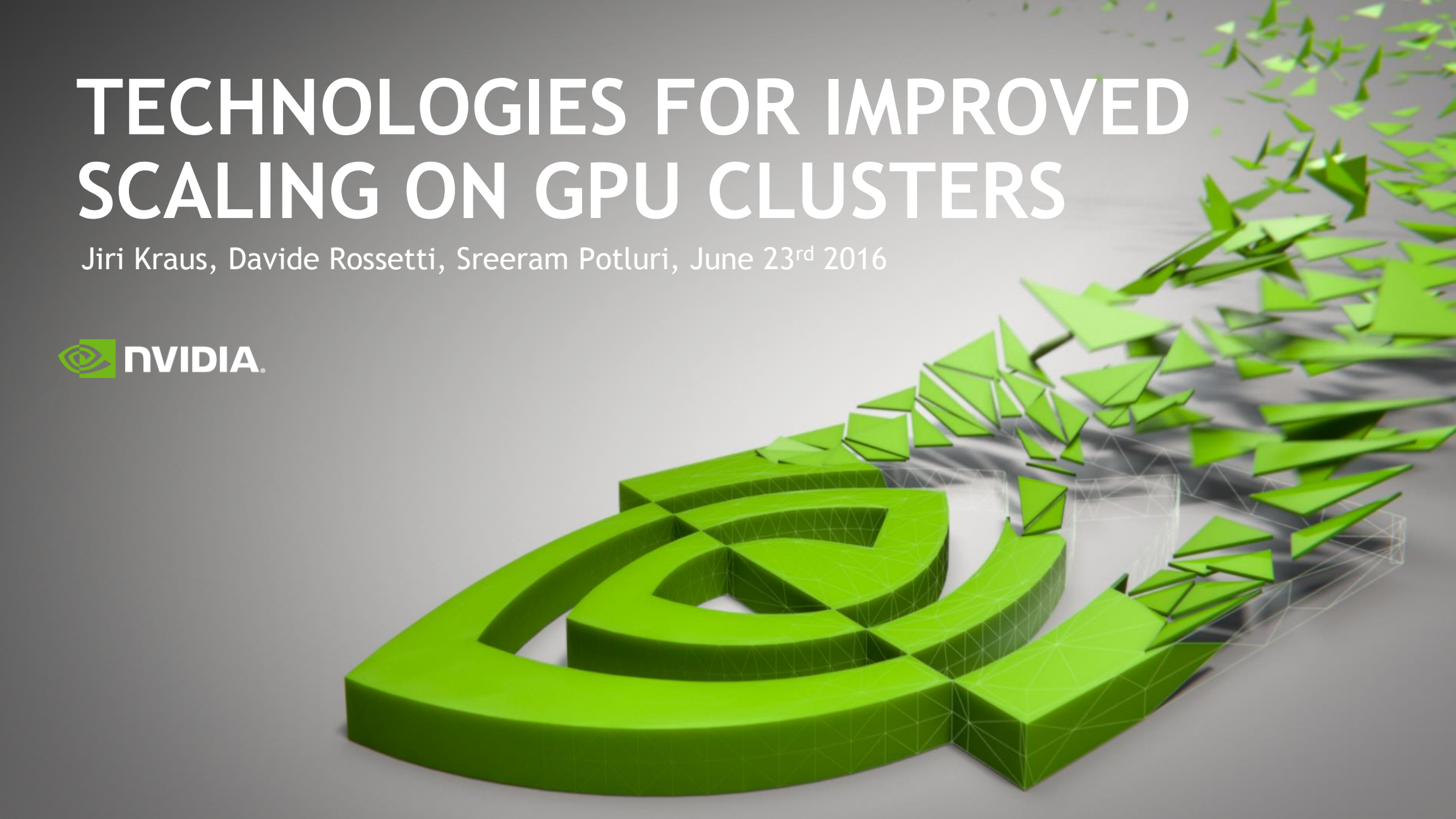
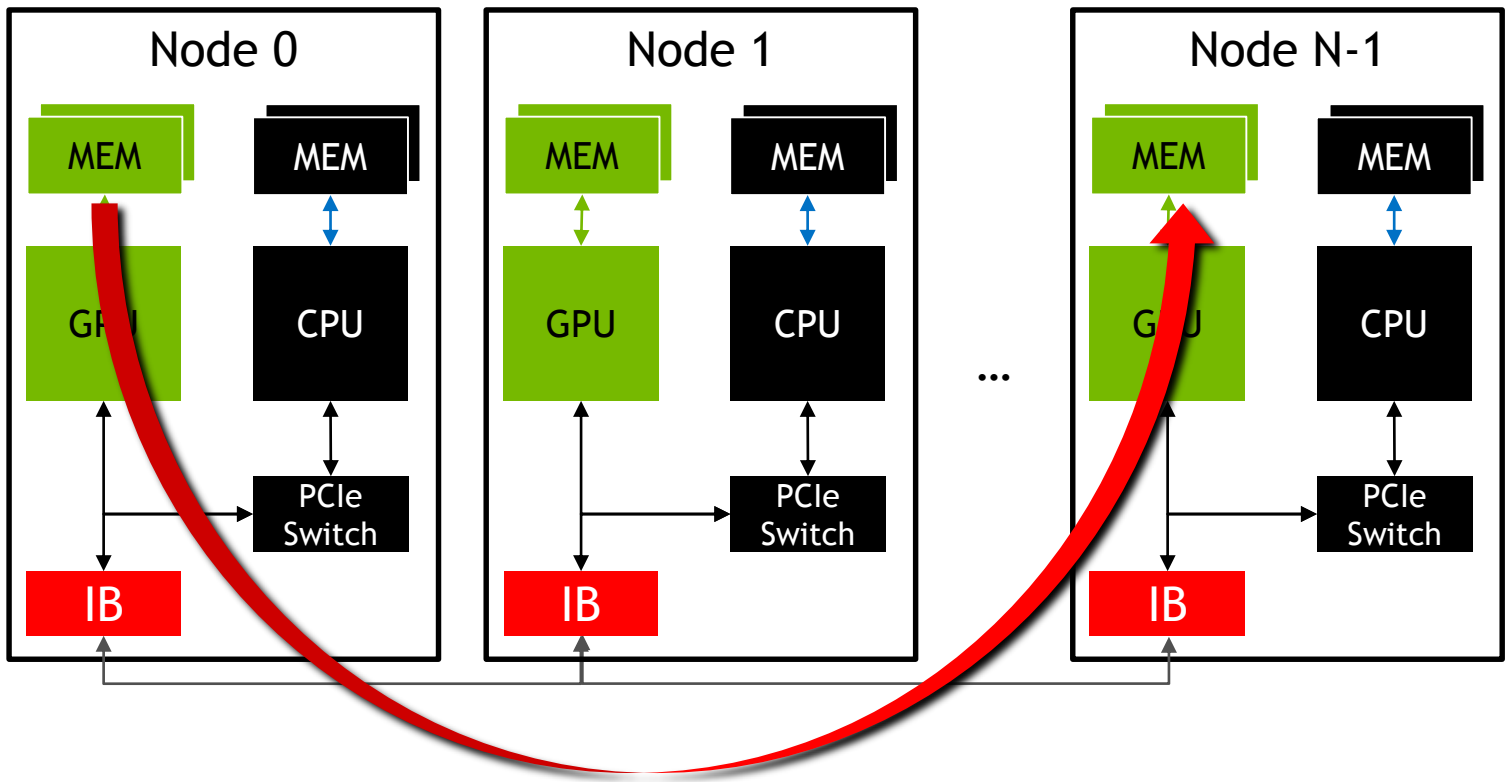
TECHNOLOGIES FOR IMPROVED SCALING ON GPU CLUSTERS

