GPU TECHNOLOGY CONFERENCE

GPUDIRECT: INTEGRATING THE GPU WITH A NETWORK INTERFACE

DAVIDE ROSSETTI, SW COMPUTE TEAM

GPUDIRECT FAMILY[1]

- GPUDirect Shared GPU-Sysmem for inter-node copy optimization
- GPUDirect P2P for intra-node, accelerated GPU-GPU memcpy
- GPUDirect P2P for intra-node, inter-GPU LD/ST access
- GPUDirect RDMA[2] for inter-node copy optimization

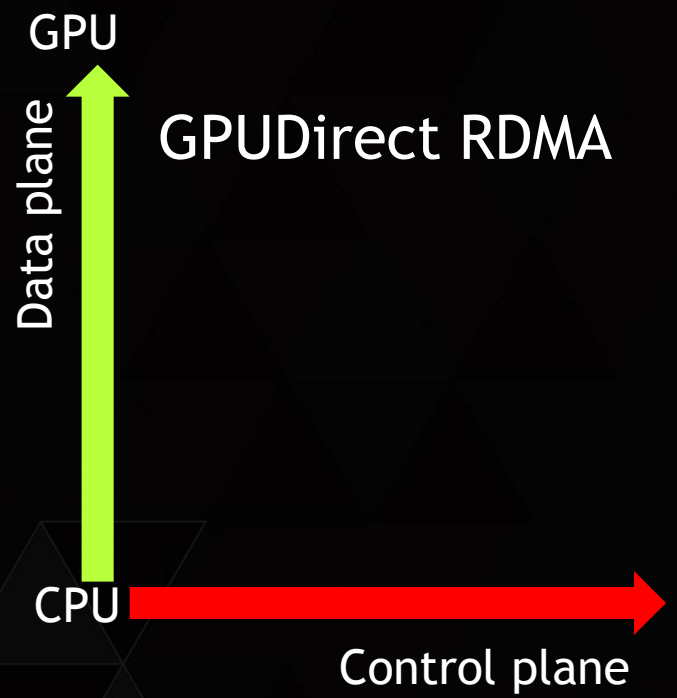
[1] developer info: https://developer.nvidia.com/gpudirect

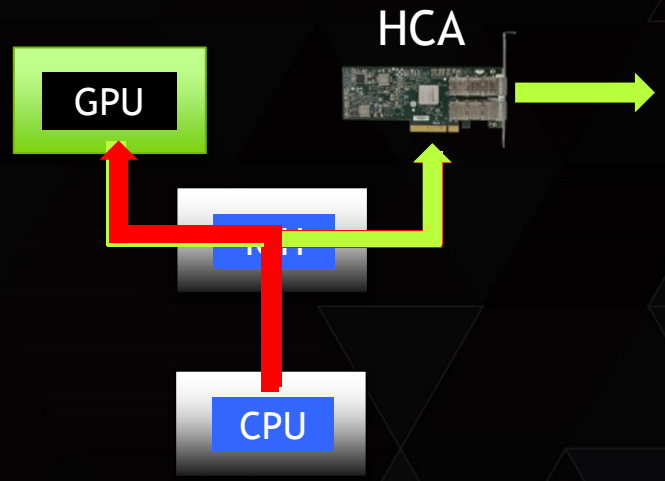
[2] http://docs.nvidia.com/cuda/gpudirect-rdma

GPUDIRECT RDMA CAPABILITIES & LIMITATIONS

- GPUDirect RDMA
  - direct HCA access to GPU memory
- CPU still driving computing + communication
  - Fast CPU needed
  - Implications: power, latency, TCO
  - Risks: limited scaling …

