Evaluation of DataSpaces in Heterogeneous In-situ workflow for GPU-MURaM at Exascale

Bo Zhang

SIParCS Intern & Rutgers University & PhD Student

Partner: Damir Pulatov

Mentor: Supreeth Suresh, Cena Miller

July 12, 2021

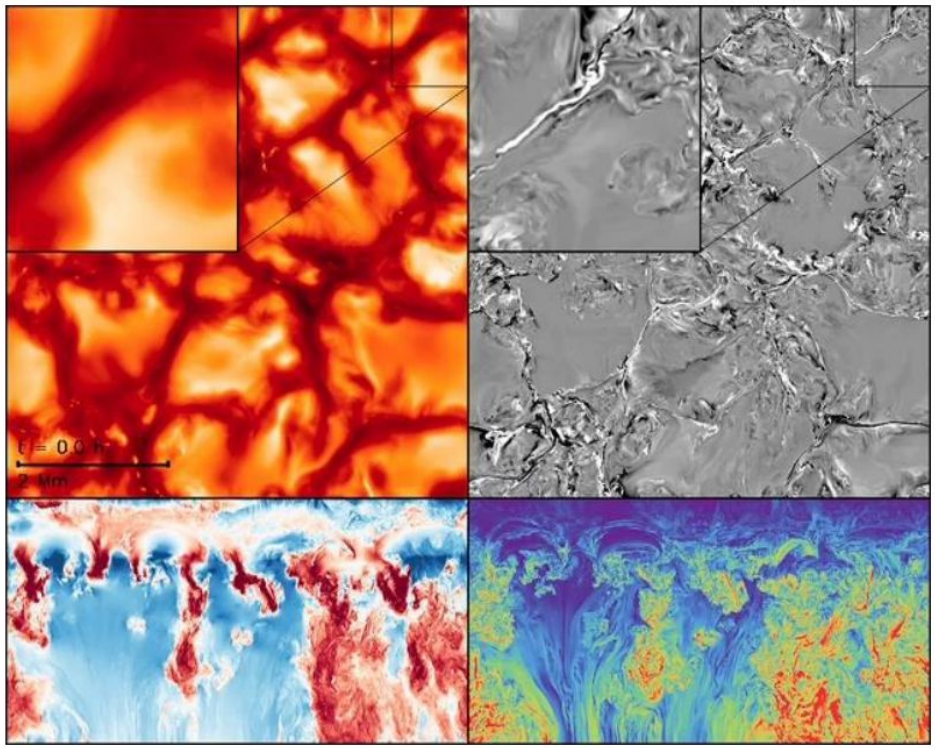
This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Outline

- Background & Motivation
- Design
- Evaluation
- Finished Work
- Lessons Learned
- Next Steps
- Q & A

Background

- MURaM is the primary solar model used for simulations of the upper convection zone, photosphere and corona.
- 100x acceleration is needed to keep up the simulation with the real time data from telescope.
- MURaM have been ported to use scalable GPUs to achieve this!
- As computation is optimized, I/O and post processing becomes the next major bottleneck.

