Argonne NATIONAL LABORATORY

System Software for Scalable Applications

Pavan Balaji (PI)

Co-PIs: William Gropp, Ewing Lusk, Rajeev Thakur

U.S. DEPARTMENT OF ENERGY

Primary Research Goals

- Understanding the performance characteristics of communication software (in particular MPI) on systems with complex hardware topologies (in particular Blue Waters)
- Specific Goals:
  - Topology-aware communication and performance characterization of MPI on Blue Waters
  - Understanding the performance of integrated MPI+GPU models, where a common runtime system can perform data transfer to/from host and GPU memories
  - Investigation of virtual accelerators where systems like Blue Waters can allocate shared XK service nodes hosting accelerators while XE nodes can transparently use them as local “virtual accelerators”

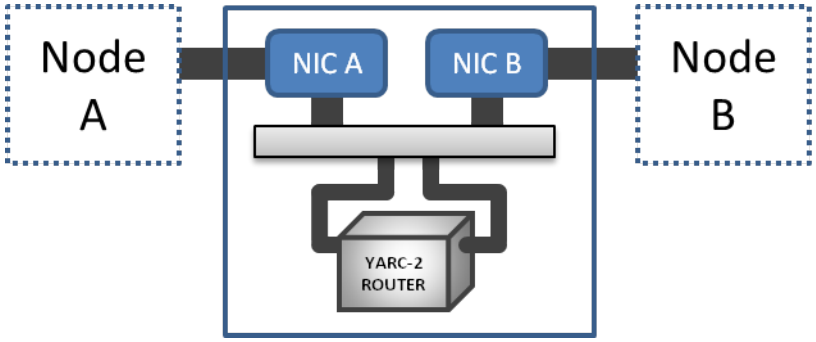
Argonne NATIONAL LABORATORY

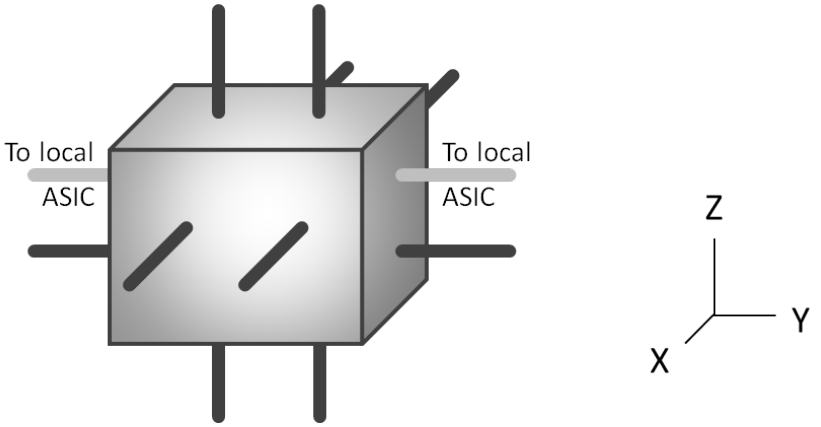
Topology-aware Communication on Blue Waters

U.S. DEPARTMENT OF ENERGY

Topology-aware Communication: Motivation

- Network properties can have a significant impact on application performance
- BW uses a 3-dimensional Cray Gemini torus featuring anisotropic properties
  - Twice the Y-dimension bandwidth in the X and Z dimensions
  - A Gemini ASIC is shared by two nodes
- **Task placement considering these properties is highly beneficial**

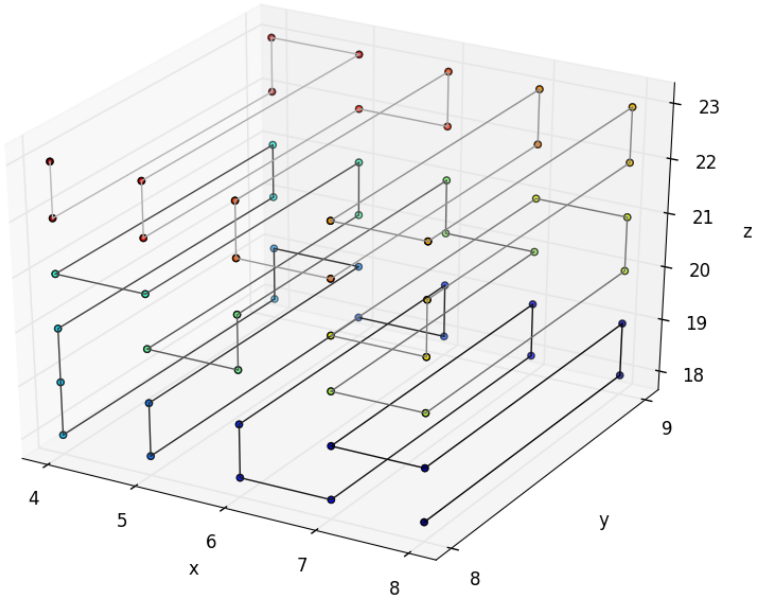




Antonio Pena, Ralf Gunter, James Dinan, Pavan Balaji, Rajeev Thakur, and William Gropp. *"Analysis of Topology-dependent MPI Performance on Gemini Networks."* The Euro MPI Users' Group Conference (Euro MPI), Sep. 15-18, 2013, Madrid, Spain