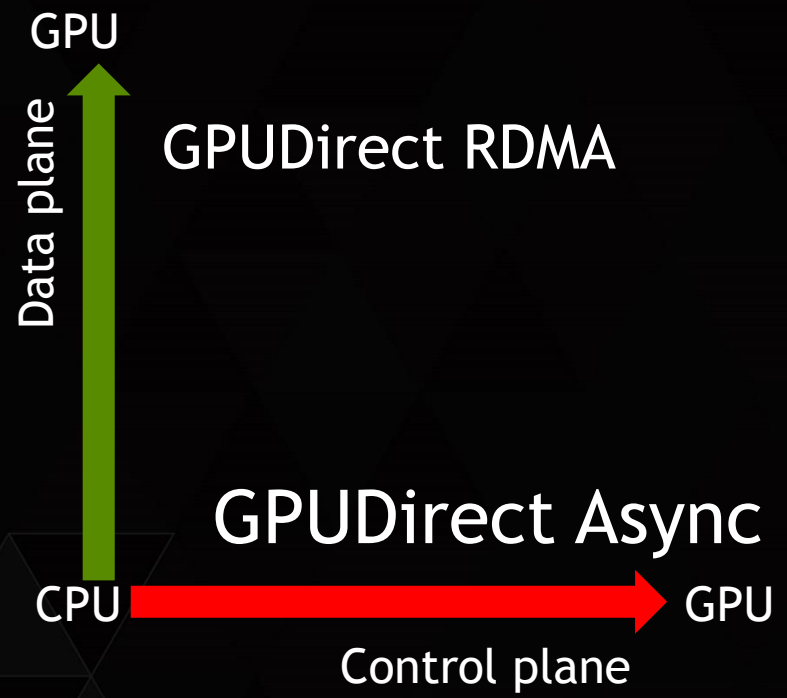
MOVING DATA AROUND

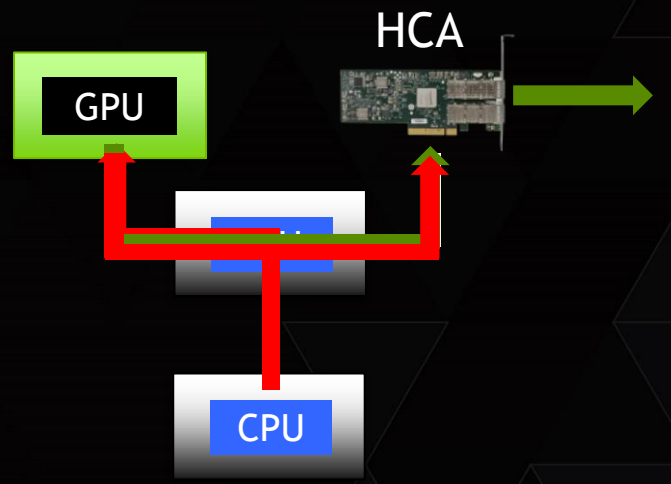




communication tasks on HCA

MEET NEXT THING

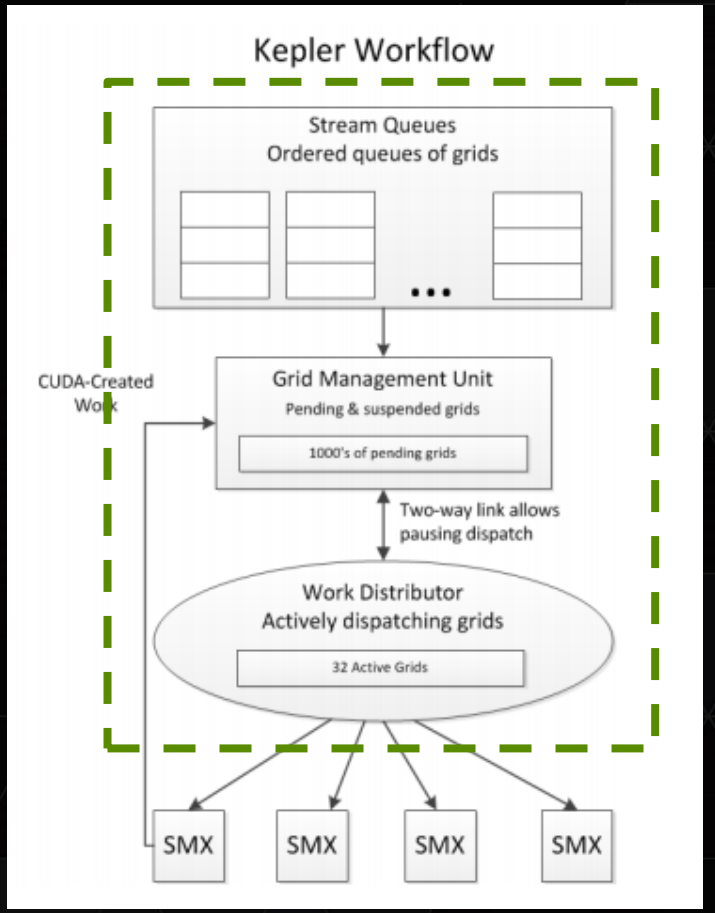




CPU prepares and queues
compute and communication tasks on GPU
GPU triggers communication on HCA
HCA directly accesses GPU memory

