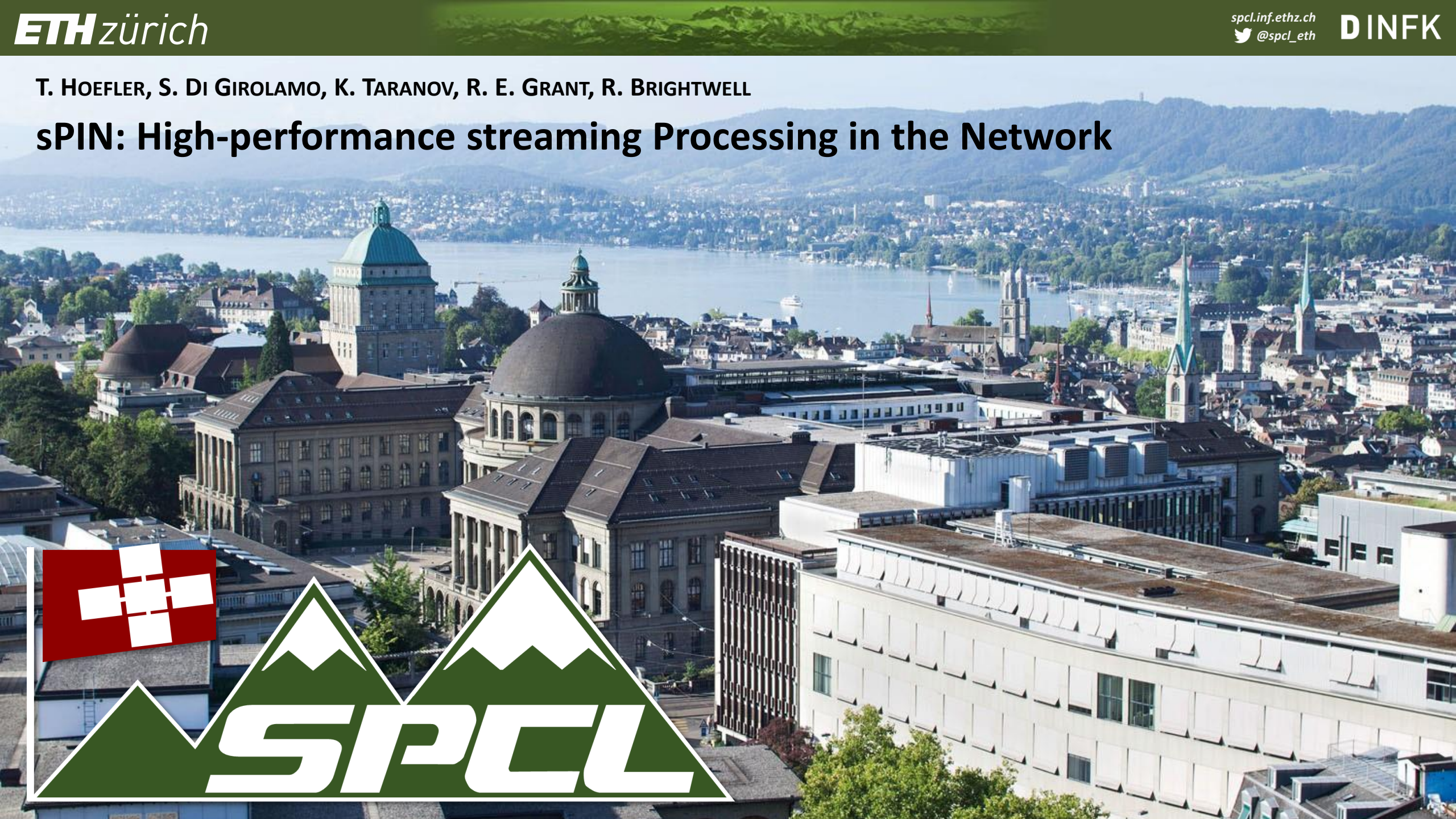
T. Hoefler, S. Di Girolamo, K. Taranov, R. E. Grant, R. Brightwell

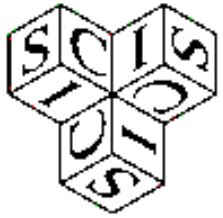
sPIN: High-performance streaming Processing in the Network



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

InfiniBand Trade Association Launches the RoCE Initiative to Advance RDMA over Converged Ethernet Solutions

RoCE™

RoCE delivers significant performance and efficiency gains to cloud, storage, virtualization and hyper-converged infrastructures

businessinsider.com

Microsoft to Drive RDMA Into Datacenters and Clouds

November 18, 2013 by Timothy Prickett Morgan

RDMA over Ethernet - the Rocky road to convergence

17 November 2015 | By Brandon Hoff

TOP500® SUPERCOMPUTER SITES

June 2017

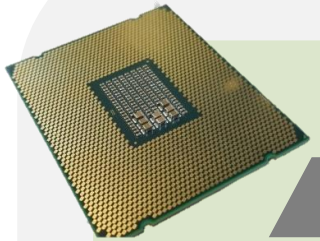
95 / top-100 systems use RDMA

>285 / top-500 systems use RDMA

Data Processing in modern RDMA networks

Remote Nodes (via network)

Local Node



Core i7 Haswell

Regs	Regs
4 cycles ↕ ~1.3ns	4 cycles ↕ ~1.3ns
L1	L1
11 cycles ↕ ~ 3.6ns	11 cycles ↕ ~ 3.6ns
L2	L2
34 cycles ↕ ~11.3ns	34 cycles ↕ ~11.3ns

L3

125 cycles ↕ ~41.6ns

Main Memory

Input buffer

RDMA NIC

RDMA Processing

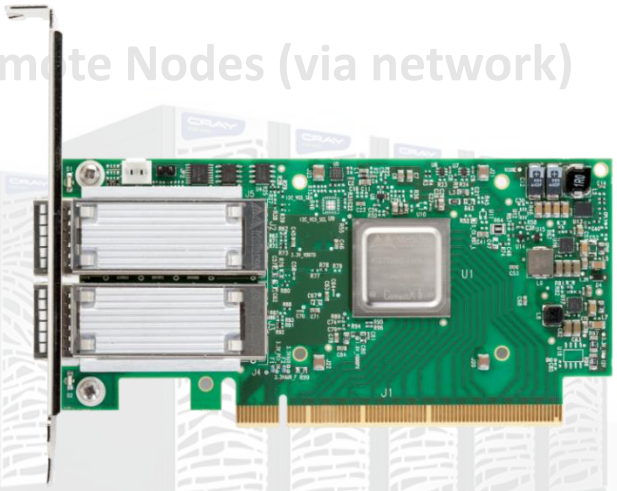
DMA Unit

PCIe bus

~ 250ns

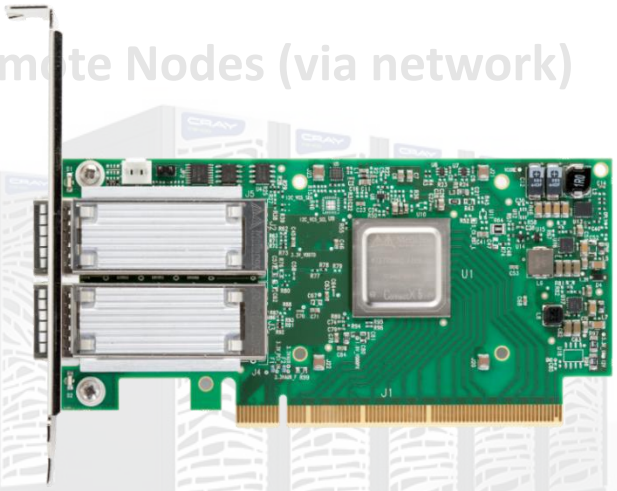
arriving packets

Data Processing in modern RDMA networks

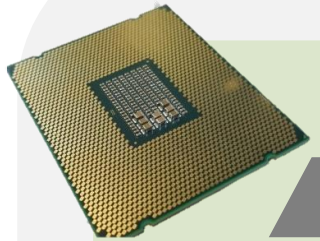


Mellanox Connect-X5: 1 msg/5ns
Tomorrow (400G): 1 msg/1.2ns

Data Processing in modern RDMA networks



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



Local Node

Core i7 Haswell

Regs
4 cycles ↕ ~1.3ns
L1
11 cycles ↕ ~ 3.6ns
L2
34 cycles ↕ ~11.3ns

Regs
4 cycles ↕ ~1.3ns
L1
11 cycles ↕ ~ 3.6ns
L2
34 cycles ↕ ~11.3ns

L3

125 cycles ~41.6ns

Main Memory

Input buffer

PCIe bus
~ 250ns

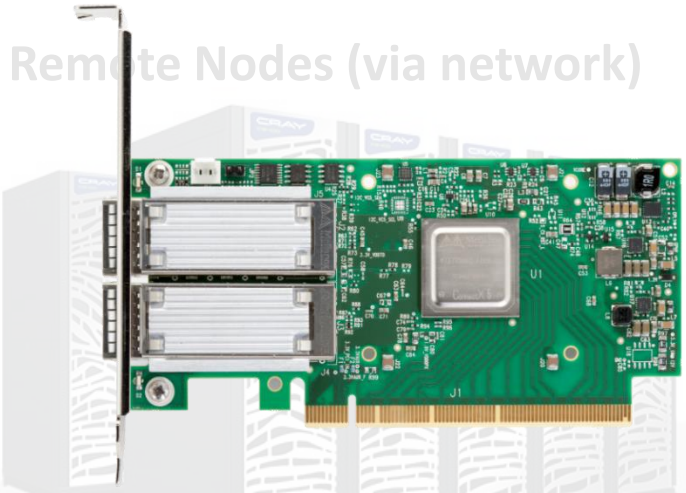
RDMA NIC

RDMA Processing

DMA Unit

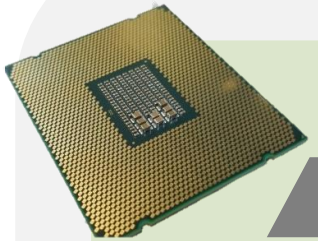
arriving packets

Data Processing in modern RDMA networks



Mellanox Connect-X5: 1 msg/5ns

Tomorrow (400G): 1 msg/1.2ns



Local Node

Core i7 Haswell

Regs
4 cycles ↕ ~1.3ns
L1
11 cycles ↕ ~ 3.6ns
L2
34 cycles ↕ ~11.3ns
L3

Regs
4 cycles ↕ ~1.3ns
L1
11 cycles ↕ ~ 3.6ns
L2
34 cycles ↕ ~11.3ns

125 cycles ↕ ~41.6ns

RDMA NIC

?