Jiri Kraus, Davide Rossetti, Sreeram Potluri, June 23rd 2016



MULTI GPU PROGRAMMING



MULTI GPU PROGRAMMING

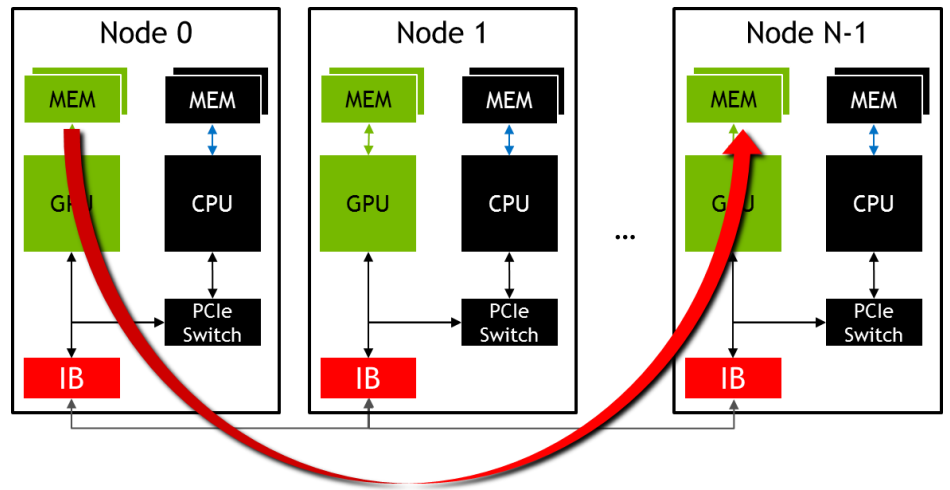
Minimizing Communication Overhead

Application:

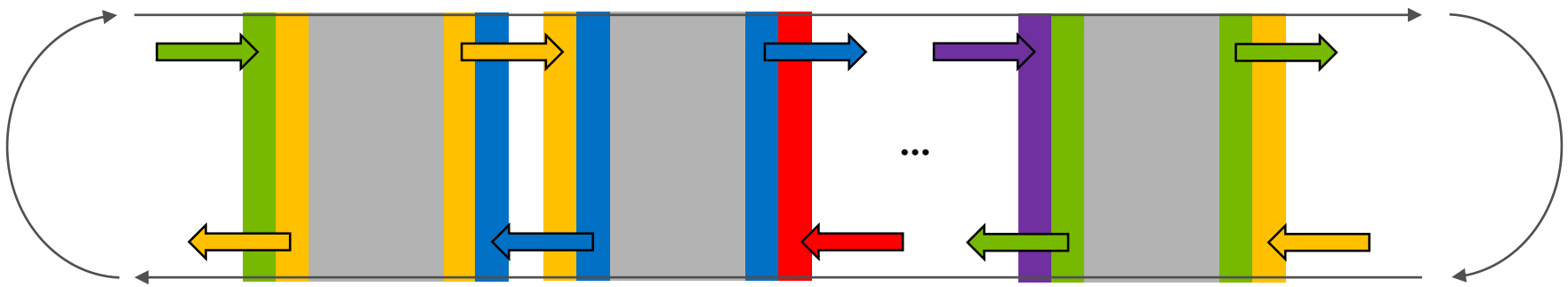
- Expose enough Parallelism
- Balance Load
- Minimize communication
- Hide communication time

MPI/System SW/HW:

- Maximize BW
- Minimize Latencies



1D RING EXCHANGE



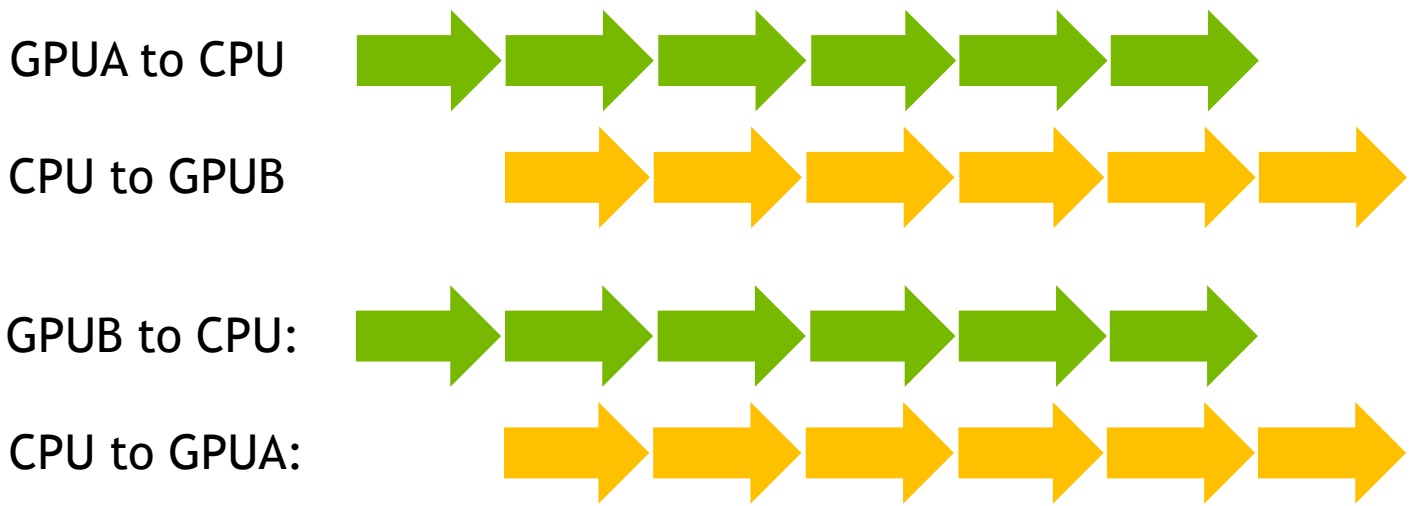
Halo updates for 1D domain decomposition with periodic boundary conditions

