

OpenACC Course

Office Hour 4: Advanced OpenACC Techniques

Jiri Kraus, NVIDIA Developer Technologies



Course Syllabus

Oct 1: Introduction to OpenACC

Oct 6: Office Hours

Oct 15: Profiling and Parallelizing with the OpenACC Toolkit

Oct 20: Office Hours

Oct 29: Expressing Data Locality and Optimizations with OpenACC

Nov 3: Office Hours

Nov 12: Advanced OpenACC Techniques

Nov 24: Office Hours

Recordings:

<https://developer.nvidia.com/openacc-course>

Answered Questions and Recordings

<https://developer.nvidia.com/openacc-course>

Questions from the last class

Q1: Does the new PGI OpenACC for Multicore feature work with code like this?

Michael Wolfe's blog post "OpenACC for Multicore CPUs"
<http://www.pgroup.com/lit/articles/insider/v6n3a1.htm>

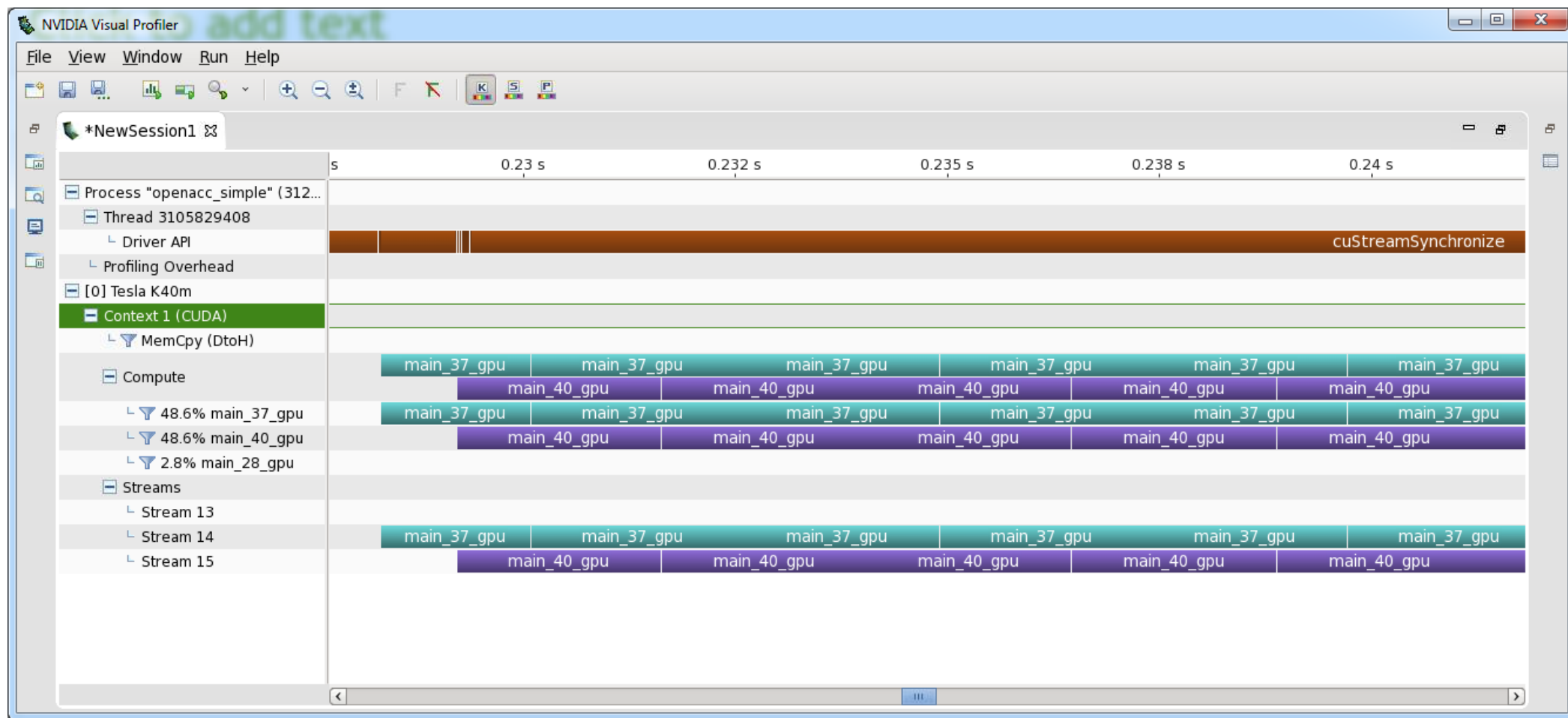
Q2: Can one achieve overlapping of two kernels?