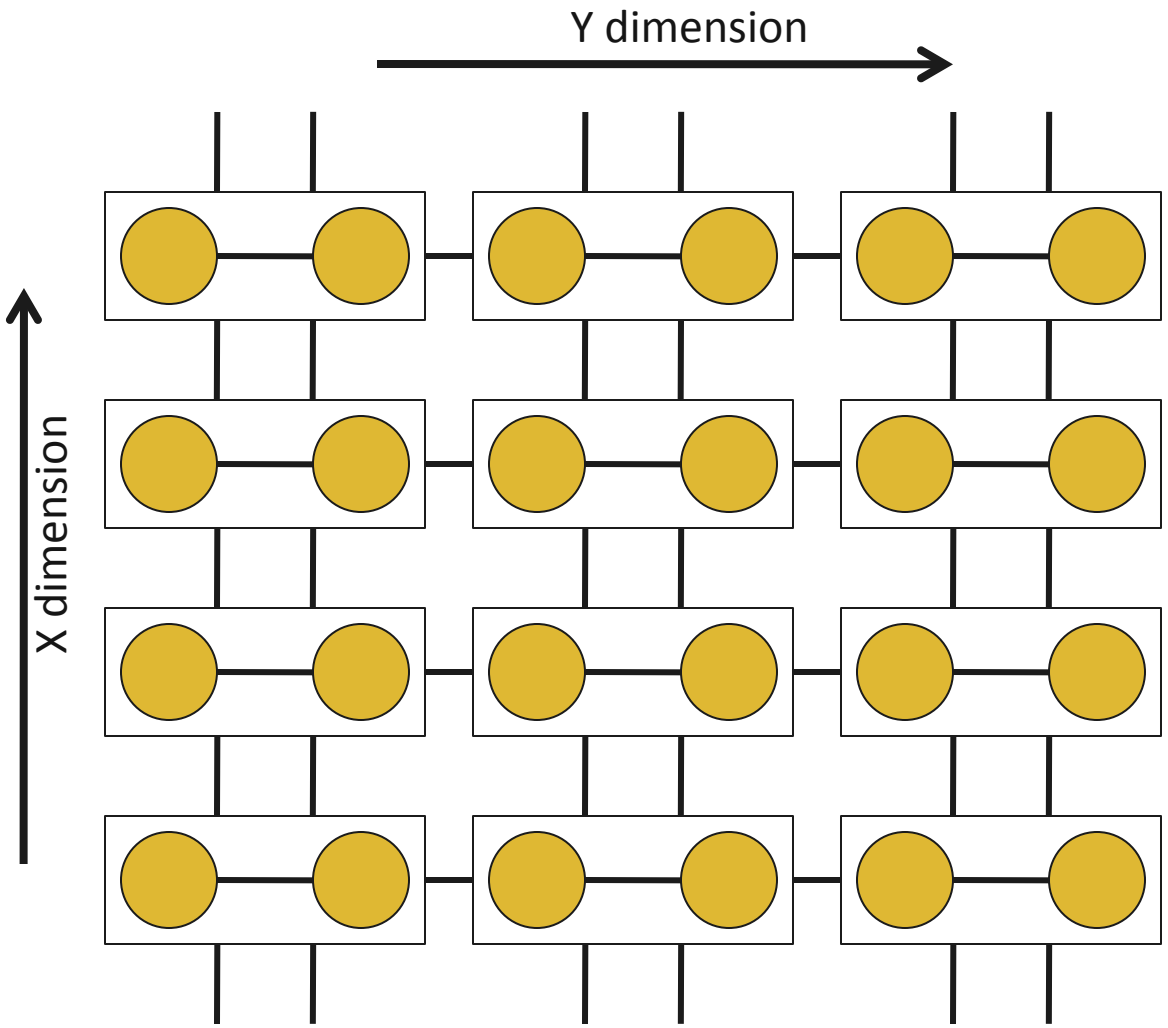
Job Placement and Rank Ordering in BW

- Cray MPICH follows the node ordering assigned by the job scheduler
- Ranks are ordered in a zigzag fashion
  - First and last ranks are adjacent
  - Decrease hop count
  - Increase bisection bandwidth
- Given that:
  - XE6 routers contain two nodes
  - Z links are faster than X links
  - Every 5th link is crossing a cabinet (slower)
  - 4 x 2 x 8 building blocks



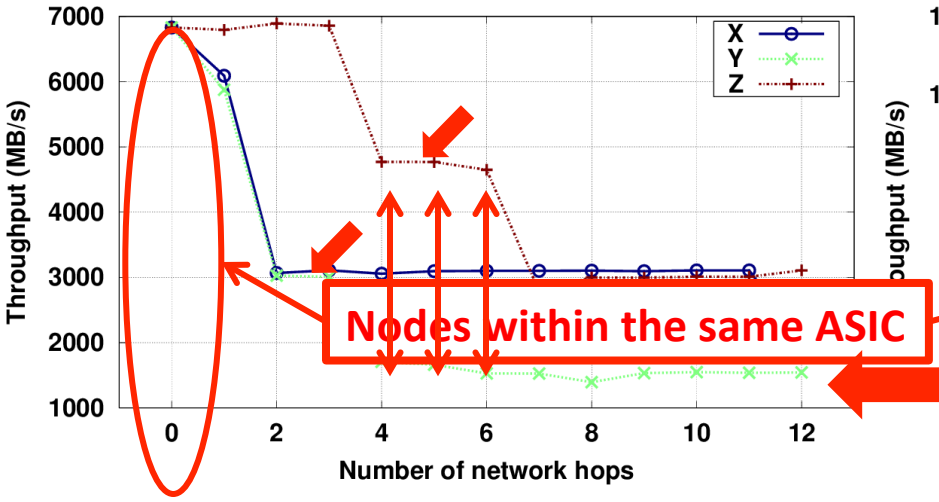
Carl Albing, Norm Troullier, Stephen Whalen, Ryan Olson, Joe Glenski, Howard Pritchard, and Hugo Mills. Scalable node allocation for improved performance in regular and anisotropic 3D torus supercomputers. In *Recent Advances in the Message Passing Interface*, volume 6960 of *LNCS*, 2011.

Blue Waters Network Layout

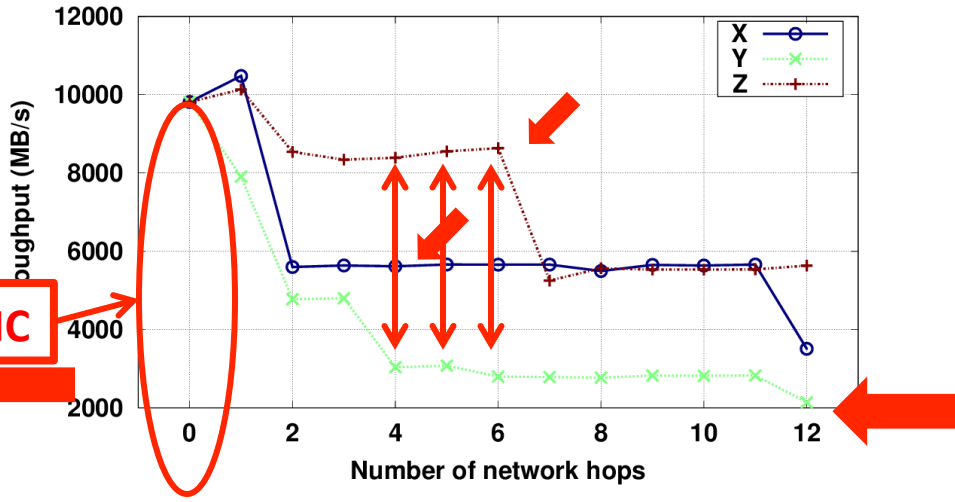


Point-to-Point Communication (single process)

- Point-to-point benchmarking
- Anisotropic behavior illustrated
- Communications in the Y direction perform significantly lower: ½ links
- Z links offer over 3x TR than Y
- X and Z: largely different behaviors
- Increase in Latency per hop: ~0.1µs



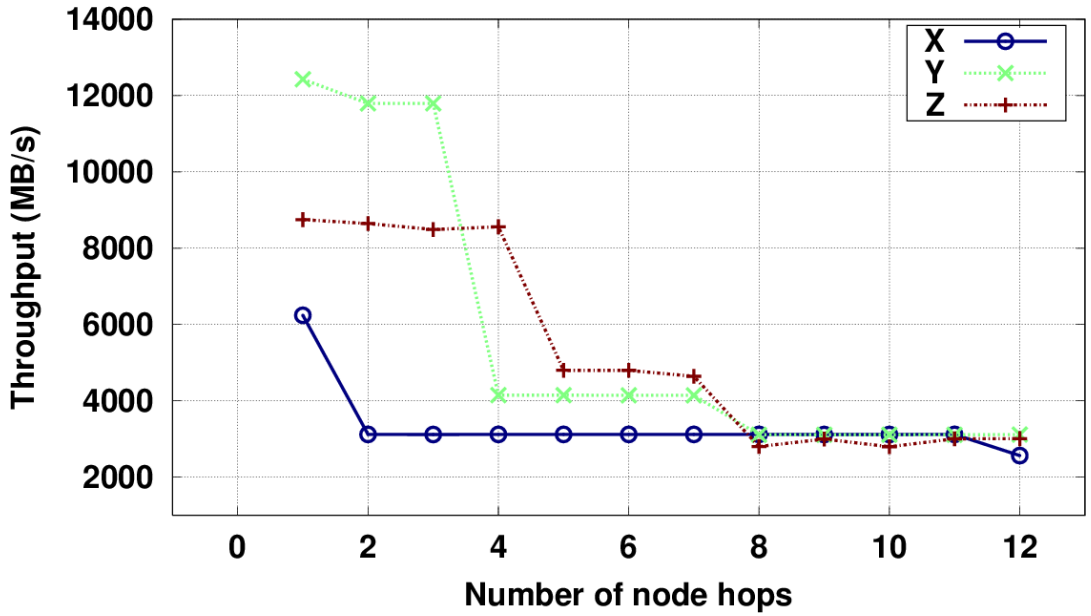
Unidirectional



Bidirectional

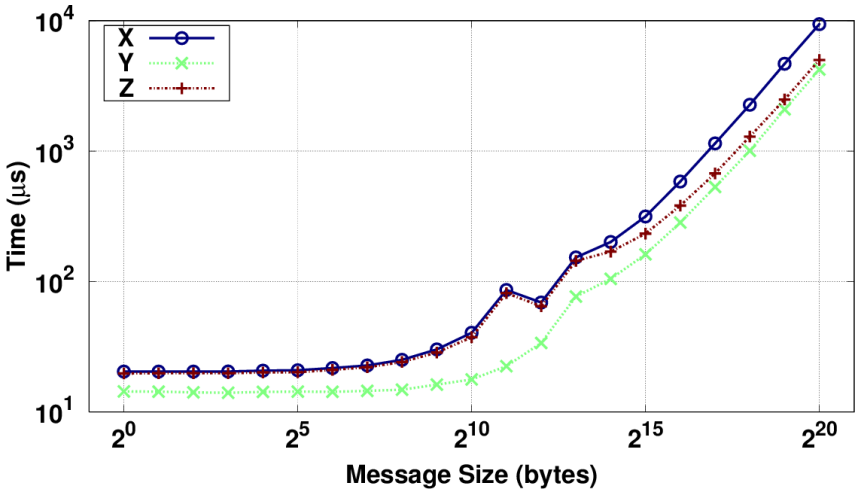
Point-to-Point Communication (multiple processes)