Unidirectional rings are important building block for collective algorithms

MAXIMIZING BANDWIDTH

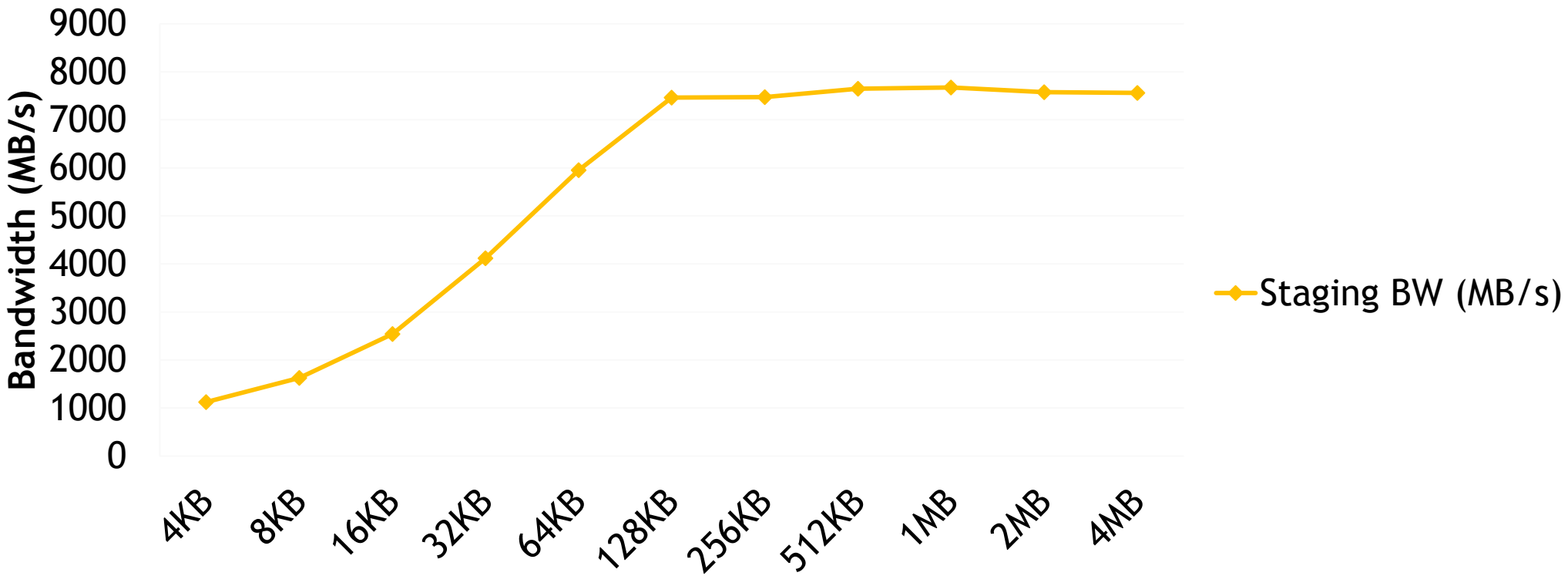
Bidirectional intra node GPU to GPU

CPU staging pipeline can achieve good GPU to GPU Bandwidth:



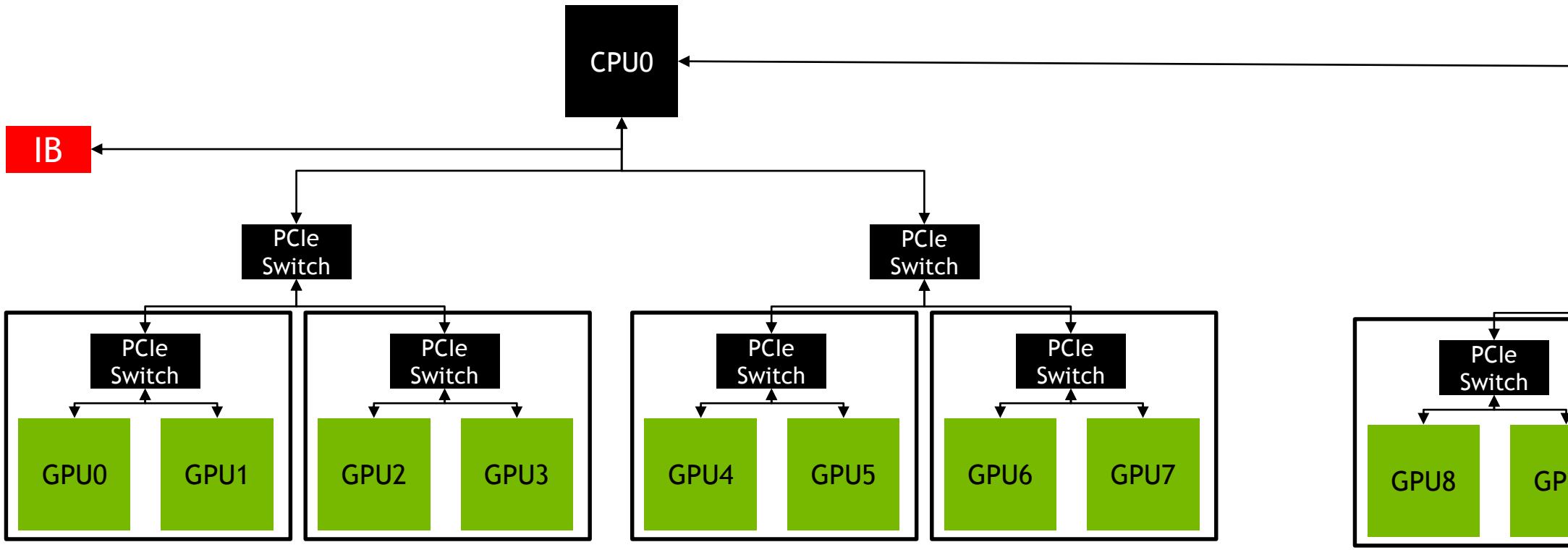
MAXIMIZING BANDWIDTH

MVAPICH2-GDR 2.2b intra-node GPU-to-GPU pt2pt BiBW



TREND TO DENSER GPU NODES

Dual Socket + 8 Tesla K80 = 16 GPUs



TREND TO DENSER GPU NODES

Dual Socket + 8 Tesla K80 = 16 GPUs

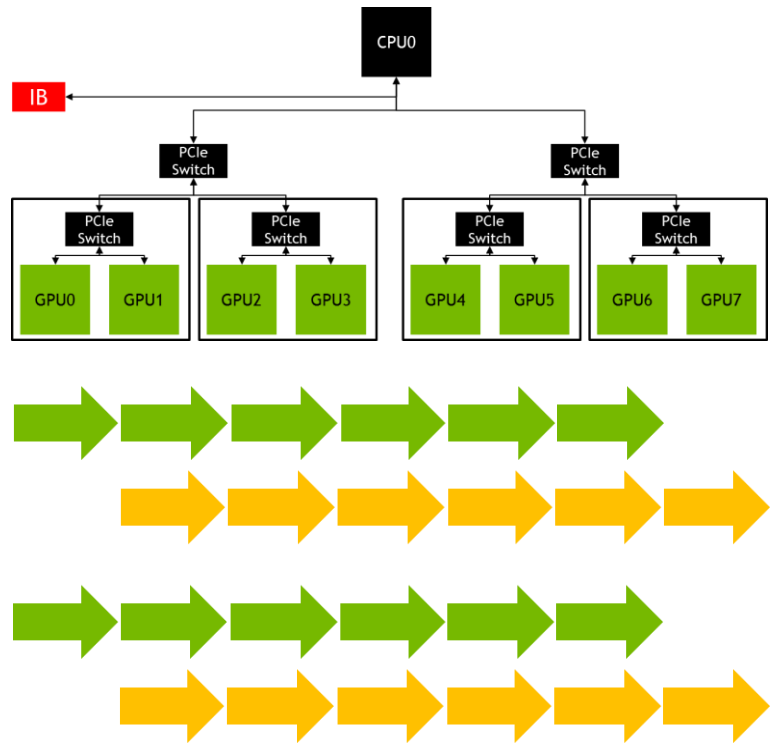
8 GPUs per CPU Socket

BiBW Staging Pipeline:

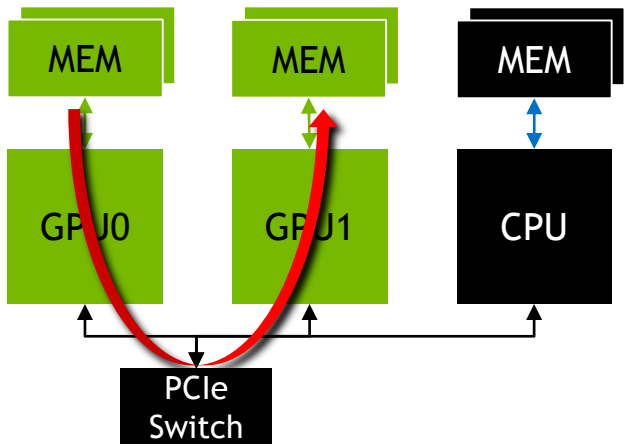
- 2 writes and 2 reads in CPU mem per Com. pair
- 9 Com. pairs (including IB and intra Socket)

CPU Memory BW: 70 GB/s

- 70 GB/s / (9*(2+2)) = 70 GB/s / 36
- 1.94 GB/s (12% of PCI-E x16 gen3)