Q3: When using device pointers with CUDA-aware MPI, are acceleration technologies like GPUDirect P2P and RDMA automatically used?

Can one achieve overlapping of two kernels?

```
35: for(int iter=0; iter<num_iters; ++iter) {
36:     #pragma acc parallel loop num_gangs(120) async(0)
37:     for (int i = 0; i < n; i++)
38:         y[i] += x[i];
39:     #pragma acc parallel loop num_gangs(120) async(1)
40:     for (int i = 0; i < n; i++)
41:         z[i] += w[i];
42: }
43: #pragma acc wait
```

Can one achieve overlapping of two kernels?



CUDA-aware MPI and GPUDirect

Example

MPI Rank 0 MPI_Send from GPU Buffer

MPI Rank 1 MPI_Recv to GPU Buffer

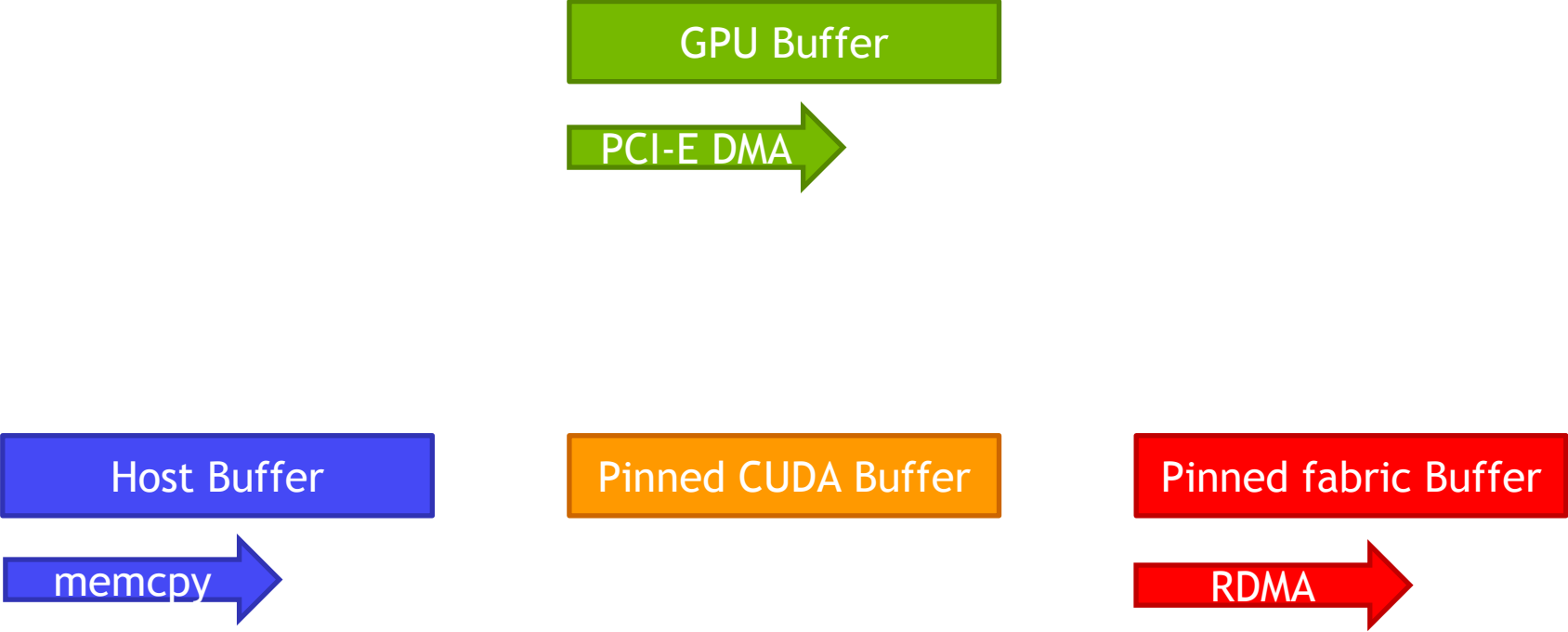
Show how CUDA+MPI works in principle

Depending on the MPI implementation, message size, system setup, ... situation might be different