- Internode aggregate transfer rate
- 2 parallel paths transfer concurrently
- Optimal node ordering and matching between MPI ranks and network topo.
- Communication saturating links greatly improve performance on Y direction
- Contiguous nodes in these experiments
- Double X and Z links become shared
  - § Aggregate TR increases for Y
- Placement of dual nodes/ASIC along Y
  - § Extra performance improvement

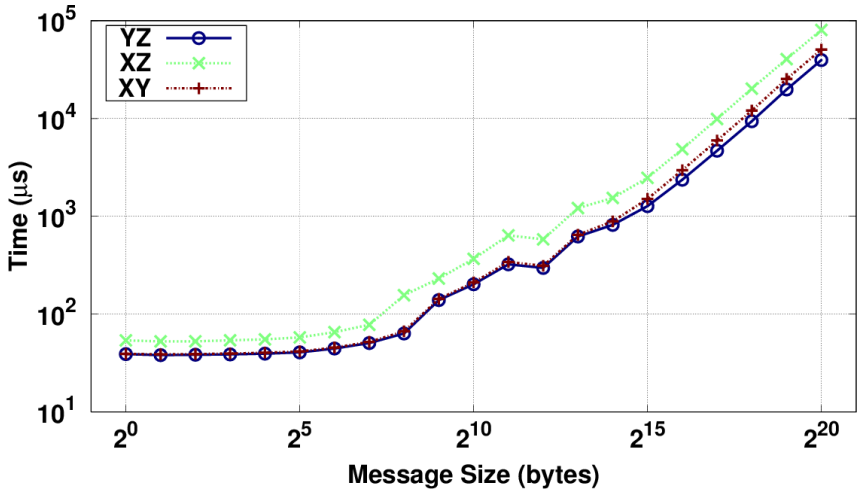


Collective Communications

- Topology matching exploited by row-wise and plane-wise collectives
- **Y direction faster!**
  - § Row-wise: up to 74% (alltoall) and 54% (allgather)
  - § Plane-wise: up to 59% (alltoall) and 53% (allgather)



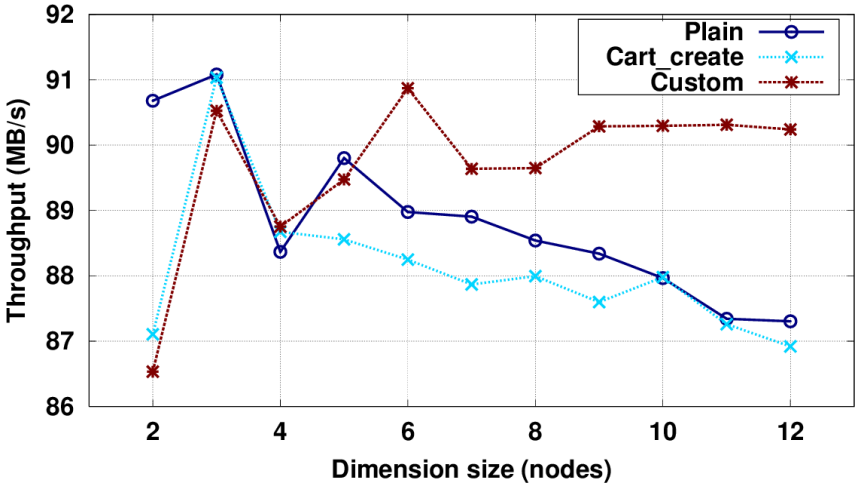
Row-wise MPI_Alltoall



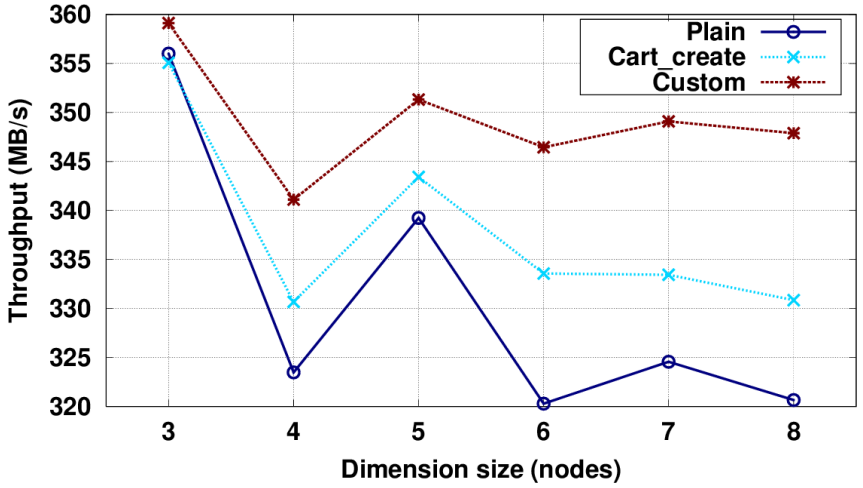
Plane-wise MPI_Alltoall

Stencil Communications

- Cray MPICH ignores the *reorder* parameter in MPI_Cart_create
  - § MPI topo. not matching network
- 2D & 3D halo exchange (contig. nodes):
  - § **Plain:** Manual ordering X-Y-Z
  - § **Cart_create:** Y-major / Z-Y-X
  - § **Custom:** MPI-network matching
- 2D:
  - § *Cart_create* worst performance
  - § *Plain* up to 1.4%; *Cart_create* 4%
- 3D:
  - § Topology matching outperforms MPI-assisted sorting up to 5%
- **Topology matching favors scalability**