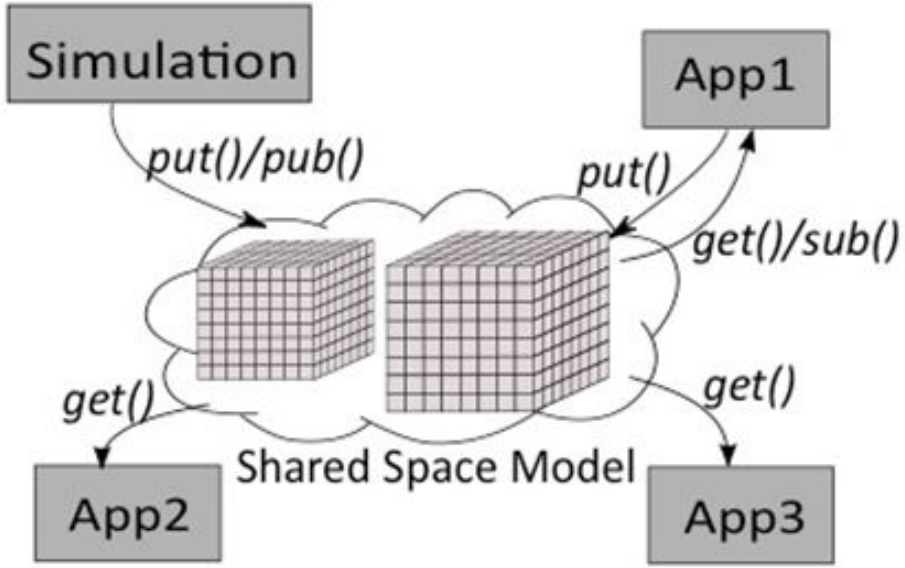


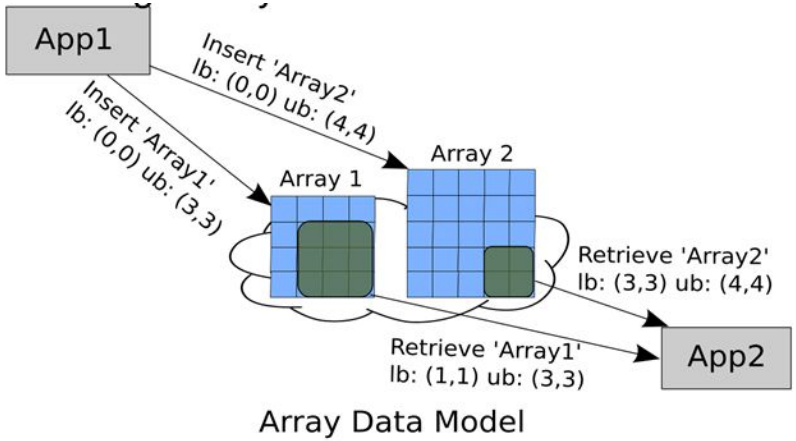
MURaM simulation of solar granulation

- Thus, both converting this workflow to an in-situ approach and a staging-based IO subsystem for this in-situ workflow are critical problems need to be addressed.

Design (Architecture)

- DataSpaces is a distributed framework provides data staging service for a coupled in-situ workflow.
- Supports tcp, verbs, gni, etc.. via OFI, UCX ...
- Use N-dimensional bounding box index, allowing flexible partial data access pattern.
- We plan to use DataSpaces as the data bridge between components in the MURaM workflow.