Two GPUs in two nodes

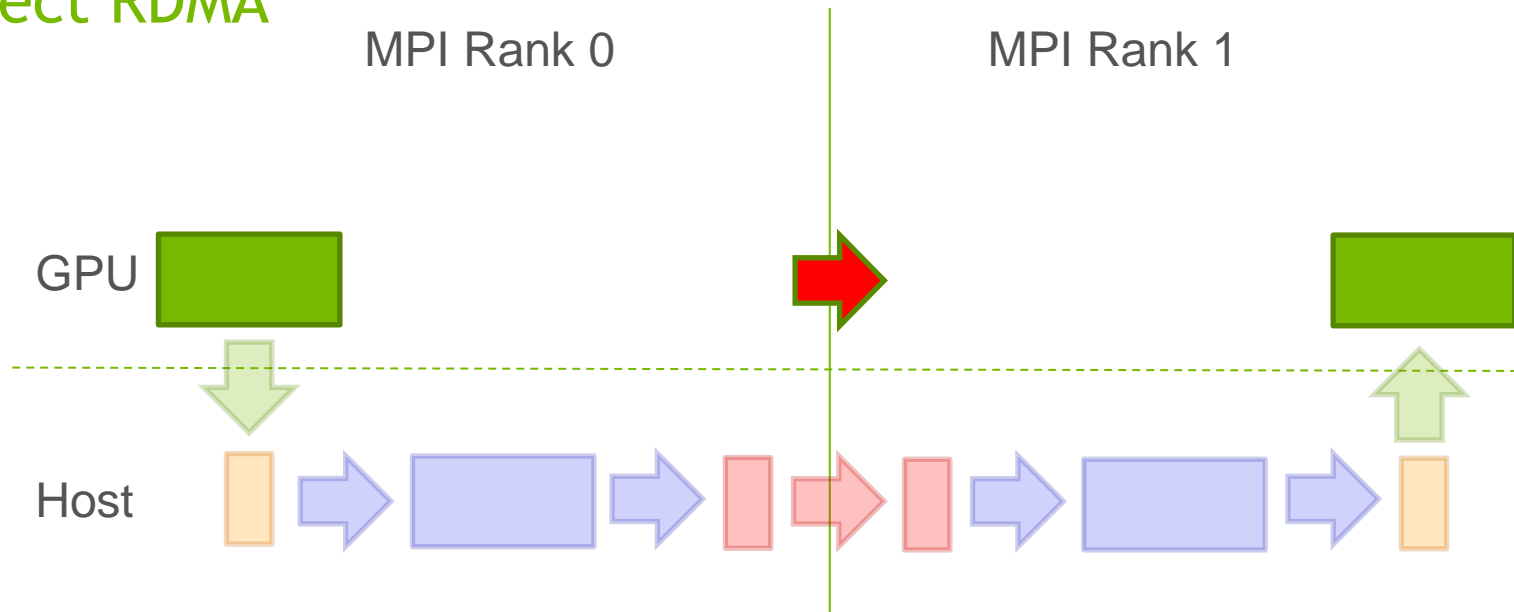
CUDA-aware MPI and GPUDirect

Example



CUDA-aware MPI and GPUDirect

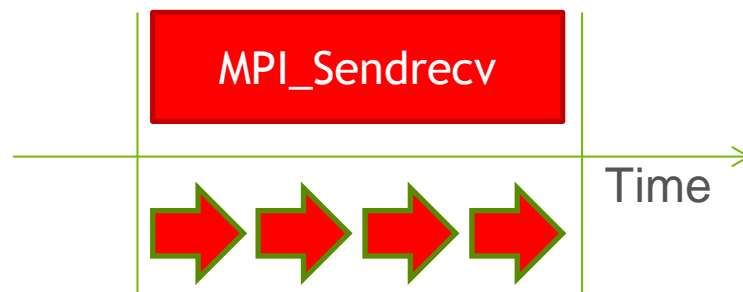
GPUDirect RDMA



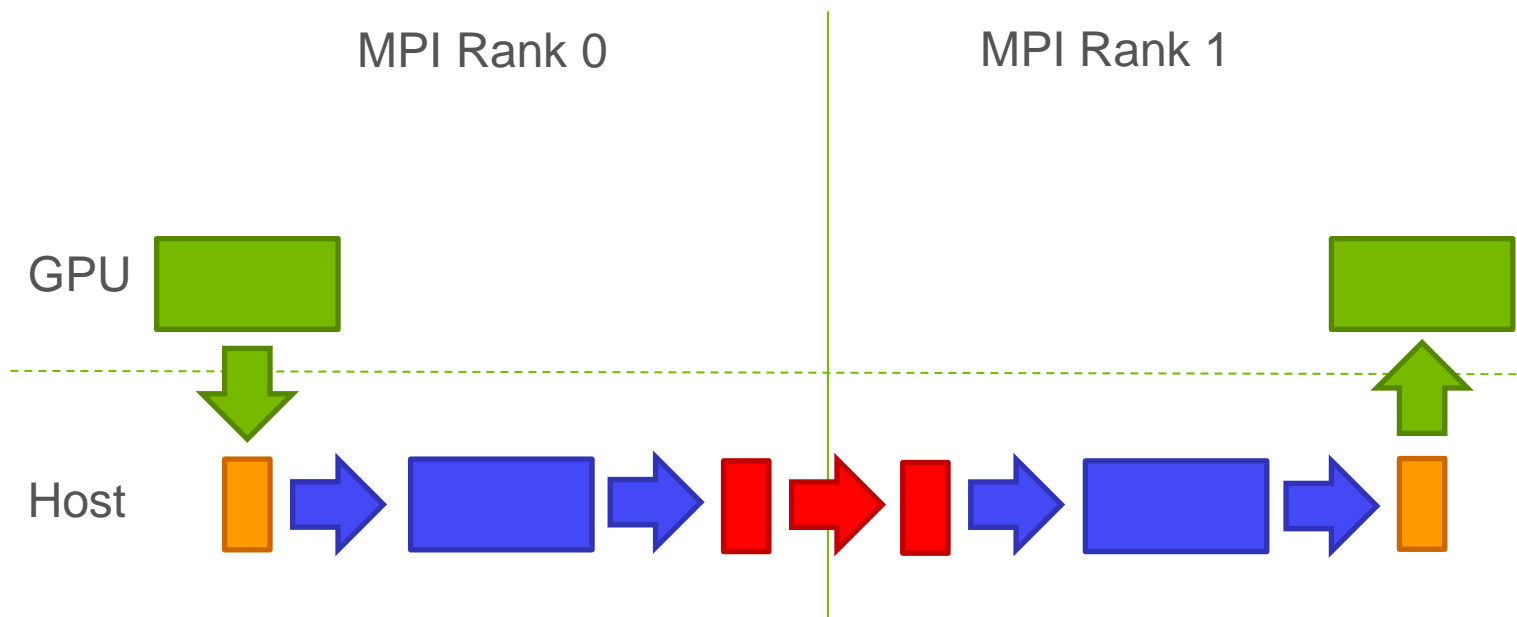
```
#pragma acc host_data use_device(s_buf)
MPI_Send(s_buf,size,MPI_CHAR,1,tag,MPI_COMM_WORLD);
#pragma acc host_data use_device(s_buf)
MPI_Recv(r_buf,size,MPI_CHAR,0,tag,MPI_COMM_WORLD,&stat);
```