PCIe bus

~ 250ns

Main Memory

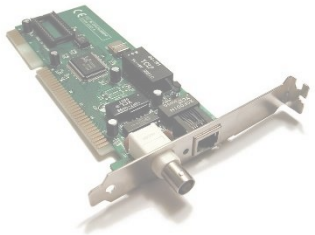
Input buffer

arriving packets

The future 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

Established Principles for Compute Acceleration

Specialization
Programmability
Libraries
Ease-of-use
Portability
Efficiency

NVIDIA CUDA
OpenCL
OpenMP 4.0

June 2017

TOP500 SUPERCOMPUTER SITES

95 / top-100 systems use RDMA

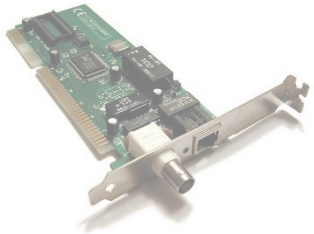
>285 / top-500 systems use RDMA

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

fully programmable NIC acceleration

Established Principles for Compute Acceleration

Specialization · Programmability · Libraries

Ease-of-use · Portability · Efficiency

NVIDIA CUDA · OpenCL · OpenMP 4.0

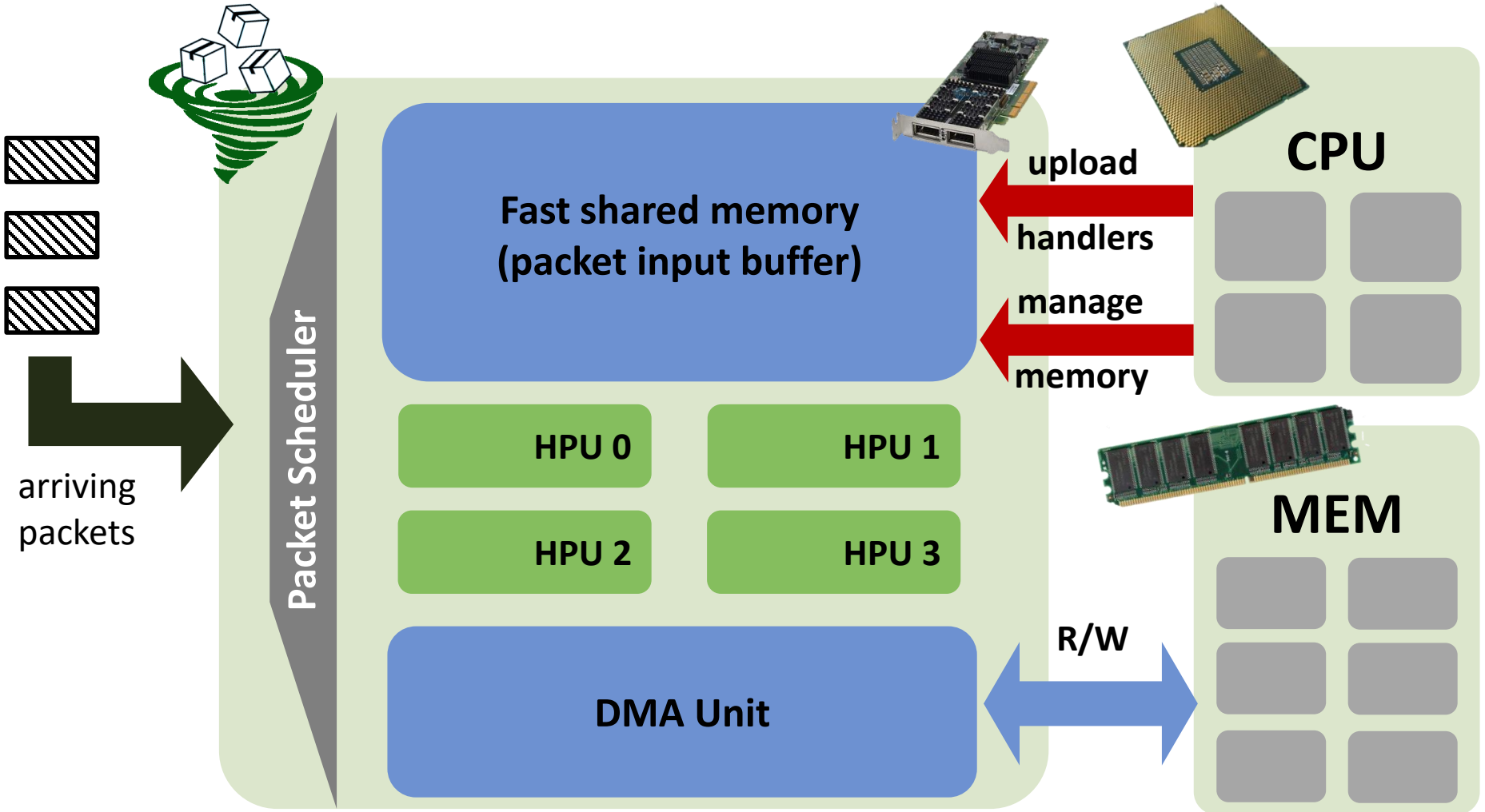
June 2017

TOP500® SUPERCOMPUTER SITES

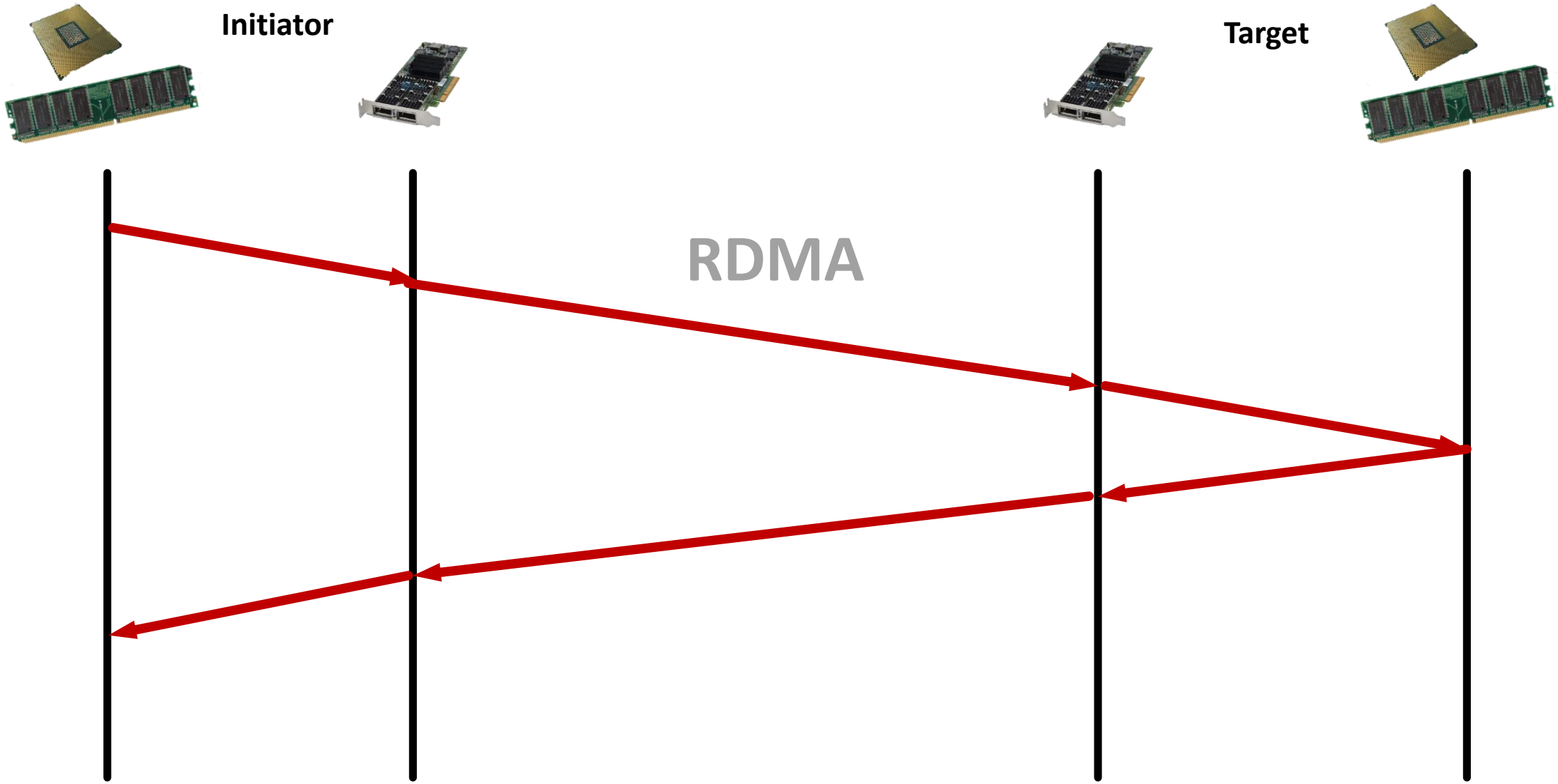
95 / top-100 systems use RDMA