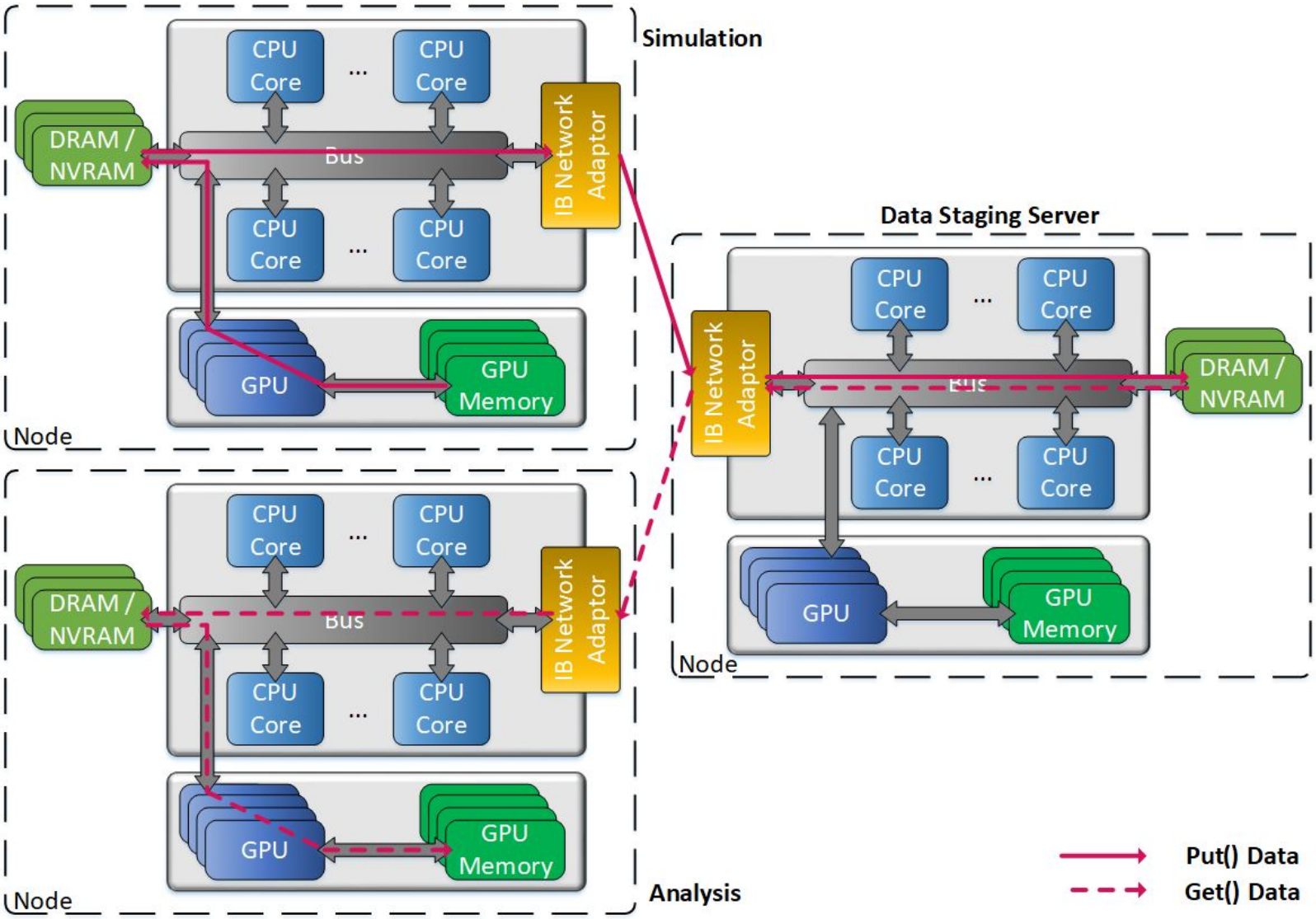




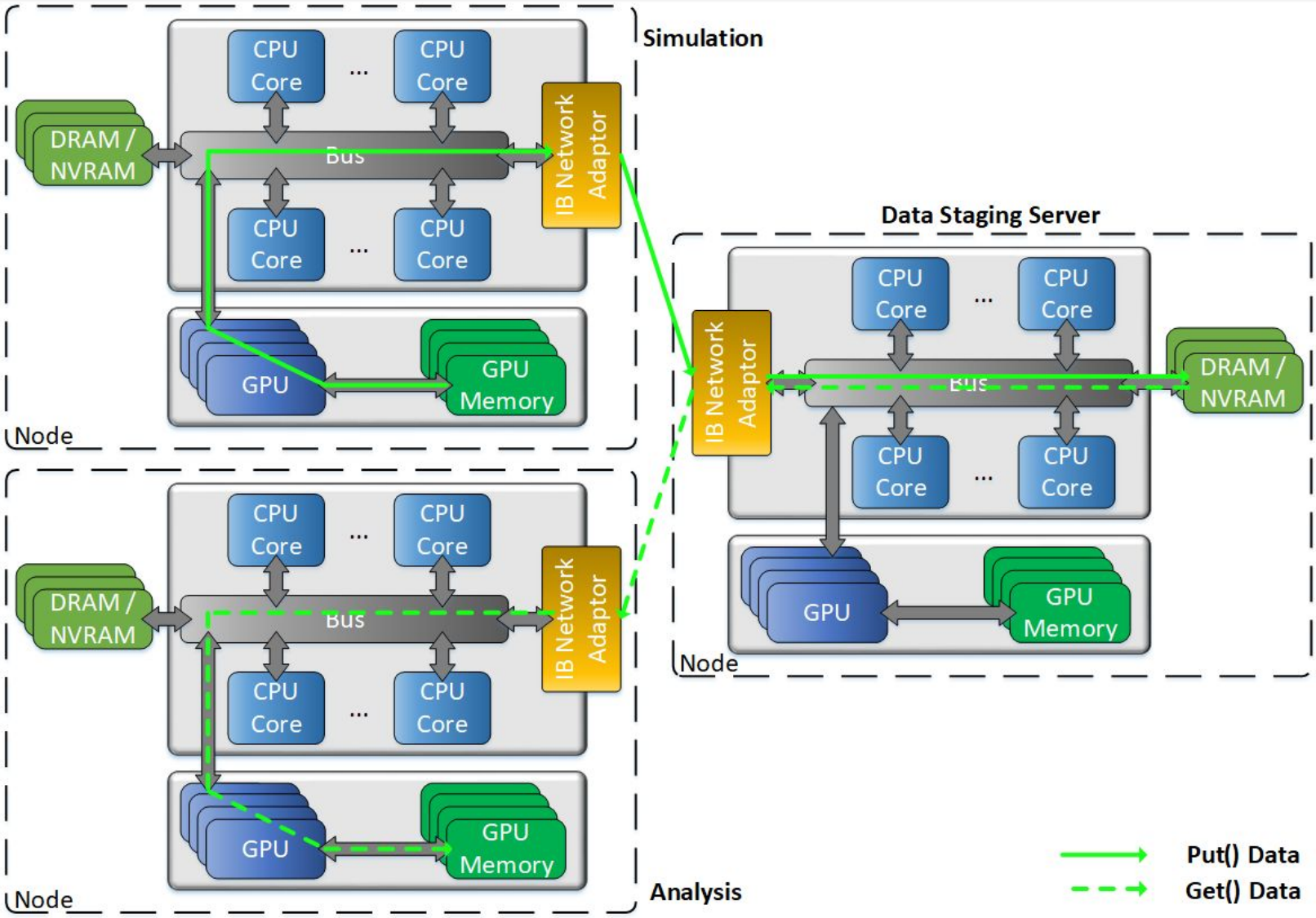
Array Data Model

lb: lower bound, ub: upper bound

Method 1: Data Flow Details (Device-Host + CPU-CPU)



