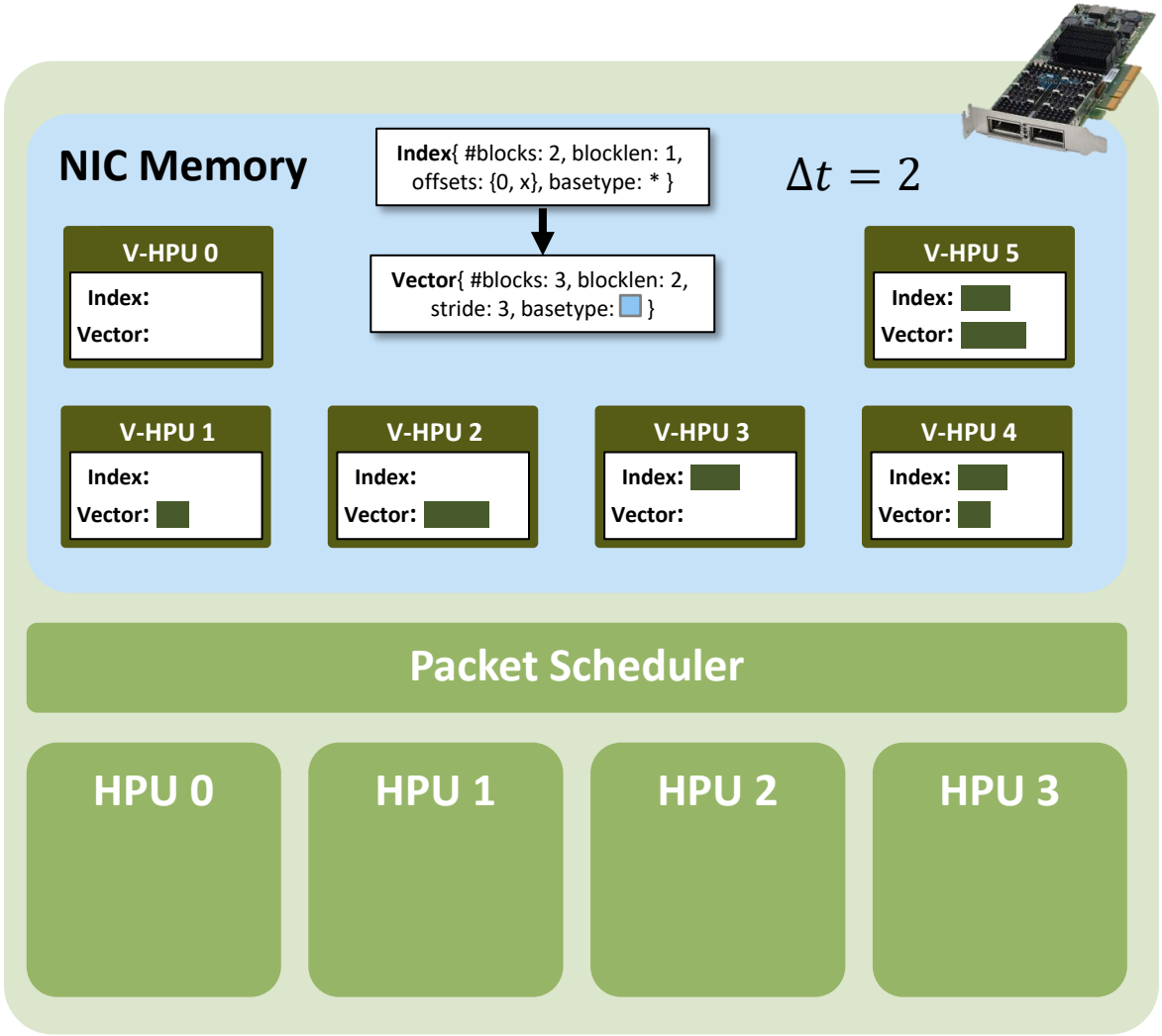
Segment

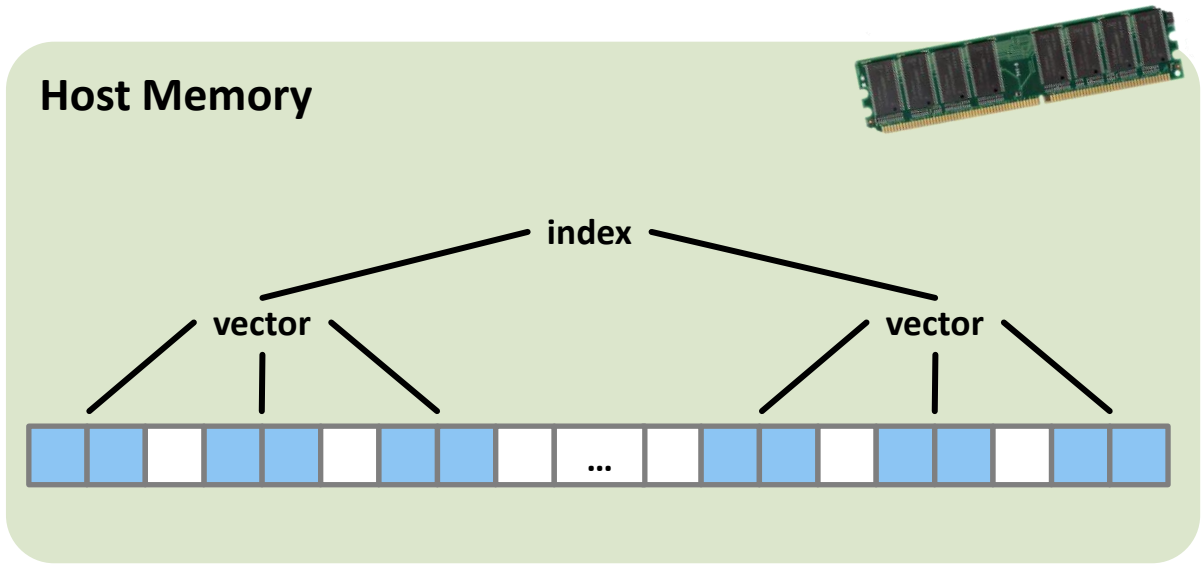
Host Memory

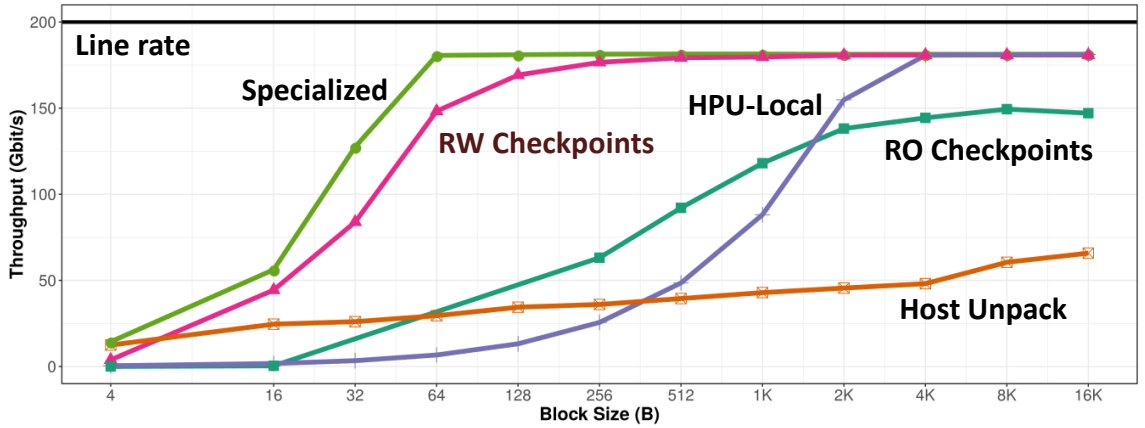


[1] Ross, Robert, et al. "Processing MPI datatypes outside MPI." European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting. Springer, Berlin, Heidelberg, 2009.

MPI Types Library on sPIN: Read-Write Checkpoints

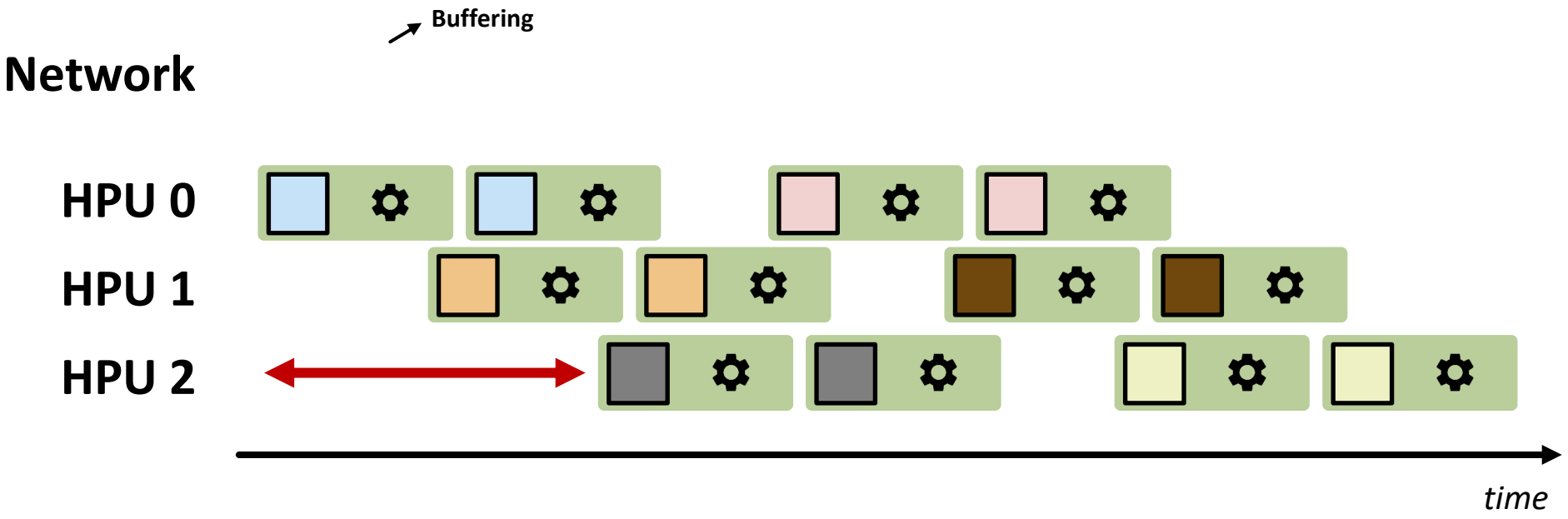






Salvatore Di Girolamo et al.: "Network-Accelerated Non-Contiguous Memory Transfers", IEEE/ACM SC19