>285 / top-500 systems use RDMA

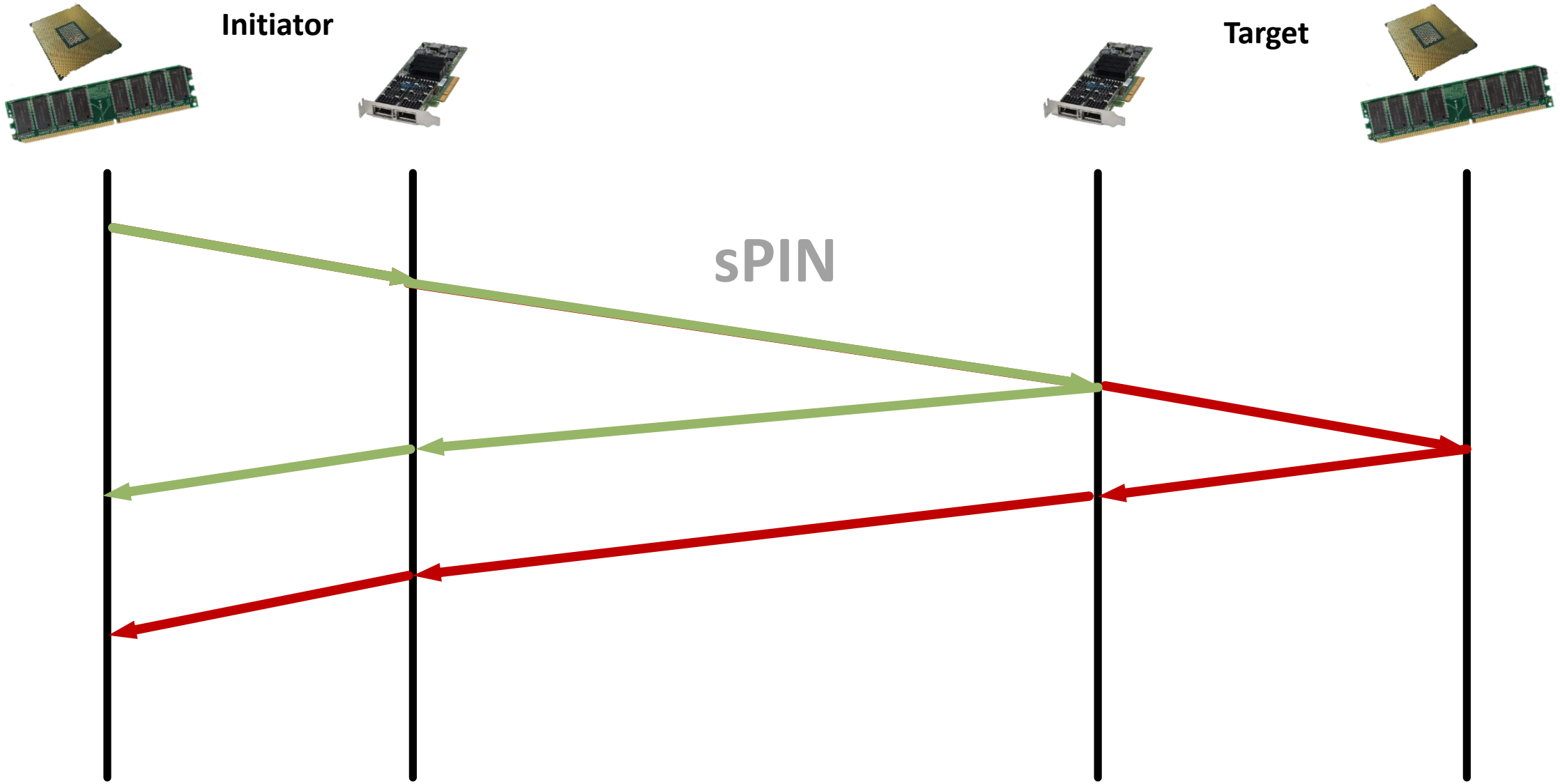
sPIN NIC - Abstract Machine Model



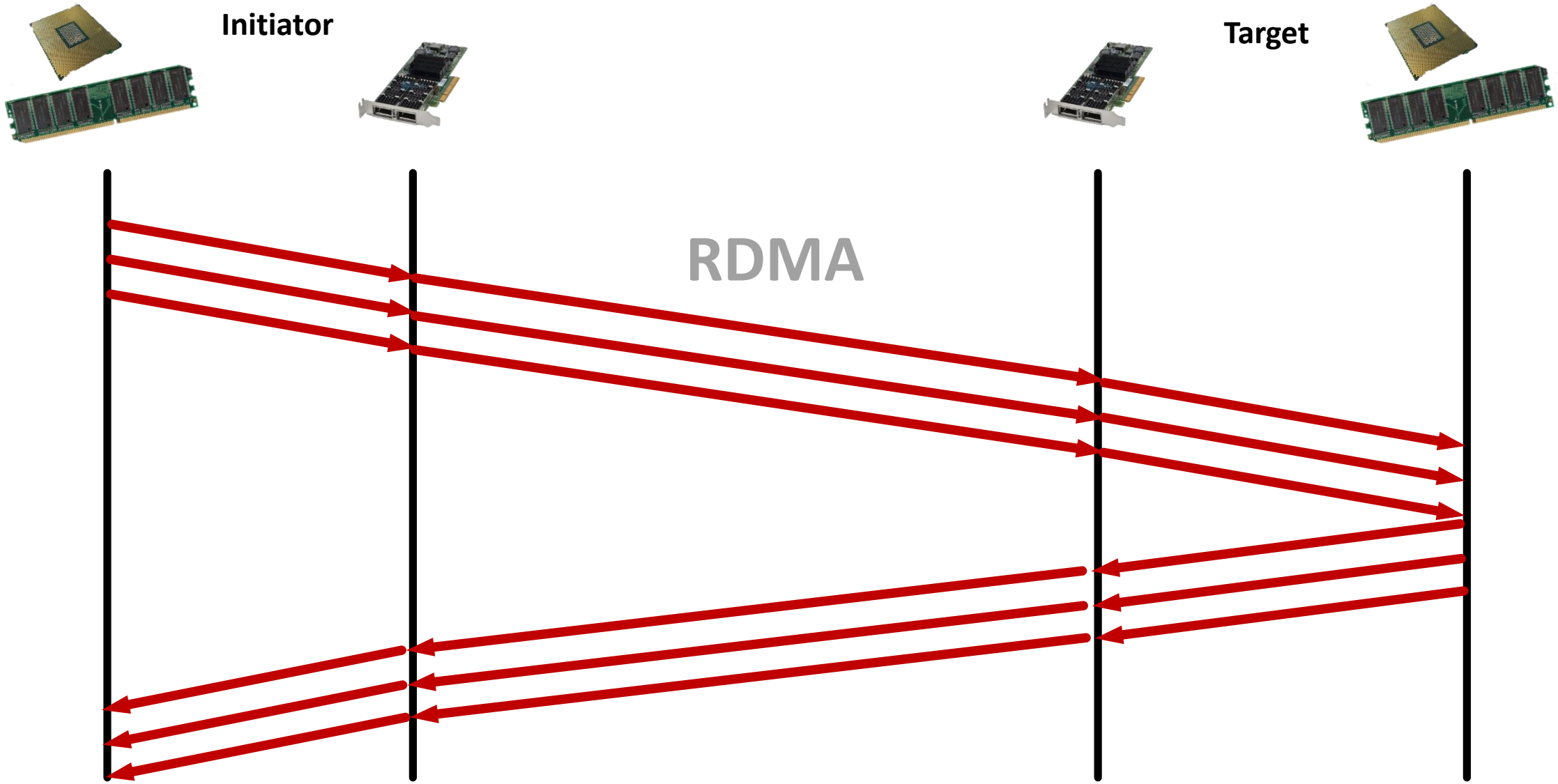
RDMA vs. sPIN in action: Simple Ping Pong



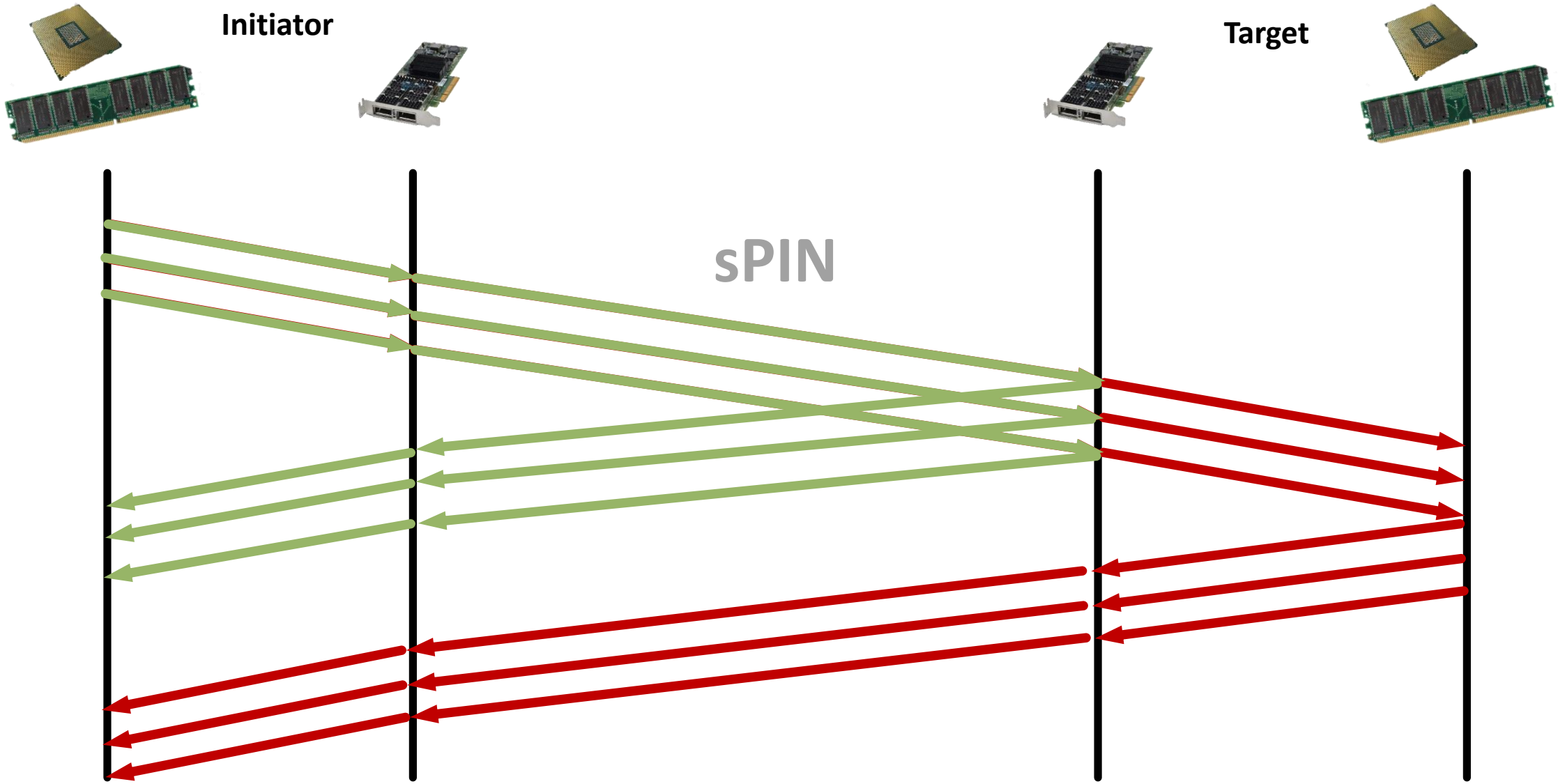
RDMA vs. sPIN in action: Simple Ping Pong



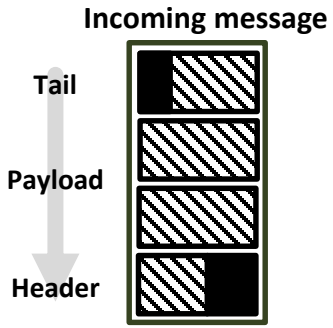
RDMA vs. sPIN in action: Streaming Ping Pong



RDMA vs. sPIN in action: Streaming Ping Pong



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

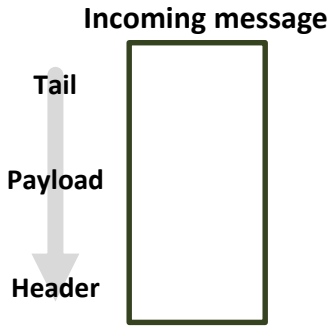
```
__handler int pp_payload_handler(const ptl_payload_t p, void * state) {
    pingpong_info_t *i = state;
    PtlHandlerPutFromDevice(p.base, p.length, 1, 0, i->source, 10, 0, NULL, 0);
    return SUCCESS;
}
```

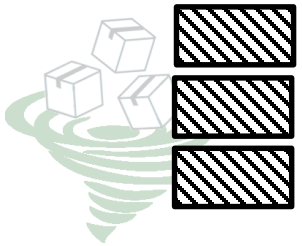
Completion handler

```
__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);
```