CPU OFF THE CRITICAL PATH

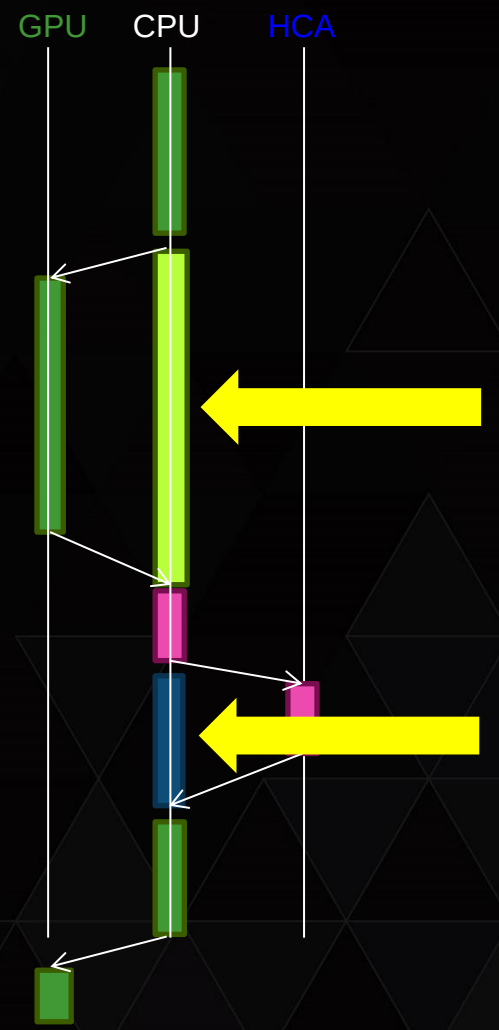
- CPU prepares work plan
  - hardly parallelizable, branch intensive
- GPU orchestrates flow
  - Runs on optimized scheduling unit
  - Same one scheduling GPU work
  - Now also scheduling network communications



KERNEL+SEND
NORMAL FLOW

```
a_kernel<<<…,stream>>>(buf);
cudaStreamSynchronize(stream);
ibv_post_send(buf);
while (!done) ibv_poll_cq(txcq);
b_kernel<<<…,stream>>>(buf);
```

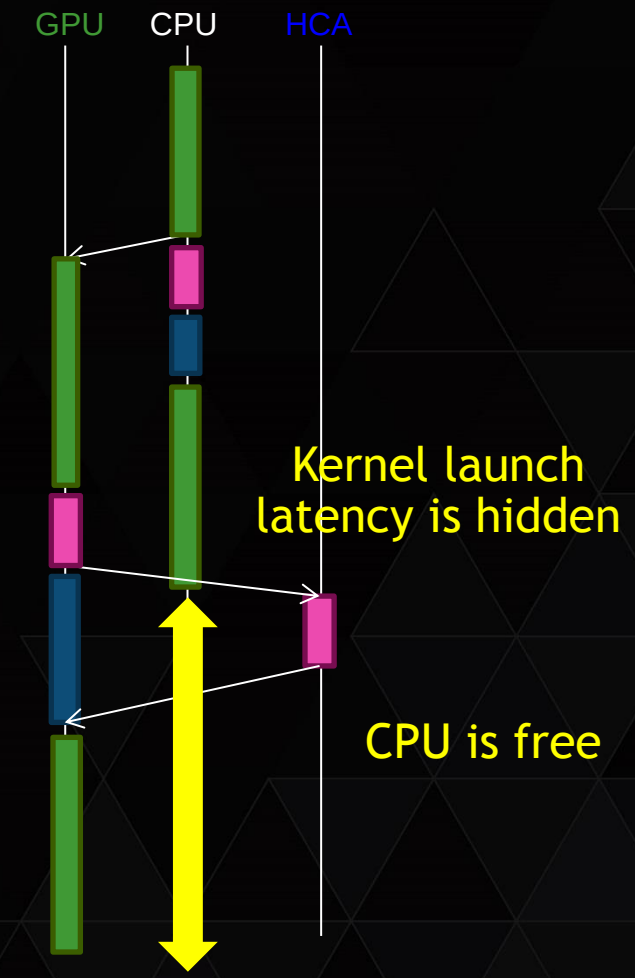
100% CPU utilization
Limited scaling!



