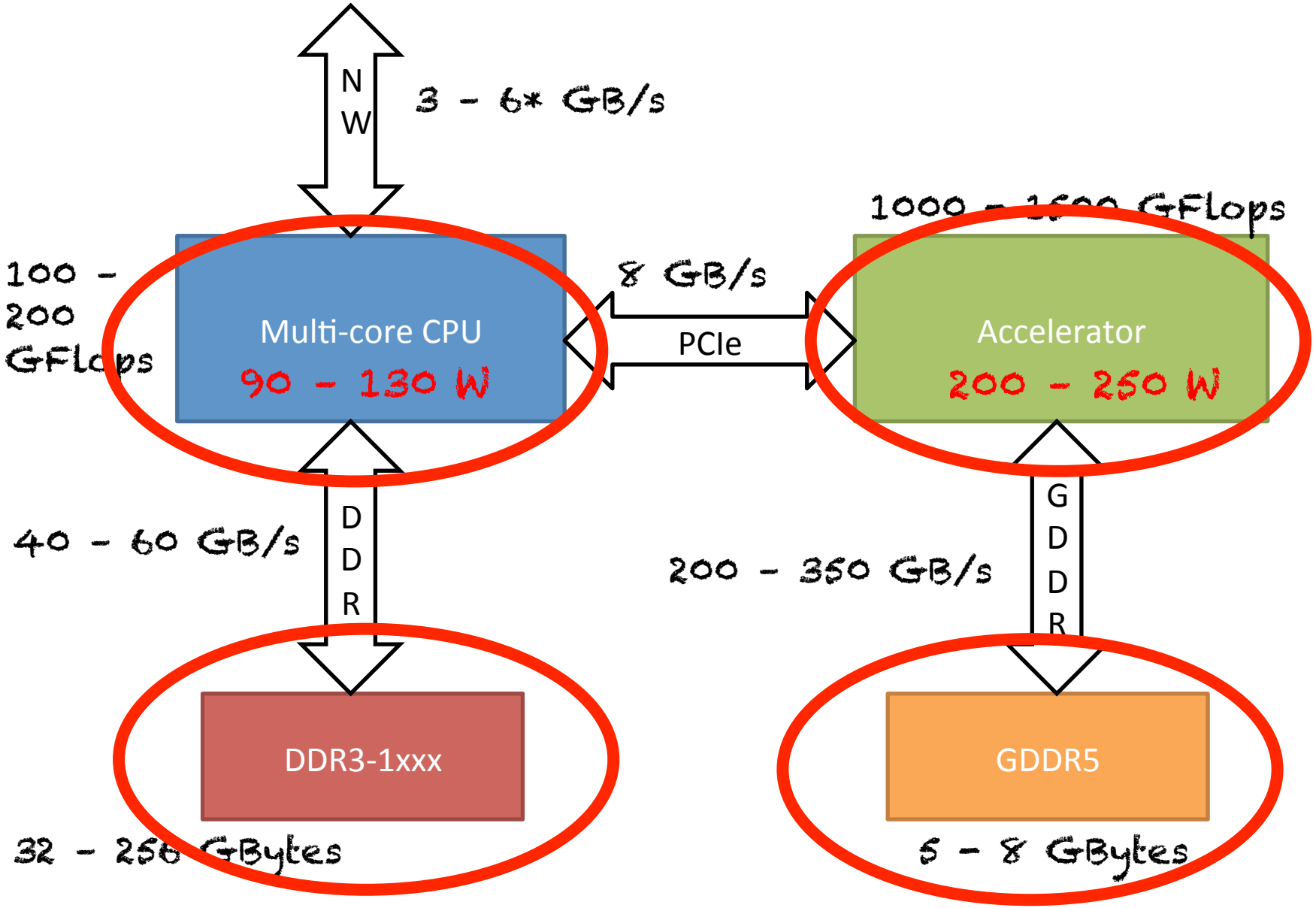
Direct MPI from NVIDIA Tesla and Intel Xeon Phi Accelerator Memories on an InfiniBand Cluster