Method 1: Data Flow Details (GPU Direct)



Evaluation

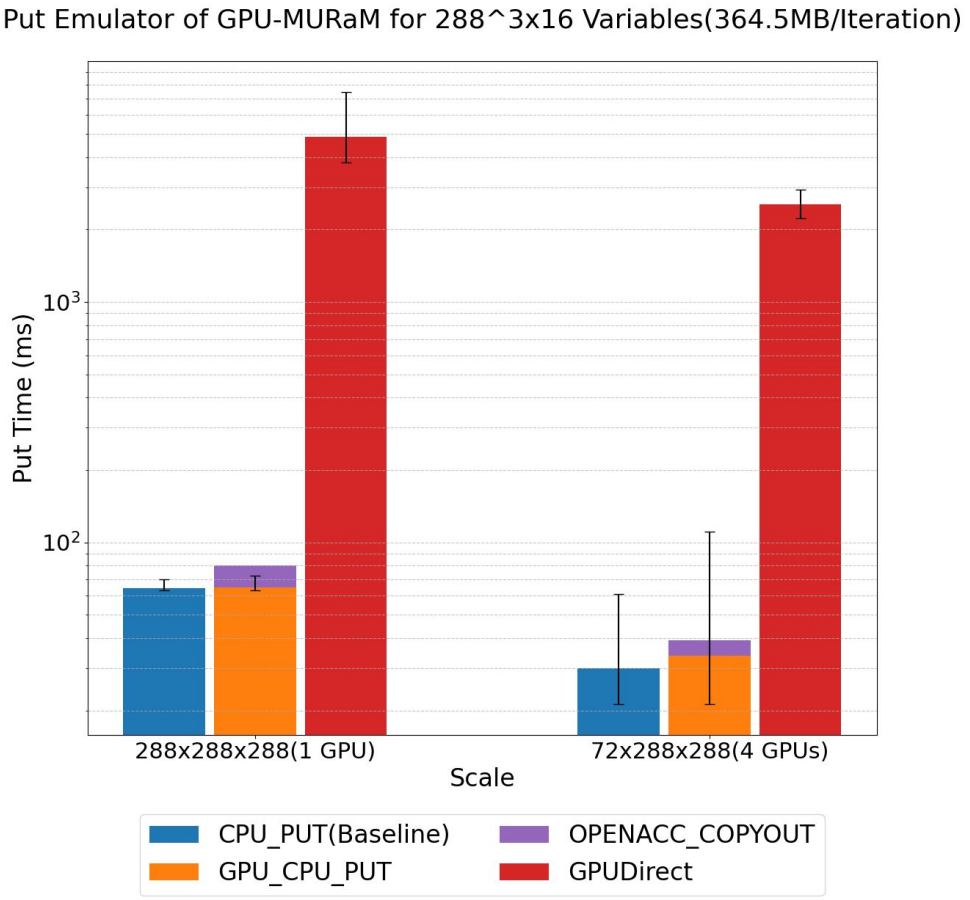
- Our experiments were performed on CASPER system.