KERNEL+SEND
GPUDIRECT ASYNC

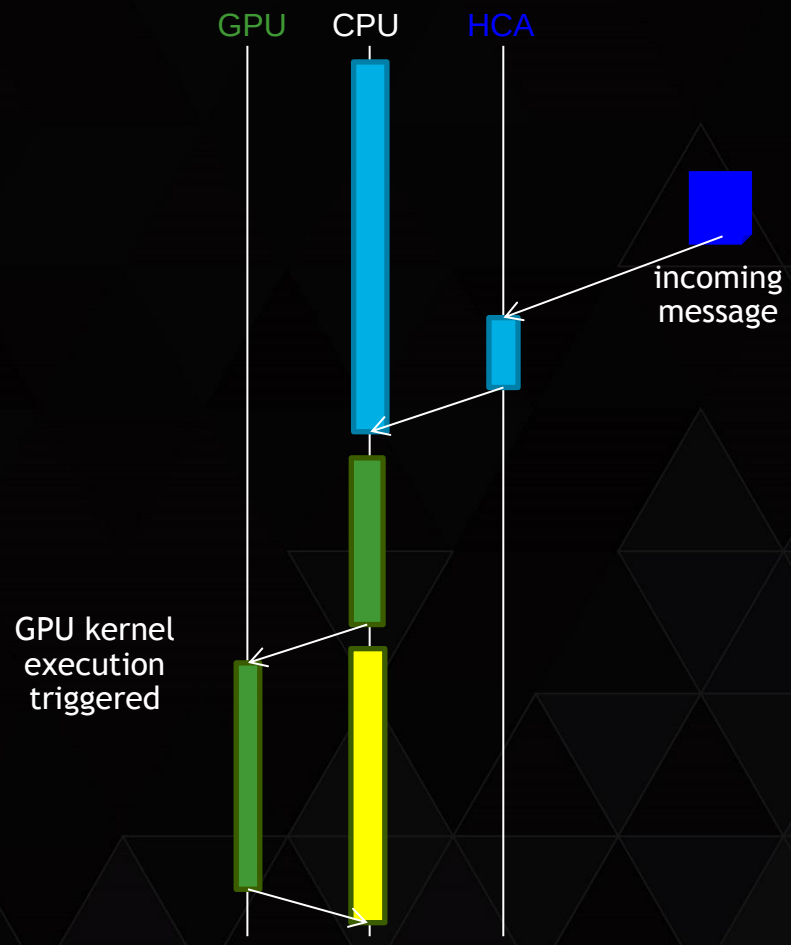
```
a_kernel<<<…,stream>>>(buf);
gds_stream_queue_send(stream,qp,buf);
gds_stream_wait_cq(stream,txcq);
b_kernel<<…,stream>>(buf);
```

No CPU in critical path!
Improve Scaling!



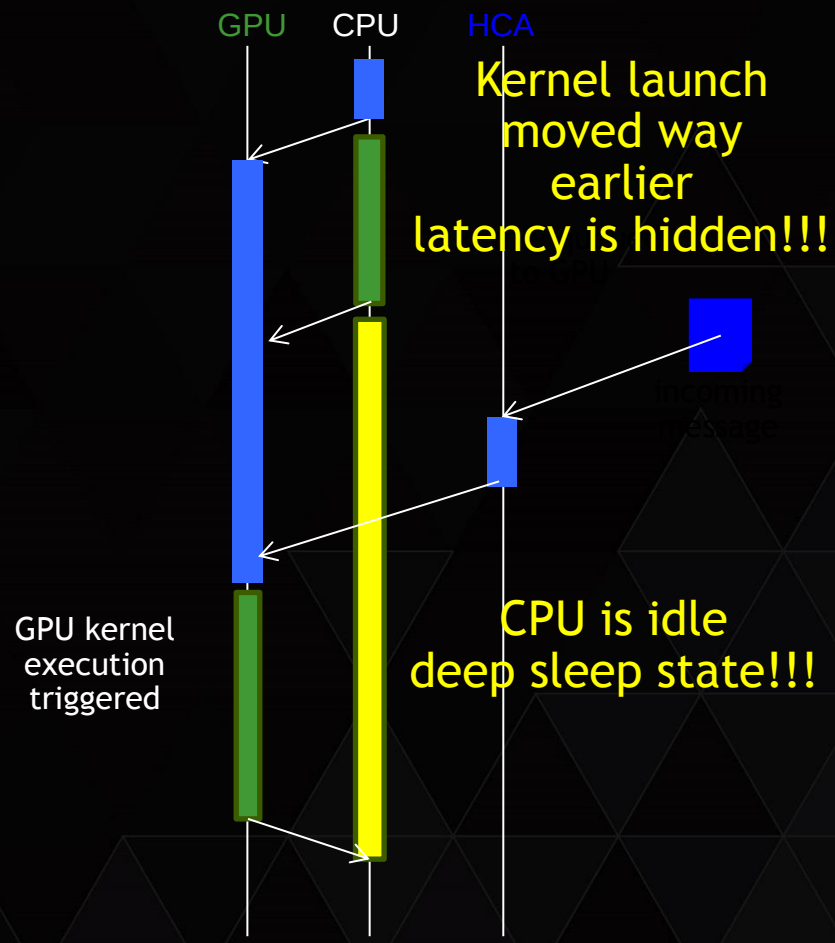
RECEIVE+KERNEL
NORMAL FLOW

```
while (!done) ibv_poll_cq();
a_kernel<<<…,stream>>>(buf);
cuStreamSynchronize(stream);
```