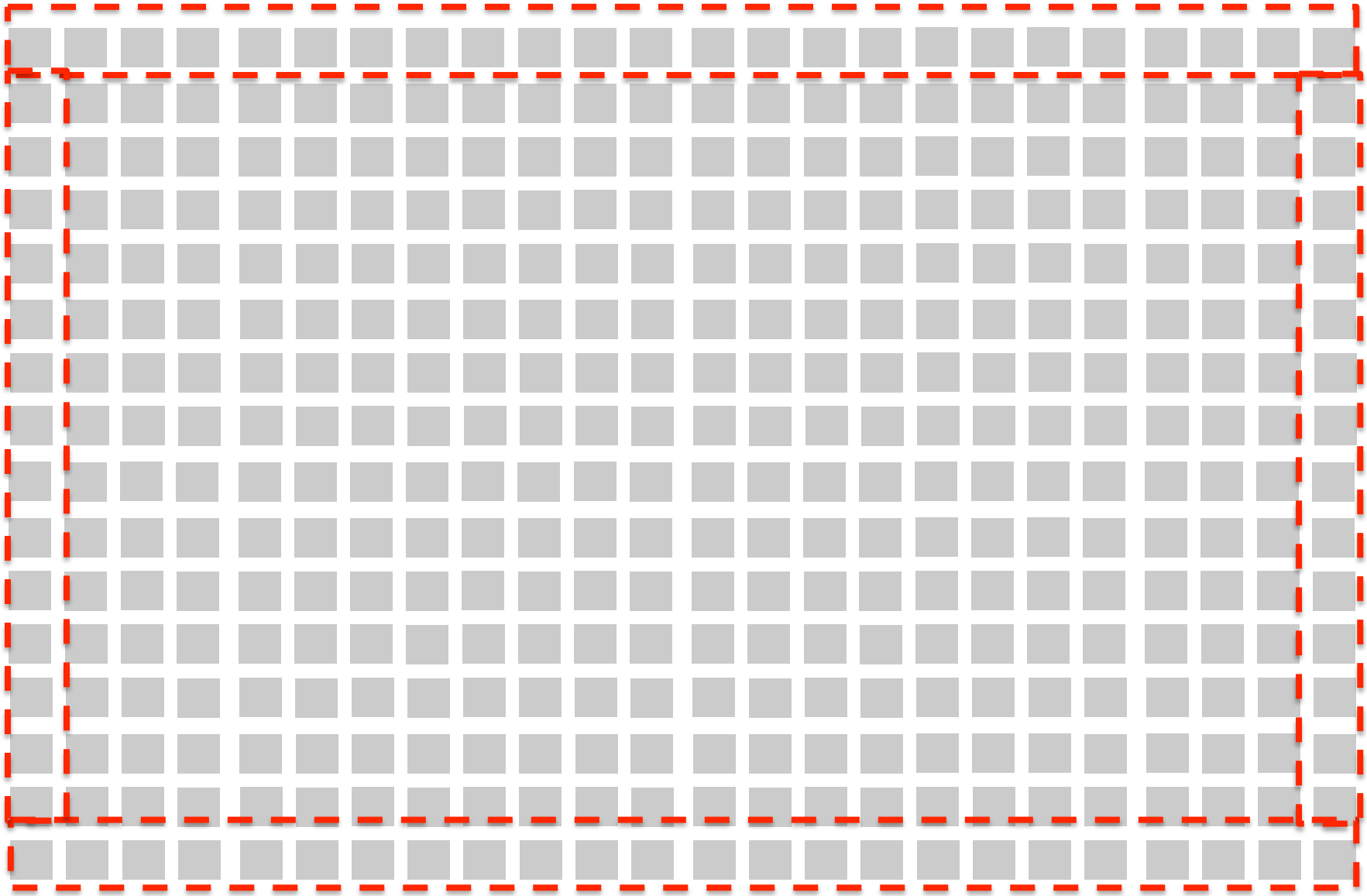
Sadaf Alam

HPC Advisory Council Meeting

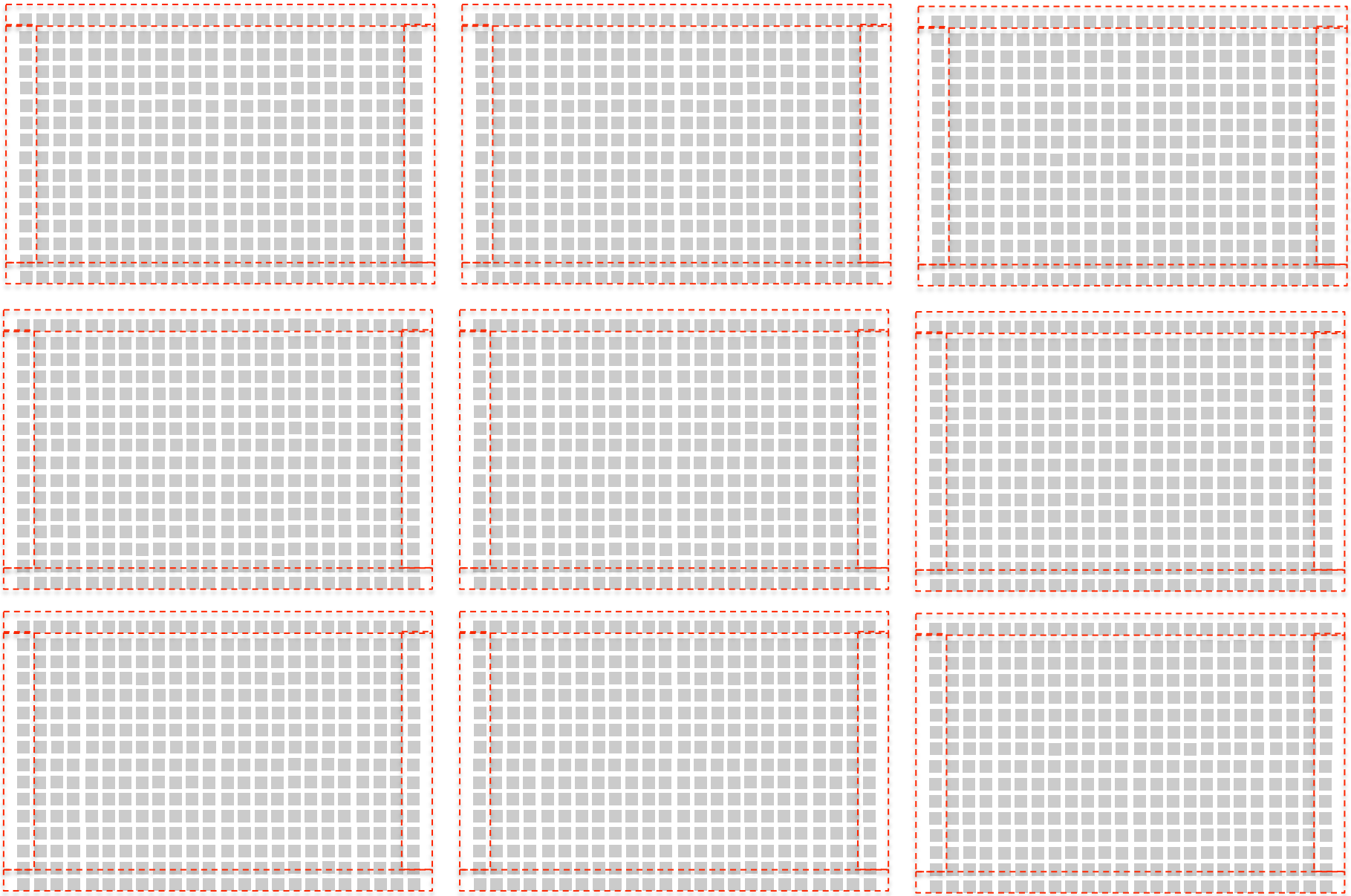
Lugano, March 2013


N
W
3 - 6* GB/s
1000 - 1500 GFlops
100 - 200 GFlops
Multi-core CPU
90 - 130 W
8 GB/s
PCIe
Accelerator
200 - 250 W
40 - 60 GB/s
D
D
R
200 - 350 GB/s
G
D
D
R
DDR3-1xxx
GDDR5
32 - 256 GBytes
5 - 8 GBytes