GPUDIRECT P2P

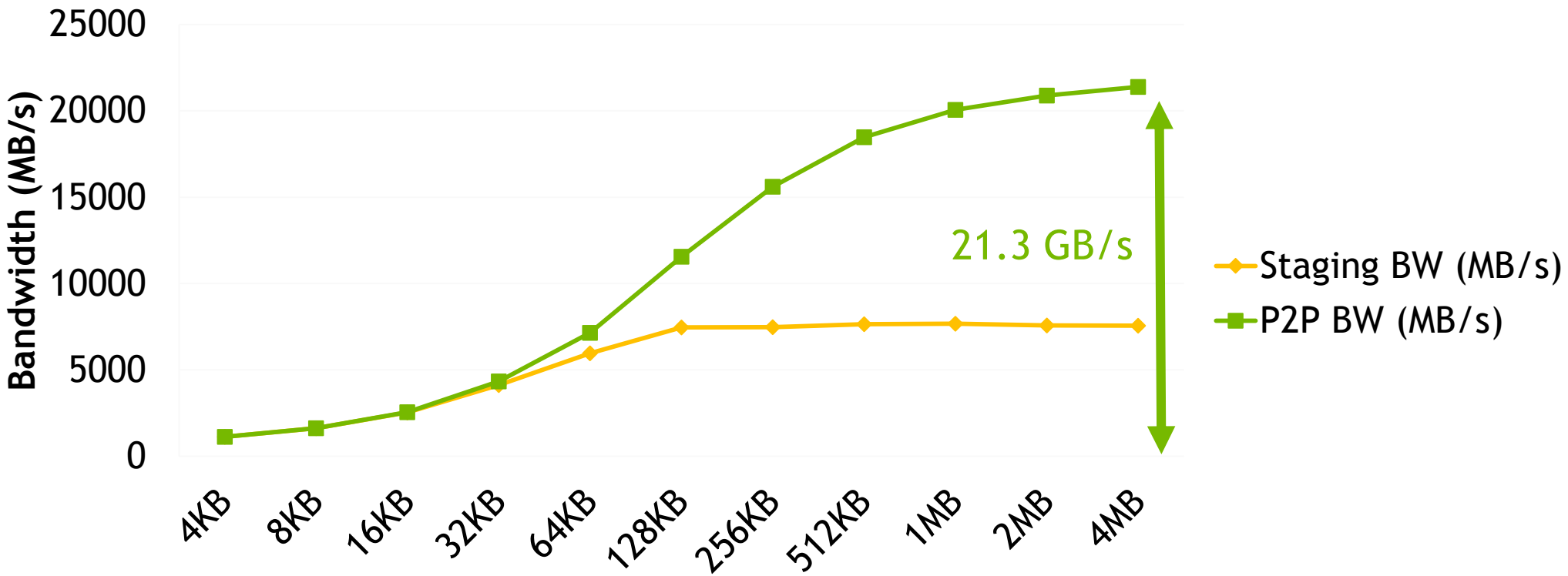


Maximizes intra node inter GPU Bandwidth

Avoids Host memory and system topology bottlenecks

MAXIMIZING BANDWIDTH

MVPAICH2-GDR 2.2b intra-node GPU-to-GPU pt2tp BiBW



NVLINK

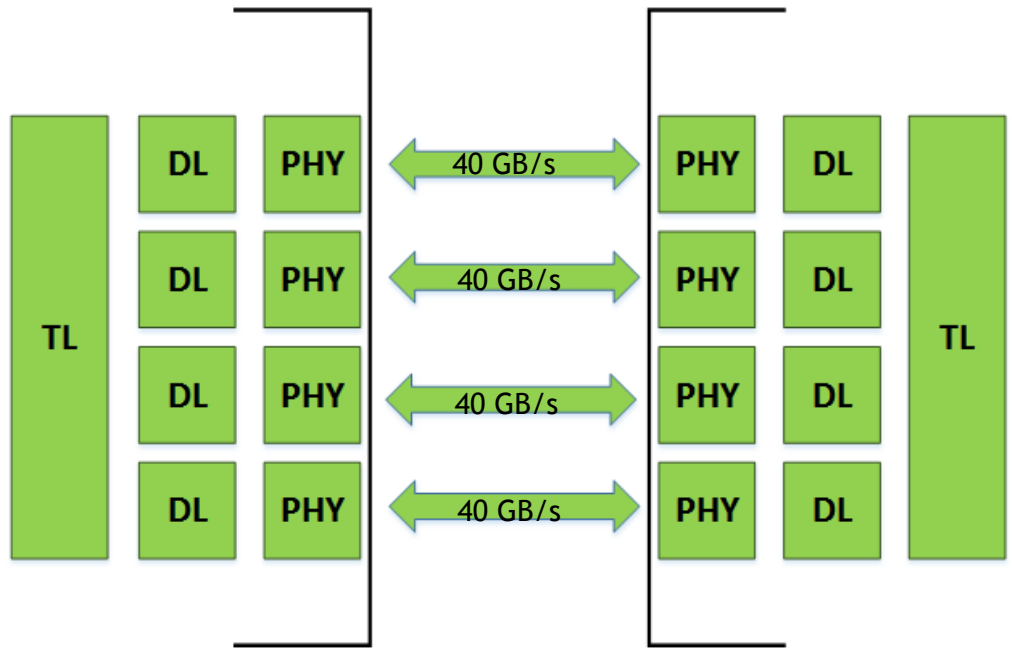
P100 supports 4 NVLinks

Up to 94% bandwidth efficiency

Supports read/writes/atomics to peer GPU

Supports read/write access to NVLink-enabled CPU

Links can be ganged for higher bandwidth



NVLink on Tesla P100

NVLINK - GPU CLUSTER

Two fully connected quads,
connected at corners

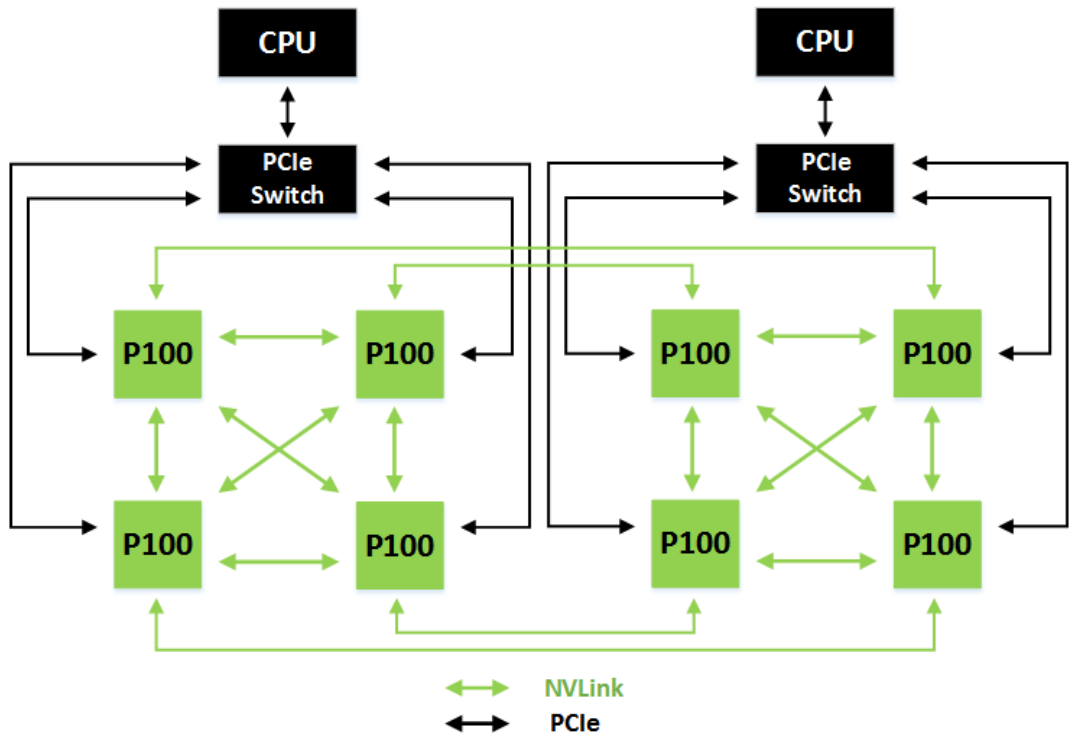
160GB/s per GPU bidirectional to Peers