RECEIVE+KERNEL
GPUDIRECT ASYNC

```
gds_stream_wait_cq(stream,rx_cq);
a_kernel<<<…,stream>>>(buf);
cuStreamSynchronize(stream);
```



USE CASE SCENARIOS

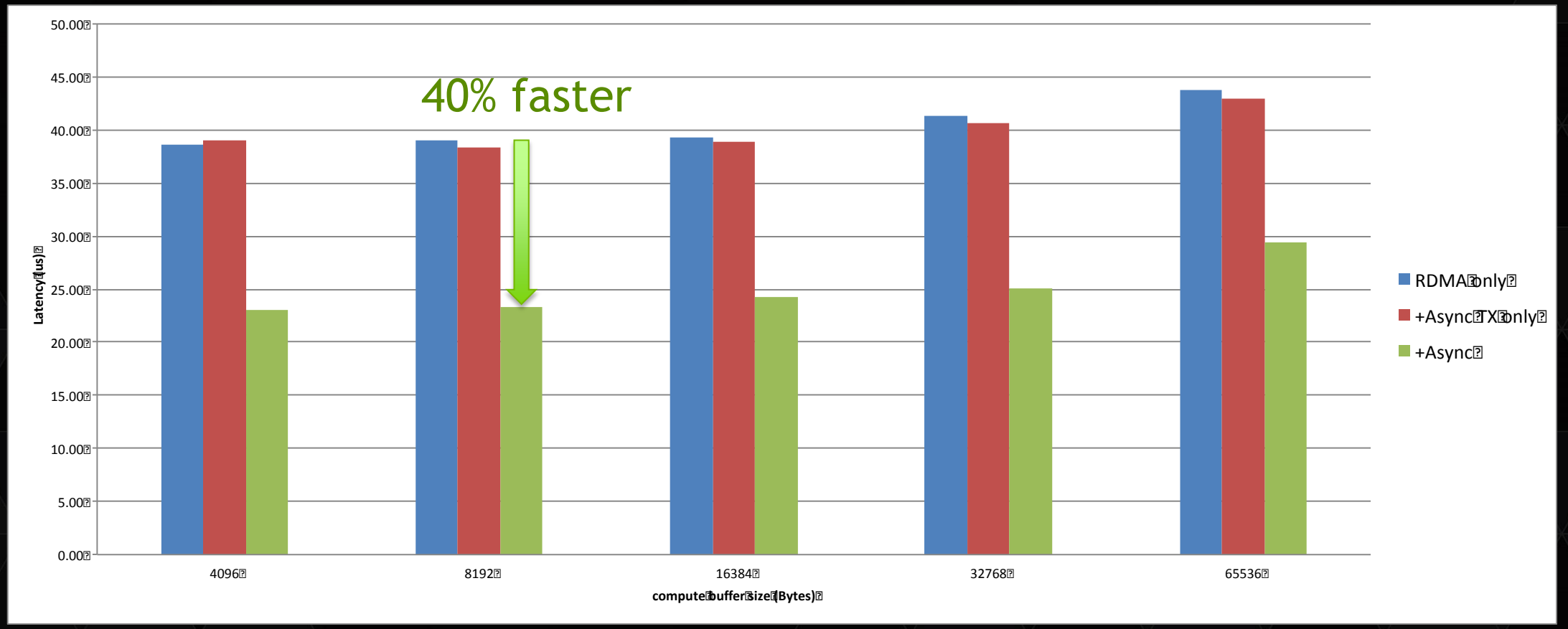
Performance mode (~ Top500)