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);
```

Possible sPIN implementations

- **sPIN is a programming abstraction, similar to CUDA or OpenCL combined with OFED or Portals 4**
  - It enables a large variety of NIC implementations!
  - For example, massively multithreaded HPUs

    Including warp-like scheduling strategies

at 400G, process more than 833 million messages/s

- **Main goal: sPIN must not obstruct line-rate**
  - Programmer must limit processing time per packet

    Little's Law: 500 instructions per handler, 2.5 GHz, IPC=1, 1 Tb/s → 25 kiB memory
  - Relies on fast shared memory (processing in packet buffers)

    Scratchpad or registers
  - Quick (single-cycle) handler invocation on packet arrival

    Pre-initialized memory & context
- **Can be implemented in most RDMA NICs with a firmware update**
  - Or in software in programmable (Smart) NICs

Mellanox TECHNOLOGIES
Innova Flex
(Kintex FPGA)

Microsoft
Catapult
(Virtex FPGA)

BROADCOM
BCM58800 SoC
(Full Linux)

Simulating a sPIN NIC – Ping Pong

- **LogGOPSim v2 [1]: combine LogGOPSim (packet-level network) with gem5 (cycle accurate CPU simulation)**

- **Network (LogGOPSim):**
  - Supports Portals 4 and MPI
  - Parametrized for future InfiniBand

    o=65ns (measured)

    g=6.7ns (150 MM/s)

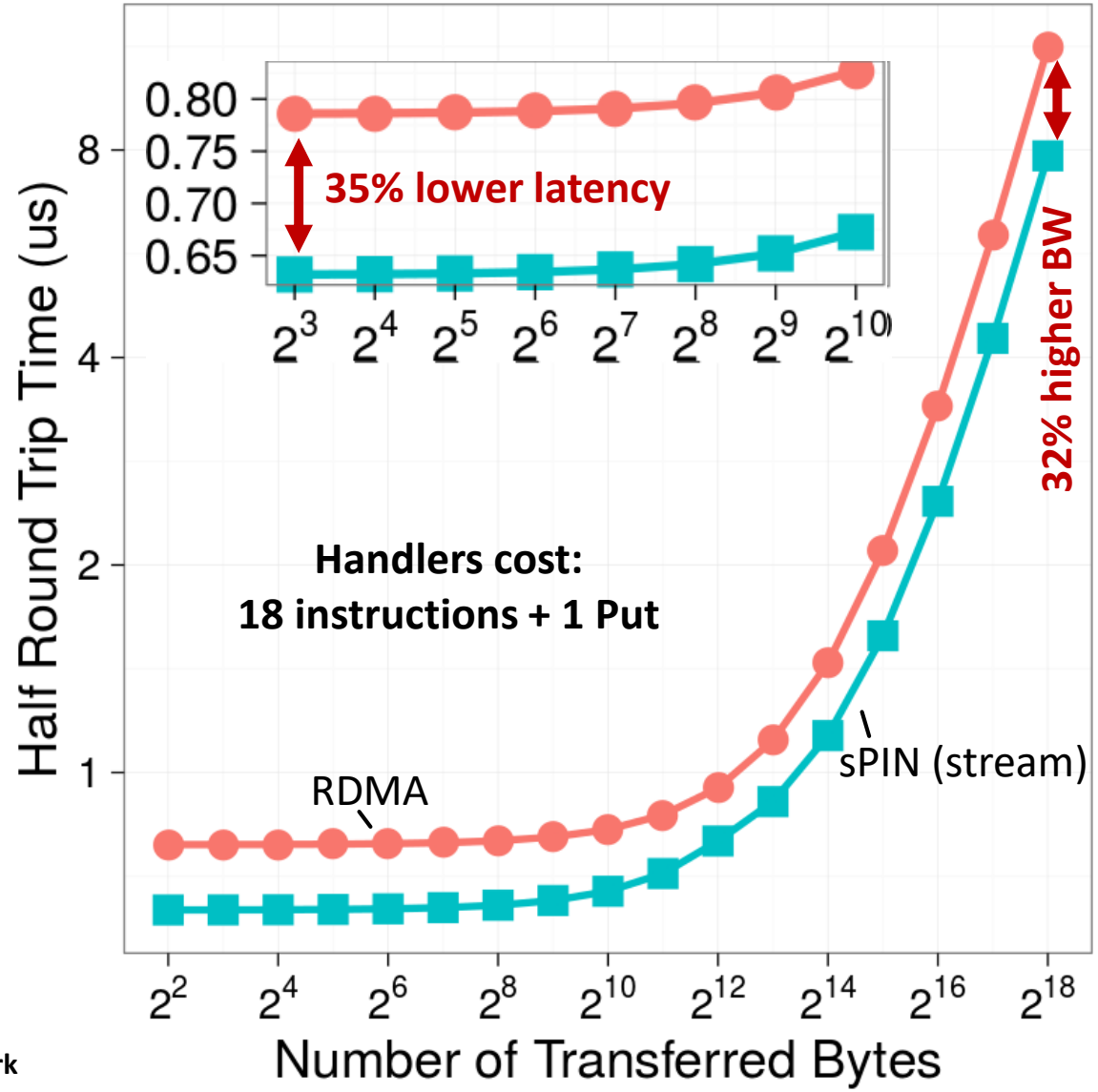
