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

Bo Zhang[1,2], Damir Pulatov[1,3], Supreeth Suresh[1], Cena Miller[1], Manish Parashar[4]

[1]National Center for Atmospheric Research [2]Rutgers University
[3]University of Wyoming [4]University of Utah

Motivation

- ❑ MPS/University of Chicago Radiative MHD(MURaM) has been ported to a scalable GPU version.
- ❑ As computation is optimized, I/O and post processing becomes the next major bottleneck.
- ❑ Creating an in-situ workflow along with a staging-based IO subsystem is a critical problem that need to be addressed.

Goal

- ❑ Build up a staging-based in-situ I/O subsystem for MURaM
- ❑ Employ GPUDirect with OpenFabric to enable direct data movement between GPUs and remote staging servers.
- ❑ Explore local staging method to efficiently use resources on the heterogeneous nodes

Introduction to DataSpaces

- ❑ We use DataSpaces as the data staging infrastructure for loosely-coupled MURaM workflow.
- ❑ DataSpaces implements a virtual shared-space abstraction that can be accessed concurrently by all applications in a coupled simulation workflow.

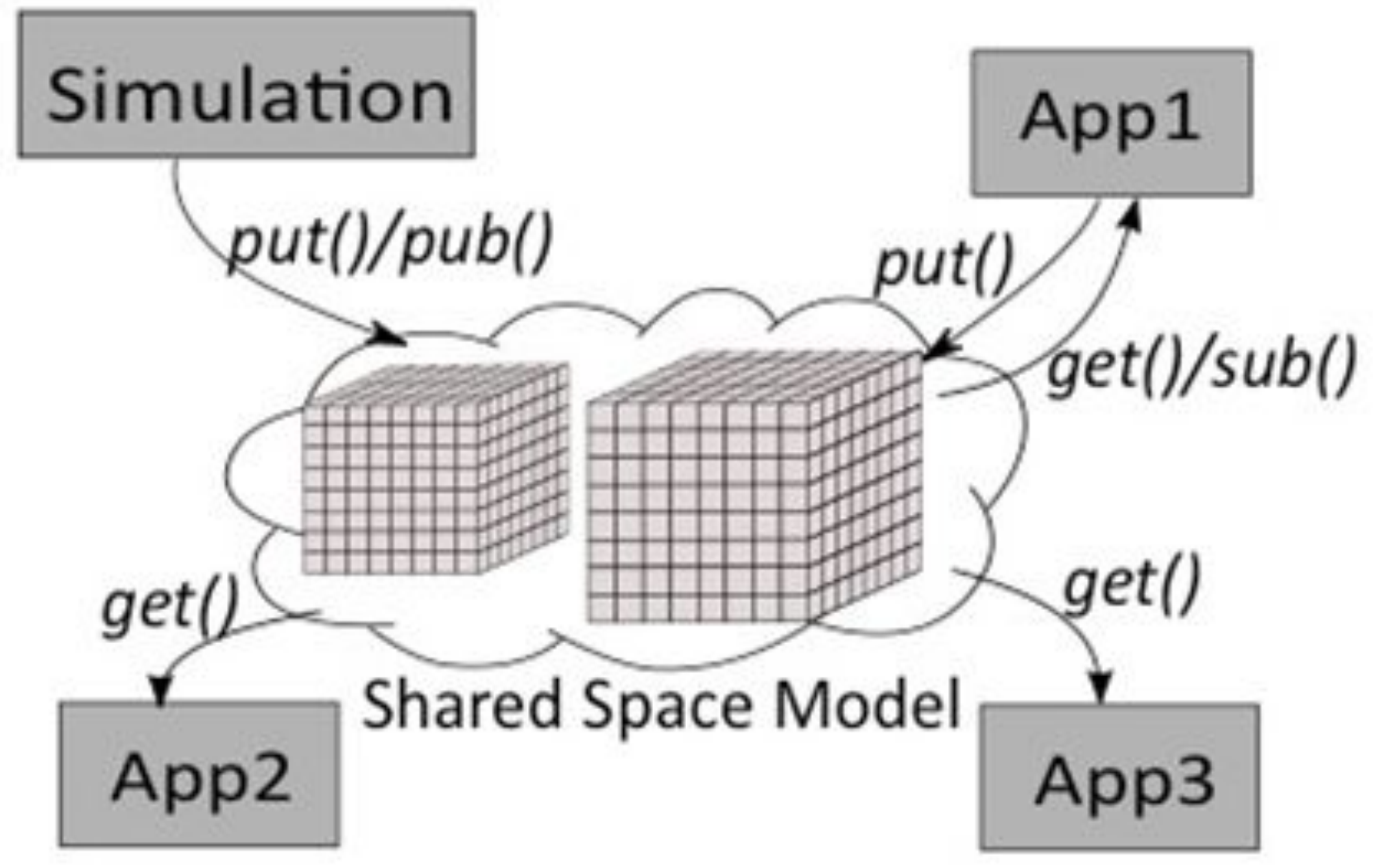

Fig 1 DataSpaces Model

- ❑ DataSpaces uses N-dimensional bounding box indexing and allows flexible partial data access pattern.