2D



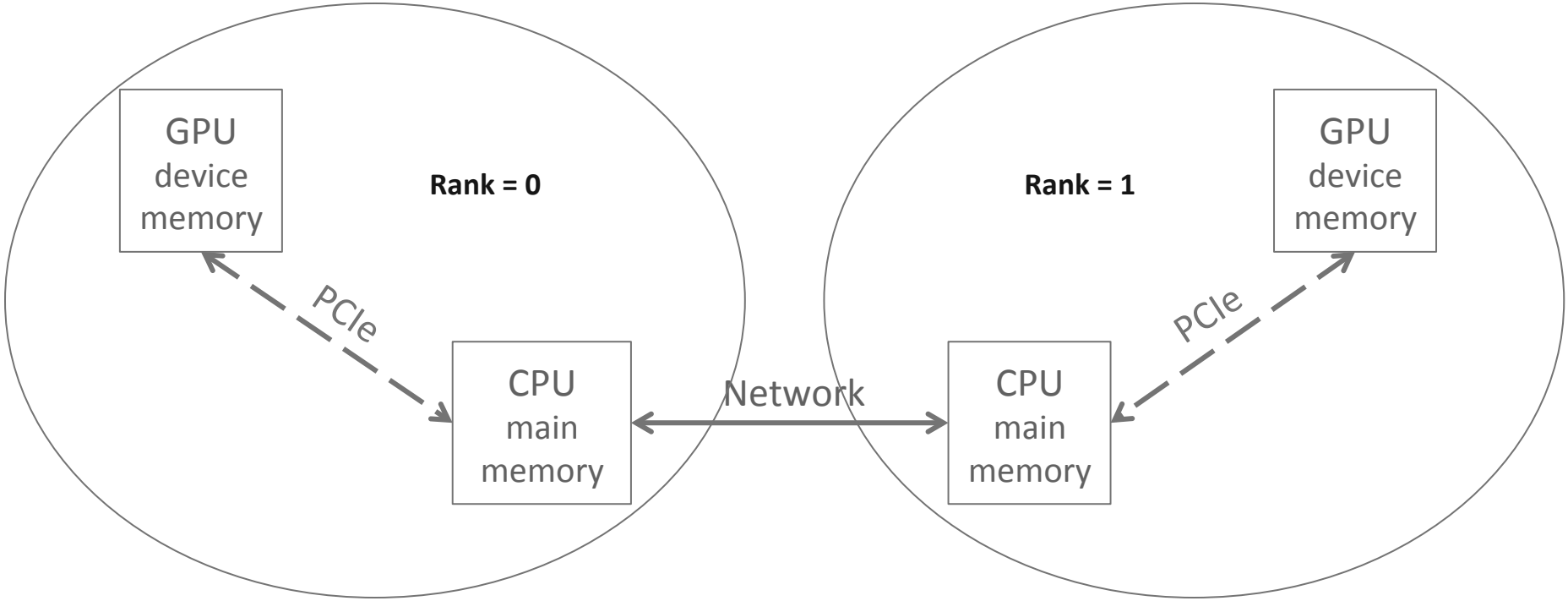
3D

Argonne NATIONAL LABORATORY

Integrated MPI-Accelerator Data Movement

U.S. DEPARTMENT OF ENERGY

Programming Heterogeneous Memory Systems (e.g: MPI+CUDA)




```
if(rank == 0)
{
  cudaMemcpy(host_buf, dev_buf, D2H)
  MPI_Send(host_buf, .. ..)
}
```

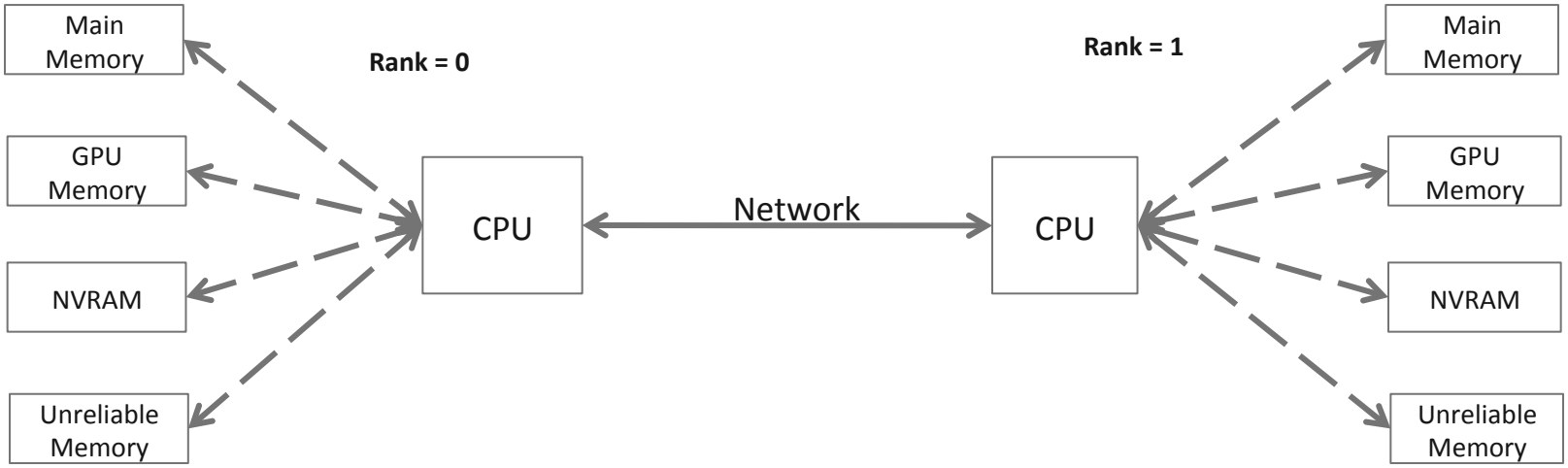
```
if(rank == 1)
{
  MPI_Recv(host_buf, .. ..)
  cudaMemcpy(dev_buf, host_buf, H2D)
}
```

Current Limitations of Programming Heterogeneous Memory Systems (e.g: MPI+CUDA)

```
if (my_rank == sender) { /* sender */
   computation_on_GPU(dev_buf);
   cudaMemcpy(host_buf, dev_buf, size, ...);
   MPI_Send(host_buf, size, ...);
} else {                    /* receiver */
   MPI_Recv(host_buf, size, ...);
   cudaMemcpy(dev_buf, host_buf, size, ...);
   computation_on_GPU(dev_buf);
}
```

- **Programmability/Productivity**: Manual data movement leading to complex code; Non-portable codes
- **Performance**: Inefficient and non-portable performance optimizations
  - Manual copy between host and GPU memory serializes PCIe, Interconnect
    - Difficult for user to do optimal pipelining or utilize DMA engine efficiently
  - Incur protocol overheads multiple times
  - Architecture-specific optimizations

DMEM: A Model for Unified Data Movement



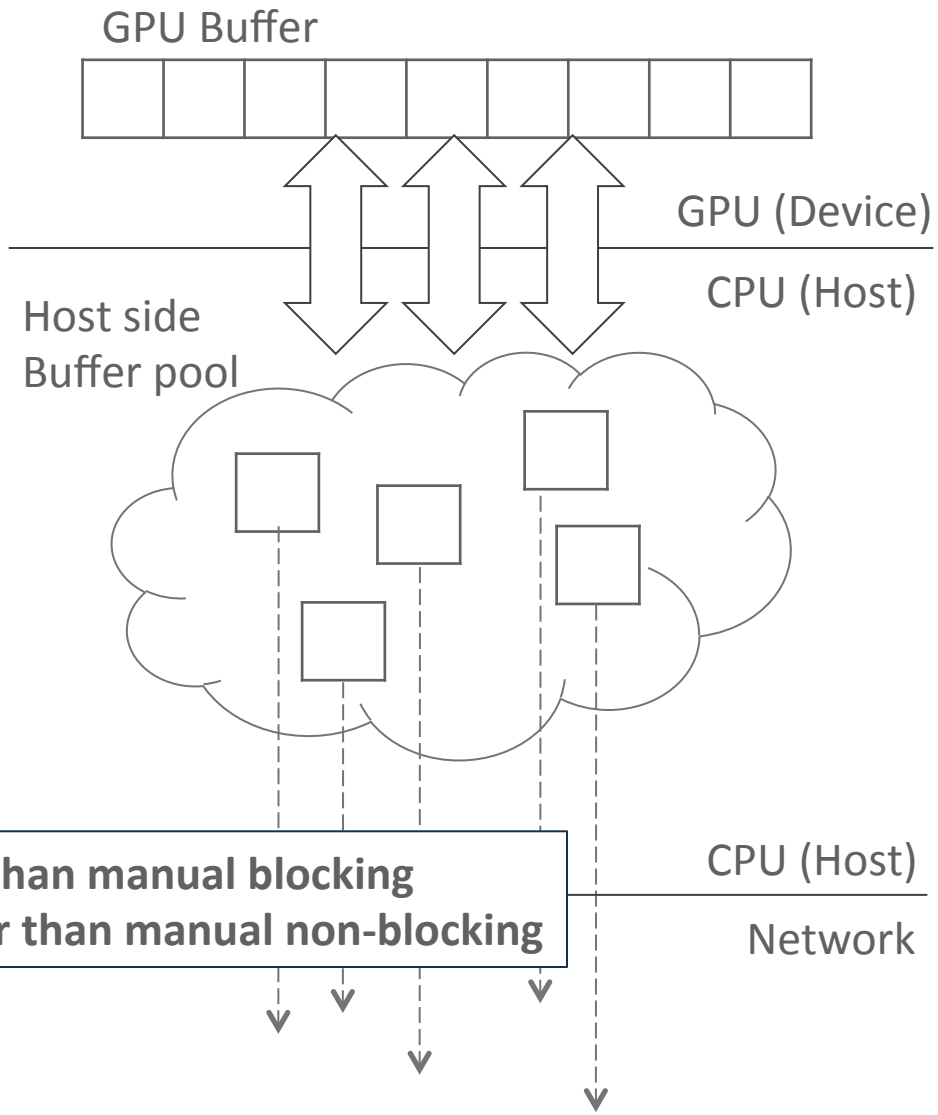

```
if(rank == 0)
{
  MPI_Send(any_buf, .. ..);
}
```