Load/store access to Peer Memory

Full atomics to Peer GPUs

High speed copy engines for bulk data copy

PCIe to/from CPU



NVLINK TO CPU

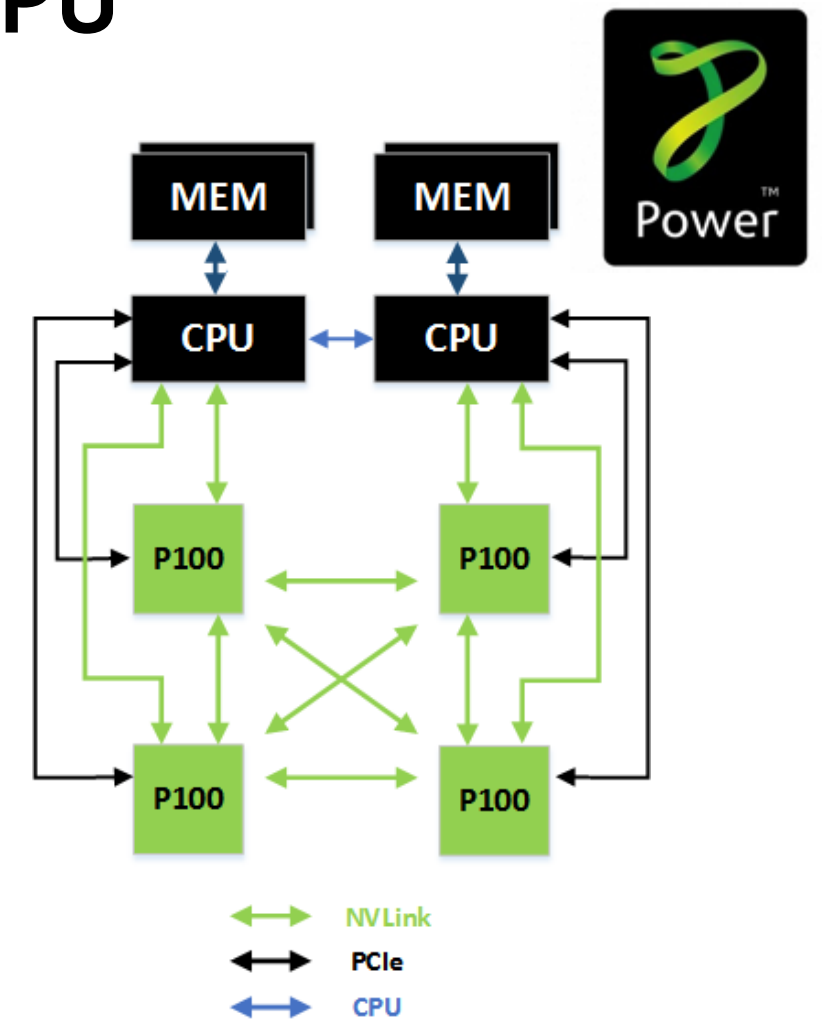
Fully connected quad

120 GB/s per GPU bidirectional for peer traffic

40 GB/s per GPU bidirectional to CPU

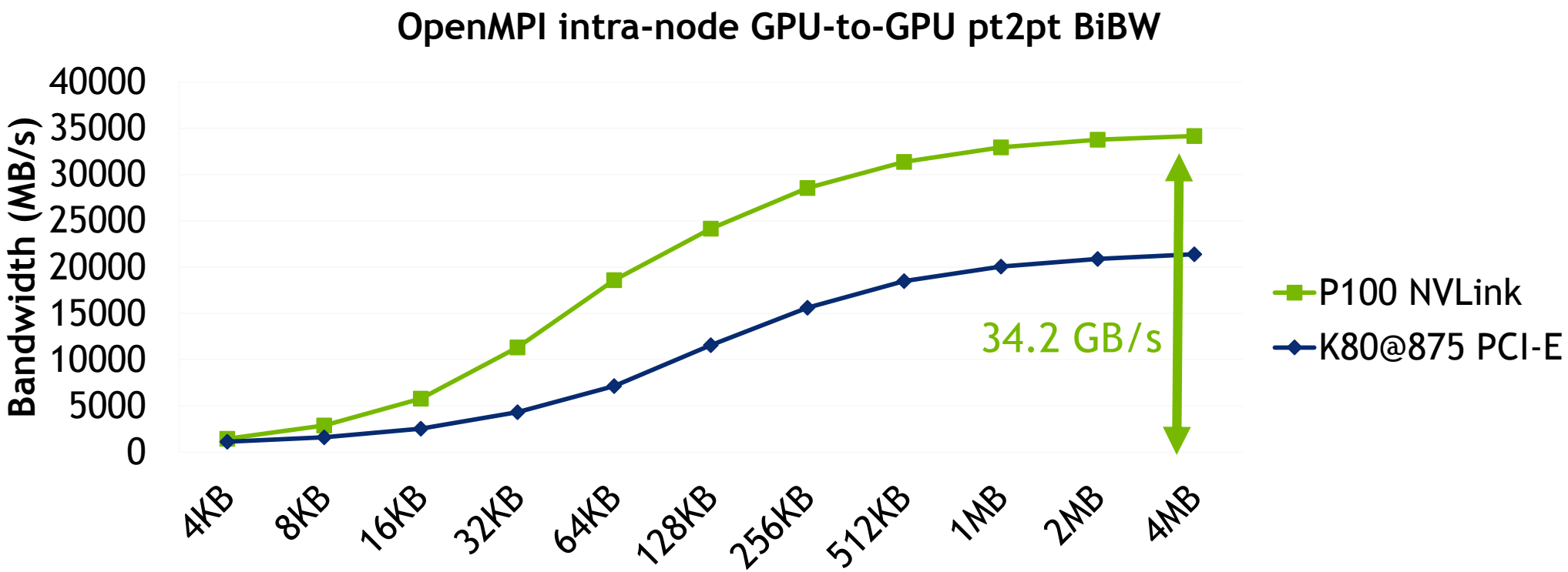
Direct Load/store access to CPU Memory

High Speed Copy Engines for bulk data movement



GPUDIRECT P2P ON PASCAL

early results, P2P thru NVLink



LATENCY

GPUDirect RDMA

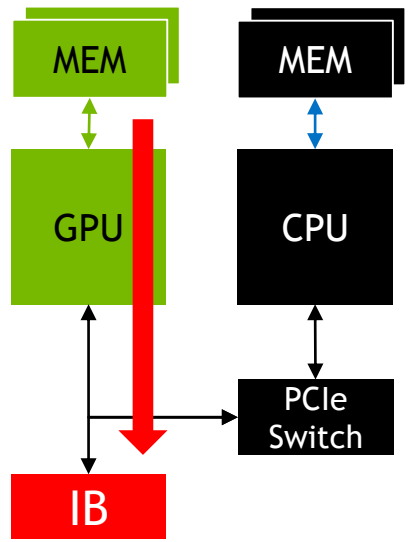
Host staging pipeline increases latency because of staging step