- Simulations nodes:
  - 768 GB DDR4-2666 memory per node
  - 2 18-core 2.6-GHz Intel Xeon Gold 6240 (Cascade Lake) processors per node
  - 2 Mellanox ConnectX-6 HDR200 InfiniBand adapters. HDR100 link on each CPU socket
  - 4 NVIDIA Tesla V100 32GB SXM2 GPUs with NVLink

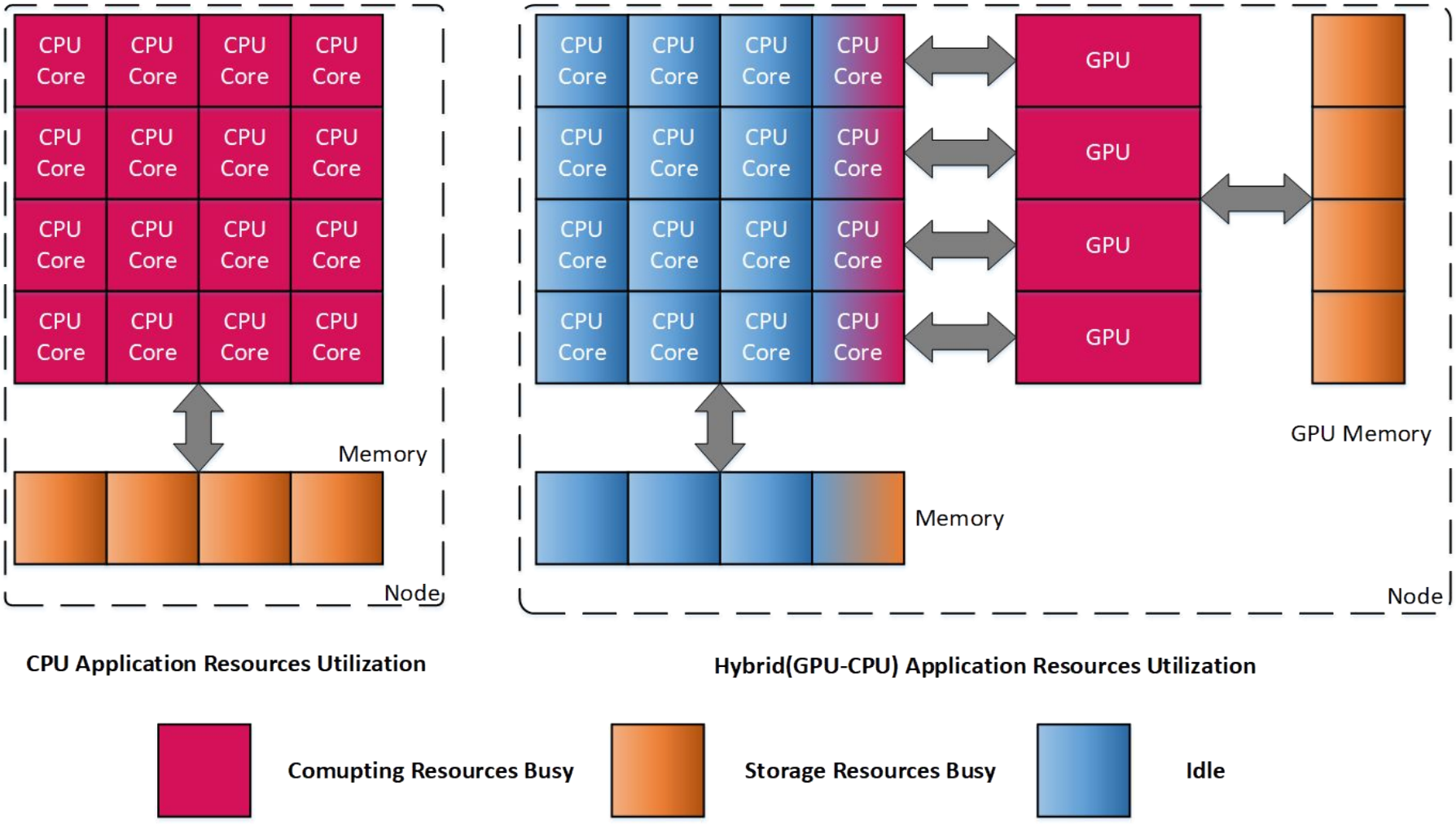
- Staging servers nodes:
  - 394 GB DDR4-2666 memory per node
  - 2 18-core 2.6-GHz Intel Xeon Gold 6240 (Cascade Lake) processors per node
  - 1 Mellanox ConnectX-6 HDR200 InfiniBand VPI adapter. HDR100 link on each CPU socket

- This is only the test code, not MuRAM!