Scenario # 1



Scenario # 2



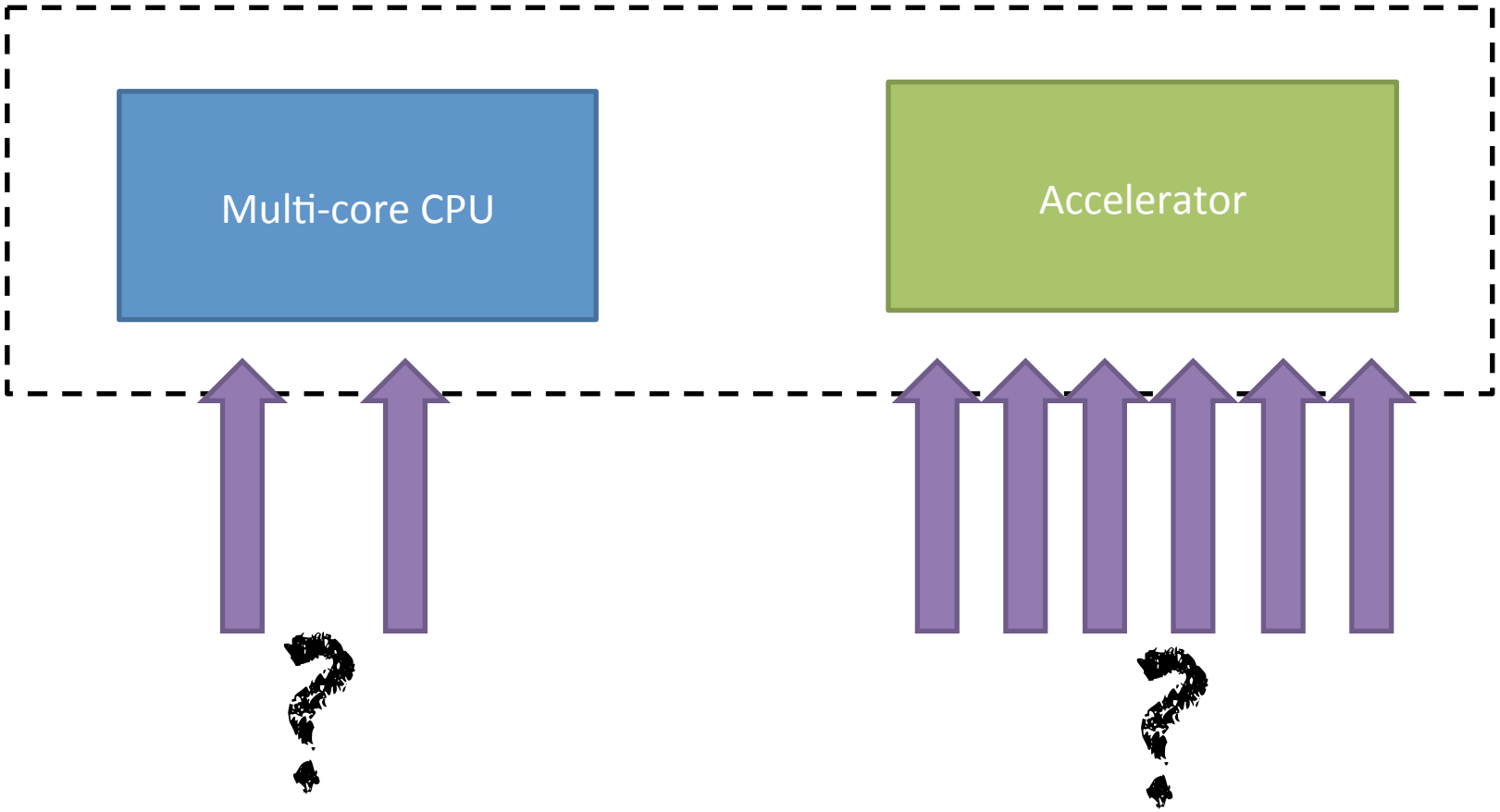
Scenario # 3

```
!$acc parallel loop
         do j=1,jmax
            do i=1,imax
               zsnd(i,j,1)=wrk2(i,j,2)
               zsnd(i,j,2)=wrk2(i,j,kmax-1)
            enddo
         enddo
!$acc end parallel loop
!$acc wait
      endif

call mpi_isend(zsnd(1,1,1), …)

!$acc parallel loop PRIVATE (s0,ss) reduction(+:wgosa)
         DO K=2,kmax-1
            DO J=2,jmax-1

call mpi_allreduce(wgosa, ...)
```

Example derived from the OpenACC version of the Himeno benchmark by Alistair Hart, Cray Inc.