- enable batching
- increase performance
- CPU available, additional GFlops

Economy mode (~ Green500)

- enable GPU IRQ waiting mode
- free more CPU cycles
- Optionally slimmer CPU

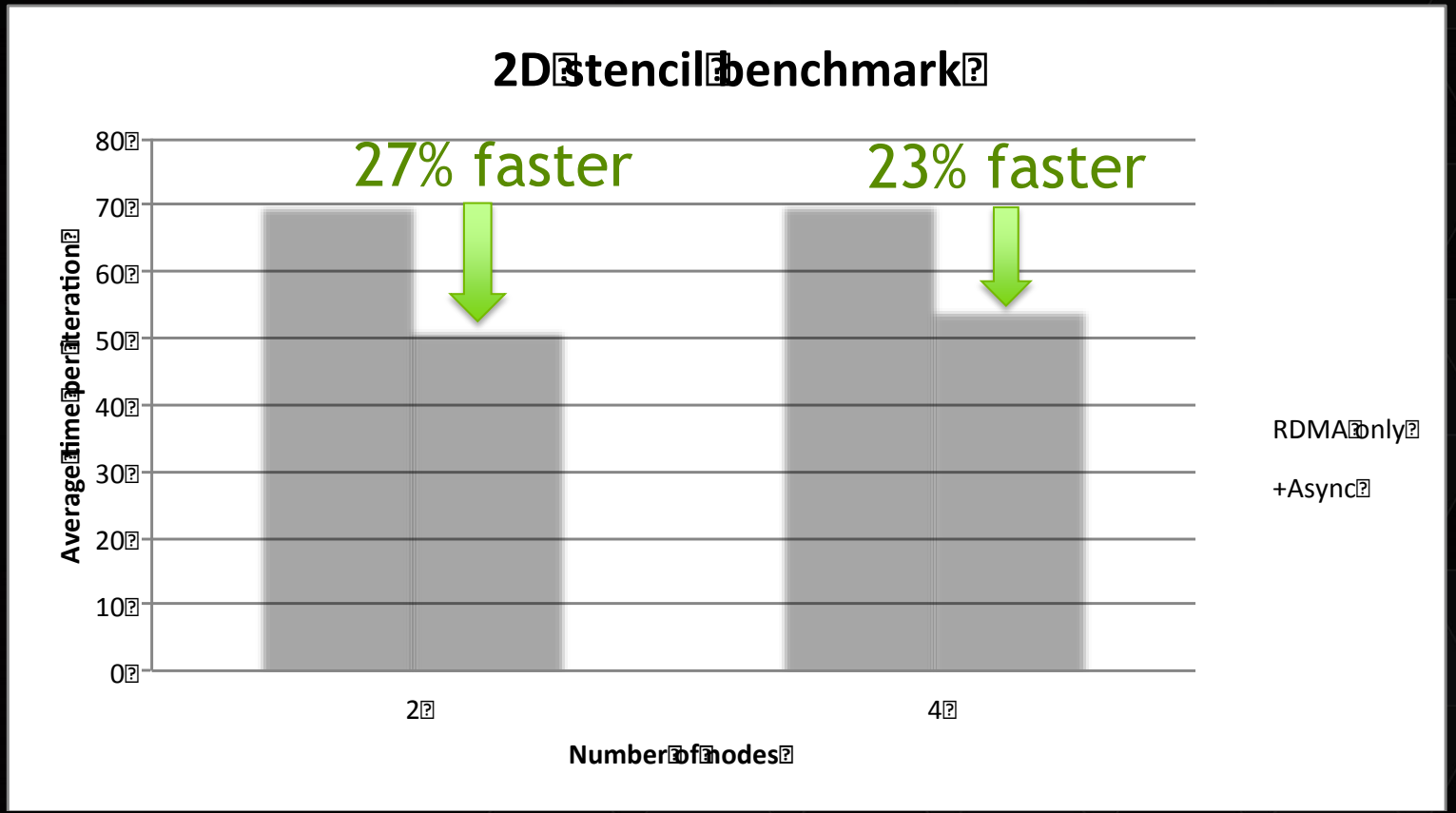
PERFORMANCE MODE



[*] modified ud_pingpong test: recv+GPU kernel+send on each side.
2 nodes: Ivy Bridge Xeon + K40 + Connect-IB + MLNX switch, 10000 iterations, message size: 128B, batch size: 20