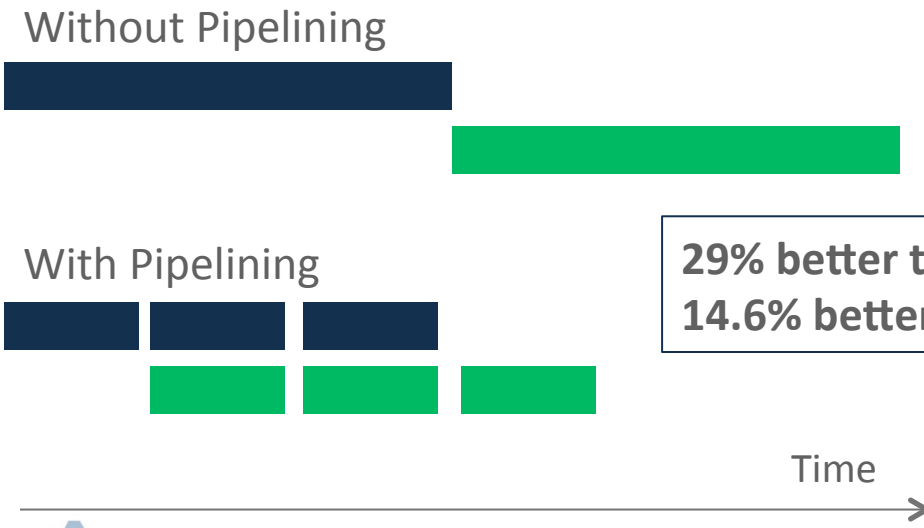
```
if(rank == 1)
{
  MPI_Recv(any_buf, .. ..);
}
```

Ashwin M. Aji, Pavan Balaji, James S. Dinan, Wu-chun Feng and Rajeev S. Thakur. Synchronization and Ordering Semantics in Hybrid MPI+GPU Programming. Workshop on Accelerators and Hybrid Exascale Systems (AsHES); held in conjunction with the IEEE International Parallel and Distributed Processing Symposium (IPDPS). May 20th, 2013, Boston, Massachusetts

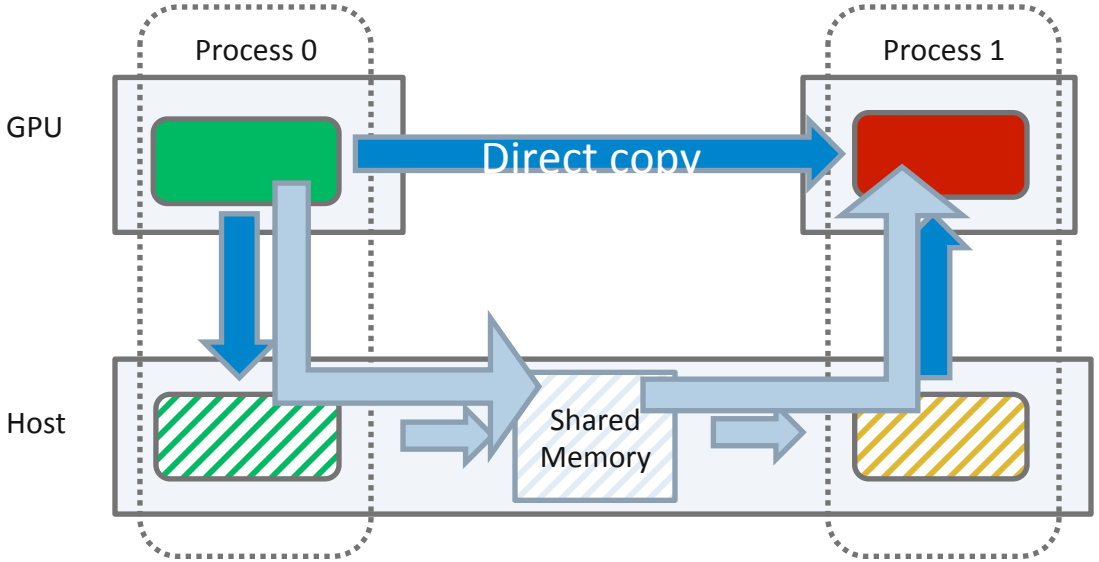
DMEM Runtime Optimizations

- Topology-aware pipelining of data
- Caching of meta-data (e.g., handles)
- Multi-stream data transfer when possible (e.g., newer accelerators)
- Architecture-specific optimizations: GPU Direct





Traditional Intranode Communication

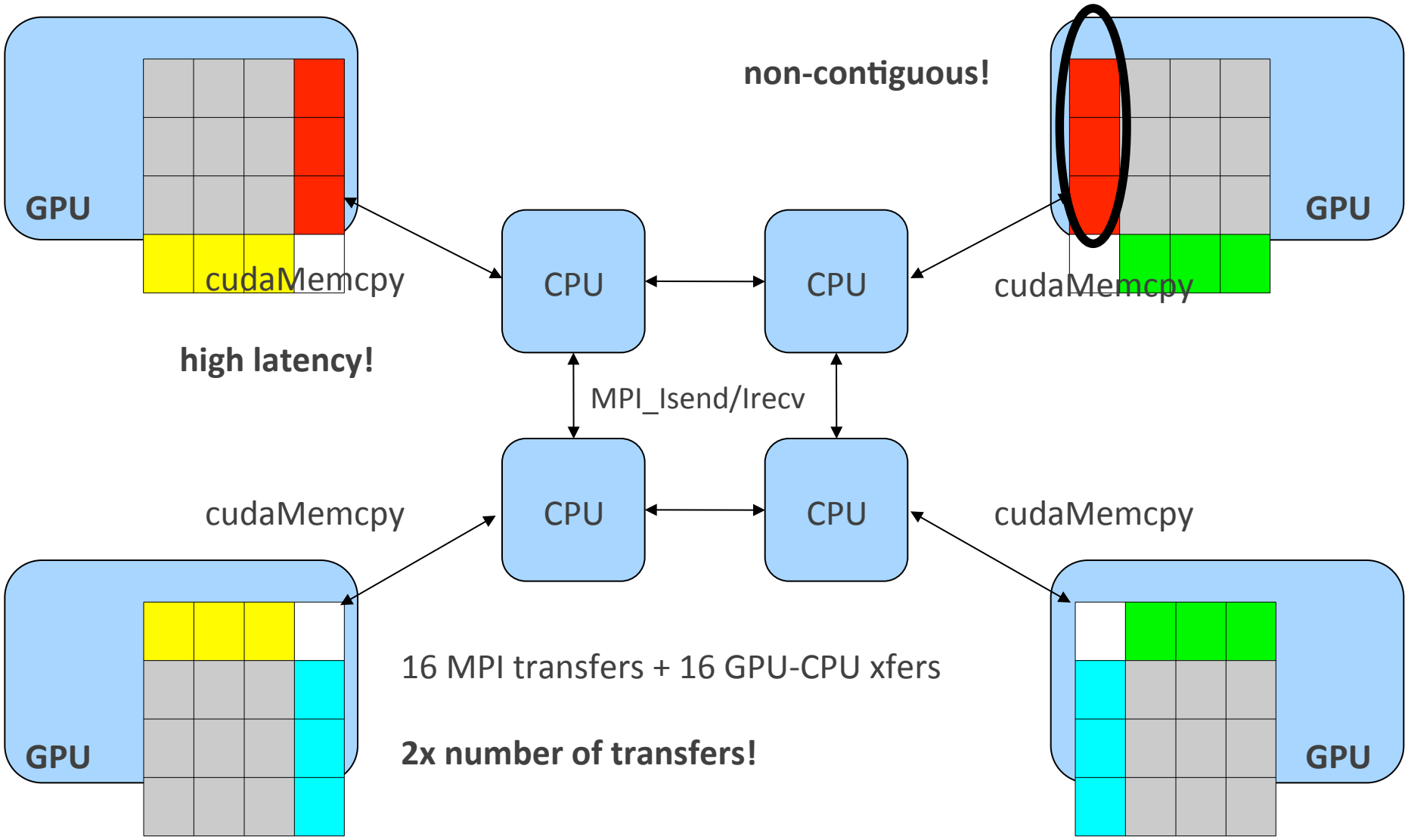


- Communication without heterogeneous memory support
  - 2 PCIe data copies + 2 main memory copies
  - Transfers are serialized