Scenario # 4

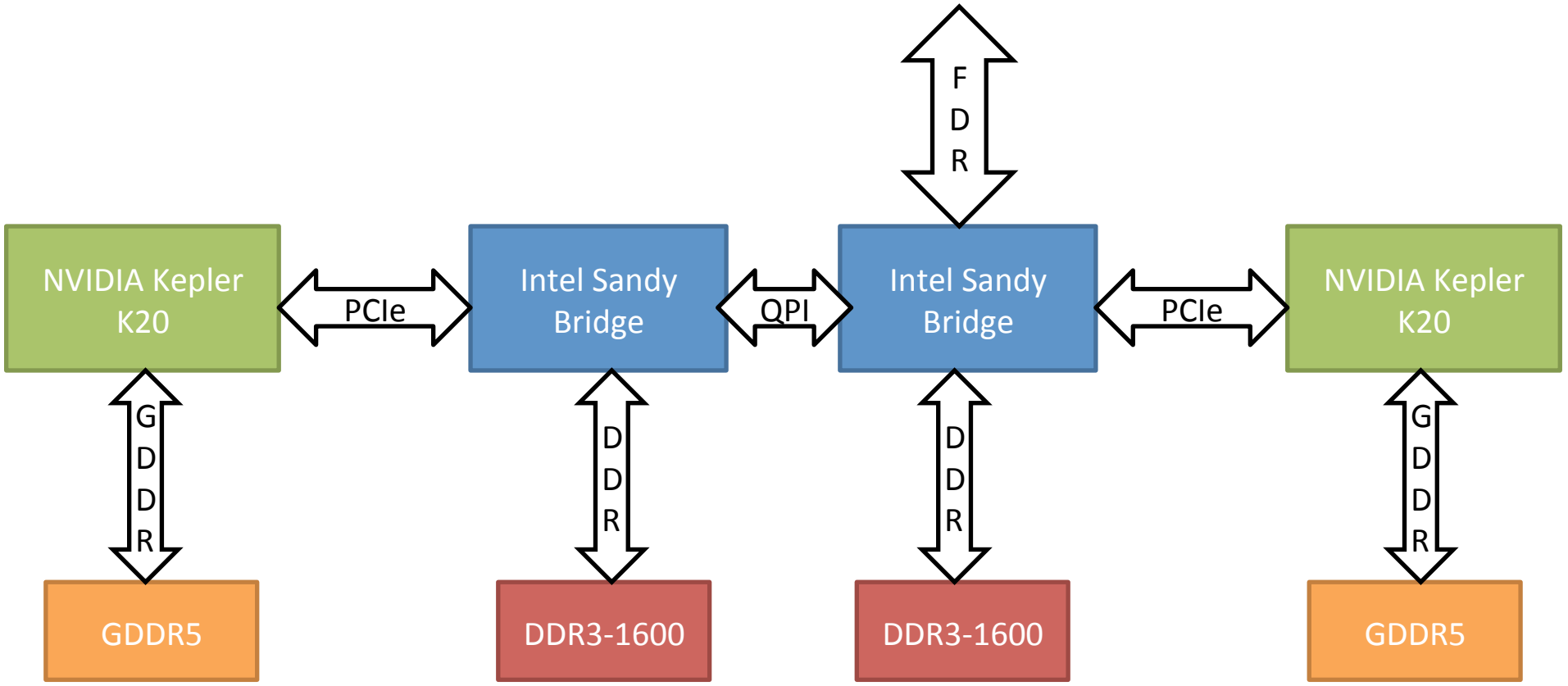
Asymmetric execution mode on Xeon Phi



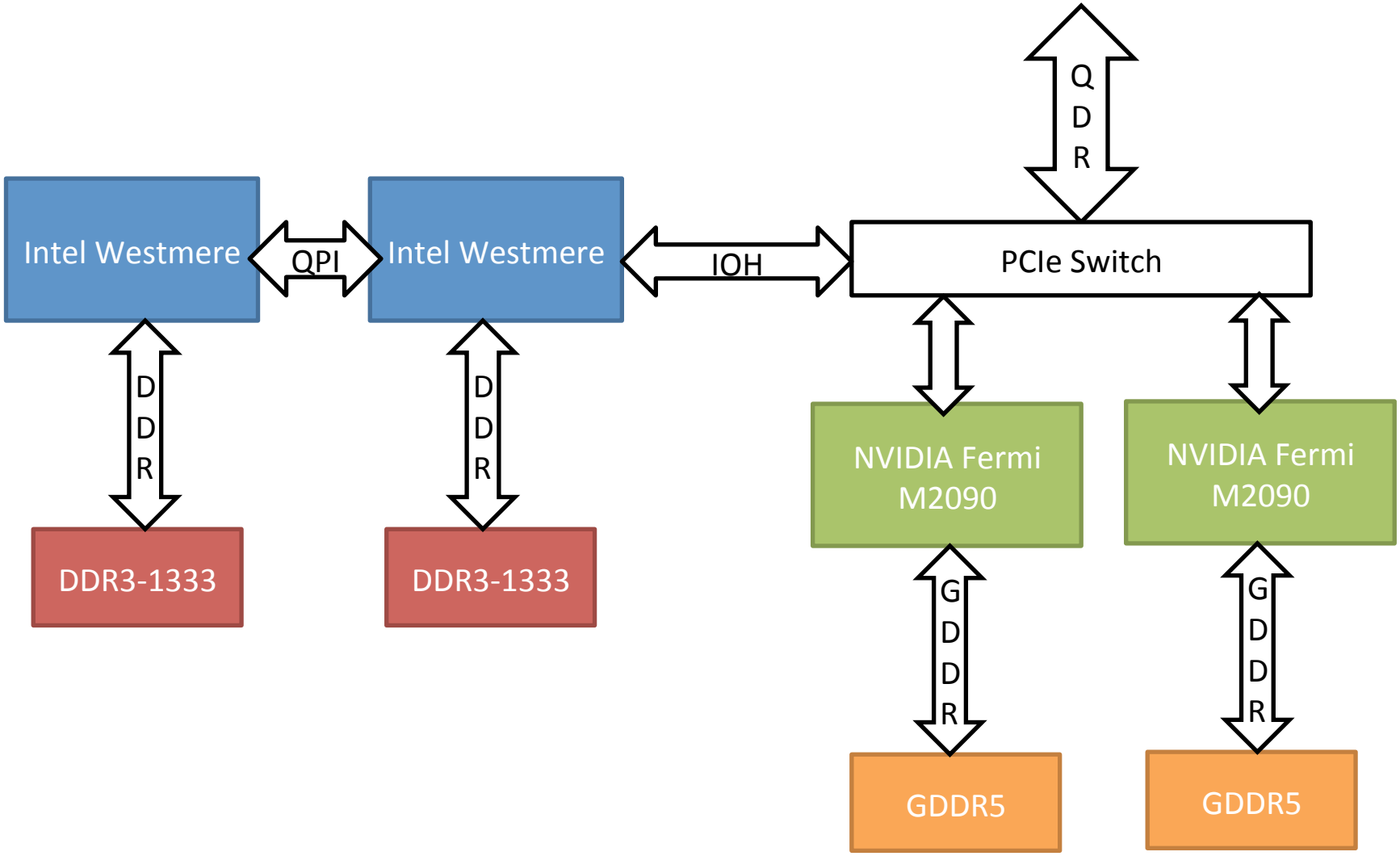
Motivation

- Improved performance for
  - Small messages transfers
  - Large messages transfers
  - Collective operations scalability
- Performance and productivity for high level interfaces e.g. OpenACC
- Load balancing in asymmetric programming modes