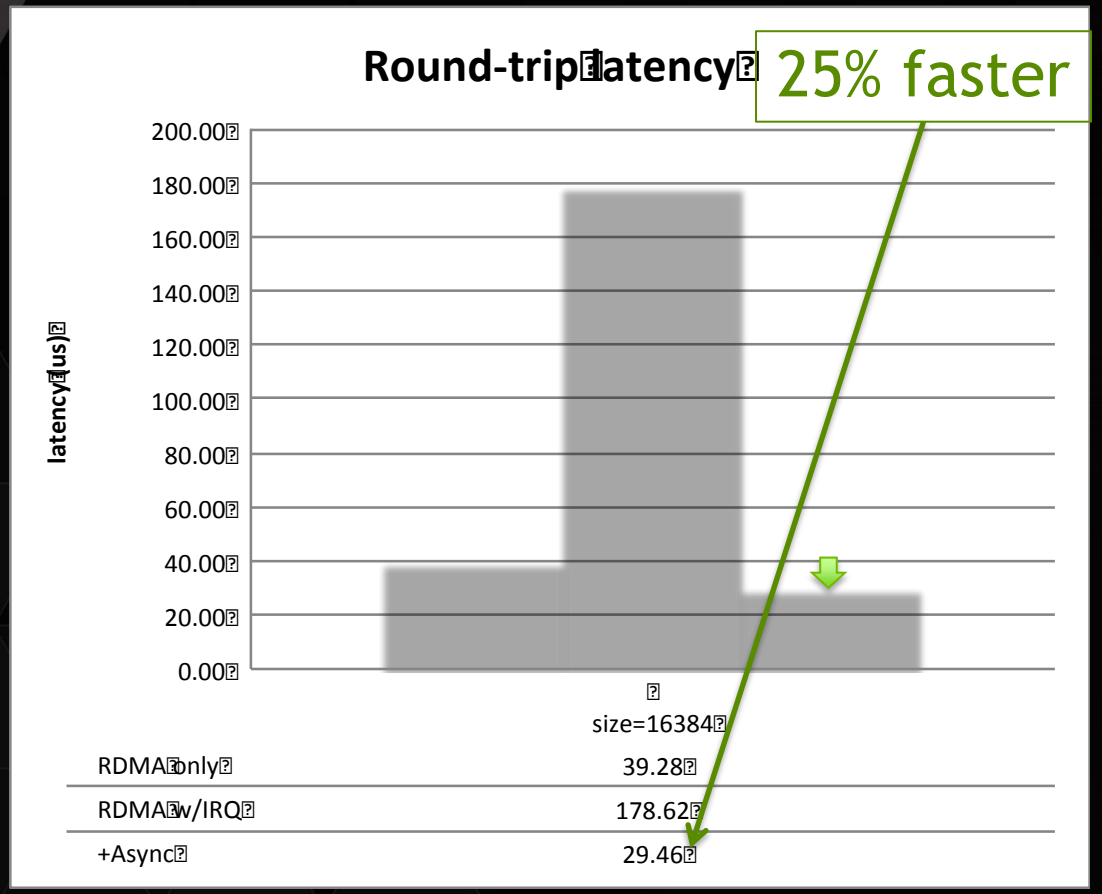
2D STENCIL BENCHMARK

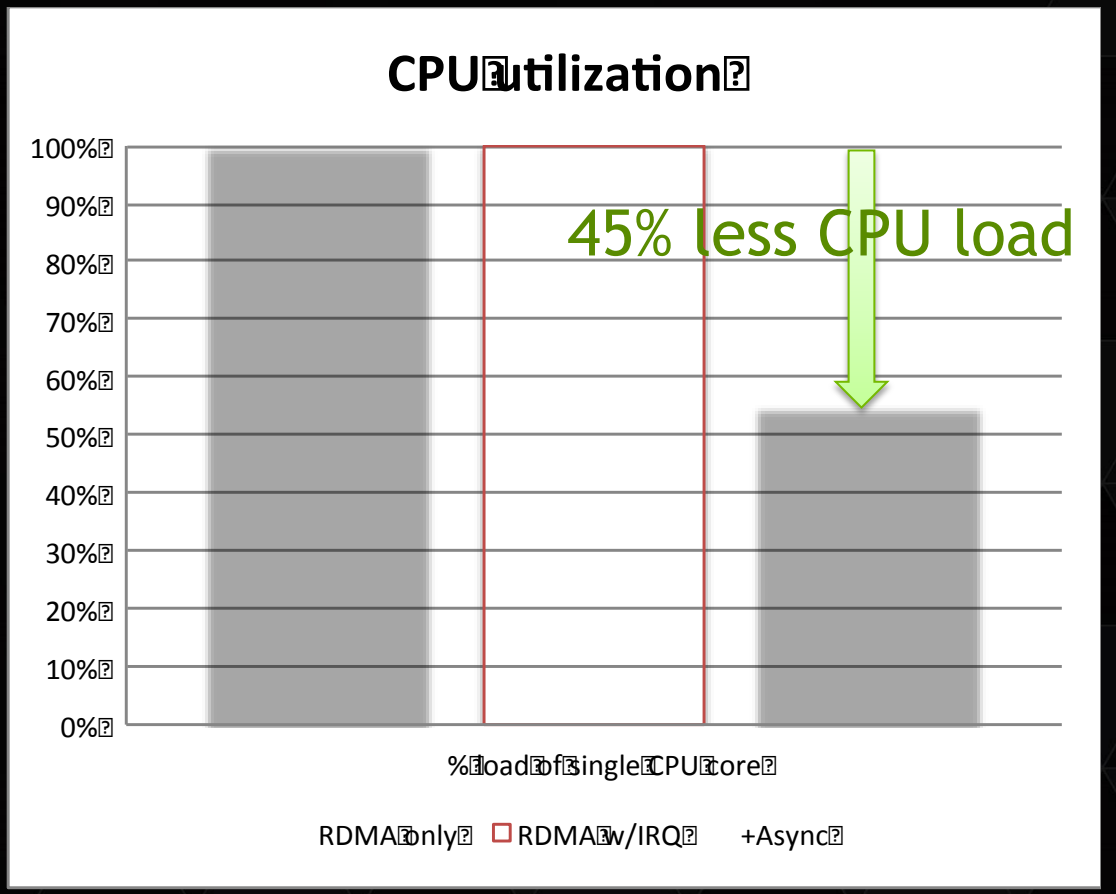
- weak scaling
- 256^2 local lattice
- 2x1, 2x2 node grids
- 1 GPU per node



[*] 4 nodes: Ivy Bridge Xeon + K40 + Connect-IB + MLNX switch

ECONOMY MODE





[*] modified ud_pingpong test, HW same as in previous slide

SUMMARY

- Meet Async, next generation of GPUDirect
- GPU orchestrates network operations
- CPU off the critical path
- **40% faster**, **45% less** CPU load

Excited about these topics ?

collaborations & jobs @NVIDIA

NVIDIA REGISTERED DEVELOPER PROGRAMS

- Everything you need to develop with NVIDIA products
- Membership is your first step in establishing a working relationship with NVIDIA Engineering
  - Exclusive access to pre-releases
  - Submit bugs and features requests
  - Stay informed about latest releases and training opportunities
  - Access to exclusive downloads
  - Exclusive activities and special offers