CUDA-aware MPI and GPUDirect

GPUDirect RDMA

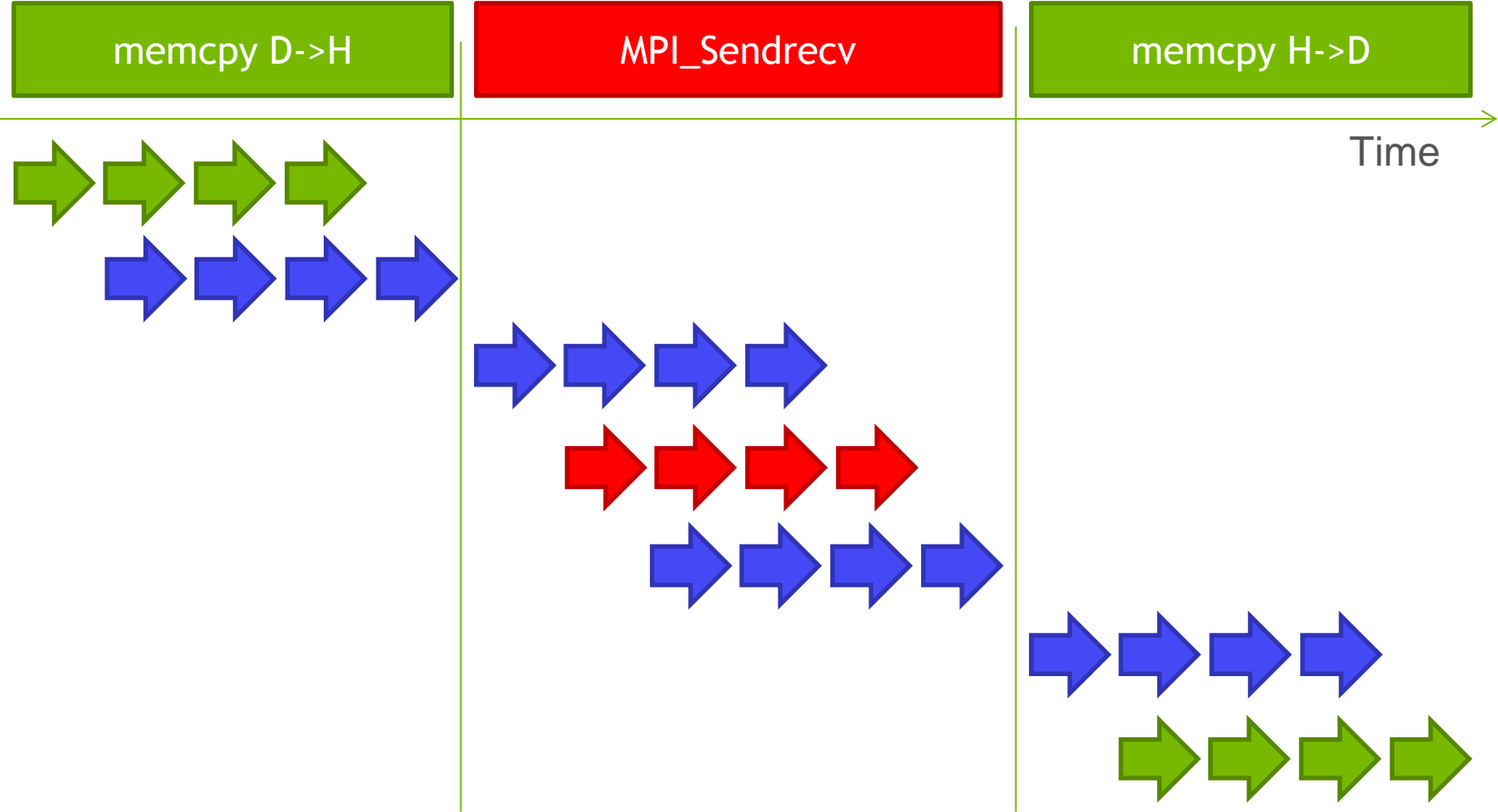


Regular MPI and GPU Buffers



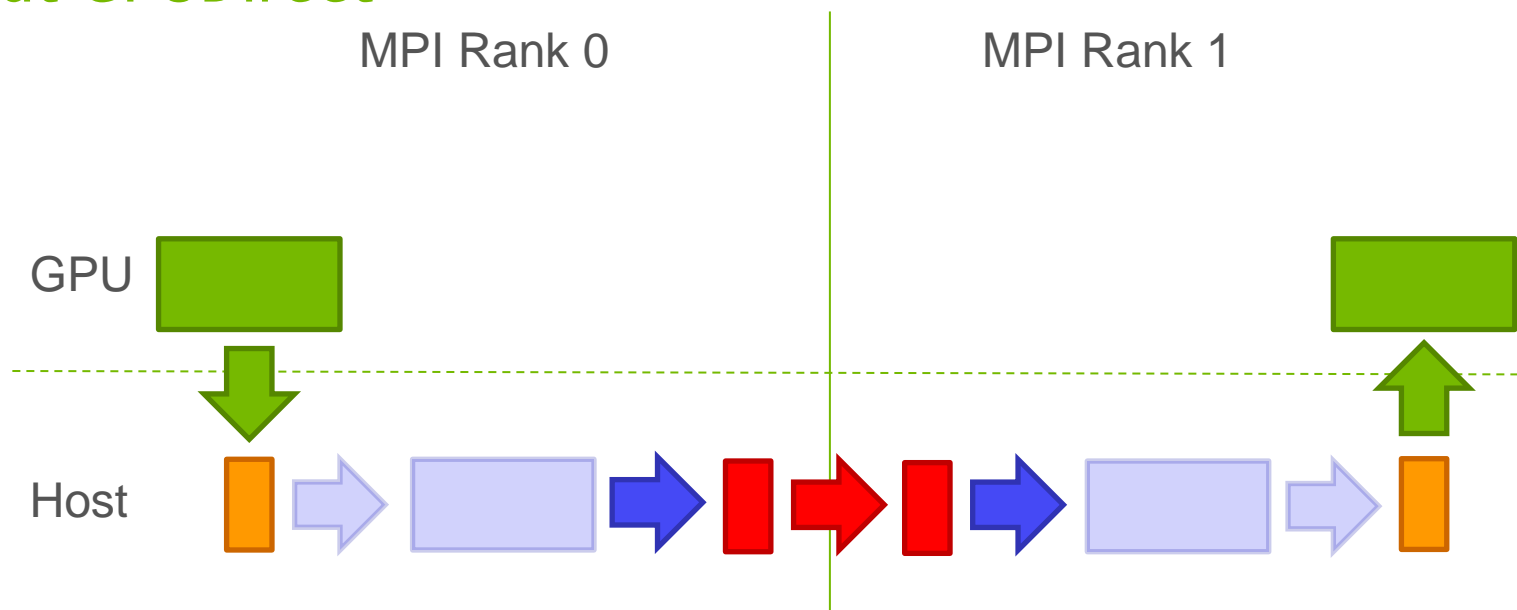
```
#pragma acc update self(s_buf[0:size])  
MPI_Send(s_buf,size,MPI_CHAR,1,tag,MPI_COMM_WORLD);  
  
MPI_Recv(r_buf,size,MPI_CHAR,0,tag,MPI_COMM_WORLD,&stat);  
#pragma acc update device(r_buf[0:size])
```