Checkpoint Interval Selection



$$T_C = T_{pkt} + \left\lceil \frac{\Delta r}{k} \right\rceil \cdot (P-1) \cdot T_{pkt} + \left\lceil \frac{n_{pkt}}{P} \right\rceil \cdot T_{PH}(\gamma)$$

1 Limit the impact of the scheduling overhead

2 Do not saturate NIC memory with checkpoints

3 Do not saturate the packet buffer

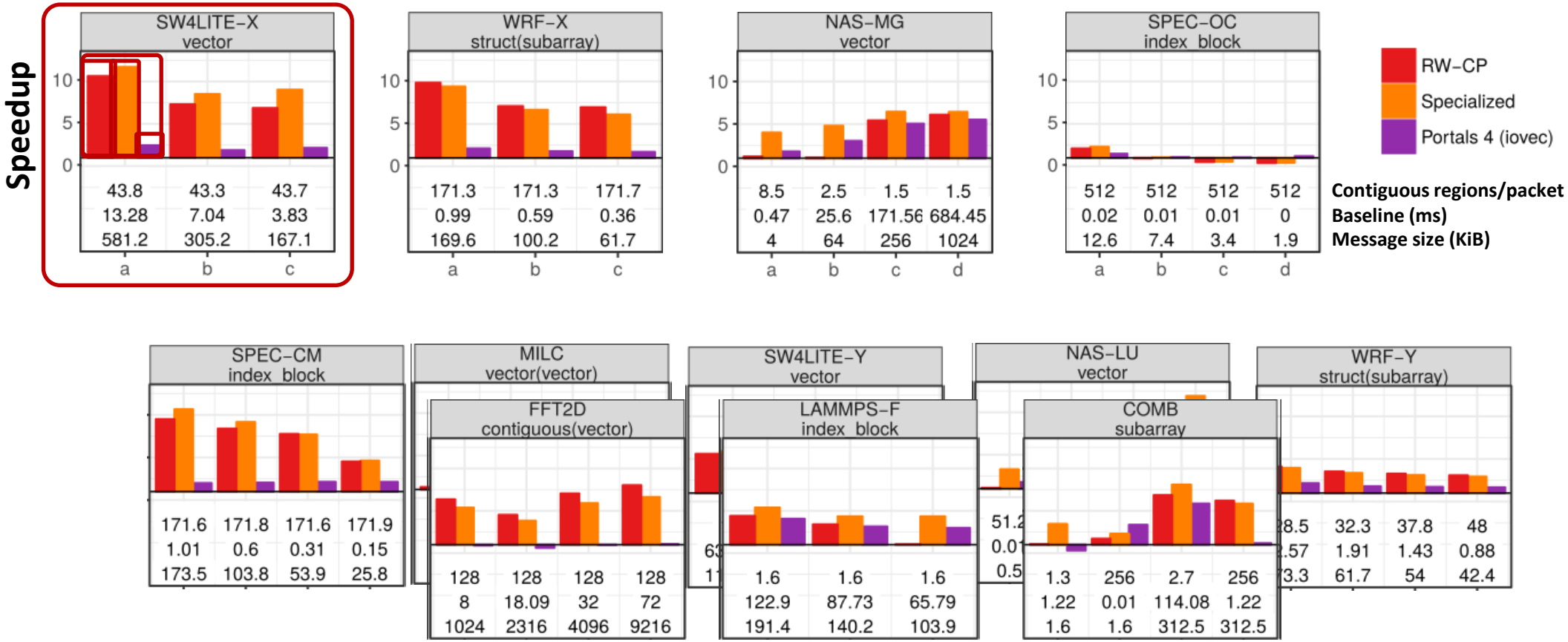
Salvatore Di Girolamo et al.: "Network-Accelerated Non-Contiguous Memory Transfers", IEEE/ACM SC19

Cray Slingshot Simulator

32 Cortex A15 @800 MHz,
single-cycle access memory

Salvatore Di Girolamo et al.: “Network-Accelerated Non-Contiguous Memory Transfers”, IEEE/ACM SC19

Real Application DDTs

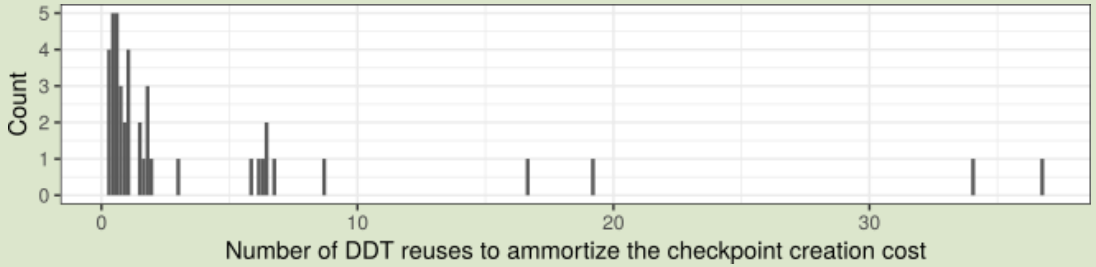


Salvatore Di Girolamo et al.: "Network-Accelerated Non-Contiguous Memory Transfers", IEEE/ACM SC19