  - Interact with other developers in the NVIDIA Developer Forums

REGISTER FOR FREE AT: developer.nvidia.com

THANK YOU

JOIN THE CONVERSATION

#GTC15

PERFORMANCE VS ECONOMY

Performance mode

PowerTOP 2.3 Overview | Idle stats | Frequency

Package		CPU 0		
C0 polling	0.0%	C0 polling	0.0%	0.0 ms
C1-IVB	0.0%	C1-IVB	0.0%	0.0 ms
C3-IVB	0.0%	C3-IVB	0.0%	0.0 ms
C6-IVB	89.1%	C6-IVB	0.0%	0.0 ms
		CPU 1		
		C0 polling	0.0%	0.0 ms
		C1-IVB	0.0%	0.0 ms
		C3-IVB	0.0%	0.0 ms
		C6-IVB	98.8%	83.5 ms

Economy mode

PowerTOP 2.3 Overview | Idle stats | Frequenc

Package		CPU 0		
C0 polling	0.0%	C0 polling	0.0%	0.0 ms
C1-IVB	0.8%	C1-IVB	7.9%	1.1 ms
C3-IVB	1.0%	C3-IVB	10.1%	1.1 ms
C6-IVB	91.3%	C6-IVB	23.2%	1.1 ms
		CPU 1		
		C0 polling	0.0%	0.0 ms
		C1-IVB	0.0%	0.0 ms
		C3-IVB	0.0%	0.0 ms
		C6-IVB	99.9%	126.1 ms

[*] modified ud_pingpong test, HW same as in previous slide, NUMA binding to socket0/core0, SBIOS power-saving profile