- MuRAM code needs huge modifications to adapt to GPU Direct. (In Progress)

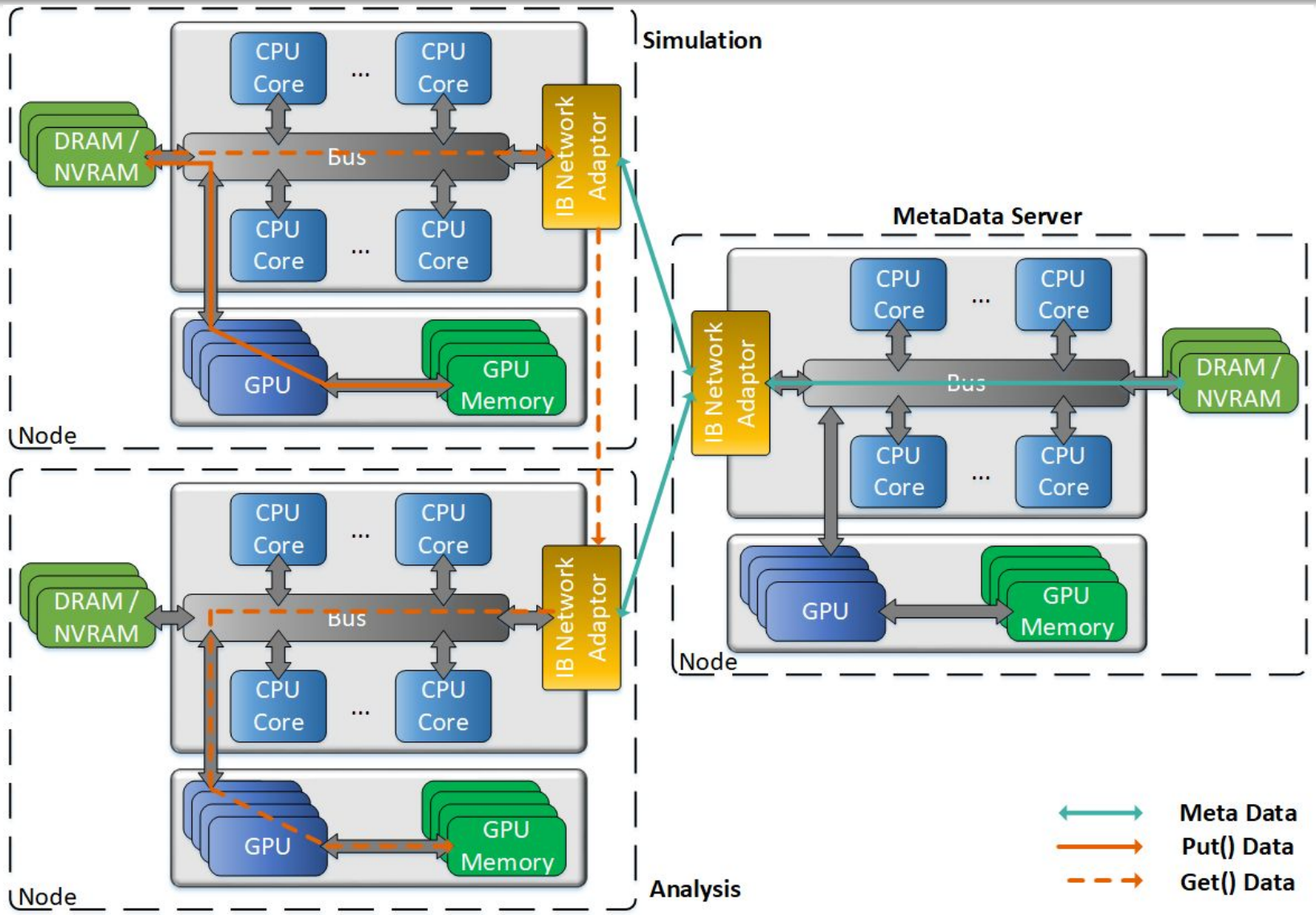


Resource Utilization Changes