References

Docan, Ciprian, Manish Parashar, and Scott Klasky. "Dataspaces: an interaction and coordination framework for coupled simulation workflows." Cluster Computing 15.2 (2012): 163-181.
Wright, Eric et al. "Refactoring the MPS/University of Chicago Radiative MHD(MURaM) Model for GPU/CPU Performance Portability Using OpenACC Directives." (2021).

Approach 1: Remote Staging with GPUDirect

Remote Staging with GPUDirect approach removes inessential data movement to host memory

Put:

1. GPU directly moves the data located in its memory to the infiniband network adaptor.
2. Data is transferred over infiniband network, reaching staging server's main memory.

Get:

1. CPU sends the get request to staging server.
2. Data is transferred over infiniband network from staging server's main memory.
3. Infiniband network adaptor directly writes the data to GPU memory.

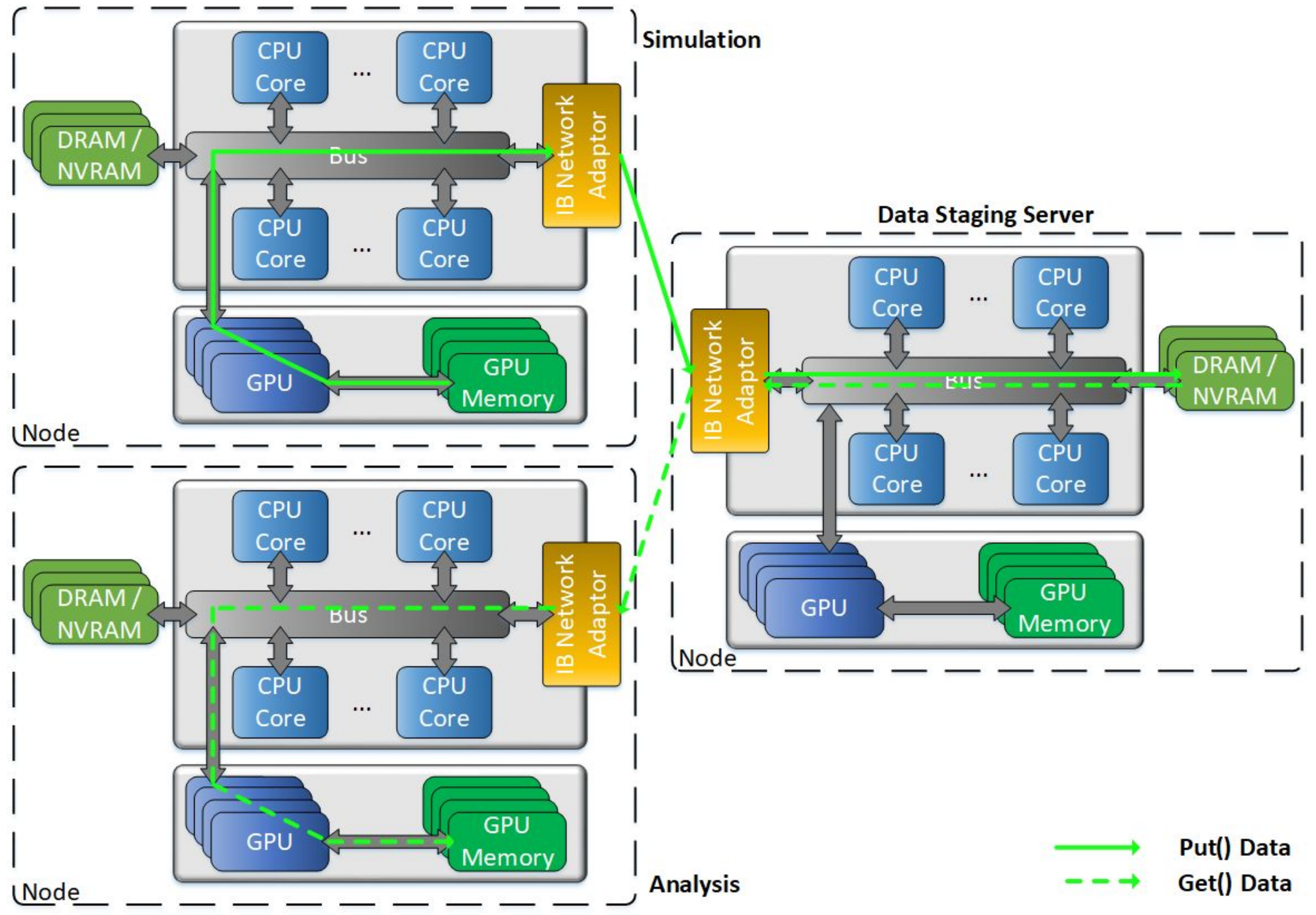

Fig 2 Remote Staging with GPUDirect

Approach 2: Local Staging

Local Staging approach efficiently uses the idle host resources while the computation workload is migrated to device