NVIDIA Kepler K20 Evaluation



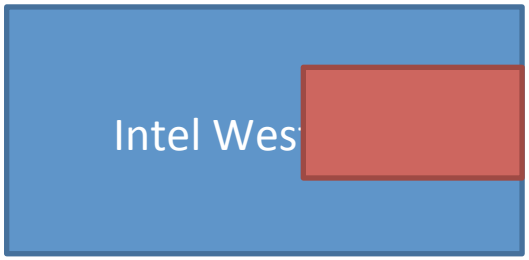
NVIDIA Fermi M2090 QDR Cluster Evaluation



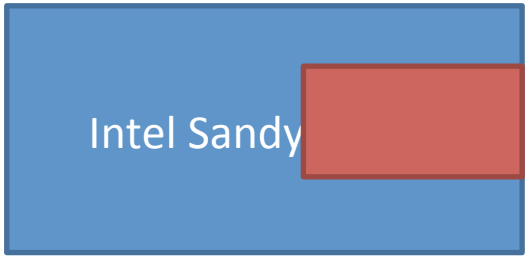
QDR Latency (8 bytes) = ~2 usec
QDR Bandwidth (1 Mbytes) = ~3 GB/s

FDR Latency (8 bytes) = ~ 1 usec
FDR Bandwidth (1 Mbytes) = ~ 6 GB/s

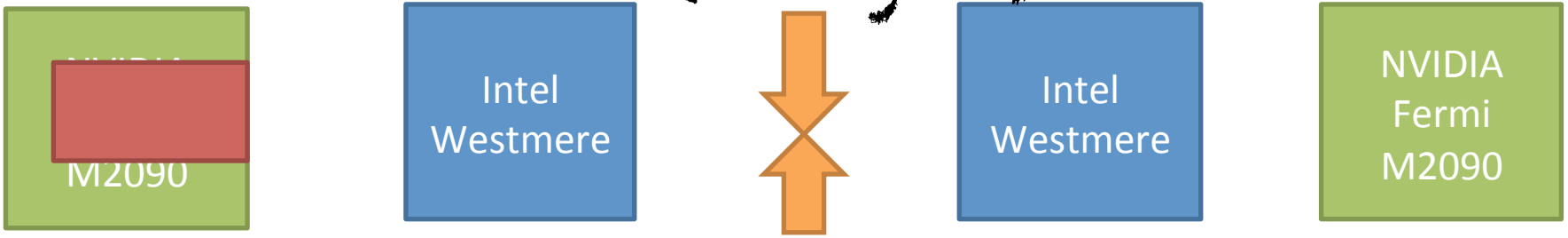
On node
Latency (8 bytes) = ~ 10 usec
Bandwidth (1 Mbytes) = ~ 5 GB/s