GPUDirect RDMA allows 3rd party devices to directly read/write GPU memory

No host staging necessary for inter node GPU to GPU messages

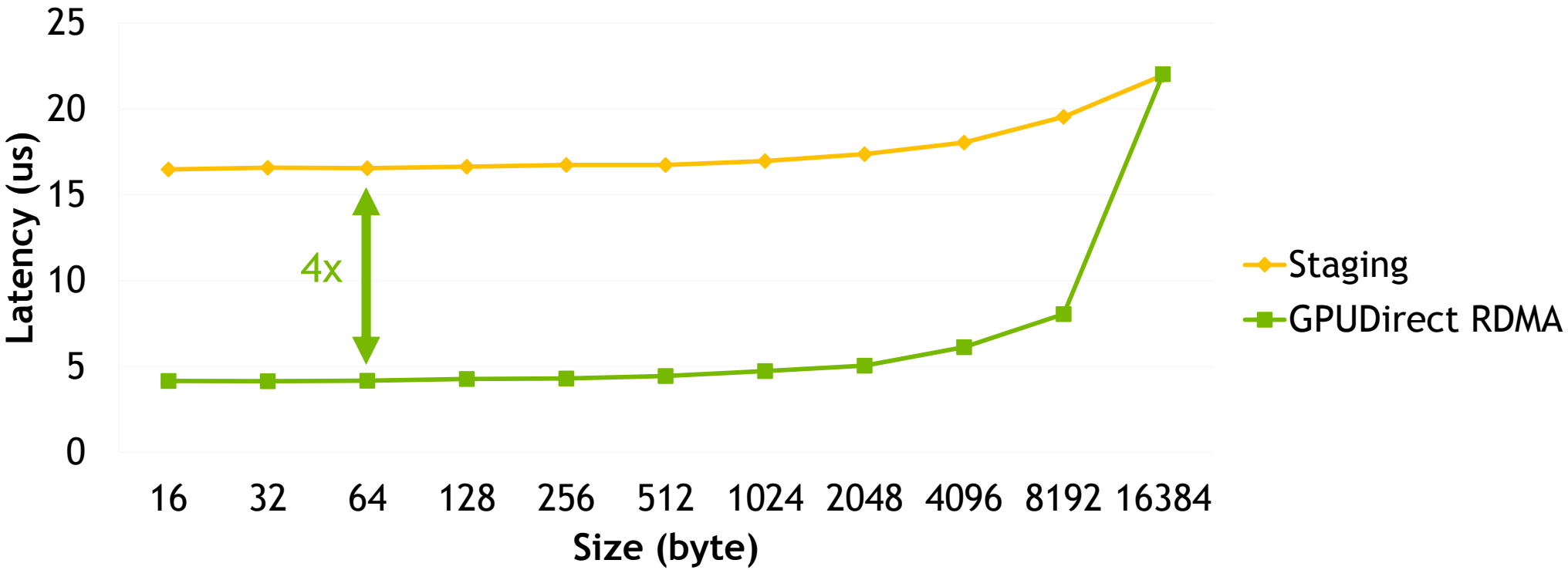
IB stack is optimized for latencies

Using Loop back for GPU to GPU messages also helps with intra node GPU to GPU latency