    G=2.5ps (400 Gib/s)

    Switch L=50ns (measured)

    Wire L=33.4ns (10m cable)

- **NIC HPU**
  - 2.5 GHz ARM Cortex A15 OOO
  - ARMv8-A 32 bit ISA
  - Single-cycle access SRAM (no DRAM)
  - Header matching m=30ns, per packet 2ns

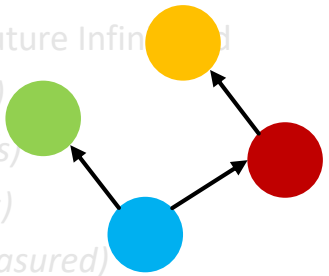
    In parallel with g!



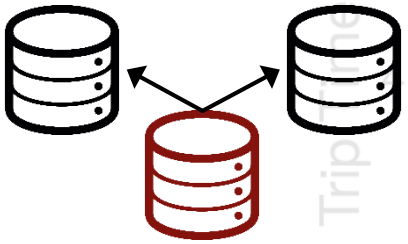
[1] S. Di Girolamo, K. Taranov, T. Schneider, E. Stalder, T. Hoefler, **LogGOPSim+gem5: Simulating Network Offload Engines Over Packet-Switched Networks**. Presented at ExaMPI'17

Simulating a sPIN NIC – Ping Pong

- LogGOPSim v2 [1]: combine LogGOPSim (packet-level network) with gem5 (cycle accurate CPU simulation)
- Network (LogGOPSim):
  - Supports Portals 4 and MPI
  - Parametrized for future Infiniband
    - *o=65ns (measured)*
    - *g=6.7ns (150 MM/s)*
    - *G=2.5ps (400 Gib/s)*
    - *Switch L=50ns (measured)*
    - *Wire L=33.4ns (10m cable)*
- NIC HPU
  - 2.5 GHz ARM Cortex A15 OOO
  - ARMv8-A 32 bit ISA
  - Single-cycle access SRAM (no DRAM)
  - Header matching m=30ns, per packet 2ns
    - *In parallel with g!*



Network Group Communication

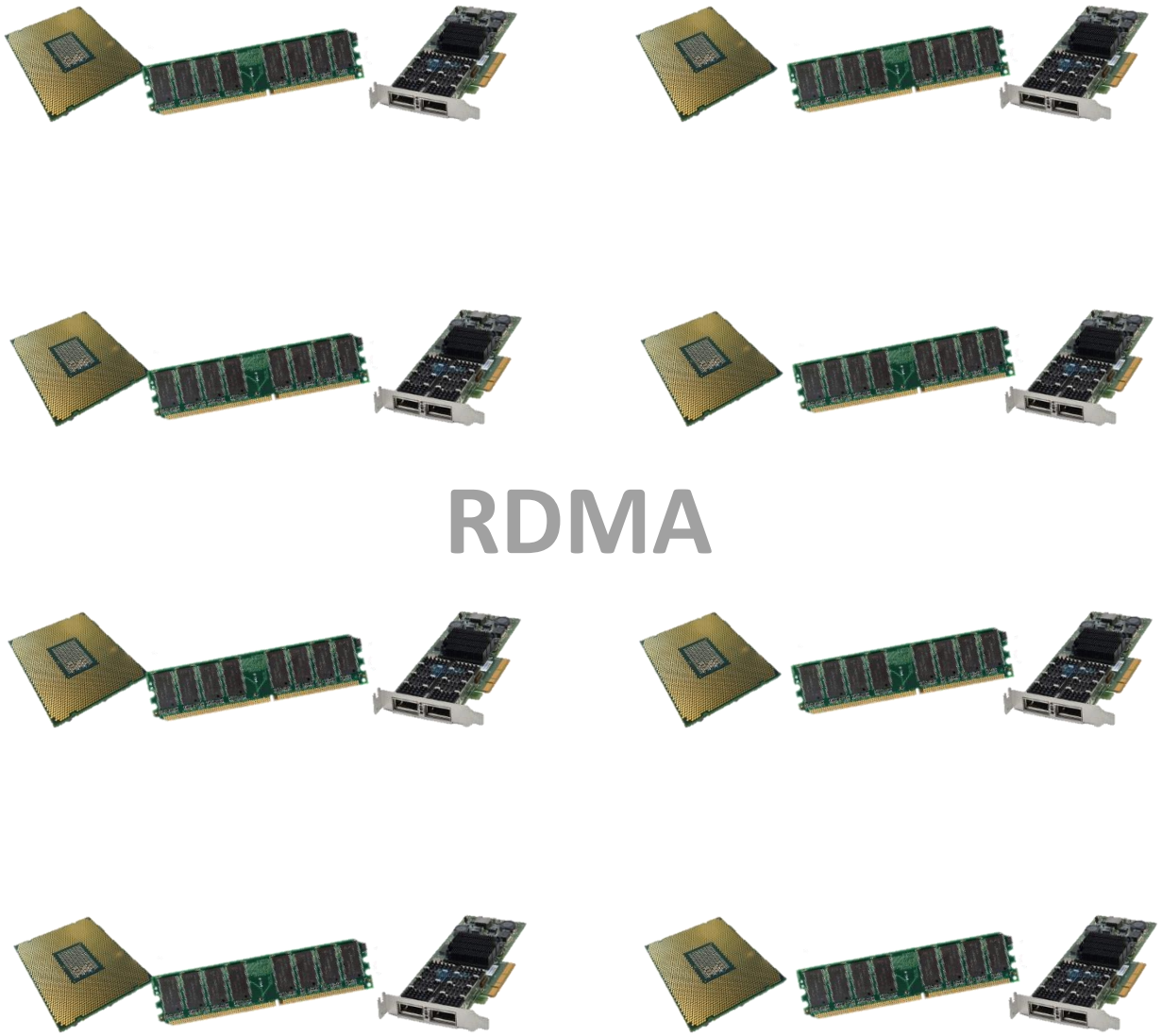


Distributed Data Management

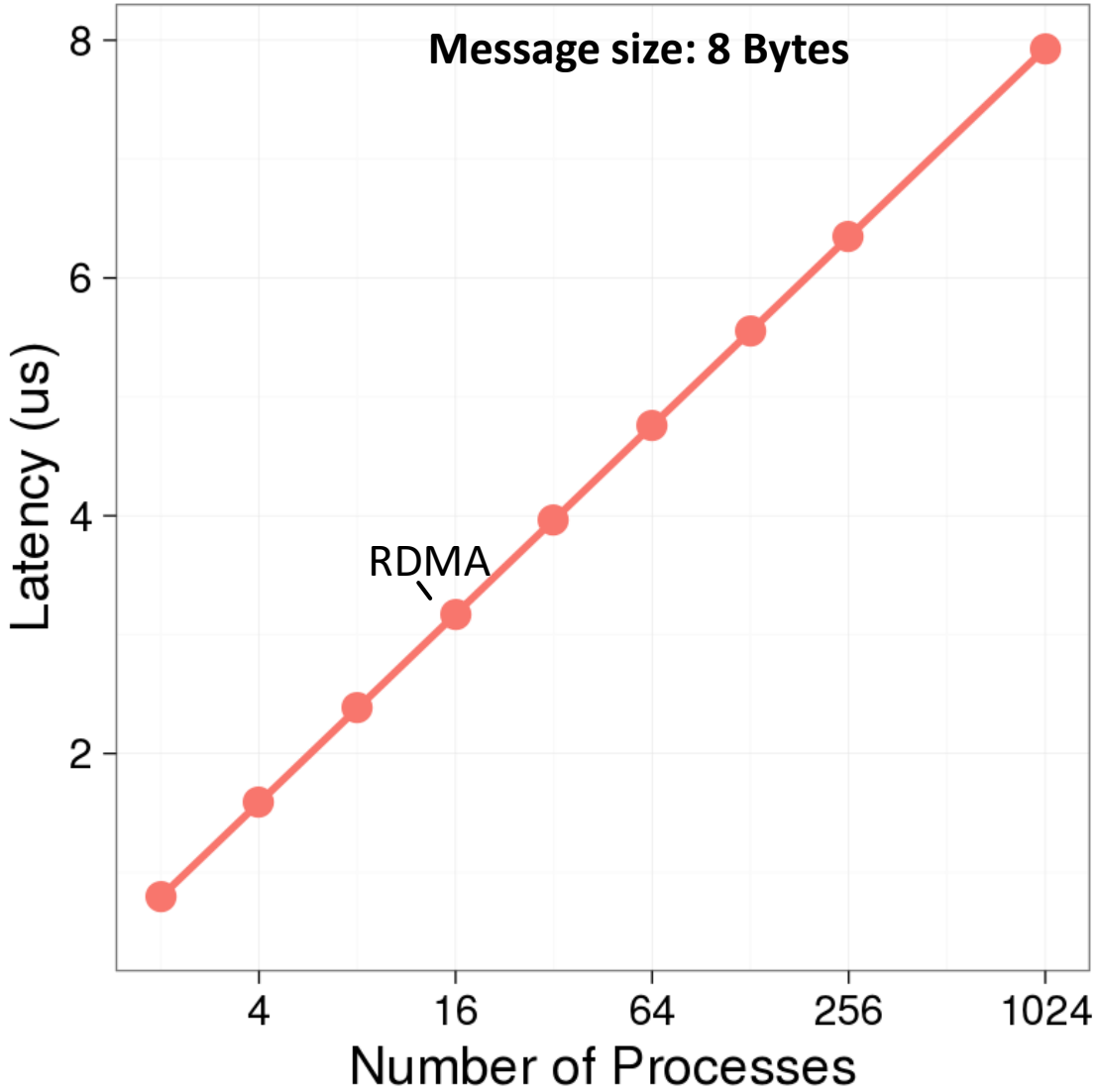
Data Layout Transformation