- Integration allows direct transfer into shared memory buffer
  - Sender and receiver drive transfer concurrently
    - Pipeline data transfer
    - Full utilization of PCIe links

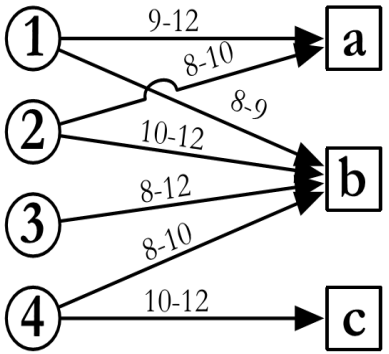
- Direct Copy: DMA-driven peer GPU copy
- Peer-to-peer data transfer between heterogeneous memory regions

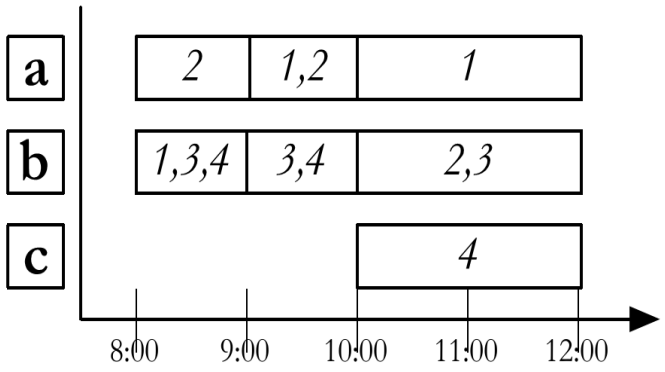
Example – 2D Stencil Computation

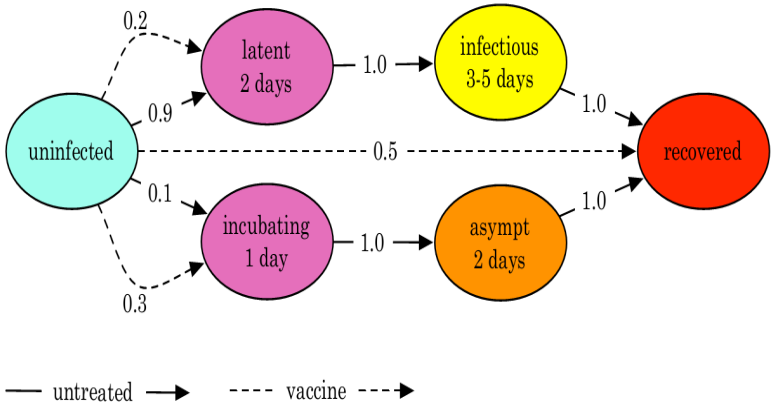


Epidemiology Simulation (EpiSimdemics)

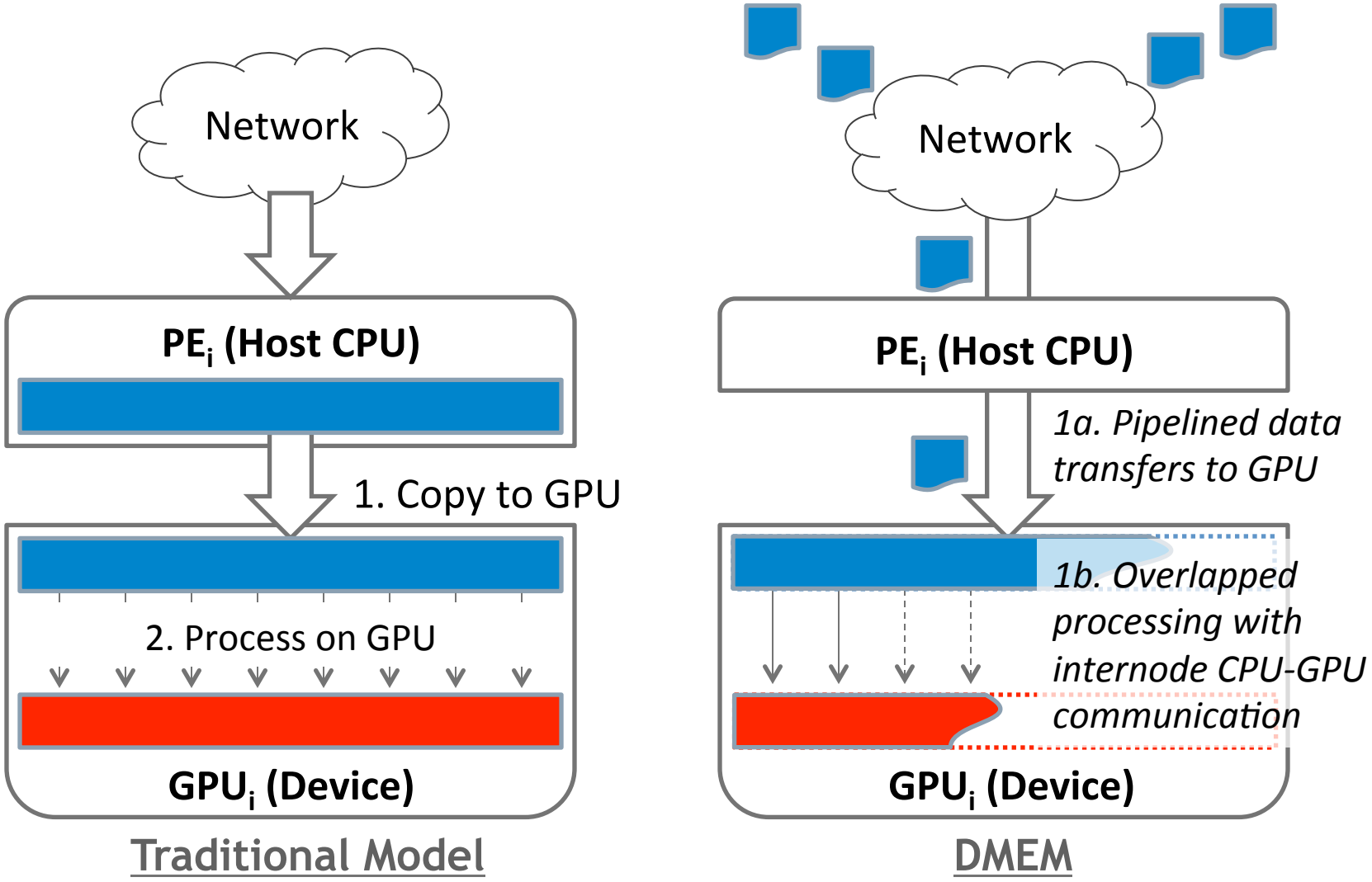
- Episimdemics models try to understand the spatio-temporal diffusion/ spread of a contagious disease through social contact networks of populations
  - Represents social networks by labeled bipartite graphs with two disjoint sets as People and Locations.
  - Duration of interaction between people is modeled using the activities and overlap of stay of different people at different locations.
- A variant of finite state machines, called probabilistic timed transition systems (PTTSs) is used to represent the within host disease propagation.