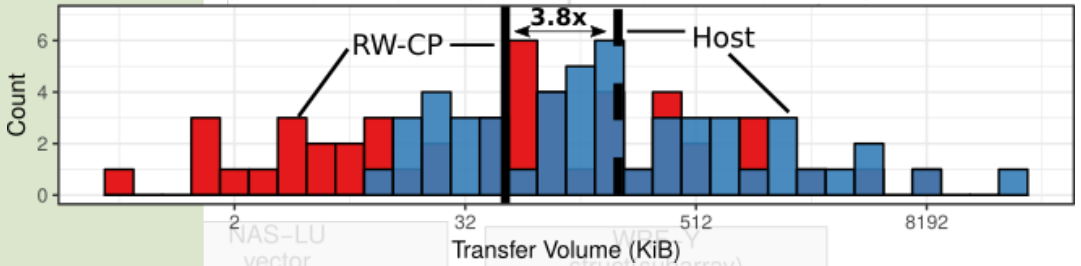
Real Applications DDTs

Checkpointing Overhead

Data Movement

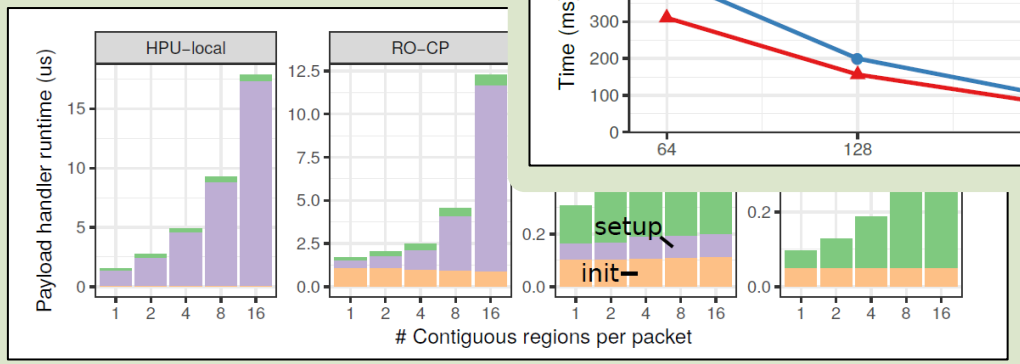


75% of the analyzed DDTs amortized after 4 reuses

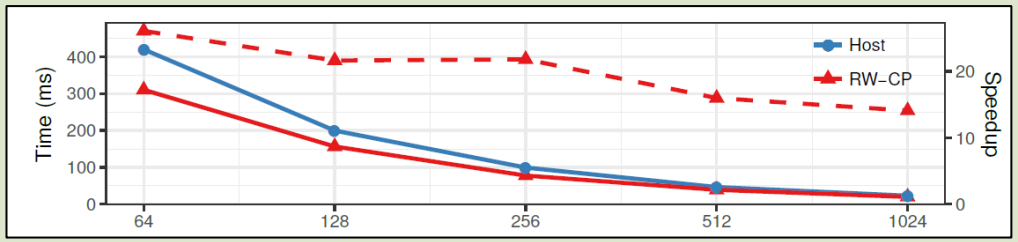


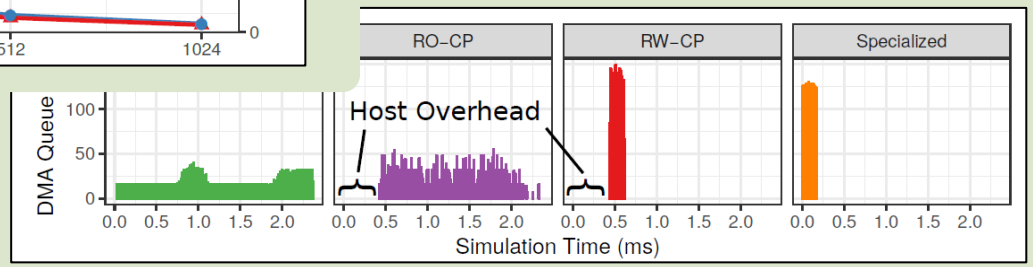
Up to 3.8x less moved data volume

Handler Analysis



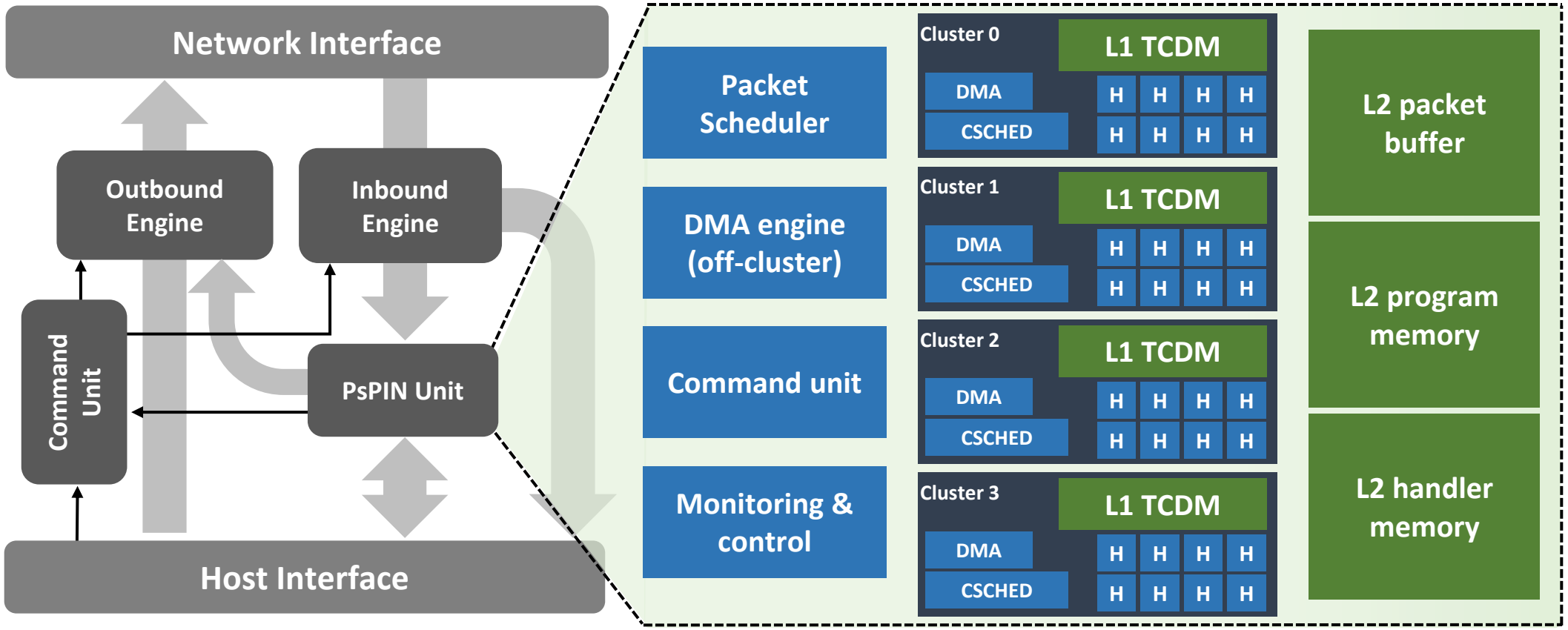
Full app speedup (FFT2D)





Salvatore Di Girolamo et al.: "Network-Accelerated Non-Contiguous Memory Transfers", IEEE/ACM SC19

PsPIN hardware implementation: sPIN on PULP

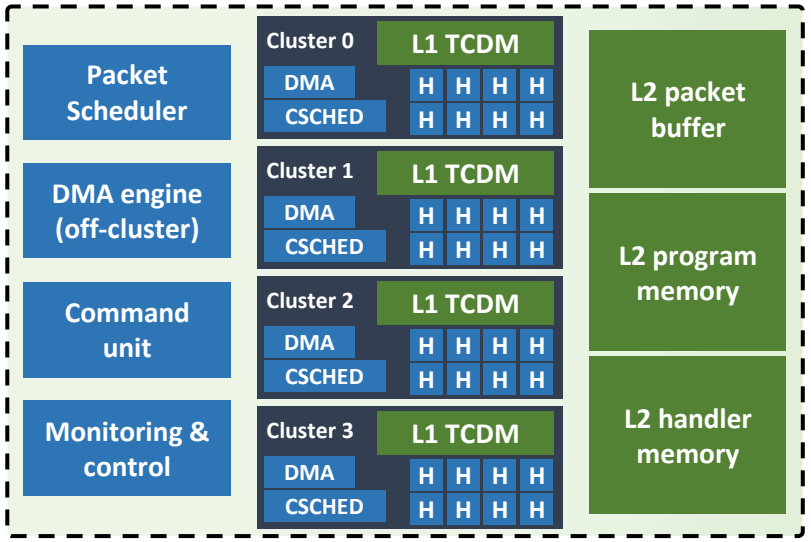


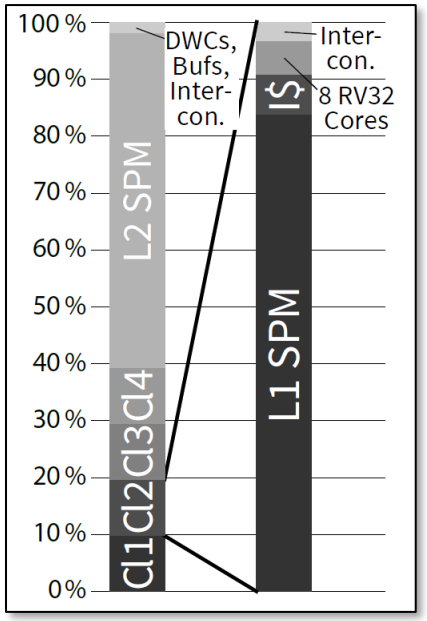
Circuit Complexity and Power Estimations

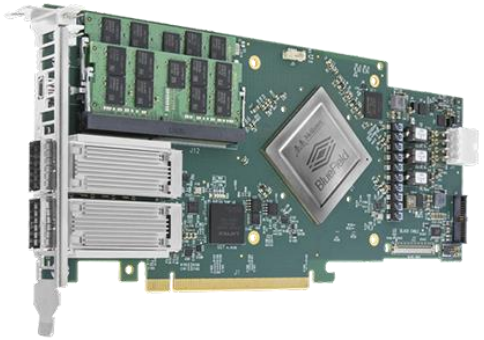
- **Processor synthesized in GlobalFoundries 22nm fully depleted silicon on insulator (FDSOI) technology**
  - Timing: 1 GHz
- **Accelerator complexity: ~95 MGE**
  - 18.5 mm2 area (assuming layout density 85%)
  - Mellanox BlueField: 16 A72 64bit cores

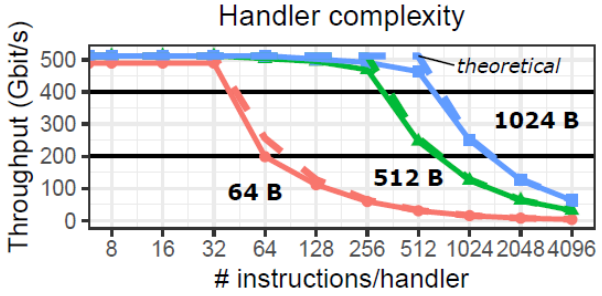
  Estimated area: 51 mm2

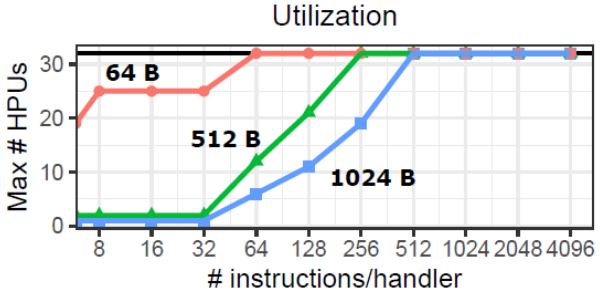
  We could have up to 64 cores and 18 MiB of memory for that area.
- **Power consumption (100% toggle rate): 6 W (not including I/O and PHY power).**











Why choosing PULP for sPIN?