[1] S. Di Girolamo, K. Taranov, T. Schneider, E. Stalder, T. Hoefler, **LogGOPSim+gem5: Simulating Network Offload Engines Over Packet-Switched Networks**. Presented at ExaMPI'17

Use Case 1: Broadcast acceleration



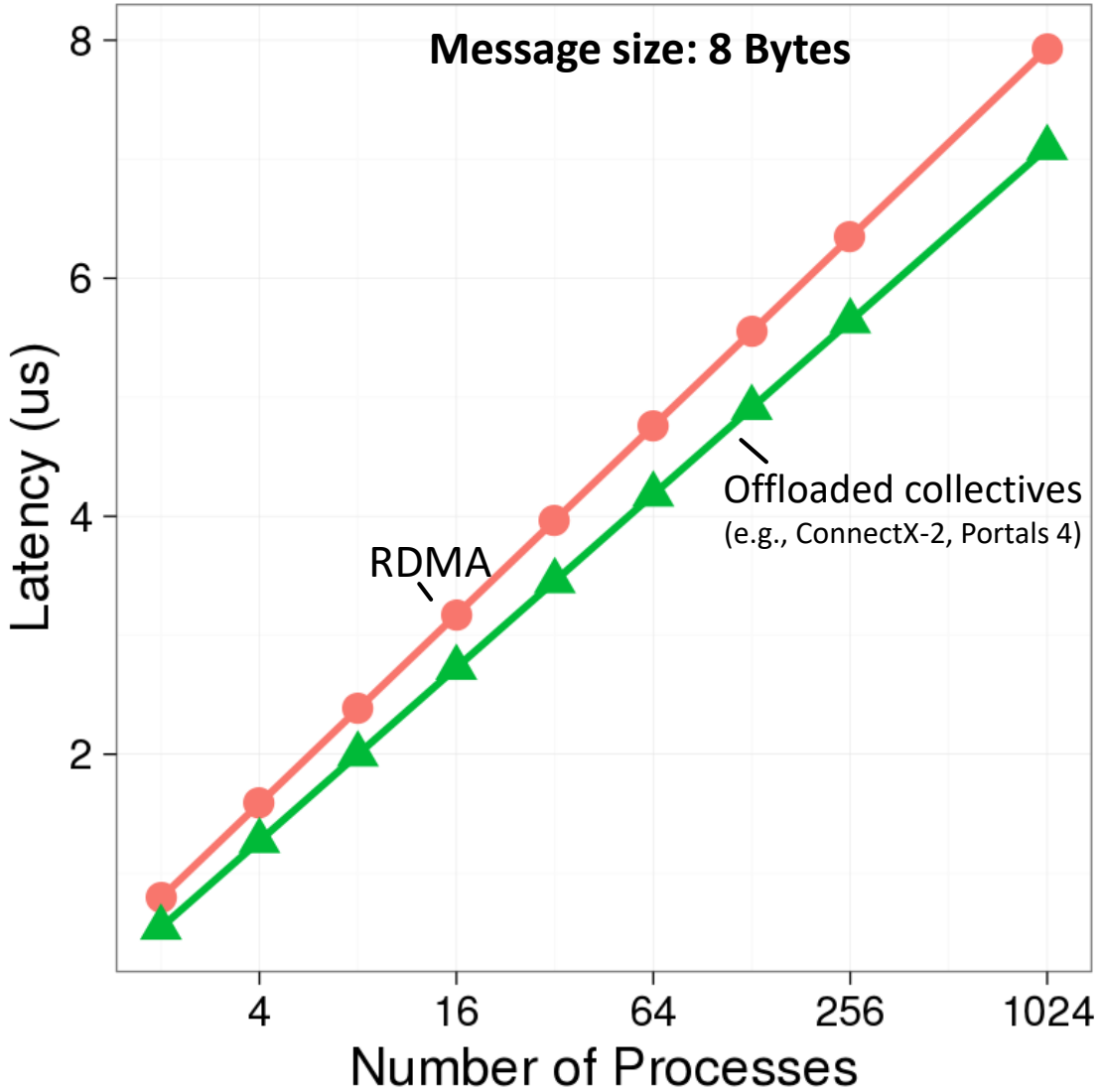
Network Group Communication



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

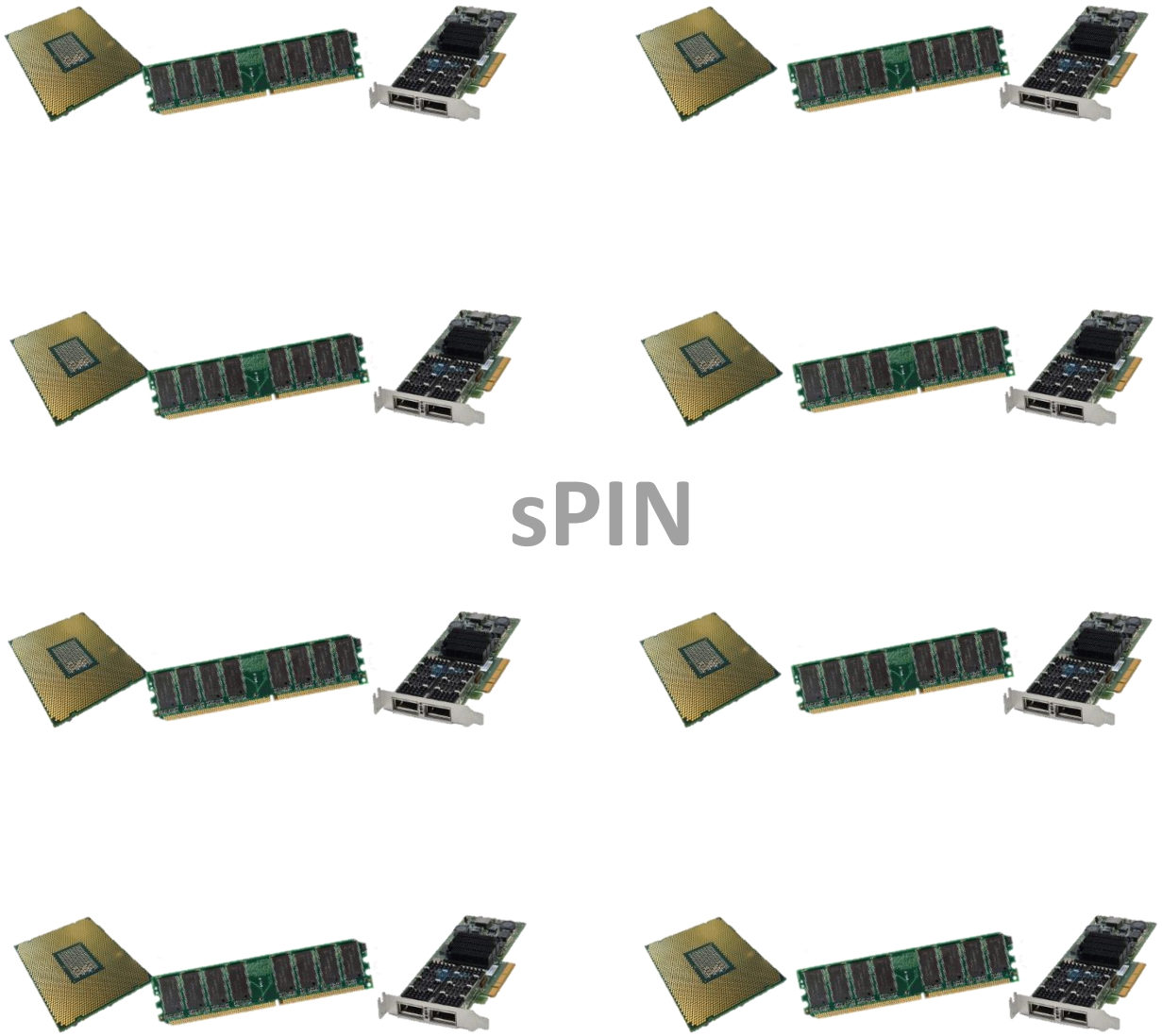
Use Case 1: Broadcast acceleration

Network Group Communication

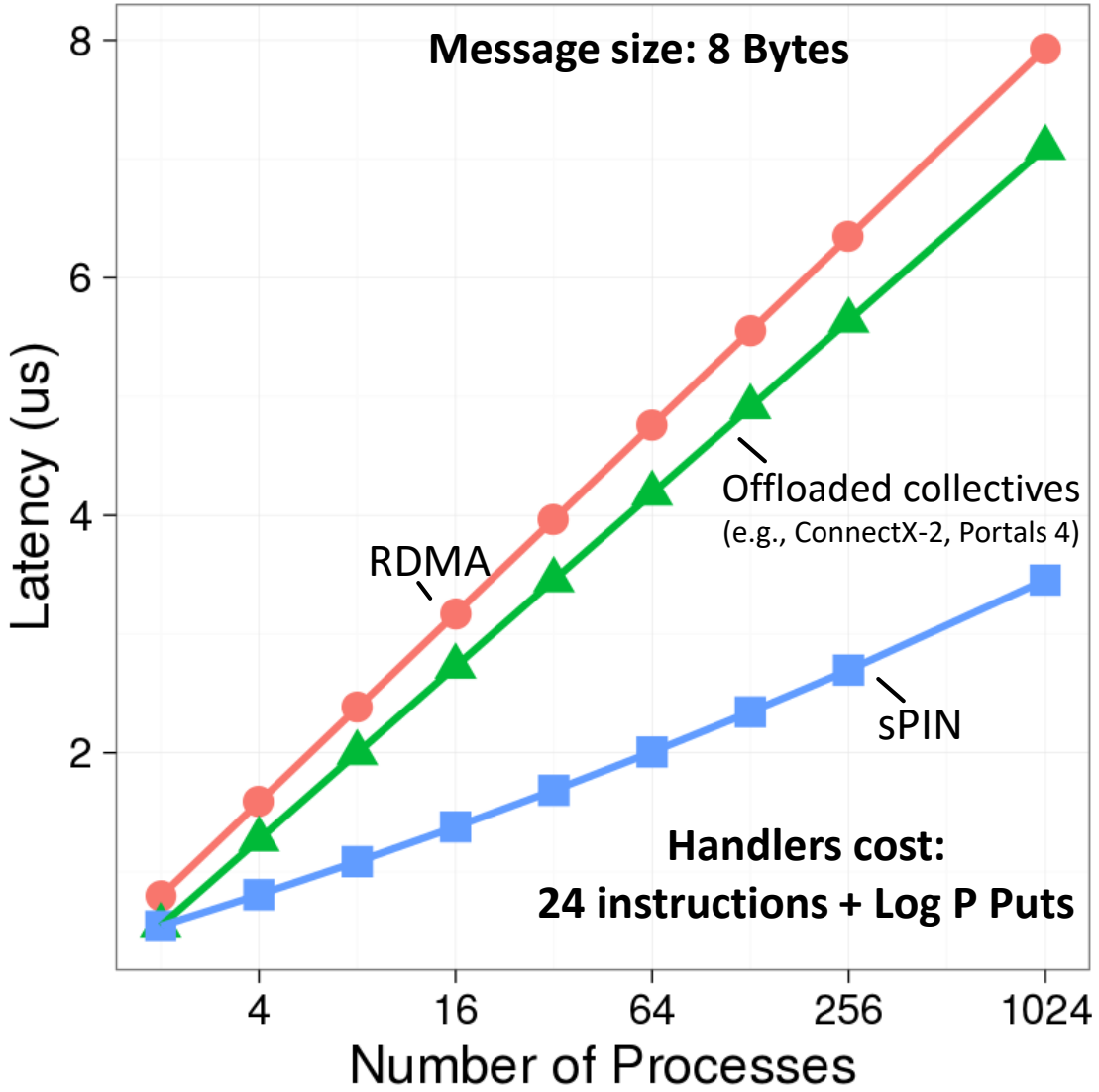


Underwood, K.D., et al., Enabling flexible collective communication offload with triggered operations. *HOTI'11*
Liu, J., et al., High performance RDMA-based MPI implementation over InfiniBand. *International Journal of Parallel Programming* 2004

Use Case 1: Broadcast acceleration



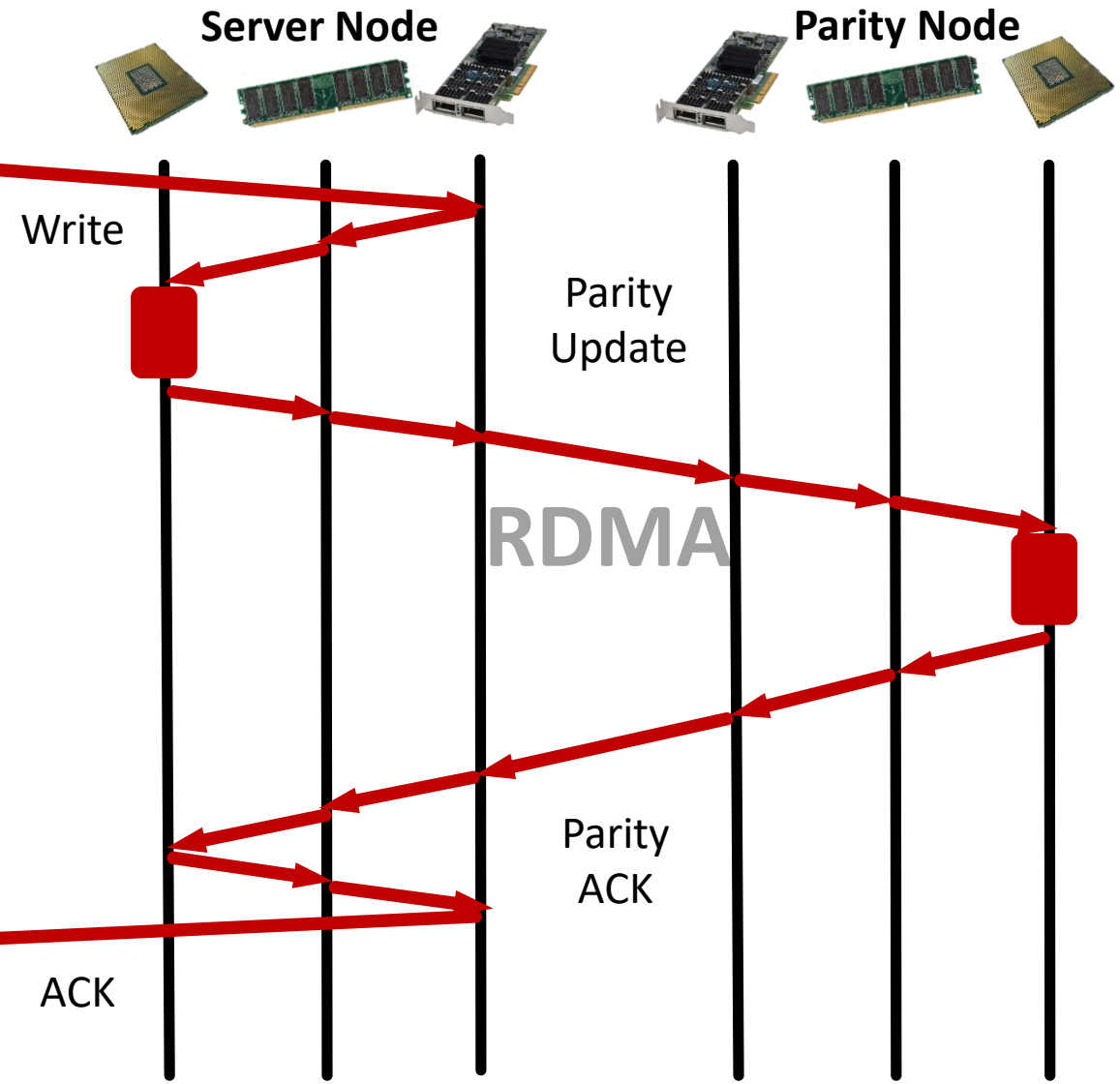
Network Group Communication

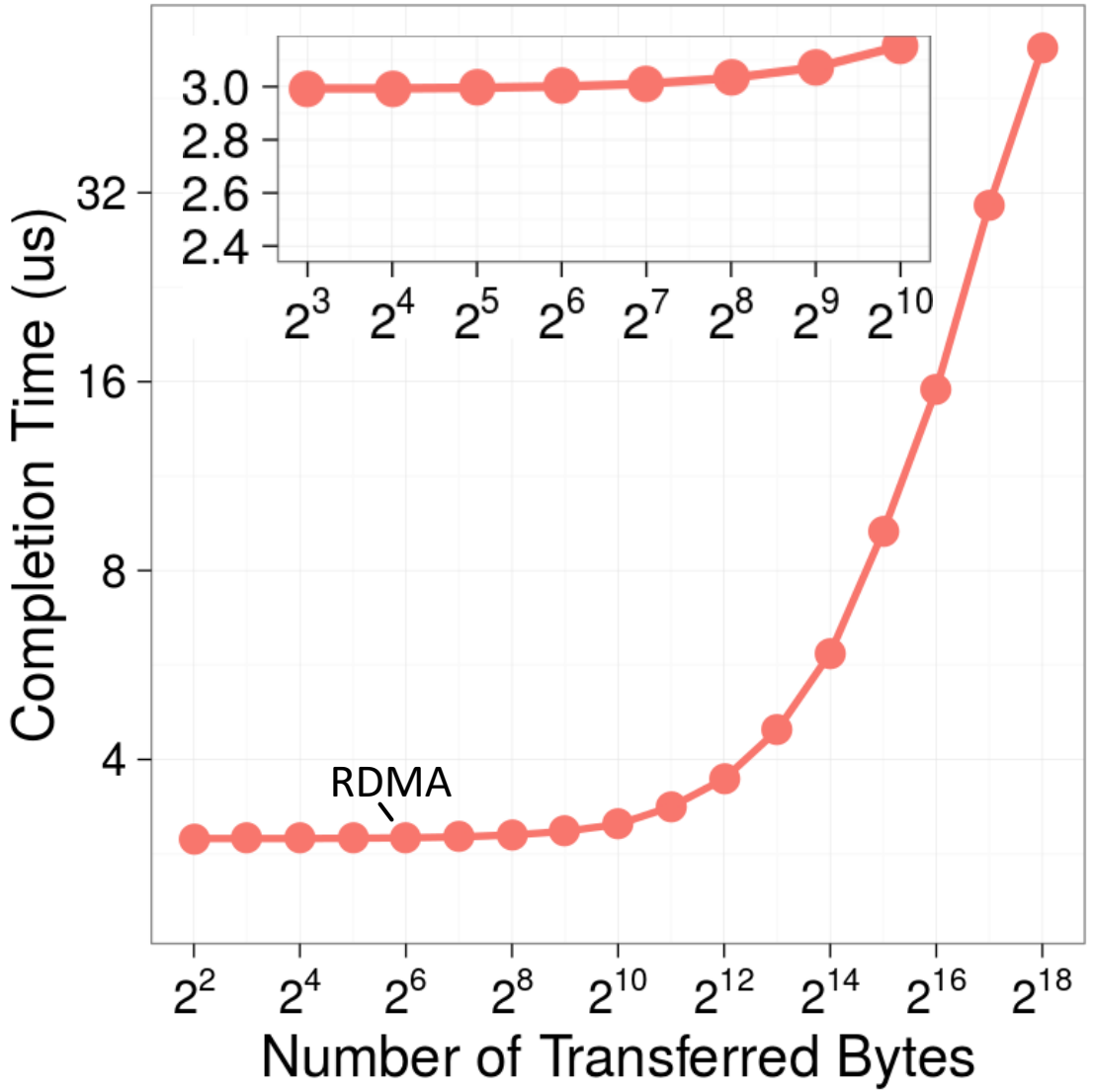


Underwood, K.D., et al., Enabling flexible collective communication offload with triggered operations. *HOTI'11*
Liu, J., et al., High performance RDMA-based MPI implementation over InfiniBand. *International Journal of Parallel Programming* 2004

Use Case 2: RAID acceleration

Distributed Data Management

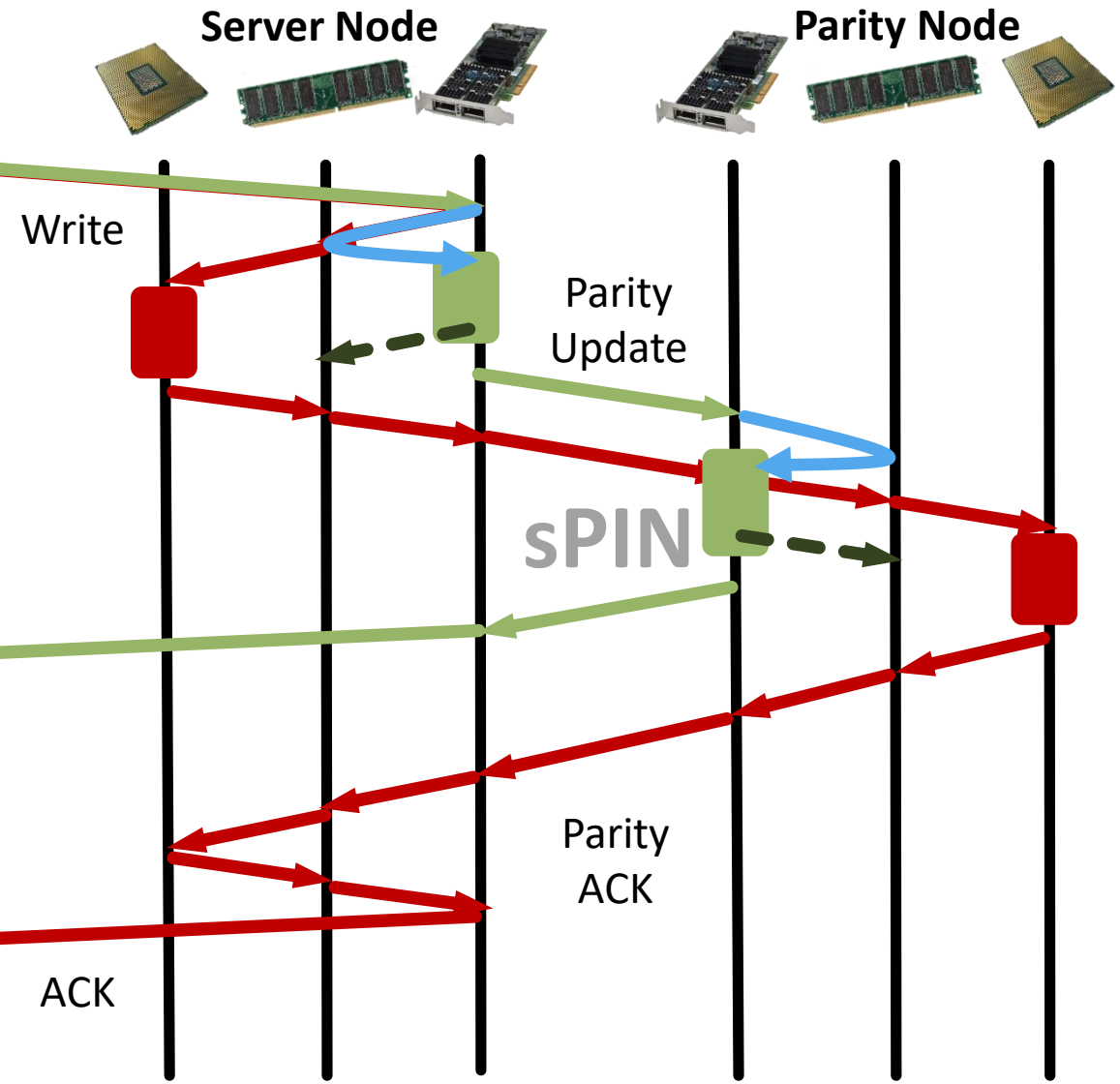


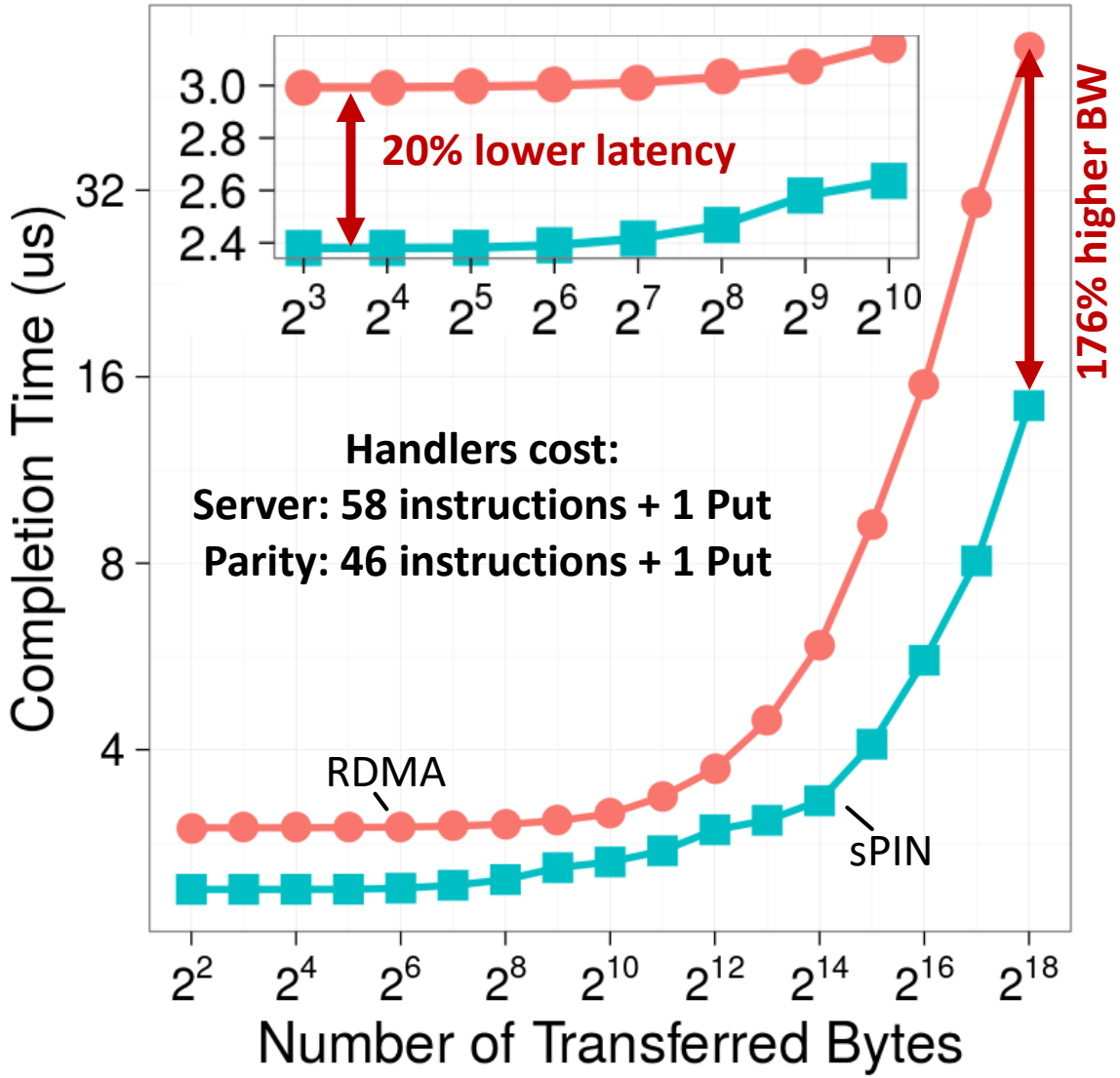


Shankar D. et al., High-performance and Resilient Key-Value Store with Online Erasure Coding for Big Data Workloads. ICDCS'17