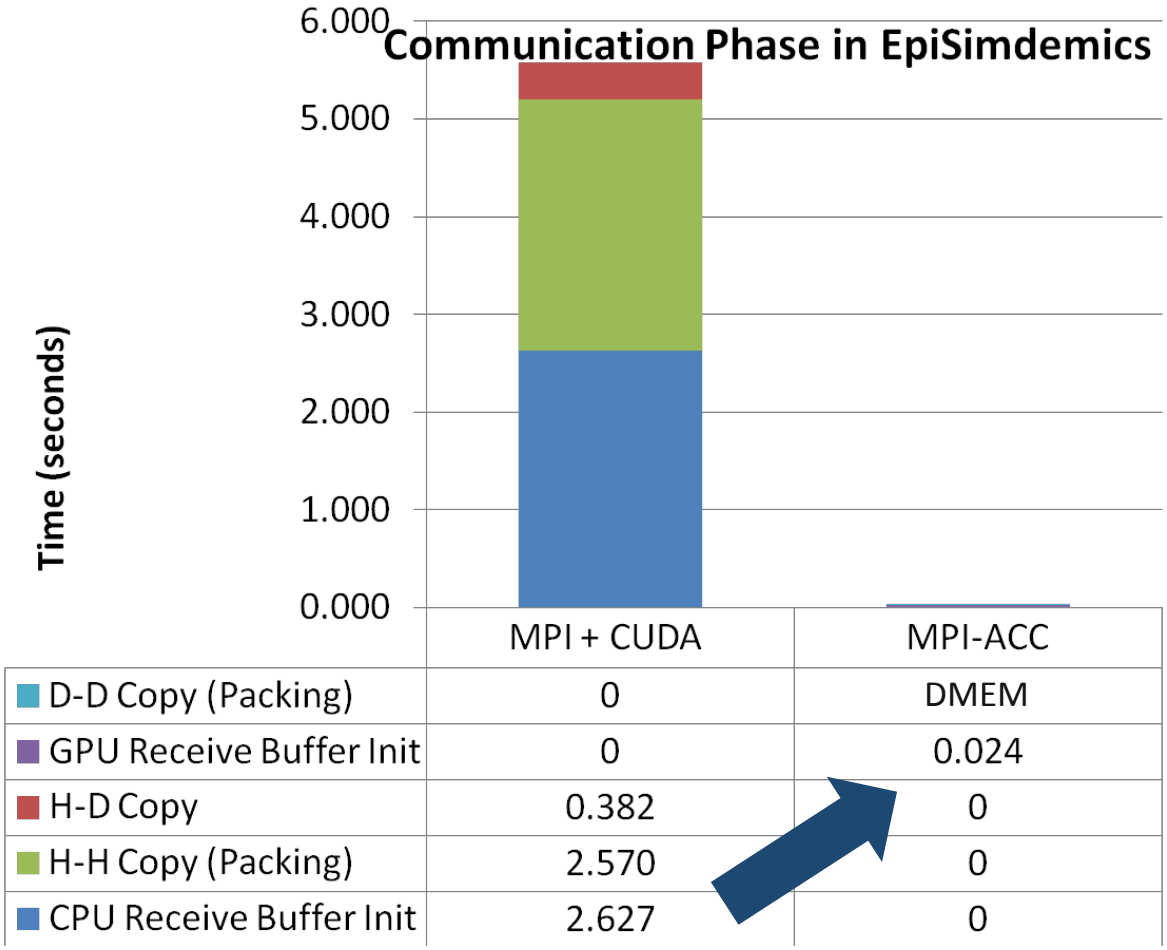




Case Study: Epidemiology



Evaluating the Epidemiology Simulation



	MPI + CUDA	MPI-ACC
D-D Copy (Packing)	0	DMEM
GPU Receive Buffer Init	0	0.024
H-D Copy	0.382	0
H-H Copy (Packing)	2.570	0
CPU Receive Buffer Init	2.627	0

- GPU has *two* orders of magnitude faster memory
- DMEM *enables* new application-level optimizations

While the work was done in the context of the Blue Waters machine, the above performance evaluation was not done on Blue Waters, but we plan to evaluate this approach on Blue Waters in the future

Argonne NATIONAL LABORATORY

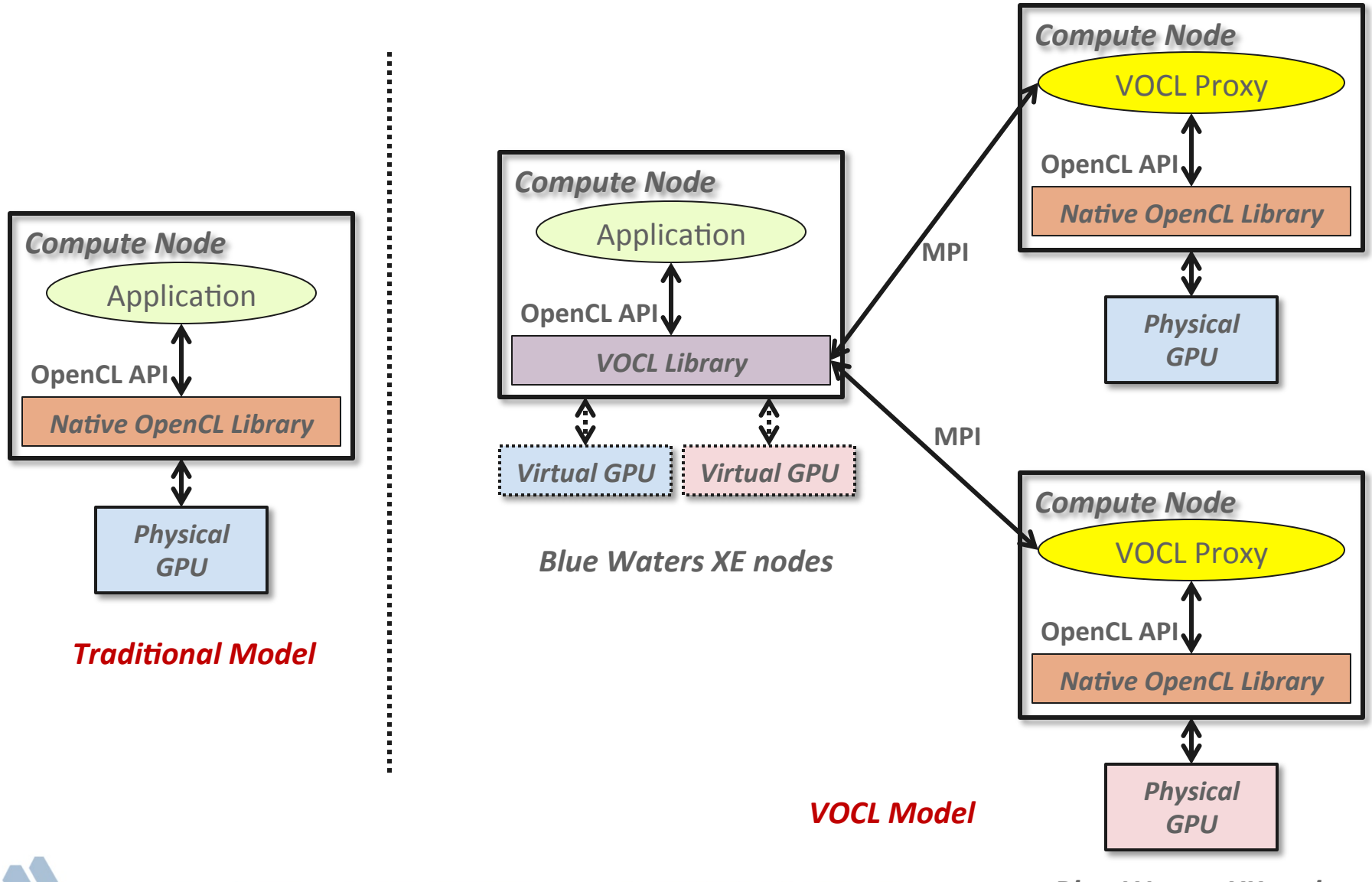
Virtual Accelerators for a Unified Accelerator View