Regular MPI and GPU Buffers



CUDA-aware MPI

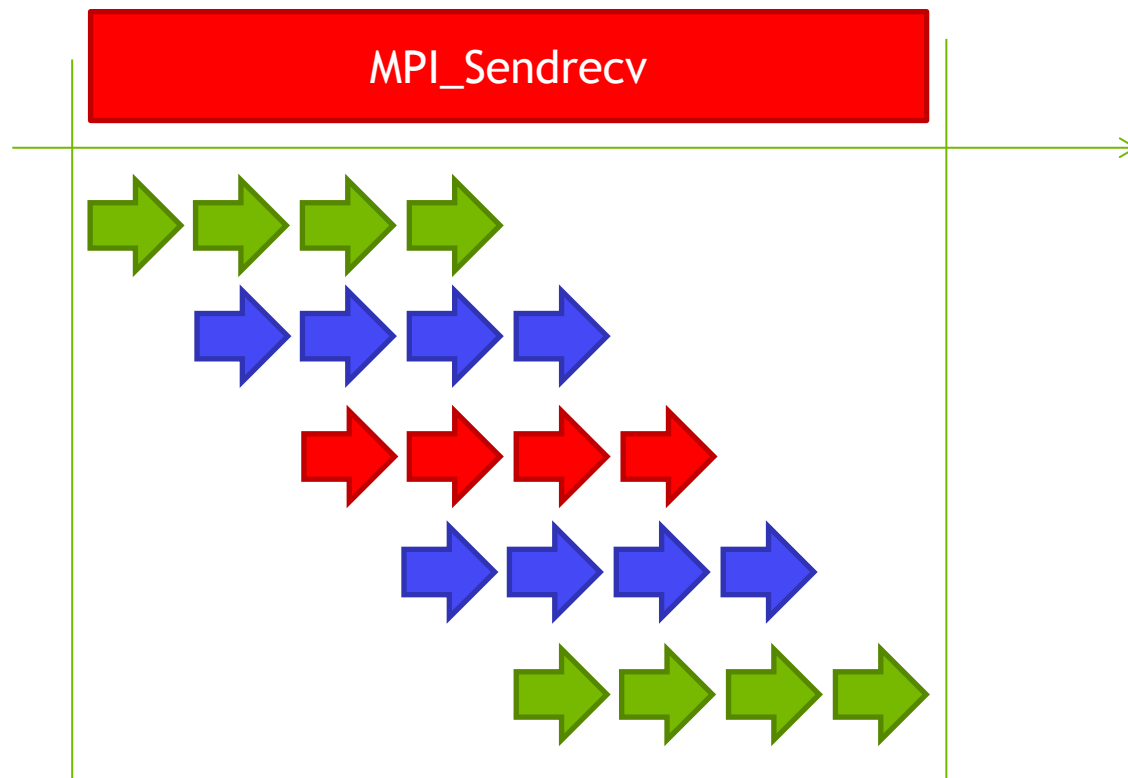
Without GPUDirect



```
#pragma acc host_data use_device(s_buf)
MPI_Send(s_buf, size, MPI_CHAR, 1, tag, MPI_COMM_WORLD);
#pragma acc host_data use_device(s_buf)
MPI_Recv(r_buf, size, MPI_CHAR, 0, tag, MPI_COMM_WORLD, &stat);
```