Architectures:

zynq: ARM Cortex-A53, 64-bit, 2-way superscalar, 1.2 GHz
ault: Intel Skylake Gold 6154, 64-bit, out-of-order execution, 3 GHz
PsPIN/RI5CY: RISC-V based, 32-bit, in-order, 1 GHz

Arch.	Tech.	Die area	PEs	Memory	Area/PE	Area/PE (scaled)
ault	14 nm	485 mm^2 [4]	18	43.3 MiB	17.978 mm^2	35.956 mm^2
zynq	16 nm	3.27 mm^2 [3]	4	1.125 MiB	0.876 mm^2	1.752 mm^2
PsPIN	22 nm	18.5 mm^2	32	12 MiB	0.578 mm^2	0.578 mm^2

Use cases:

Data reduction

Single message aggregation

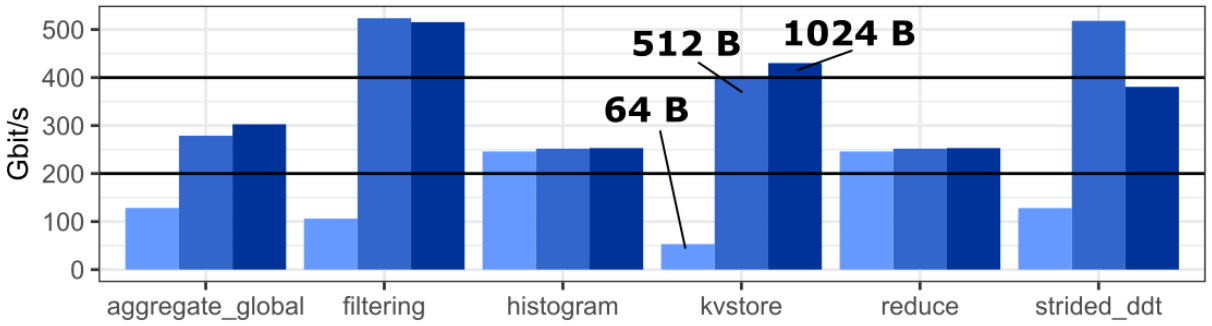
Packet filtering/rewriting

KV store cache

Strided datatypes

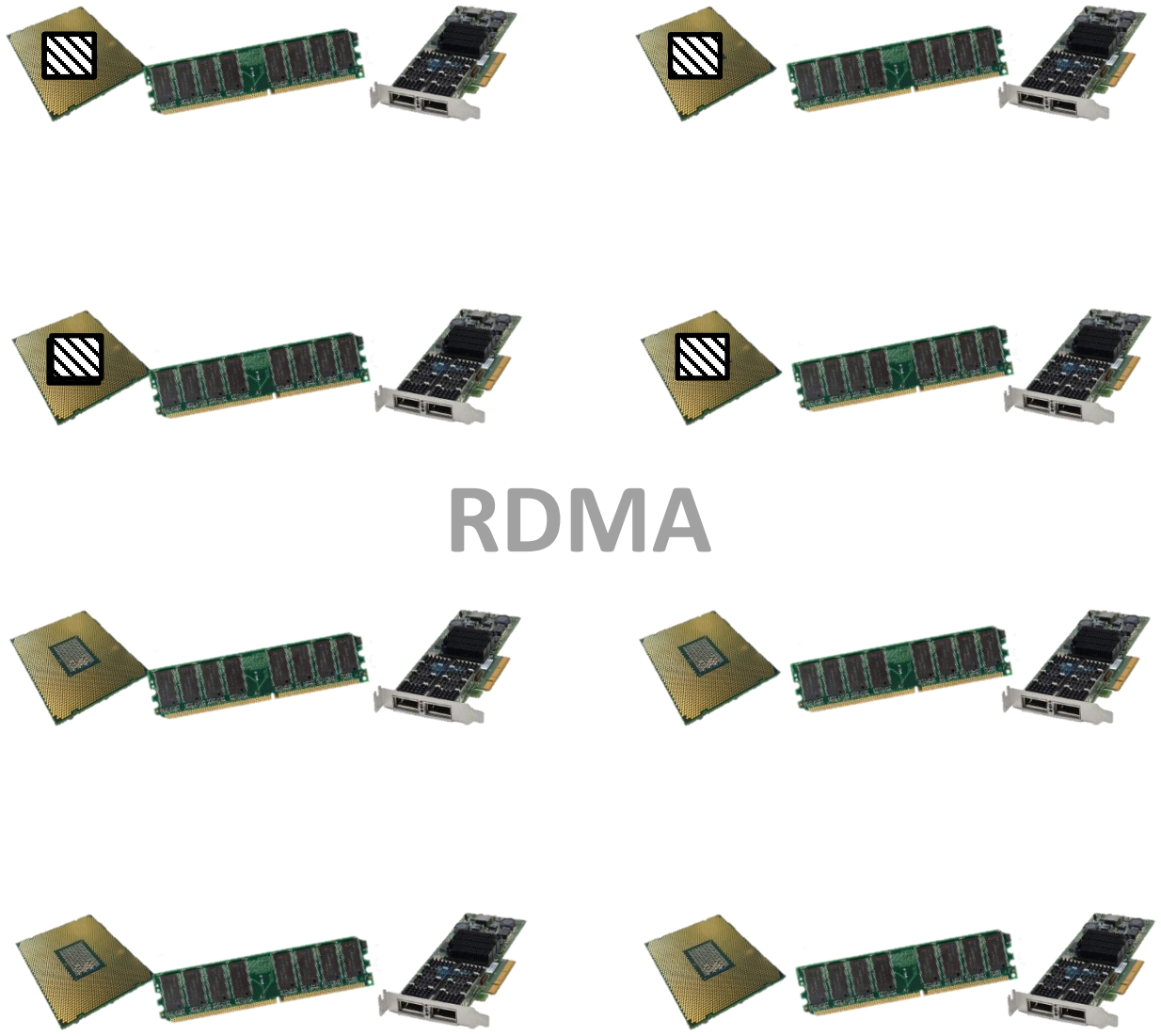
Histogram

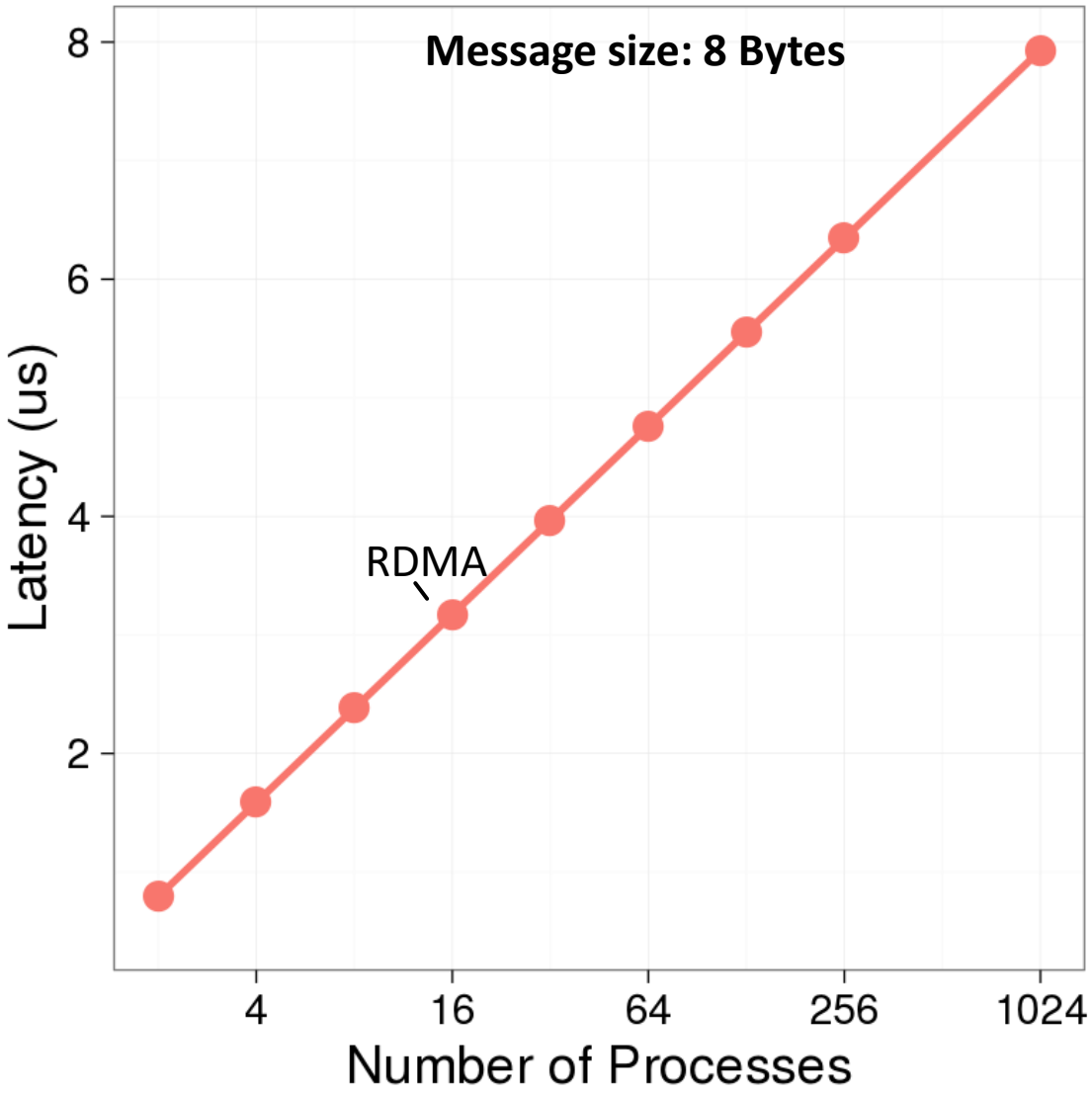
Actual throughput on PsPIN:



Illustrating broadcast acceleration with sPIN

Network Group Communication

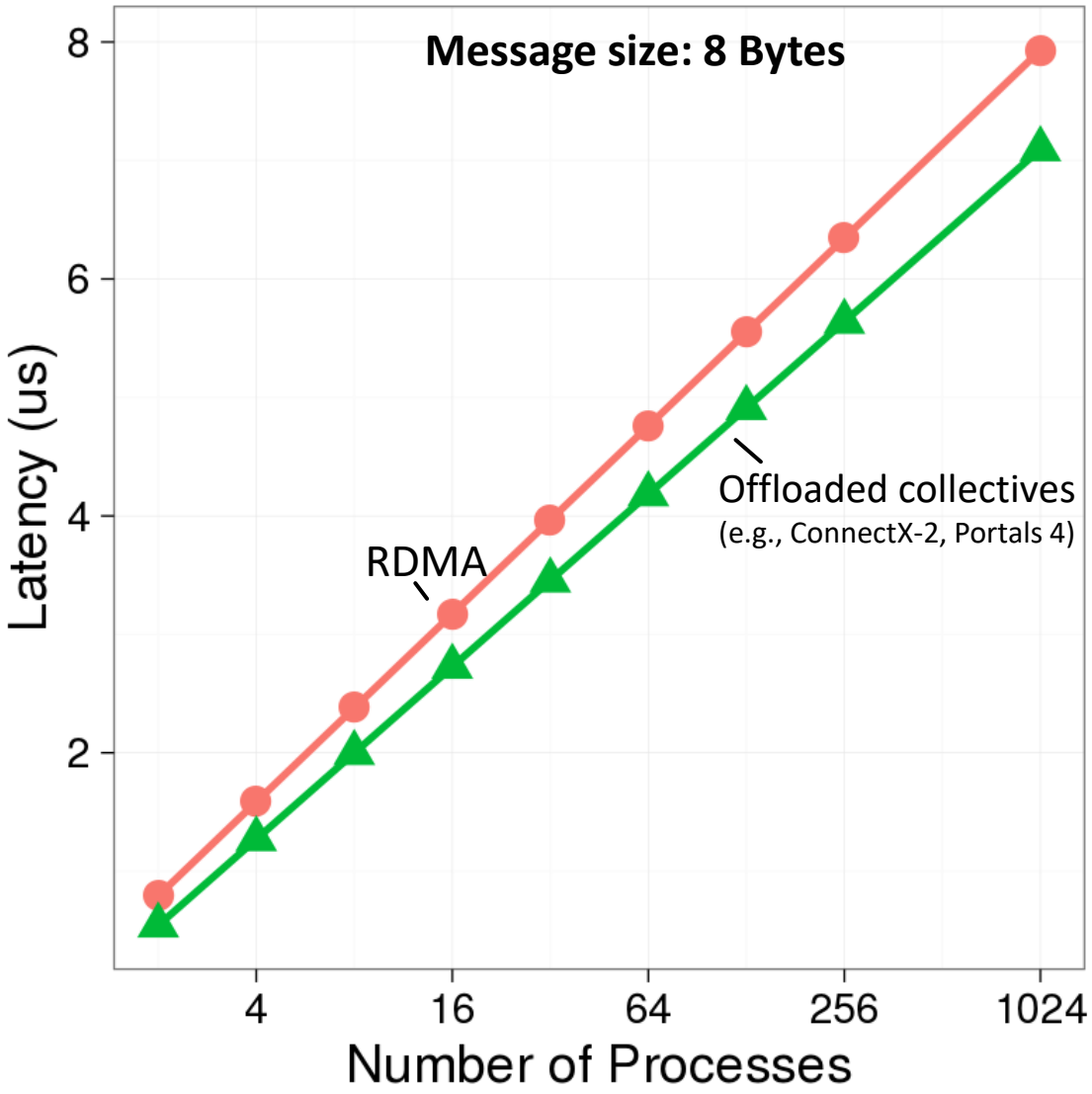




Liu, J., et al., High performance RDMA-based MPI implementation over InfiniBand. *International Journal of Parallel Programming* 2004