Use Case 2: RAID acceleration

Distributed Data Management

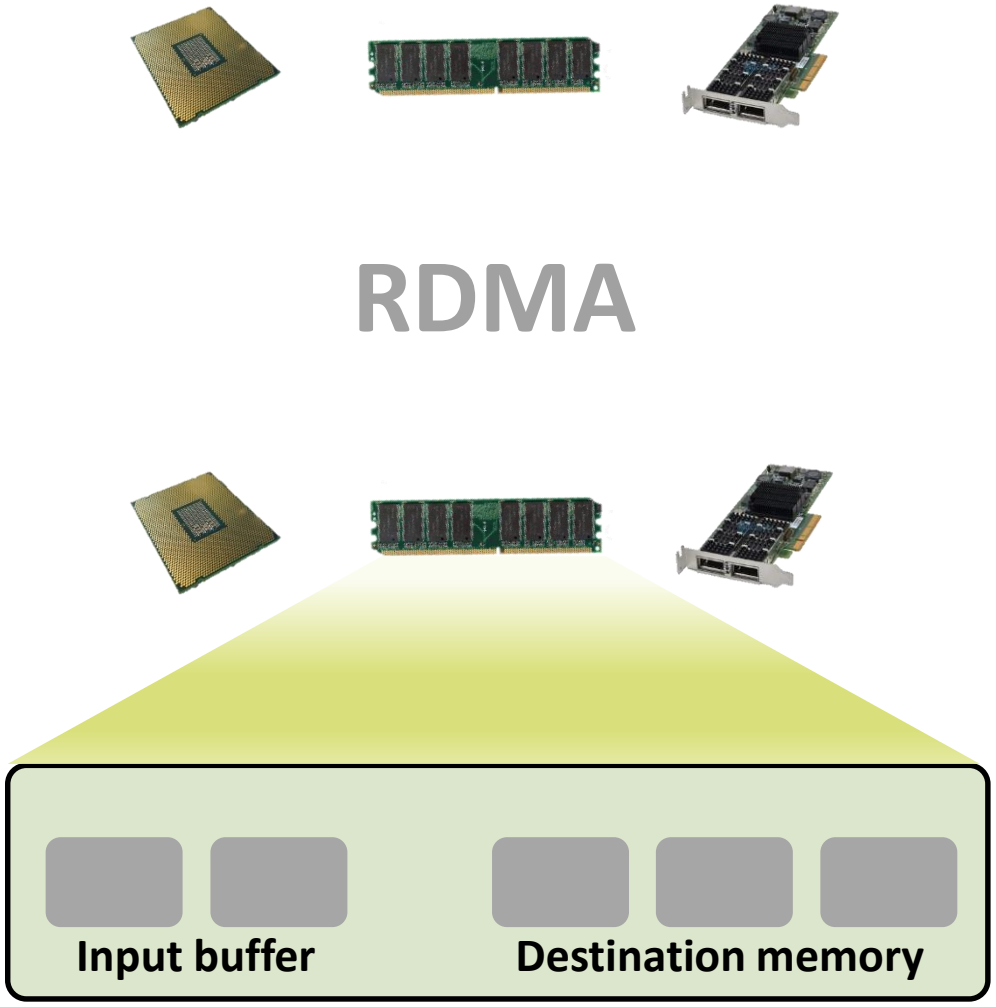


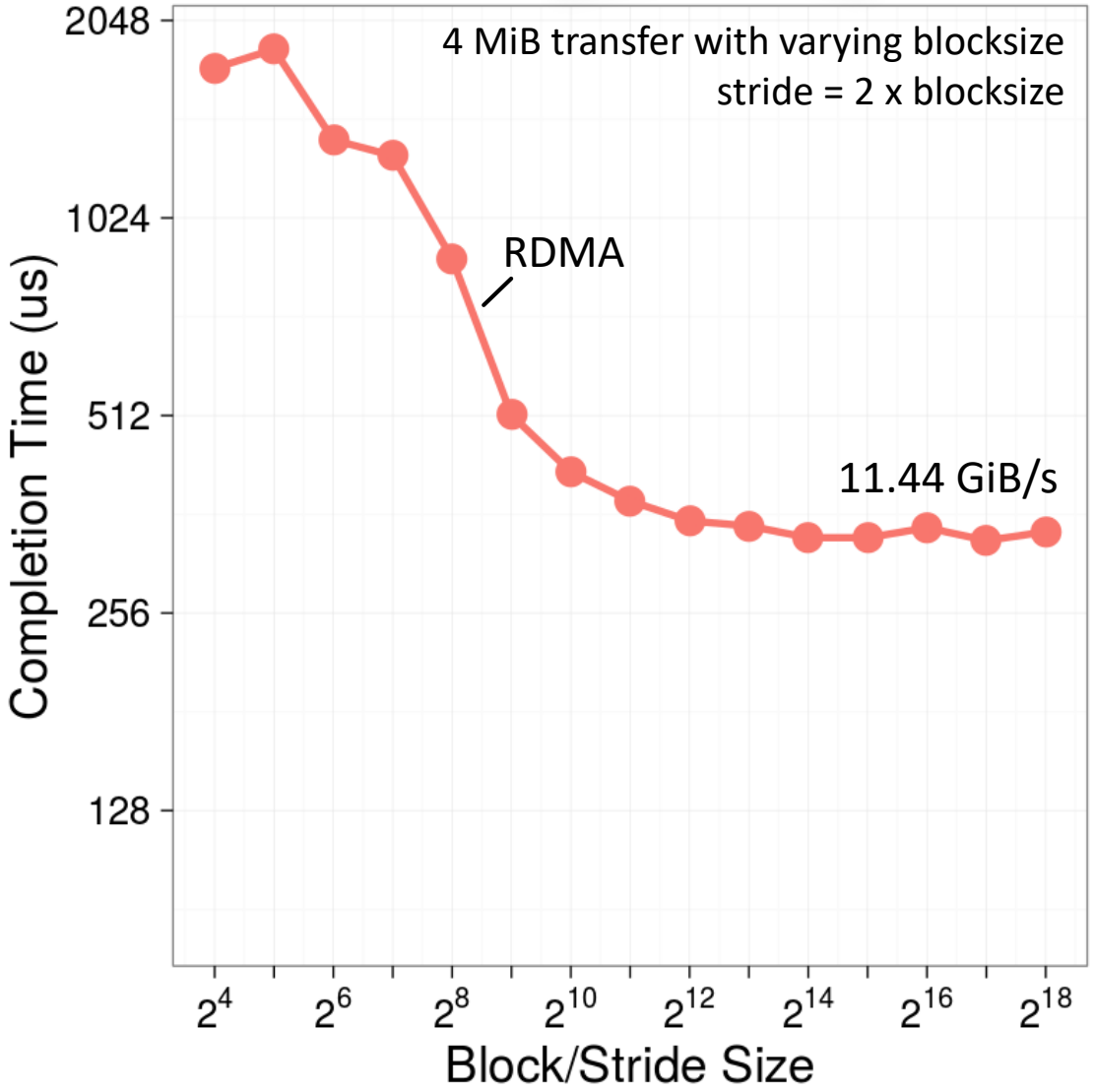


Shankar D. et al., High-performance and Resilient Key-Value Store with Online Erasure Coding for Big Data Workloads. ICDCS'17

Use Case 3: MPI Datatypes acceleration

Data Layout Transformation

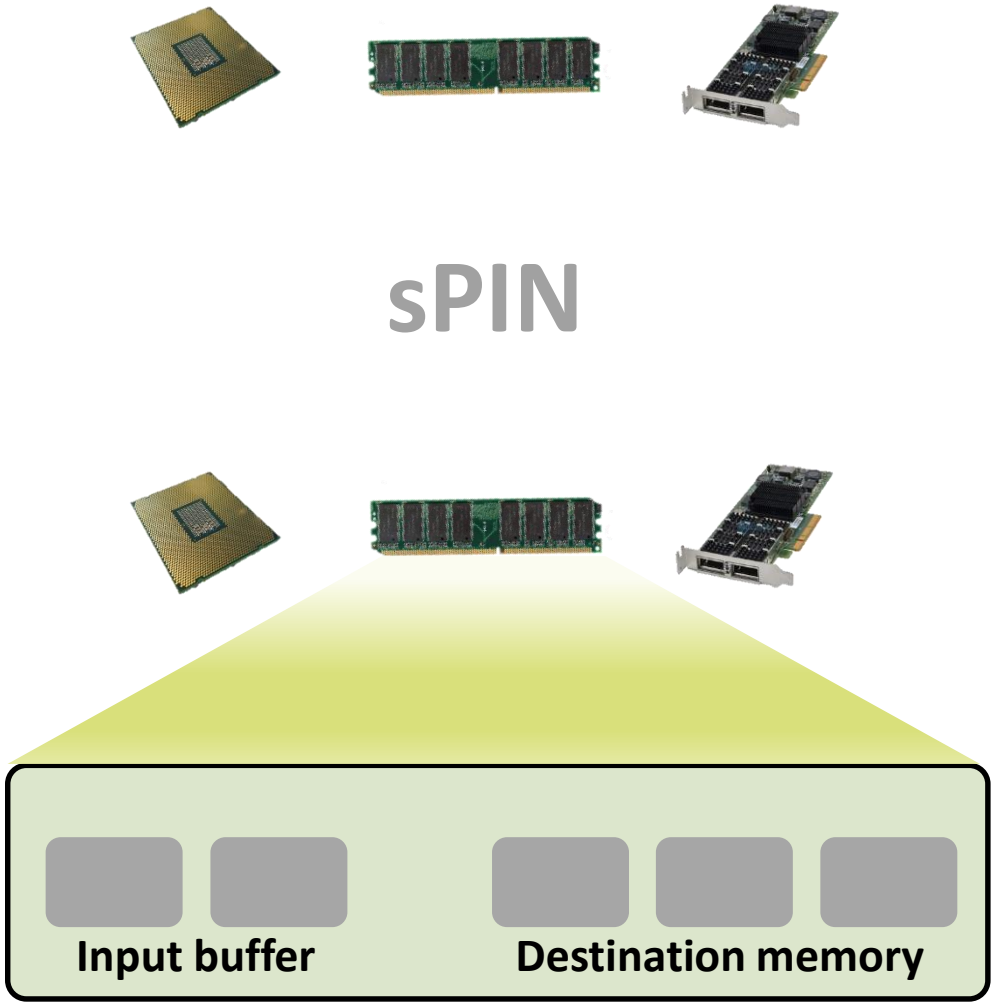


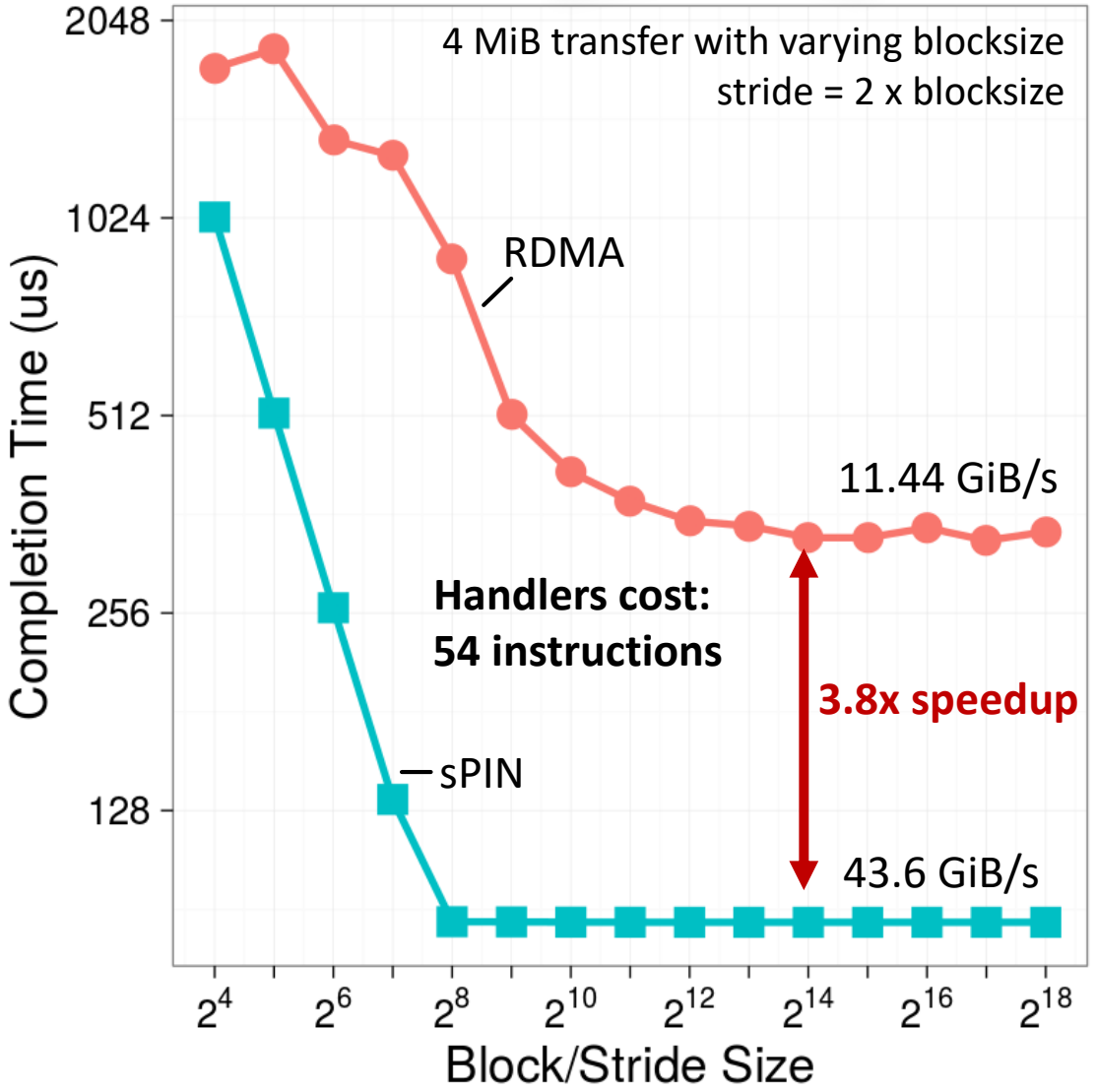


Gropp, W., et al., March. Improving the performance of MPI derived datatypes. *MPIDC'99*

Use Case 3: MPI Datatypes acceleration

Data Layout Transformation





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

Further results and use-cases

Use Case 4: MPI Rendezvous Protocol

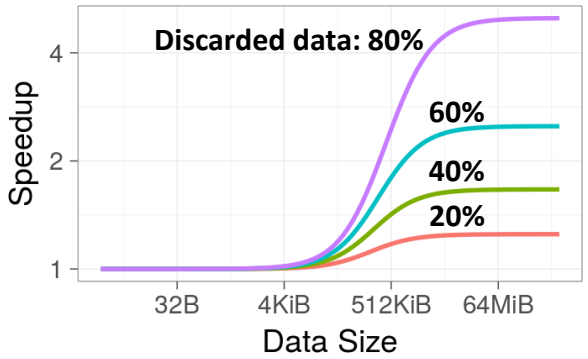
program	p	msgs	ovhd	ovhd	red
MILC	64	5.7M	5.5%	1.9%	65%
POP	64	772M	3.1%	2.4%	22%
coMD	72	5.3M	6.1%	2.4%	60%
coMD	360	28.1M	6.5%	2.8%	58%
Cloverleaf	72	2.7M	5.2%	2.4%	53%
Cloverleaf	360	15.3M	5.6%	3.2%	42%

Use Case 5: Distributed KV Store

41% lower latency

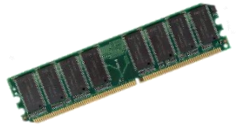
Kalia, A., et al., Using RDMA efficiently for key-value services. In *ACM SIGCOMM Computer Communication Review, 2014*

Use Case 6: Conditional Read



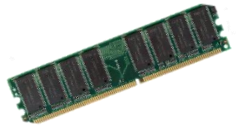