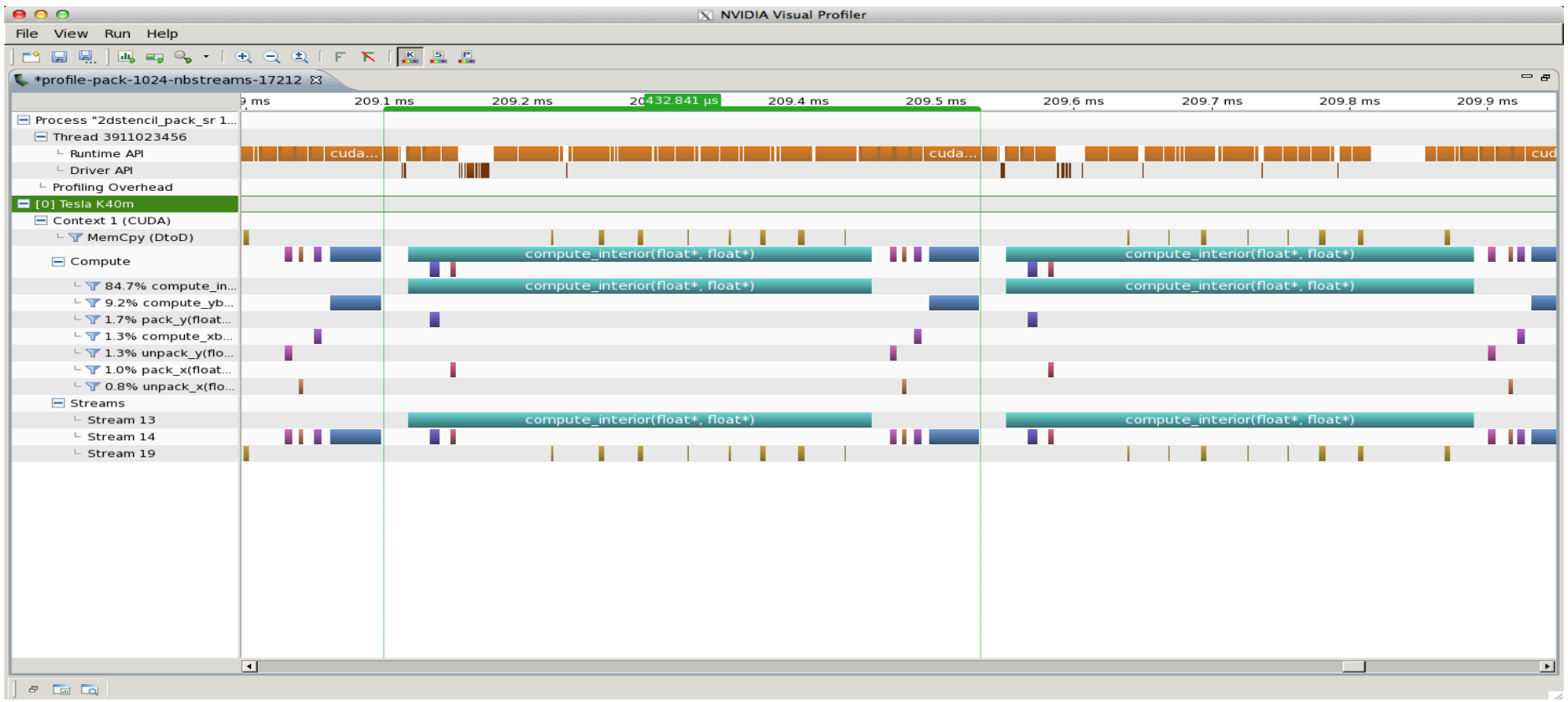


MINIMIZING LATENCY

MVAPICH2-GDR 2.2b intra-node GPU-to-GPU pt2pt latency

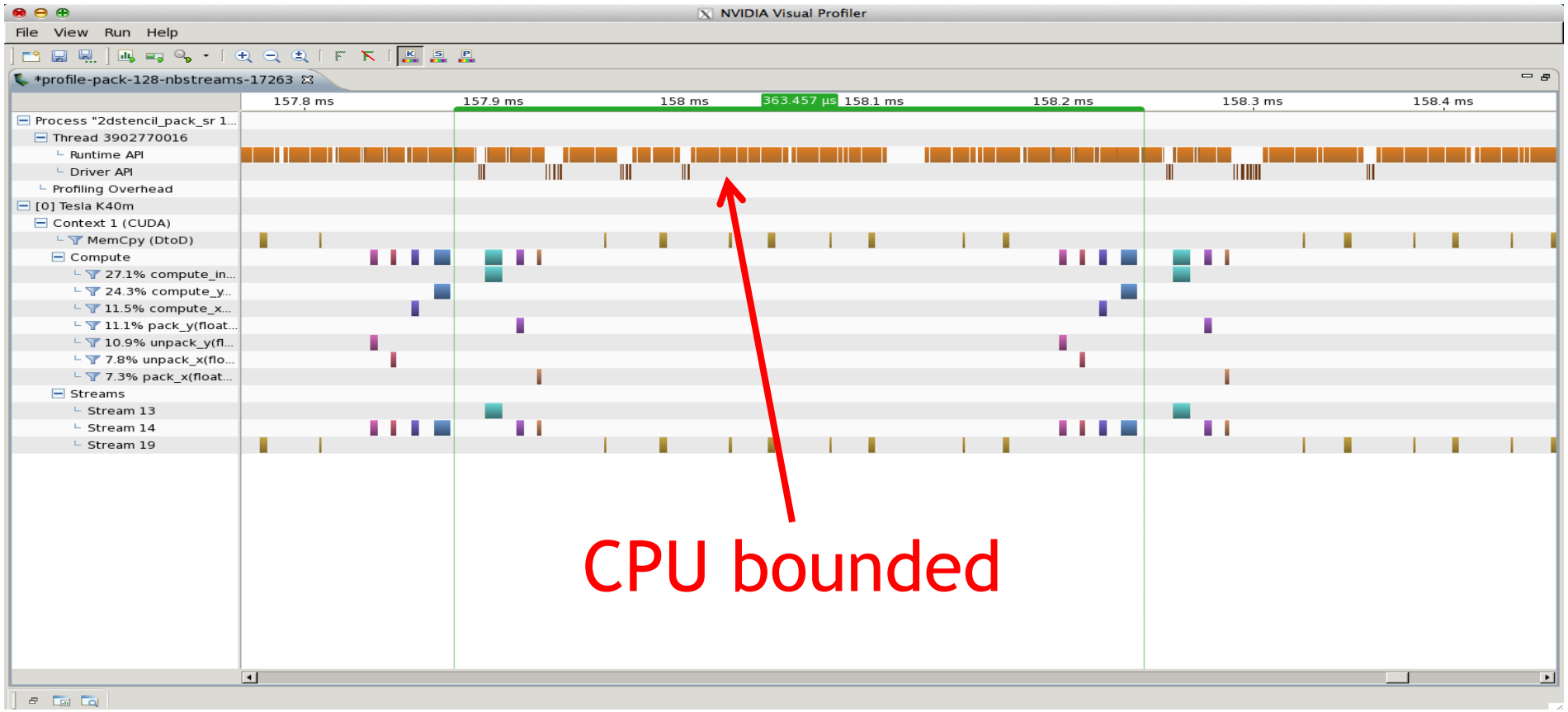


SCALING LIMITER CPU



(Time marked for one step, Domain size/GPU – 1024, Boundary – 16, Ghost Width – 1)

SCALING LIMITER CPU



(Time marked for one step, Domain size/GPU – 128, Boundary – 16, Ghost Width – 1)

USING NORMAL MPI

```
while ( t < T ) {
  //...
  for (int i=0; i<num_neighbors;++i) {
    MPI_Irecv( ... ,req);
    compute_boundary<<<grid,block,0,stream>>>(...);
    cudaStreamSynchronize(stream);
    MPI_Isend( ... ,req + 1 );
    MPI_Waitall(...,req,...);
  }
  //...
}
```