On node
Latency (8 bytes) = ~ 9 usec
Bandwidth (1 Mbytes) = ~ 5 GB/s

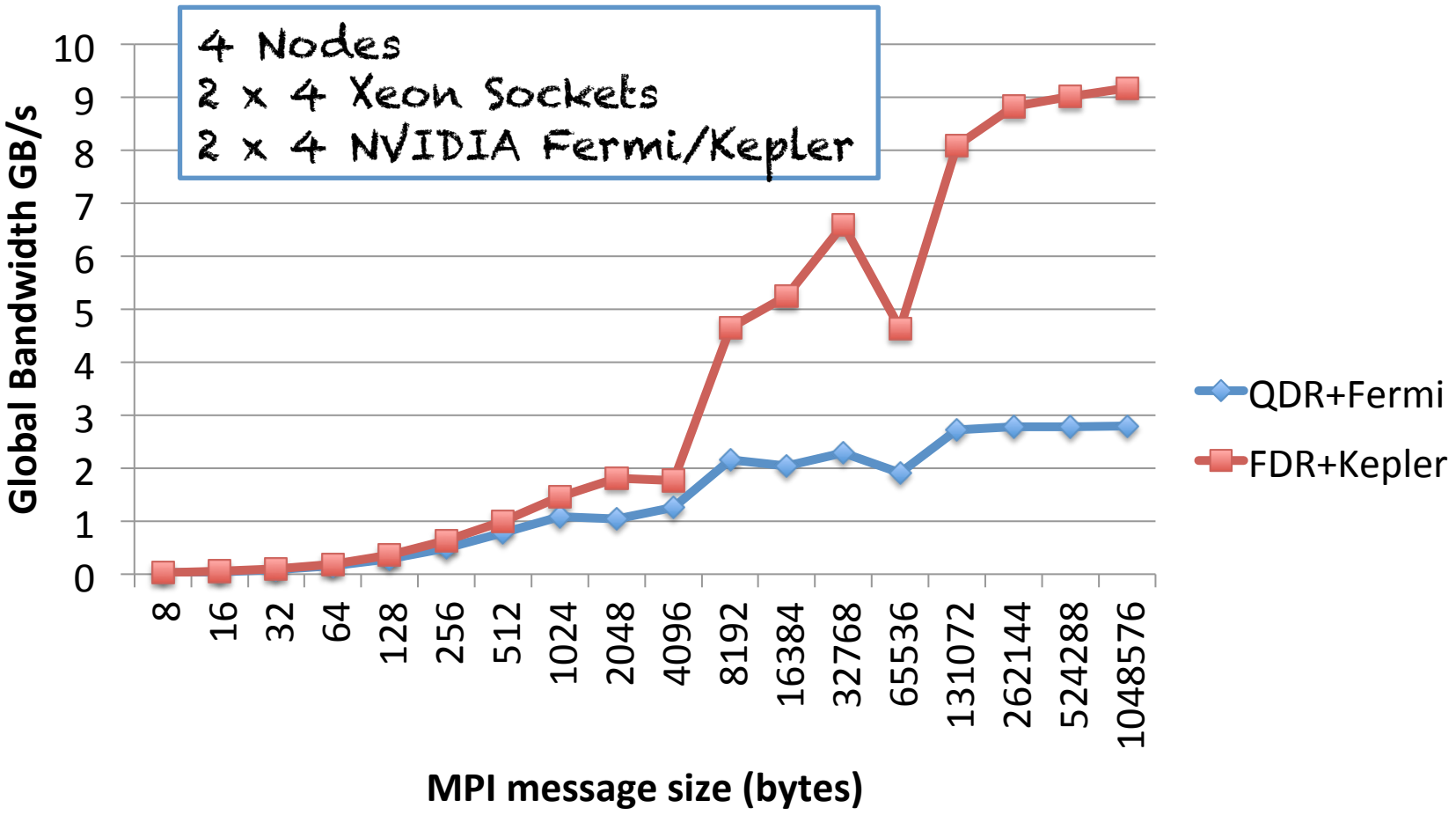


QDR Network
Latency (8 bytes) = ~ 20 usec
Bandwidth (1 Mbytes) = ~ 3 GB/s



FDR Network
Latency (8 bytes) = ~ 18 usec
Bandwidth (1 Mbytes) = ~ 6 GB/s

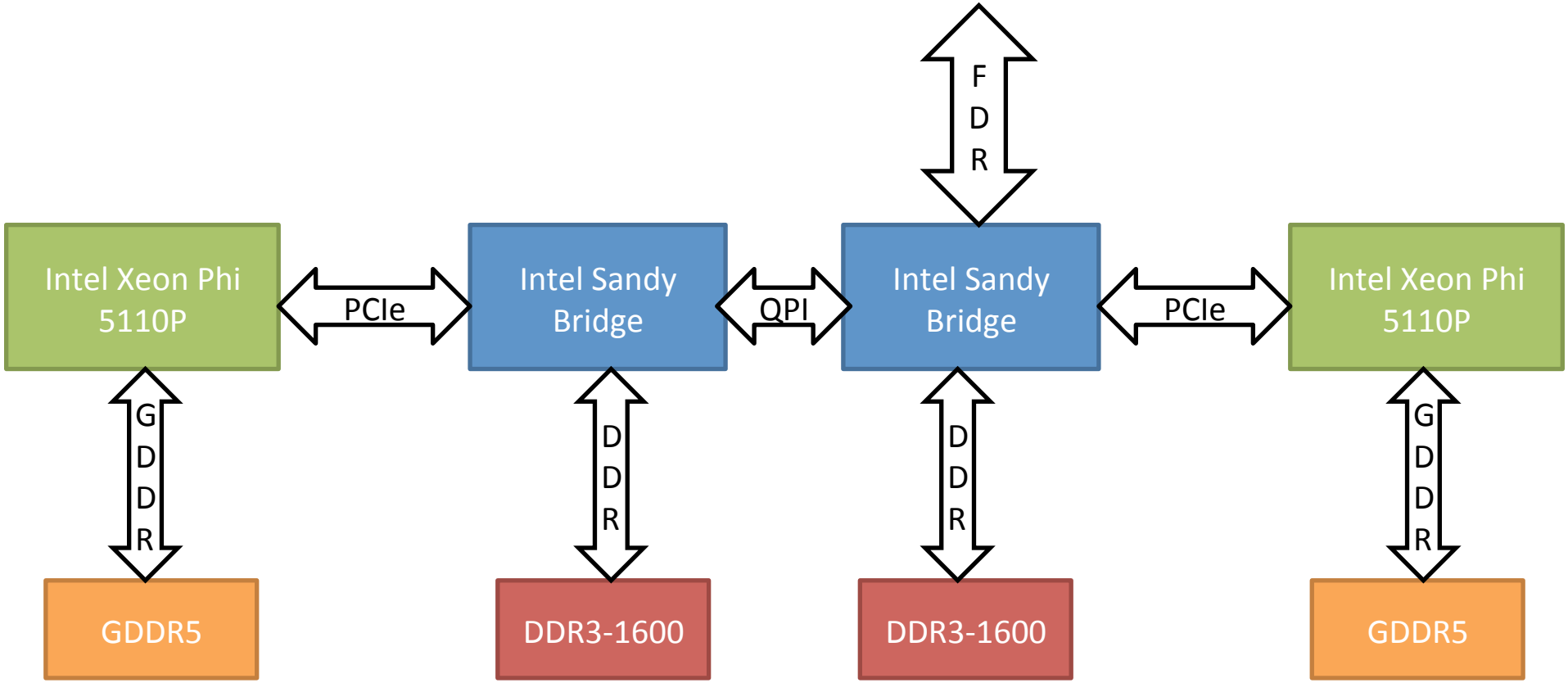
Global Bandwidth (Collective Operations)