Illustrating broadcast acceleration with sPIN

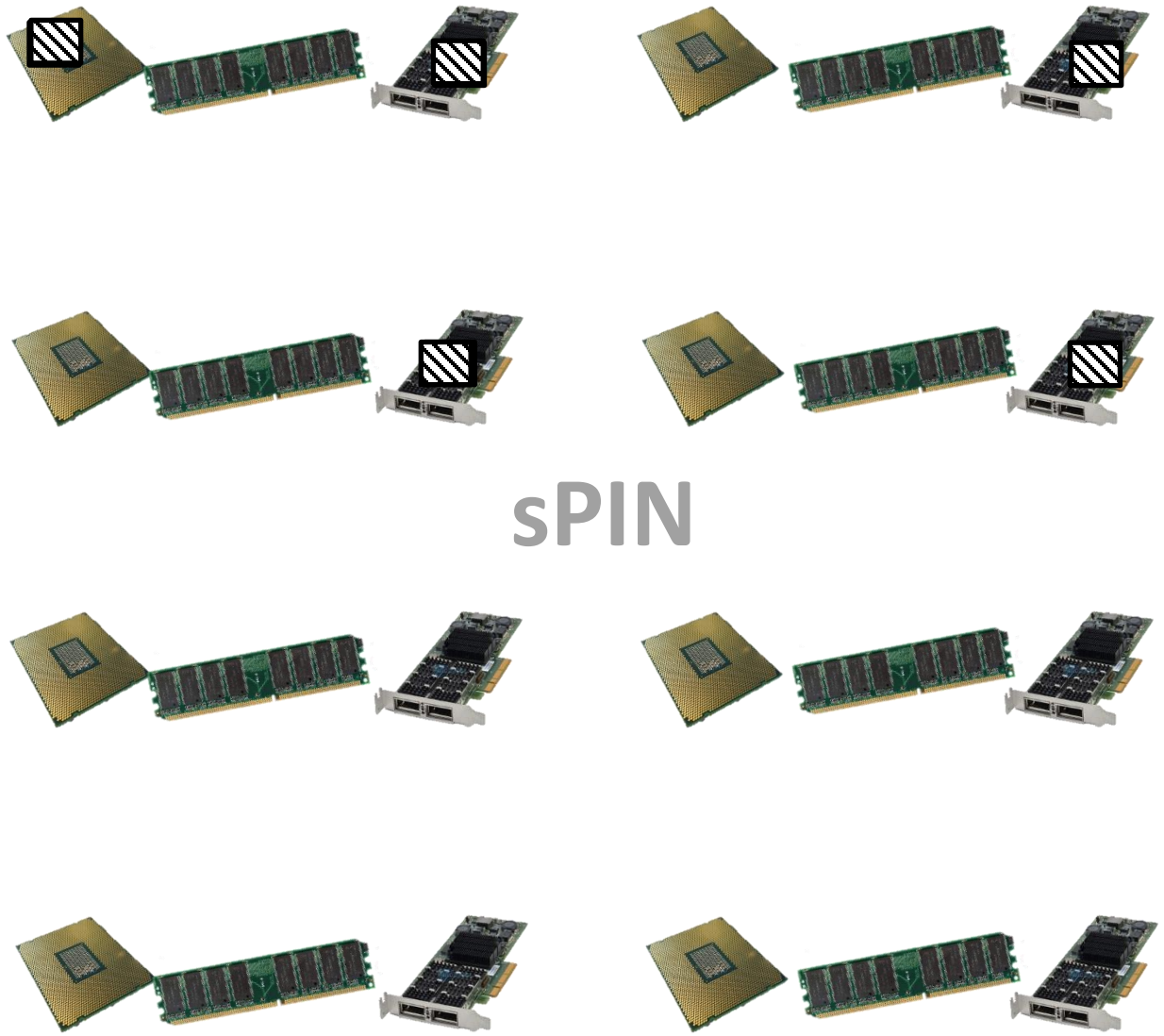
Network Group Communication

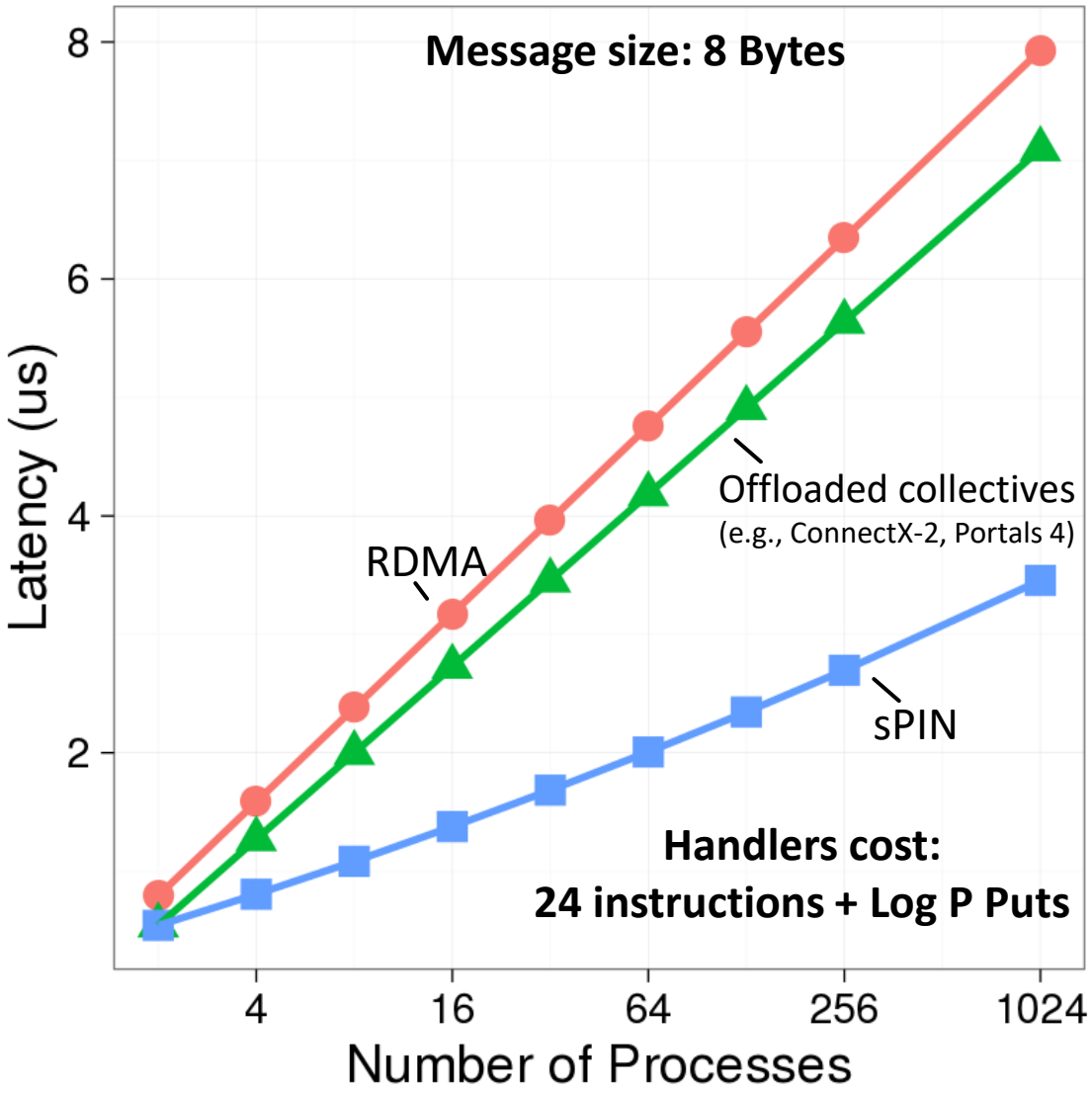


Underwood, K.D., et al., Enabling flexible collective communication offload with triggered operations. *HOTI'11*
Liu, J., et al., High performance RDMA-based MPI implementation over InfiniBand. *International Journal of Parallel Programming* 2004

Illustrating broadcast acceleration with sPIN

Network Group Communication





Underwood, K.D., et al., Enabling flexible collective communication offload with triggered operations. *HOTI'11*
Liu, J., et al., High performance RDMA-based MPI implementation over InfiniBand. *International Journal of Parallel Programming* 2004

Further results and use-cases

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

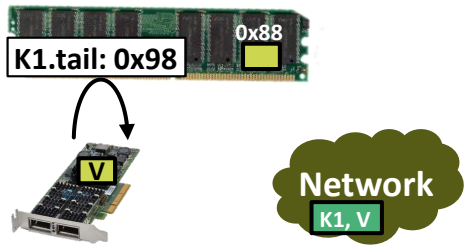
program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

Use Case 5: Distributed KV Store



41% lower latency

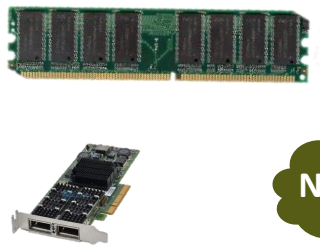
Kalia, A., et al., Using RDMA efficiently for key-value services. In *ACM SIGCOMM Computer Communication Review, 2014*

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

Use Case 5: Distributed KV Store

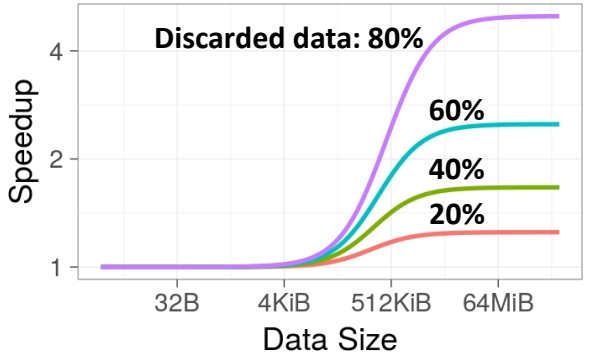


Network

41% lower latency

Kalia, A., et al., Using RDMA efficiently for key-value services. In *ACM SIGCOMM Computer Communication Review, 2014*

Use Case 6: Conditional Read



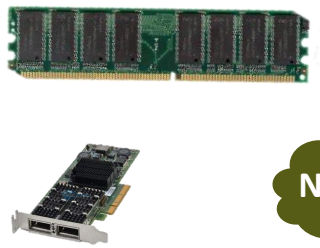
Barthels, C., et al., Designing Databases for Future High-Performance Networks. *IEEE Data Eng. Bulletin, 2017*

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

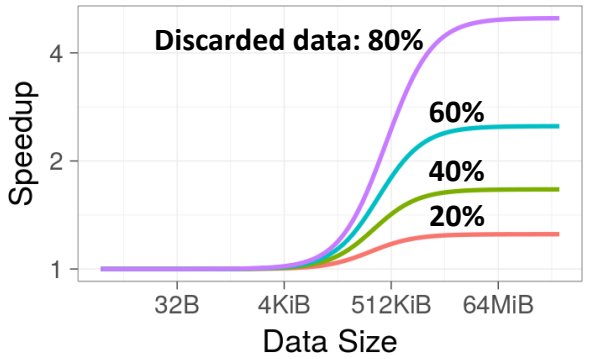
Use Case 5: Distributed KV Store



41% lower latency

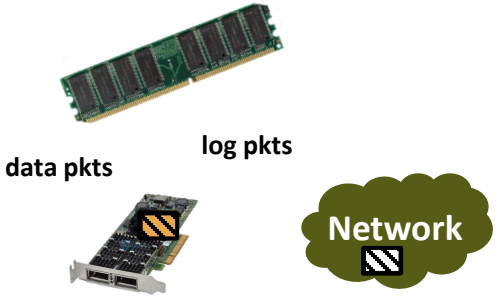
Kalia, A., et al., Using RDMA efficiently for key-value services. In *ACM SIGCOMM Computer Communication Review, 2014*

Use Case 6: Conditional Read



Barthels, C., et al., Designing Databases for Future High-Performance Networks. *IEEE Data Eng. Bulletin, 2017*

Use Case 7: Distributed Transactions



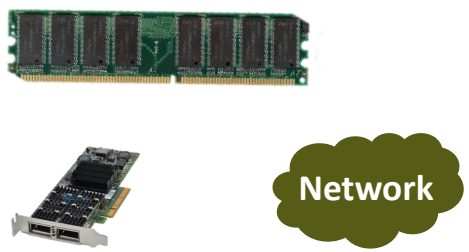
Dragojević, A, et al., No compromises: distributed transactions with consistency, availability, and performance. SOSP'15

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

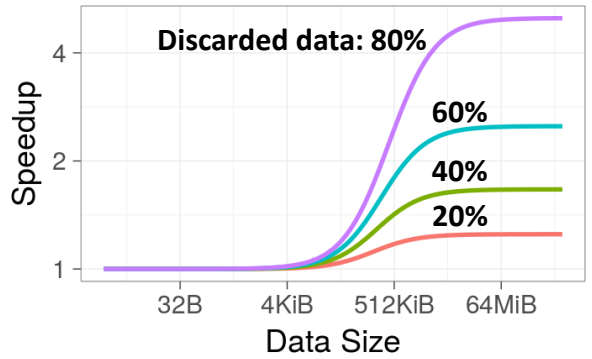
Use Case 5: Distributed KV Store



41% lower latency

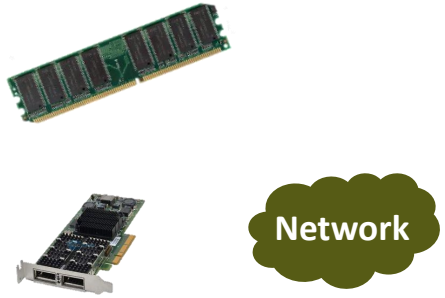
Kalia, A., et al., Using RDMA efficiently for key-value services. In *ACM SIGCOMM Computer Communication Review, 2014*

Use Case 6: Conditional Read



Barthels, C., et al., Designing Databases for Future High-Performance Networks. *IEEE Data Eng. Bulletin, 2017*

Use Case 7: Distributed Transactions



Dragojević, A, et al., No compromises: distributed transactions with consistency, availability, and performance. SOSP'15

Use Case 8: FT Broadcast

Bosilca, G., et al., Failure Detection and Propagation in HPC systems. SC'16

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

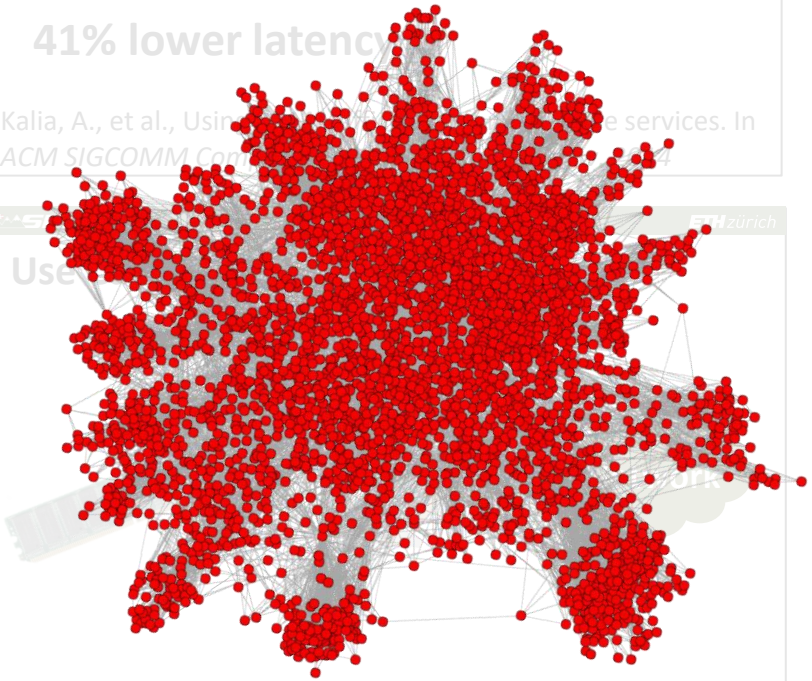
Use Case 5: Distributed KV Store

The Next 700 sPIN use-cases

… just think about sPIN graph kernels ….