GPUDIRECT ASYNC

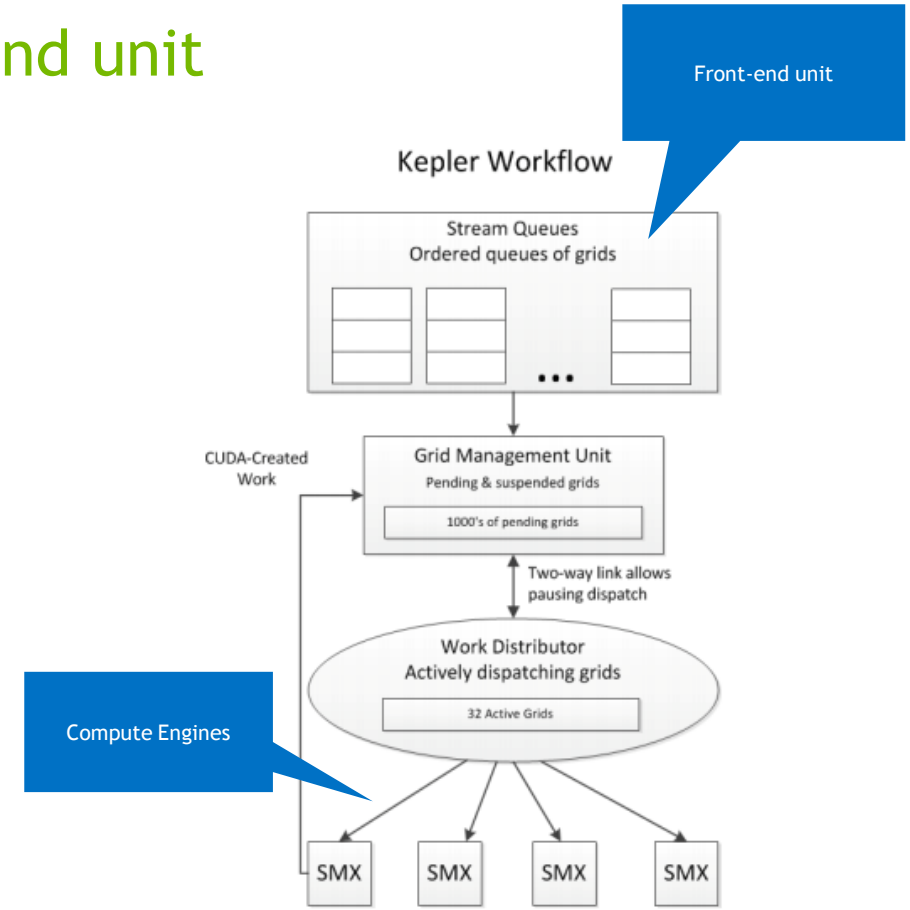
expose GPU front-end unit

CPU prepares work plan

- hardly parallelizable, branch intensive
- GPU orchestrates flow

Runs on optimized front-end unit

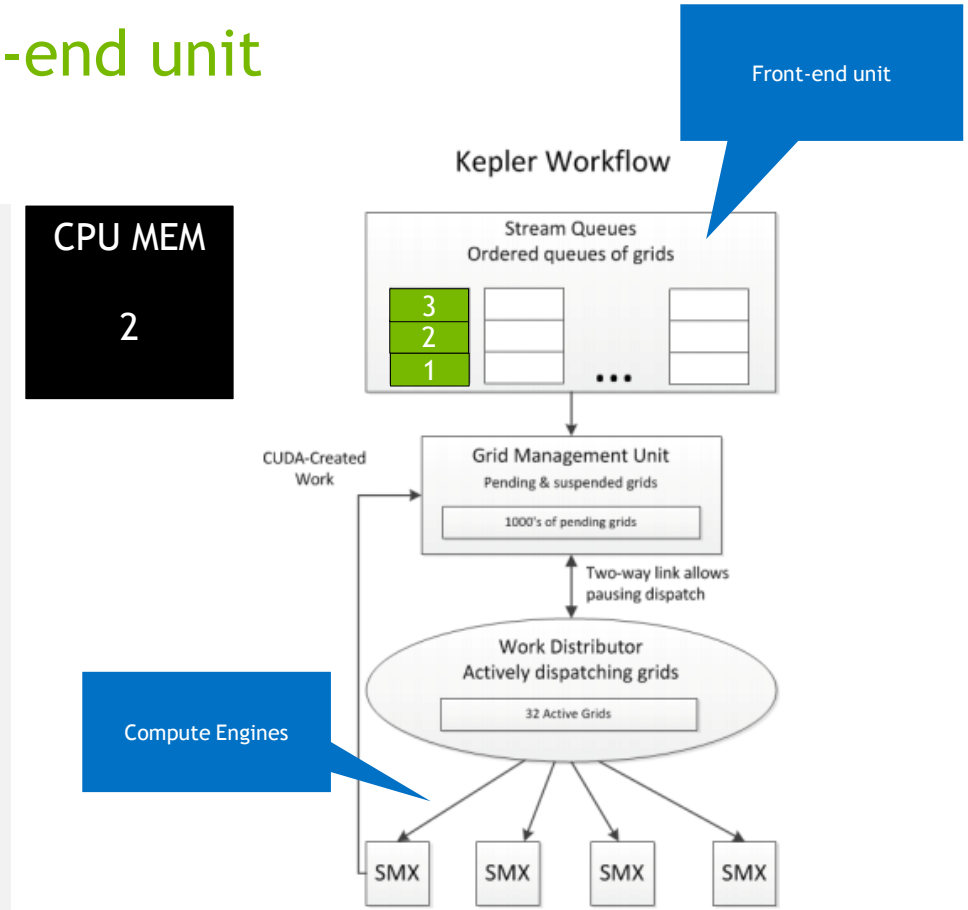
- Same one scheduling GPU work
- Now also scheduling network communications



GPUDIRECT ASYNC

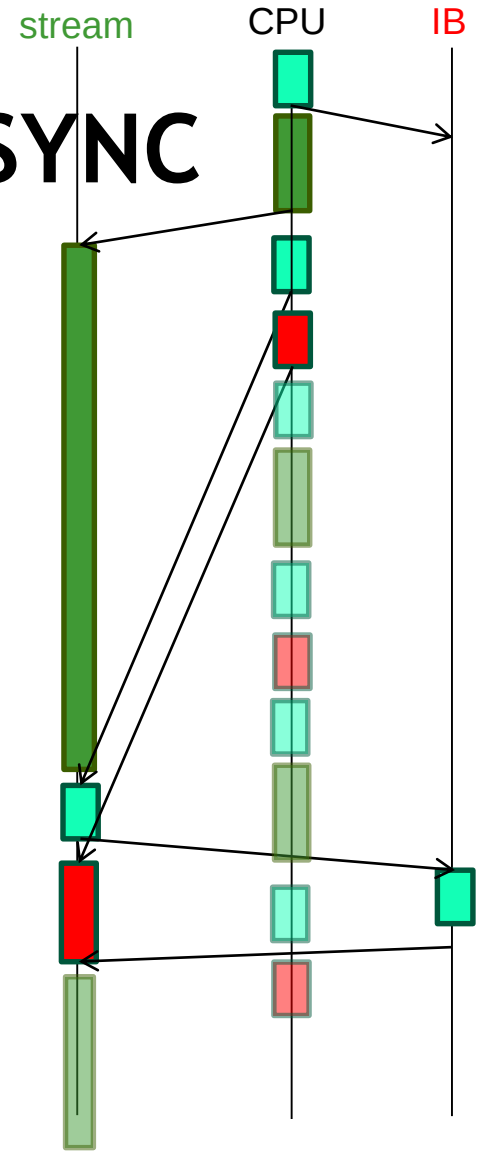
expose GPU front-end unit

```
*(volatile uint32_t*)h_flag = 0;
//...
cuStreamWaitValue32(stream, d_flag, 1,
  CU_STREAM_WAIT_VALUE_EQ);
calc_kernel<<<GSZ,BSZ,0,stream>>>();
cuStreamWriteValue32(stream, d_flag, 2, 0);
//...
*(volatile uint32_t*)h_flag = 1;
//...
cudaStreamSynchronize(stream);
assert(*(volatile uint32_t*)h_flag == 2);
```



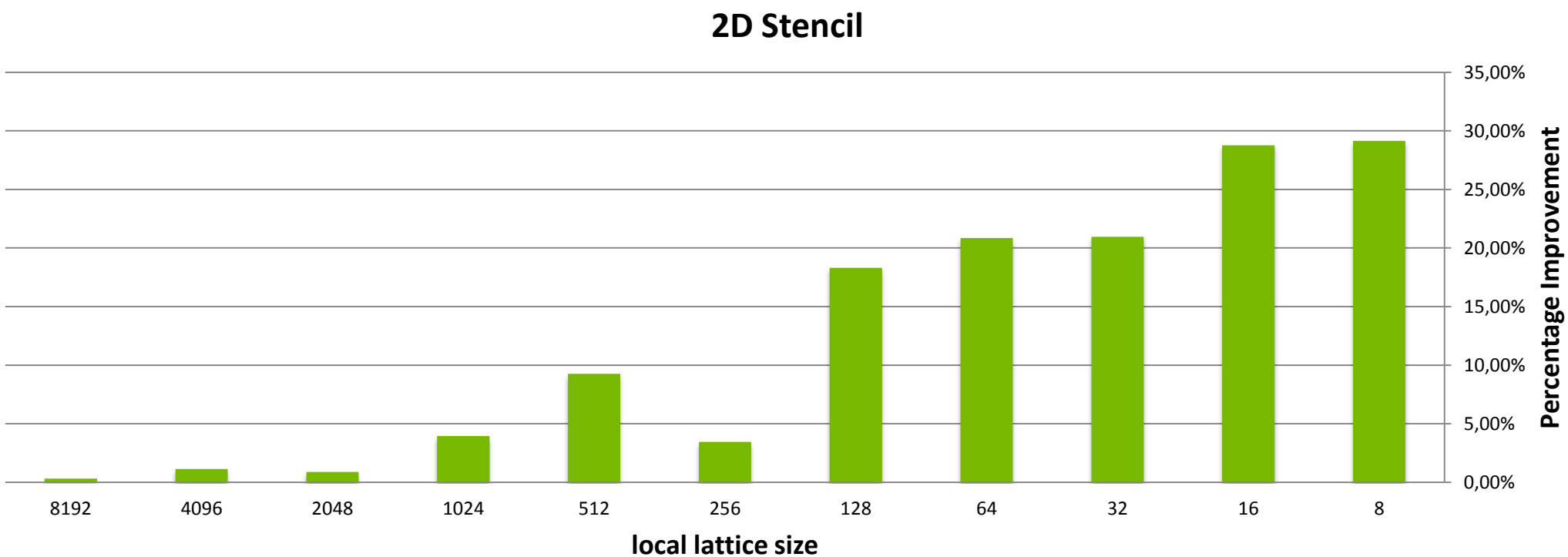
USING STREAM-ORDERED API + ASYNC

```
while ( t < T ) {
  //...
  for (int i=0; i<num_neighbors;++i) {
    mp_irecv( ... ,req);
    compute_boundary<<<grid,block,0,stream>>>(...);
    mp_isend_on_stream( ... ,req + 1, stream );
    mp_wait_all_on_stream( ...,req,stream );
  }
  //...
}
```



2D STENCIL PERFORMANCE

weak scaling, RDMA vs. RDMA+Async



four nodes, IVB Xeon CPUs, K40m GPUs, Mellanox Connect-IB FDR, Mellanox FDR switch

GPUDIRECT ASYNC + INFINIBAND

preview release of components

CUDA Async extensions, with CUDA 8

Peer-direct async extension, in MLNX OFED 3.x, soon