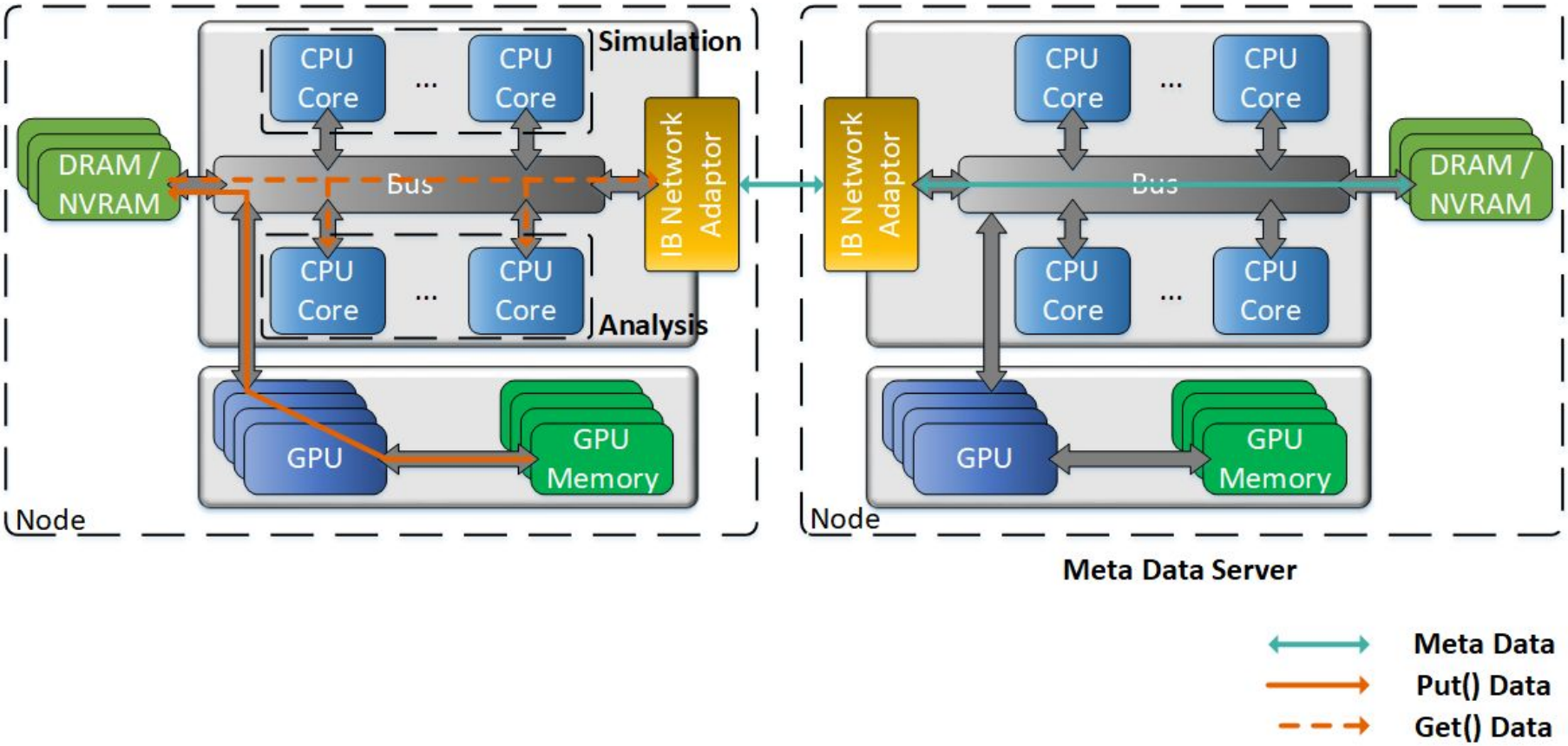


Instead of having a set of dedicated data staging servers, we can distribute the staging region to data producer's CPU memory.