41% lower latency

Kalia, A., et al., Usin… e services. In *ACM SIGCOMM Con…*



Use Case 6: Conditional Read

Barthels, C., et al., Designing Databases for Future High-Performance Networks. *IEEE Data Eng. Bulletin, 2017*

Use Case 7: Distributed Transactions

Dragojević, A, et al., No compromises: distributed transactions with consistency, availability, and performance. SOSP'15

Use Case 8: …

Bosilca, G., et al., Failure Detection and Propagation in HPC systems. SC'16

Use Case 9: Distributed Consensus

István, Z., et al., Consensus in a Box: Inexpensive Coordination in Hardware. NSDI'16

Next step - pushing sPIN into network switches?

- **Needs to be carefully vetted!**
  - Can we achieve our goals with P4?
- **What else needs to be fixed before we go into the network?**
- **We chose to investigate network noise first**
  - Not enough time here but let me give you a brief overview.

Network noise analysis and mitigation

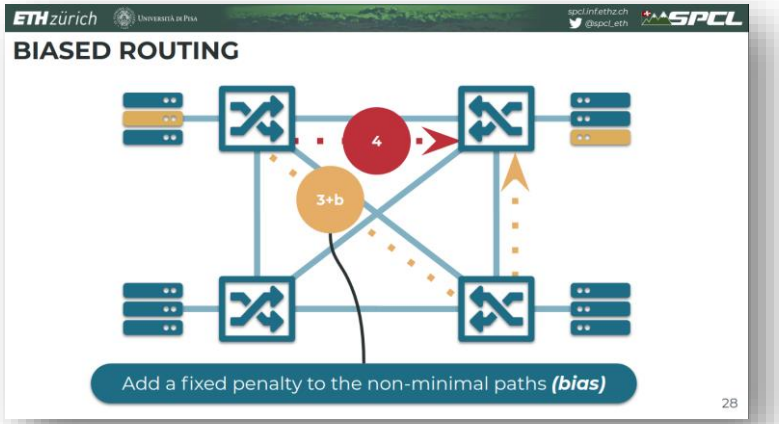
Mitigating Network Noise on Dragonfly Networks through Application-Aware Routing

Daniele De Sensi
desensi@di.unipi.it
University of Pisa
Pisa, Italy
ETH Zurich
Zurich, Switzerland

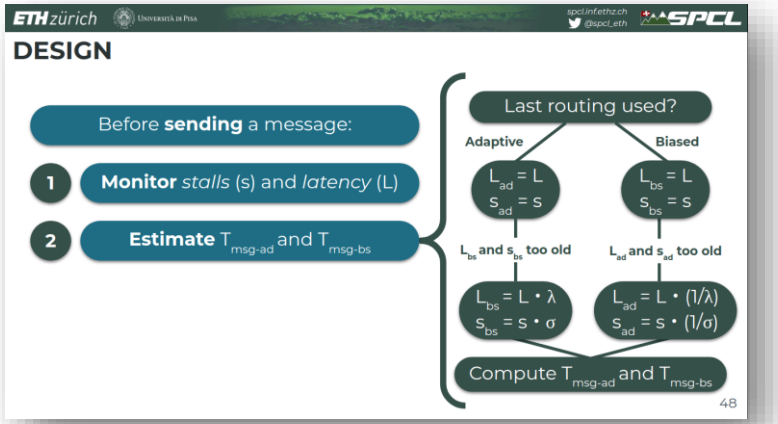
Salvatore Di Girolamo
salvatore.digirolamo@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

Torsten Hoefler
htor@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

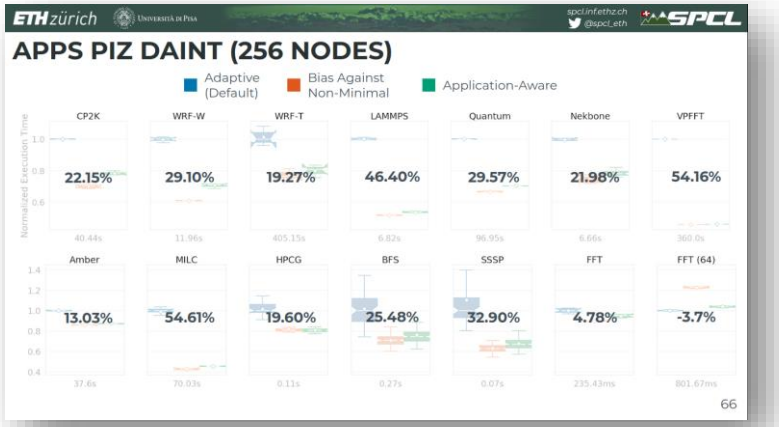
Analysis of the impact of adaptive routing on network noise



Design and implementation of a transparent solution



Improvements up to 55% on real applications



Daniele De Sensi et al.: "Mitigating Network Noise on Dragonfly Networks through Application-Aware Routing", IEEE/ACM SC19

Slingshot the Exascale Interconnect


An In-Depth Analysis of the Slingshot Interconnect

Daniele De Sensi
Department of Computer Science
ETH Zurich
ddesensi@ethz.ch

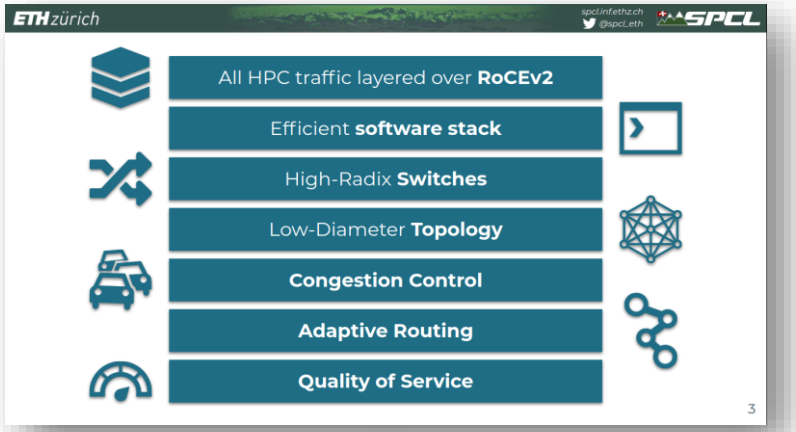
Salvatore Di Girolamo
Department of Computer Science
ETH Zurich
salvatore.digirolamo@inf.ethz.ch

Kim H. McMahon
Hewlett Packard Enterprise
kim.mcmahon@hpe.com

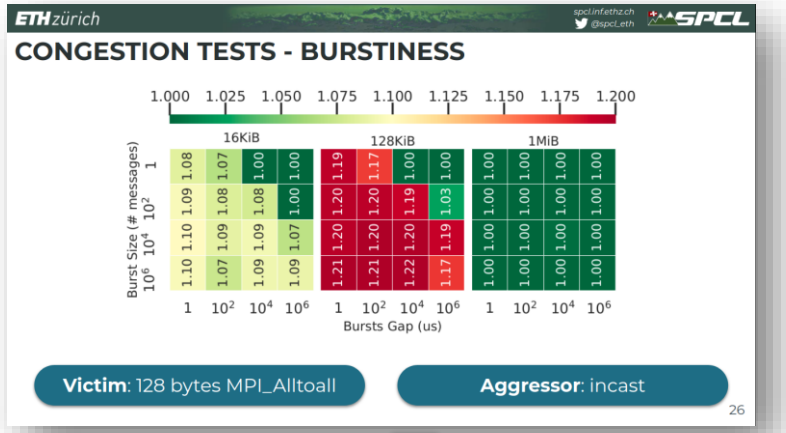
Duncan Roweth
Hewlett Packard Enterprise
duncan.roweth@hpe.com

Torsten Hoefler
Department of Computer Science
ETH Zurich
torsten.hoefler@inf.ethz.ch

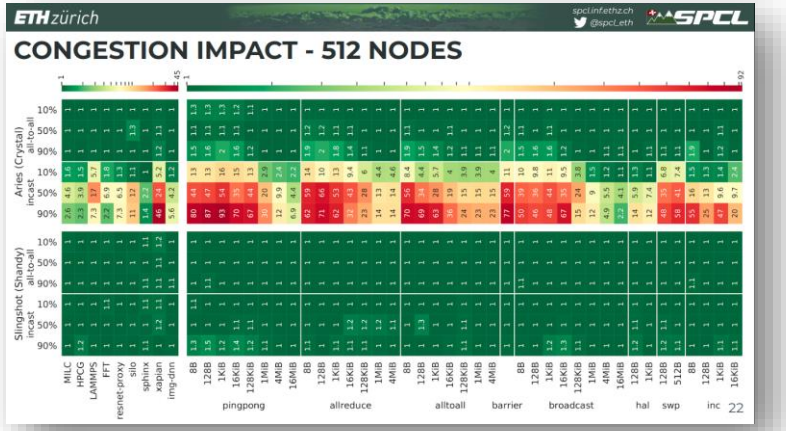
Description of the main features of the interconnect