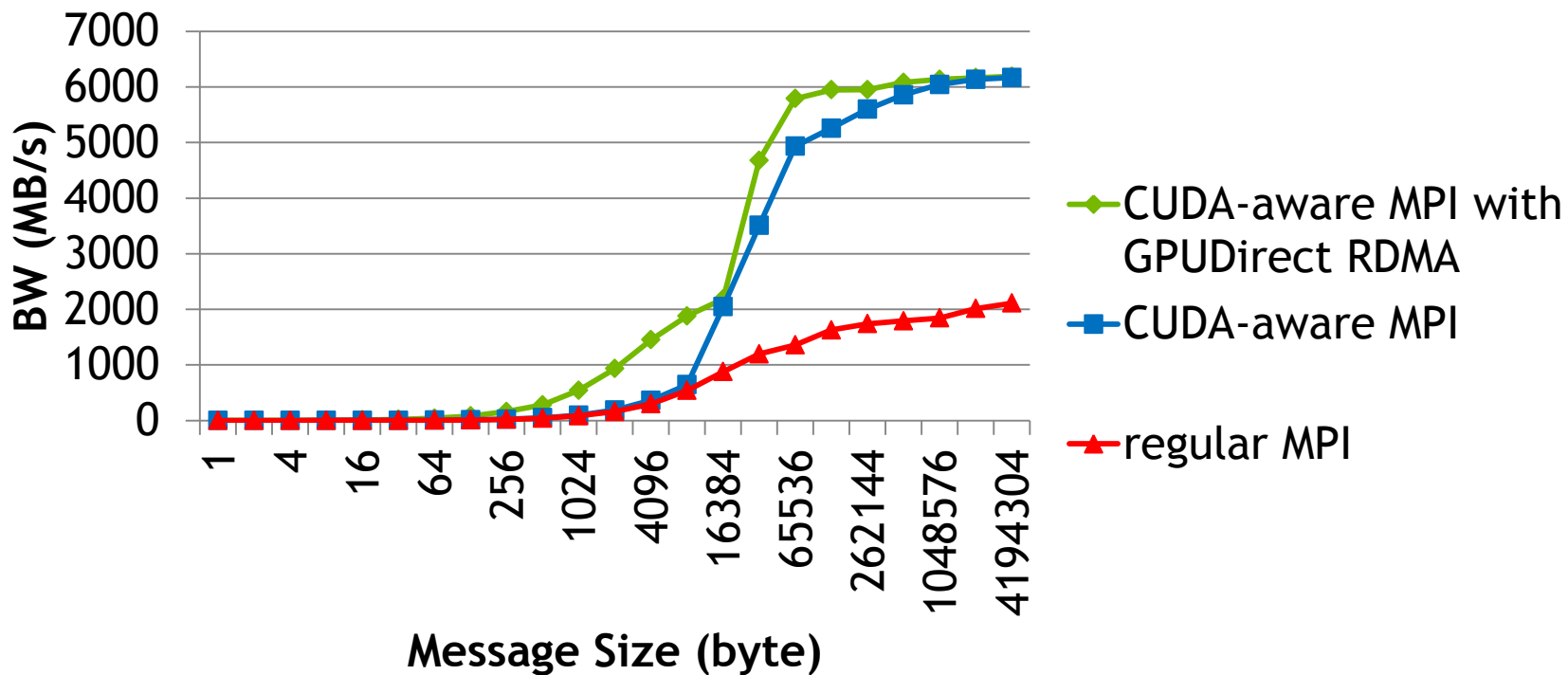
CUDA-aware MPI

Without GPUDirect



CUDA-aware MPI and GPUDirect

OpenMPI 1.8.4 MLNX FDR IB (4X) Tesla K40@875



Latency (1 byte) 19.79 us 17.97 us 5.70 us

Where to find help

- OpenACC Course Recordings and Q&A - <https://developer.nvidia.com/openacc-course>
- OpenACC on StackOverflow - <http://stackoverflow.com/questions/tagged/openacc>
- OpenACC Toolkit - <http://developer.nvidia.com/openacc>

Additional Resources:

- Parallel Forall Blog - <http://devblogs.nvidia.com/parallelforall/>
- GPU Technology Conference - <http://www.gputechconf.com/>
- OpenACC Website - <http://openacc.org/>