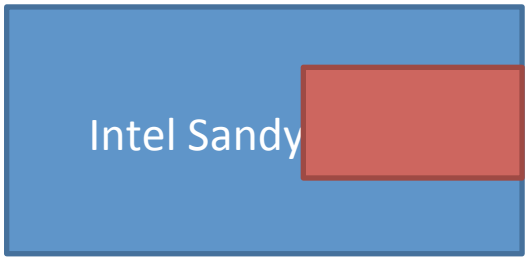
Tips and Tricks

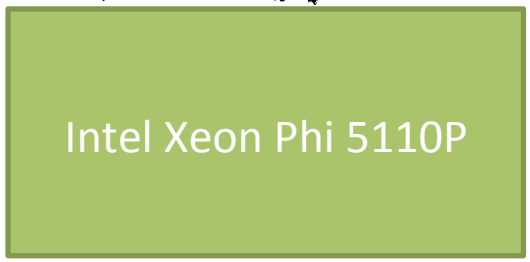
- MVAPICH2 version 1.9a2
- Evolution of GPU aware MPI
  - Early results: e.g. ptp latency = ~40 usec
  - Current results: e.g. ptp latency = ~ 20 usec
- Bandwidth improvements
  - Set environment variable
- Dependency on CUDA driver
- GPU and CPU binding on multi-GPU systems
- Misc. (MV2_USE_CUDA & mpirun_rsh)

Intel Xeon Phi Evaluation work in progress using beta software stack (IMPI)

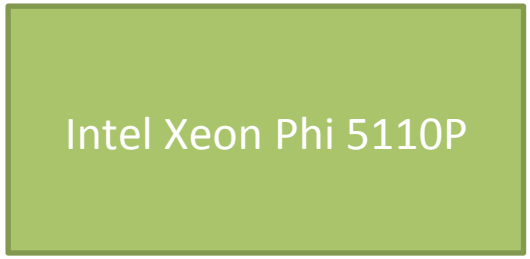


On node
Latency (8 bytes) = ~ 4 usec
Bandwidth (1 Mbytes) = ~ 5 GB/s





On MIC
Latency (8 bytes) = ~ 3 usec
Bandwidth (1 Mbytes) = ~ 2 GB/s



FDR network (mic0)
Latency (8 bytes) = ~ 6 usec
Bandwidth (1 Mbytes) = ~ 1 GB/s

FDR Network (mic1)
Latency (8 bytes) = ~ 9 usec
Bandwidth (1 Mbytes) = ~ 0.3 GB/s

Tips and Tricks

- Ensure that you have the latest software stack
  - MPSS
  - Intel MPI
  - SCIF setting
  - ...
- Find out how to specify host list incl. MICs
- Pay attention to the warning messages
  - May need to set environment variables
- Affinities on host and Xeon Phi matter

Challenges and Next Steps

- Consistent GPU-aware MPI implementation
- GPUDirect-RDMA (host memory bypass)
- Xeon Phi aware MPI in offload mode?
- Parallel file I/O ...

References

- http://mvapich.cse.ohio-state.edu/overview/mvapich2/
- http://software.intel.com/en-us/mic-developer
- https://developer.nvidia.com/category/zone/cuda-zone

Acknowledgements

- MVAPICH2 development team at the Ohio State University

- CSCS staff (benchmark development and system administration)

- Vendor support: Intel, Mellanox and NVIDIA