In-depth benchmarking procedure that can be ported to other interconnect

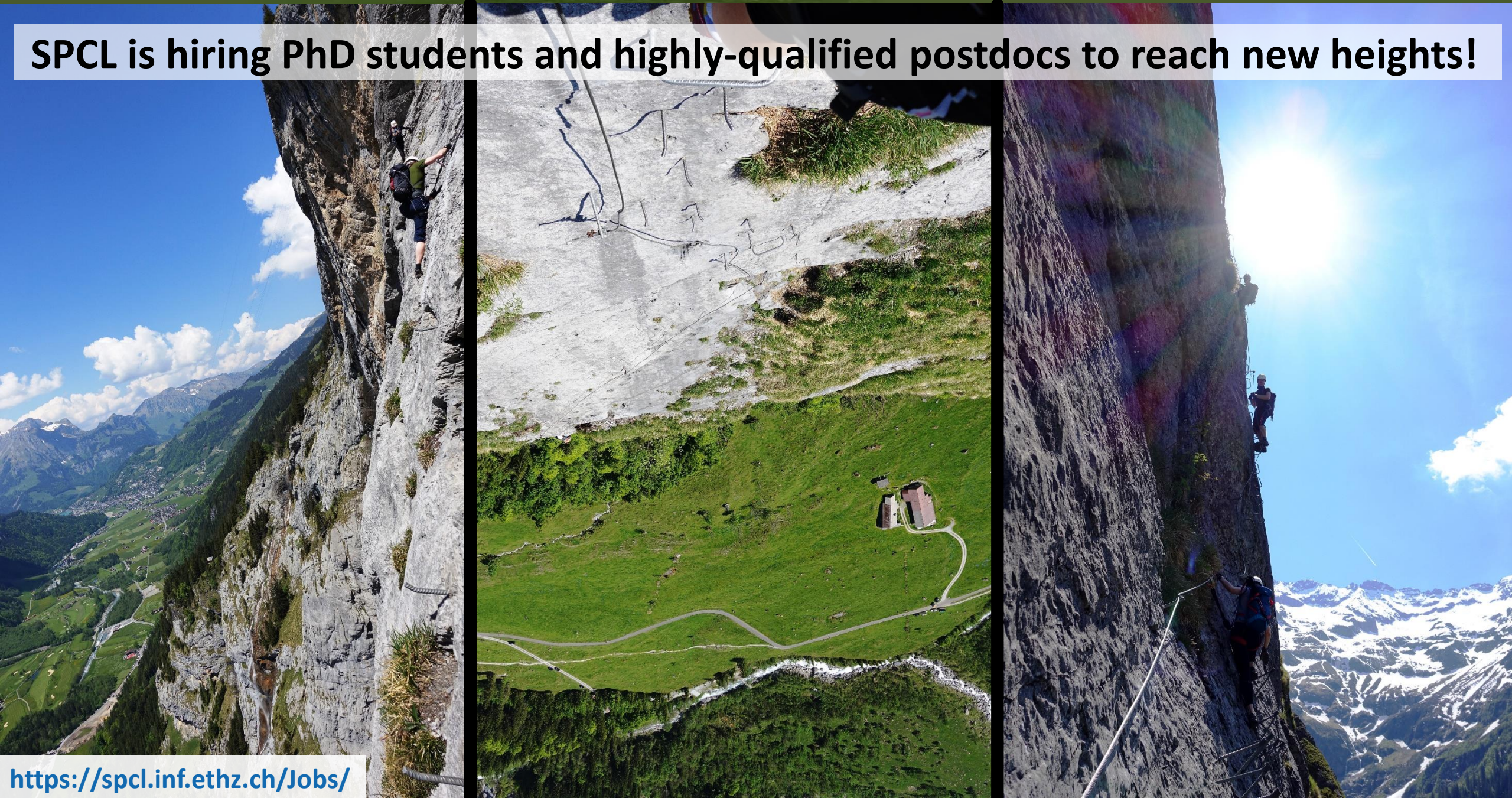


Detailed results on performance, congestion control, and quality of service, on microbenchmarks, HPC, and DC applications



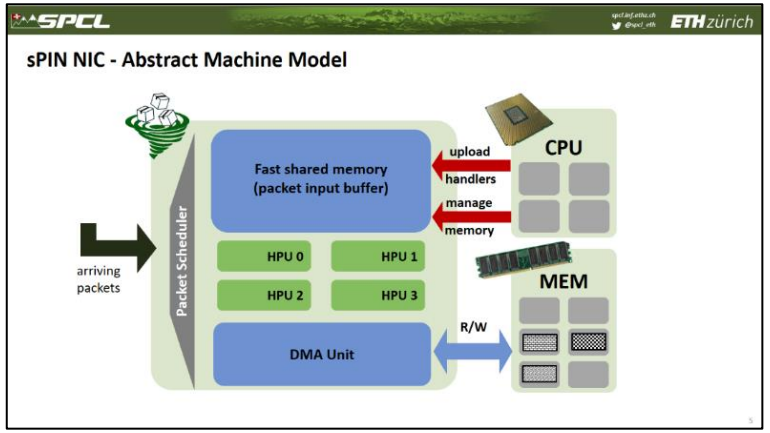
Daniele De Sensi et al.: "An In-Depth Analysis of the Slingshot Interconnect", IEEE/ACM SC20

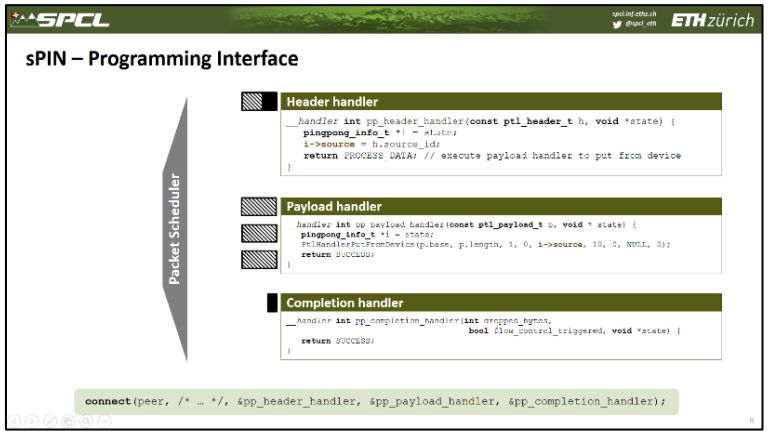
SPCL is hiring PhD students and highly-qualified postdocs to reach new heights!



https://spcl.inf.ethz.ch/Jobs/

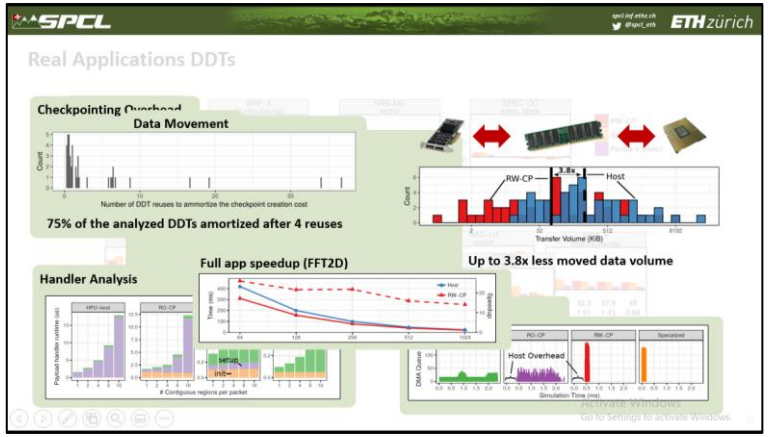
sPIN *Streaming Processing in the Network* for Network Acceleration

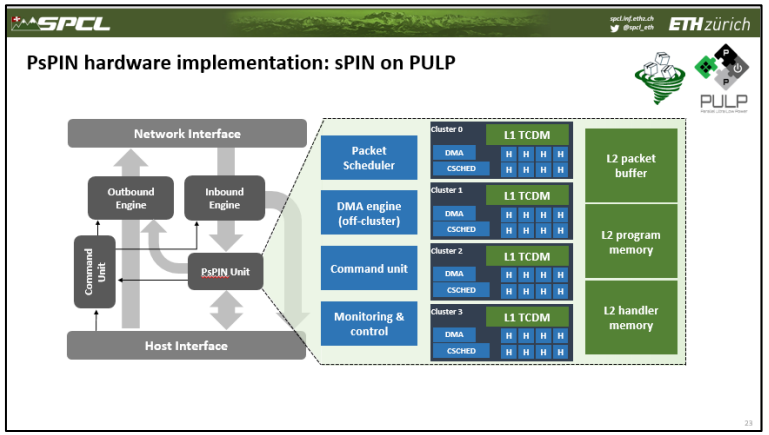




sPIN

beyond RDMA





Full specification: https://arxiv.org/abs/1709.05483

Try it out: https://spcl.inf.ethz.ch/Research/Parallel_Programming/sPIN/

Backup Slides

But why PULP/RISC-V?

- RISC-V is an open source ISA
  - Allows and supports extensions

    Doing this in ARM may be complex and expensive

- PULP aims to provide high performance per Watt
  - Energy efficient
  - Provides tight control over compute and data movement schedule
  - Fits well the sPIN abstract machine model (e.g., removing cache coherency on ARM could be painful)
  - PULP is actively researched + we can leverage ISS group expertise at ETH