Barthels, C., et al., Designing Databases for Future High-Performance Networks. *IEEE Data Eng. Bulletin, 2017*

Use Case 7: Distributed Transactions



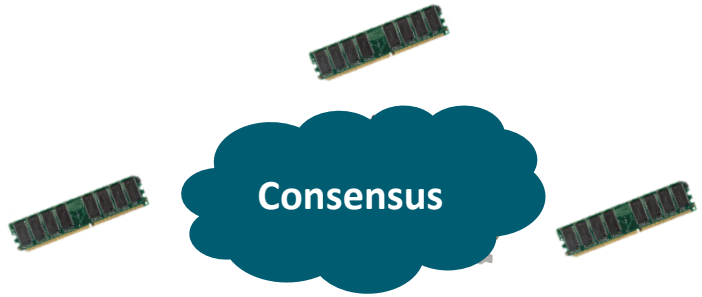
Dragojević, A, et al., No compromises: distributed transactions with consistency, availability, and performance. SOSP'15

Use Case 8: FT Broadcast



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

Use Case 9: Distributed Consensus



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

Further results and use-cases

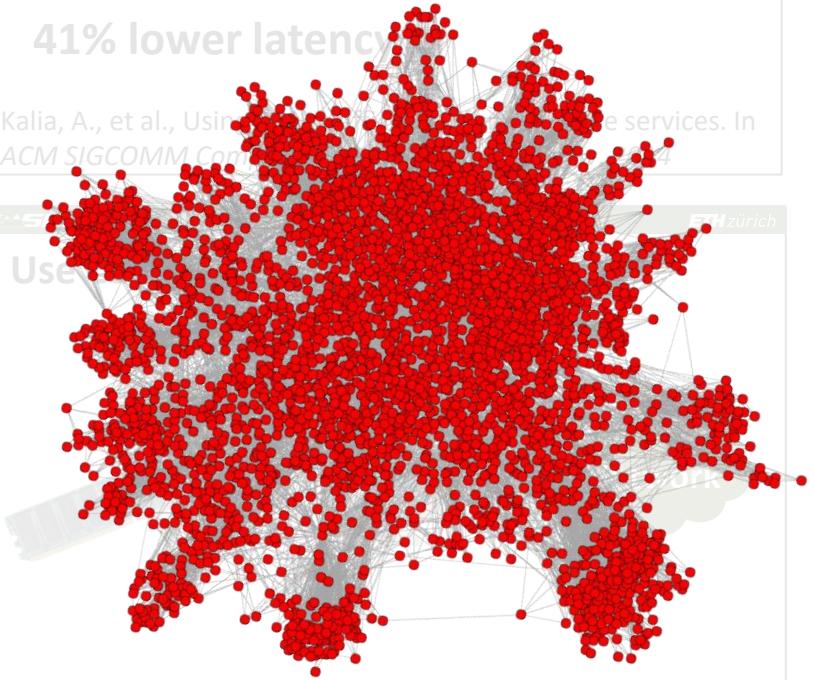
Use Case 4: MPI Rendezvous Protocol

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., Using … 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

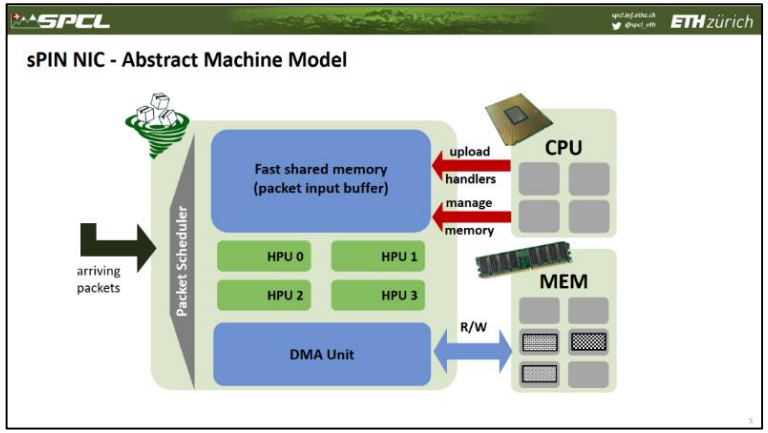
Use Case 8: …

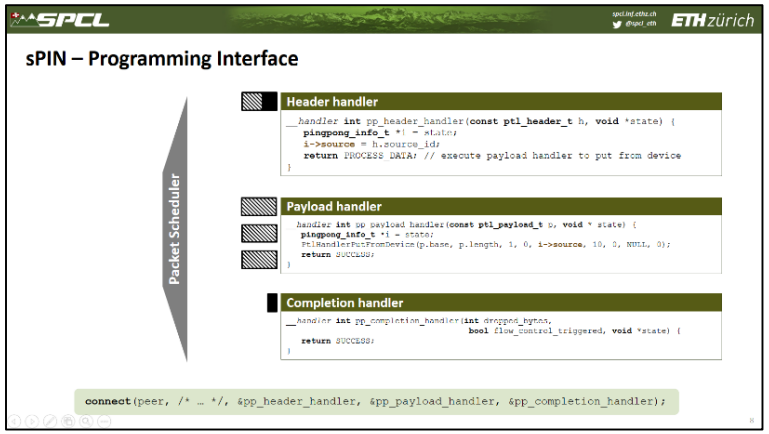
Bosilca, G., et al., Failure Detection and Propagation in HPC systems. SC'16

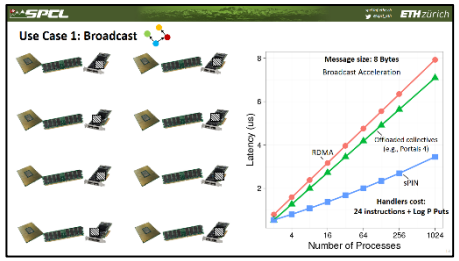
Use Case 9: Distributed Consensus

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

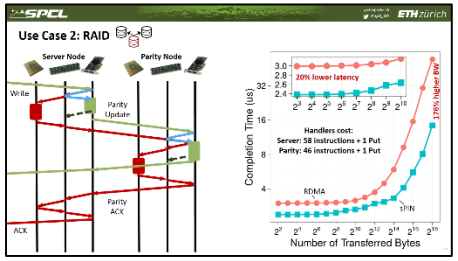
sPIN *Streaming Processing in the Network* for Network Acceleration

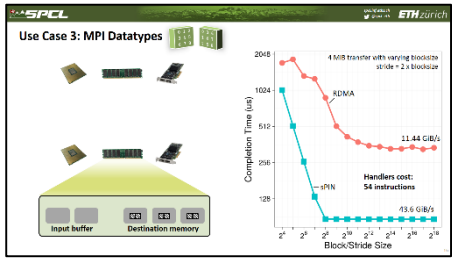


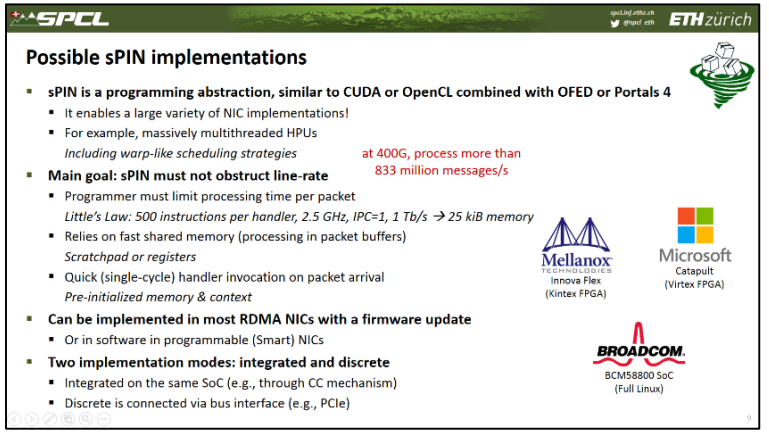




sPIN  **beyond RDMA**







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

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