U.S. DEPARTMENT OF ENERGY

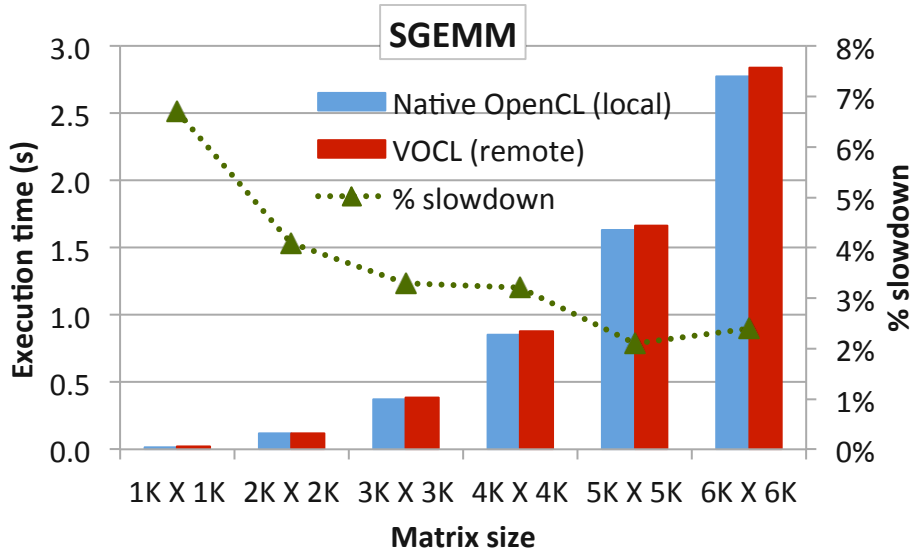
Virtualized Accelerator Units

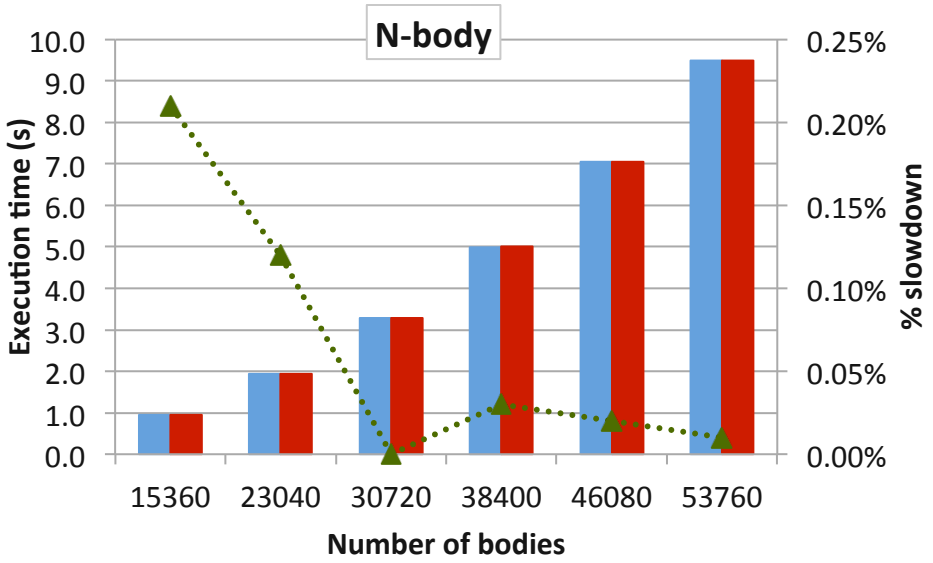
- Explicit movement of data or computation to ensure locality
  - Already a requirement for accelerators
  - Is increasingly becoming a requirement for NUMA architectures (e.g., problems with OpenMP)
- Currently, this is a “performance requirement” for CPU architectures, but a “correctness requirement” for accelerators
- Idea: If data or computation has to be moved explicitly anyway, why does the data+computation need to move to a local NUMA domain or accelerator? Why can’t the runtime system move it somewhere else if more appropriate?
  - Data/computation staging to use remote accelerators
  - Migration if needed (though this can be expensive)

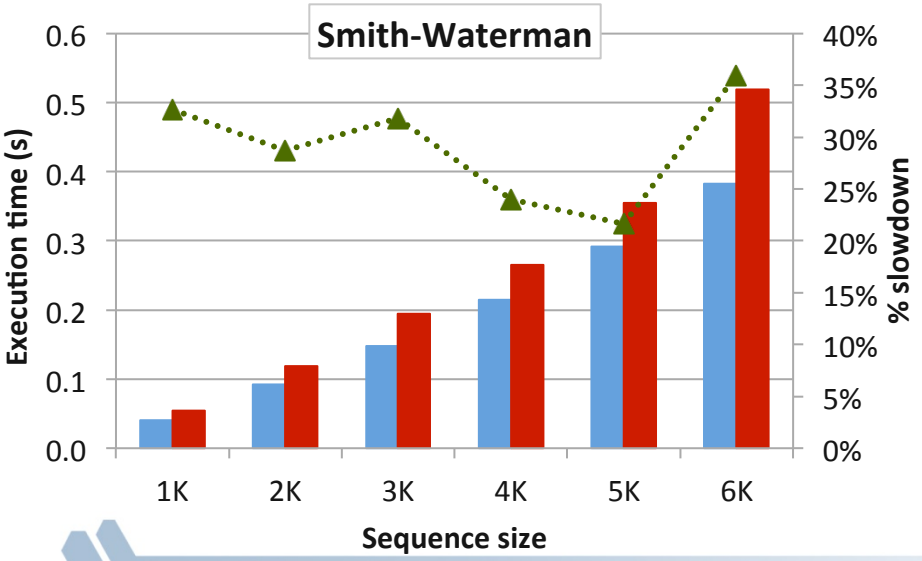
Virtual OpenCL (VOCL) Framework

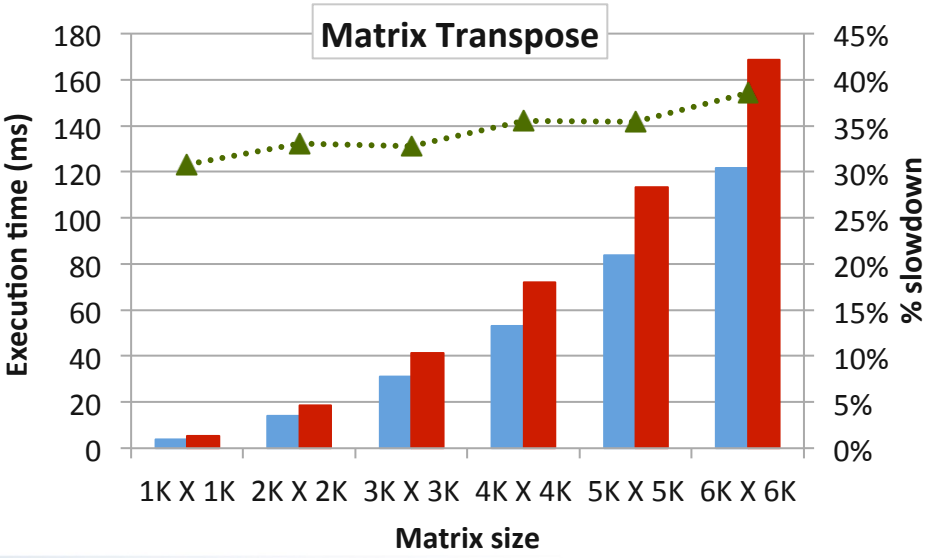


Performance Overhead of Real Application Kernels (single precision)









Summary

- Blue Waters provide an interesting, but a complex hardware architecture
  - Complex and heterogeneous communication performance trends along different dimensions
  - Heterogeneity within the node (CPU and GPU) as well as across nodes (XE and XK)
  - Understanding data and computation movement trends and requirements is critical
- We investigated a small subset of performance characteristics, with focus on both CPUs and GPUs, in the context of MPI

Argonne NATIONAL LABORATORY

Thank You!

Joint effort between:

Programming Models and Runtime Systems Group (Argonne)

Contact: balaji@mcs.anl.gov

William Gropp's group (UIUC)

Contact: wgropp@illinois.edu