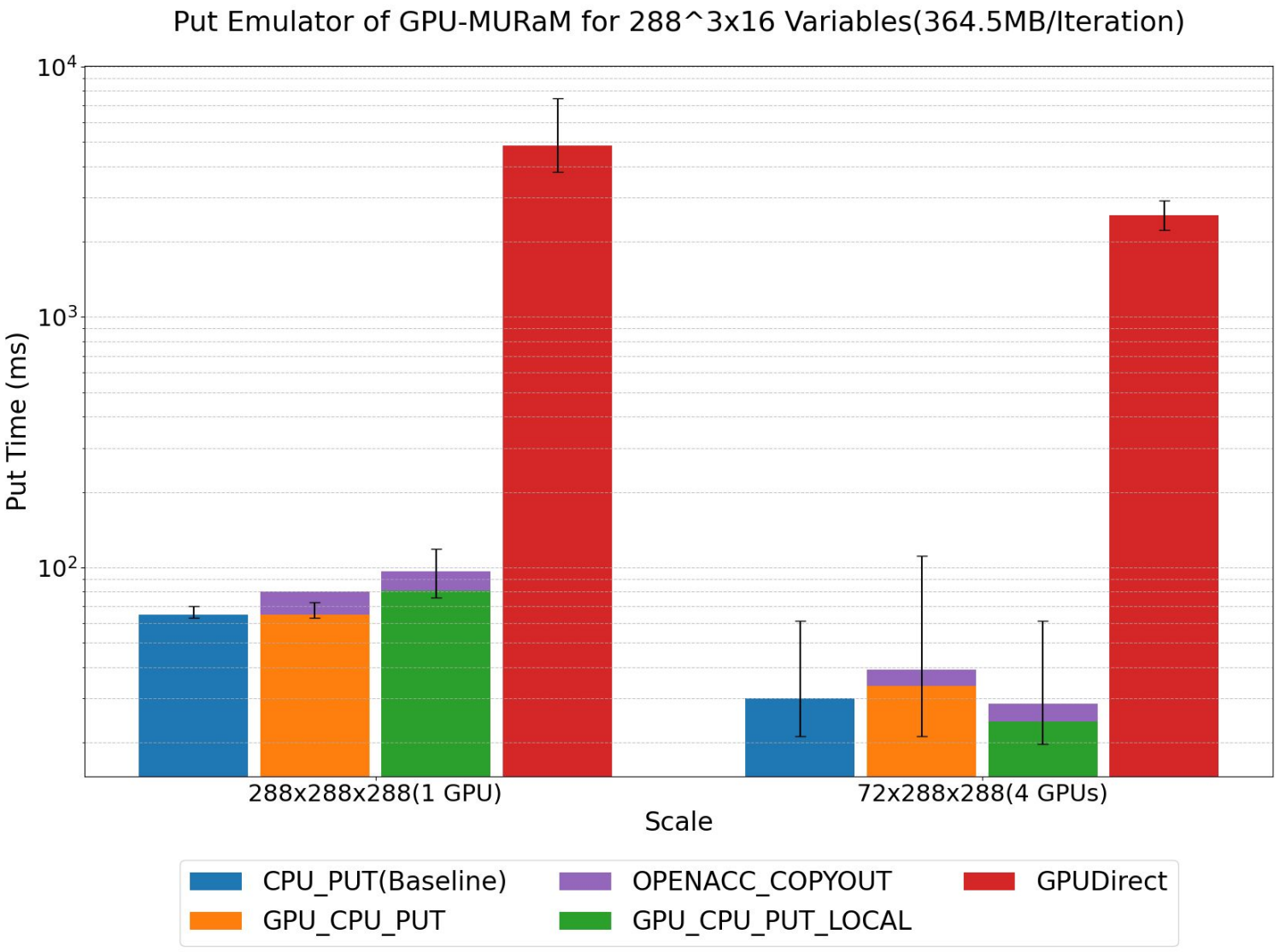
Method 2: Data Flow Details (Local Data Storage)



Method 2: Data Flow Details (Local Data Storage & Co-located Apps)



Evaluation


Put Emulator of GPU-MURaM for 288^3x16 Variables(364.5MB/Iteration)
Put Time (ms)
10^4
10^3
10^2
288x288x288(1 GPU)
72x288x288(4 GPUs)
Scale
CPU_PUT(Baseline)
GPU_CPU_PUT
OPENACC_COPYOUT
GPU_CPU_PUT_LOCAL
GPUDirect

Finished Work

So far, the work I finished includes:

- Learning about MURaM simulation I/O
- Building DataSpaces on CASPER
- Integrating DataSpaces as a staging backend to MURaM simulation.
- Exploring the GPUDirect over Infiniband options as the network layer
- Implementing the test code for proposed methods
- Collecting evaluation results

All the detailed documentation could be found in my wiki space:

https://wiki.ucar.edu/pages/viewpage.action?spaceKey=~bz186&title=SIPARCS-Project6-InSitu

Lessons Learned

So far, some Lessons I learned includes:

- How MURaM simulation I/O behaves
- How to run MURaM on CASPER
- Environmental variables and special configurations for building DataSpaces together with MURaM on CASPER
- GPUDirect over Infiniband functionality resides in rdma-core
- The network card configuration is completely different between the GPU nodes and CPU nodes on CASPER
- How to profile using Nvidia's toolkit.
- Performance of GPUDirect over Infiniband is still bad; needs optimization

Next Steps

For the next stage, my planned work includes:

- Migrating all the methods which are finished in test code to the MuRAM
- Collecting the evaluation results for MuRAM
- Reproduce the result on another machine (Frontrea at TACC)
- Profiling the test code and try to find out the bottleneck
- Replacing the network substrate from OFI to MPI to check the performance

Acknowledgement

Administrative Support

- AJ Lauer
- Virginia Do
- Jerry Cyccone
- Max Cordes Galbraith

Research Support

- John Dennis
- Matthias Rempel
- Haiying Xu

Technical Support

- Brian Vanderwende

Q & A

Thanks for listening!