Put:

1. GPU copys the data to main memory.
2. CPU updates the meta data to server.

Get:

1. CPU sends the get request to meta data server.
2. Meta data server redirects the request to storage client.
3. Data is transferred over infiniband network from storage client's main memory.
4. Infiniband network adaptor directly writes the data to GPU memory.

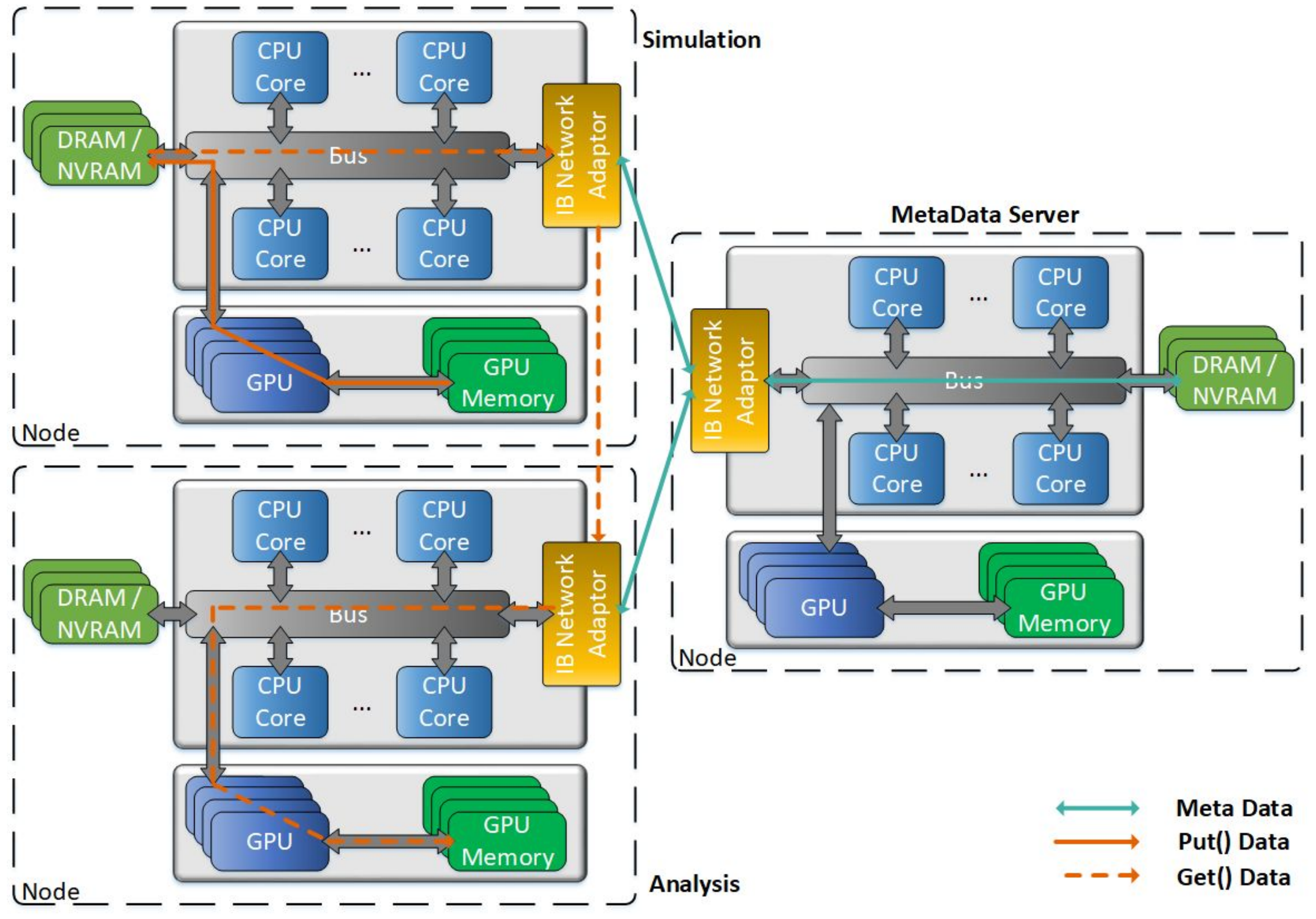

Fig 3 Local Staging

Resource Utilization Changes

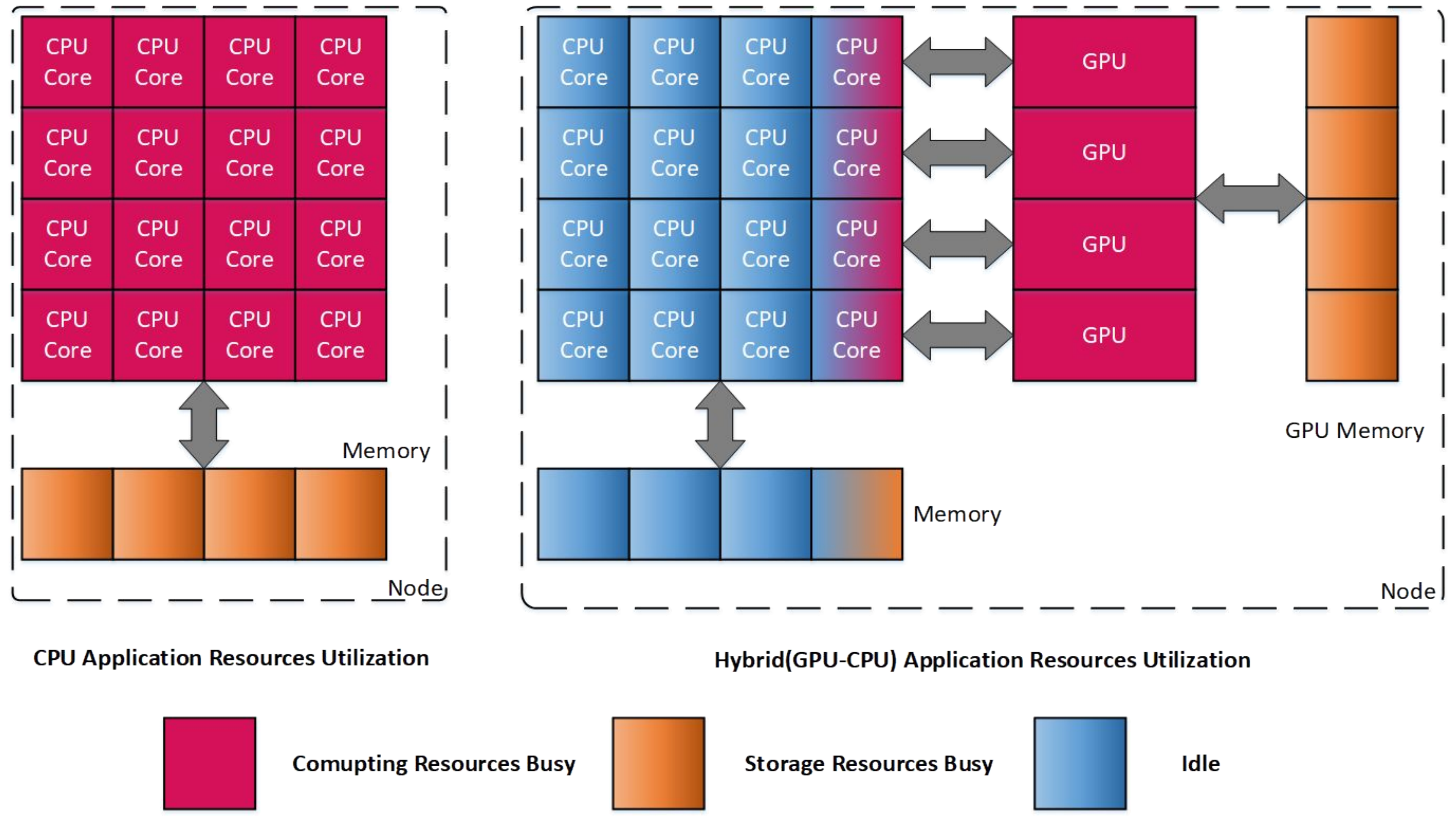
CPU Core
Memory
Node
GPU
GPU Memory
CPU Application Resources Utilization
Hybrid(GPU-CPU) Application Resources Utilization
Computing Resources Busy
Storage Resources Busy
Idle

Fig 4 Resource Utilization Changes due to Porting Applications to GPU

Results & Future work

Result:

- ➢ Remote Staging with GPUDirect performs 100x slower than the baseline
- ➢ Local Staging performs better than baseline at small scale (<100MB)

Future Work:

- ➢ Profile the Remote Staging with GPUDirect approach to find out the bottleneck
- ➢ Optimize Local Staging approach to achieve the speed of OpenACC copyout
- ➢ Migrate these approaches to GPU-MURaM production code

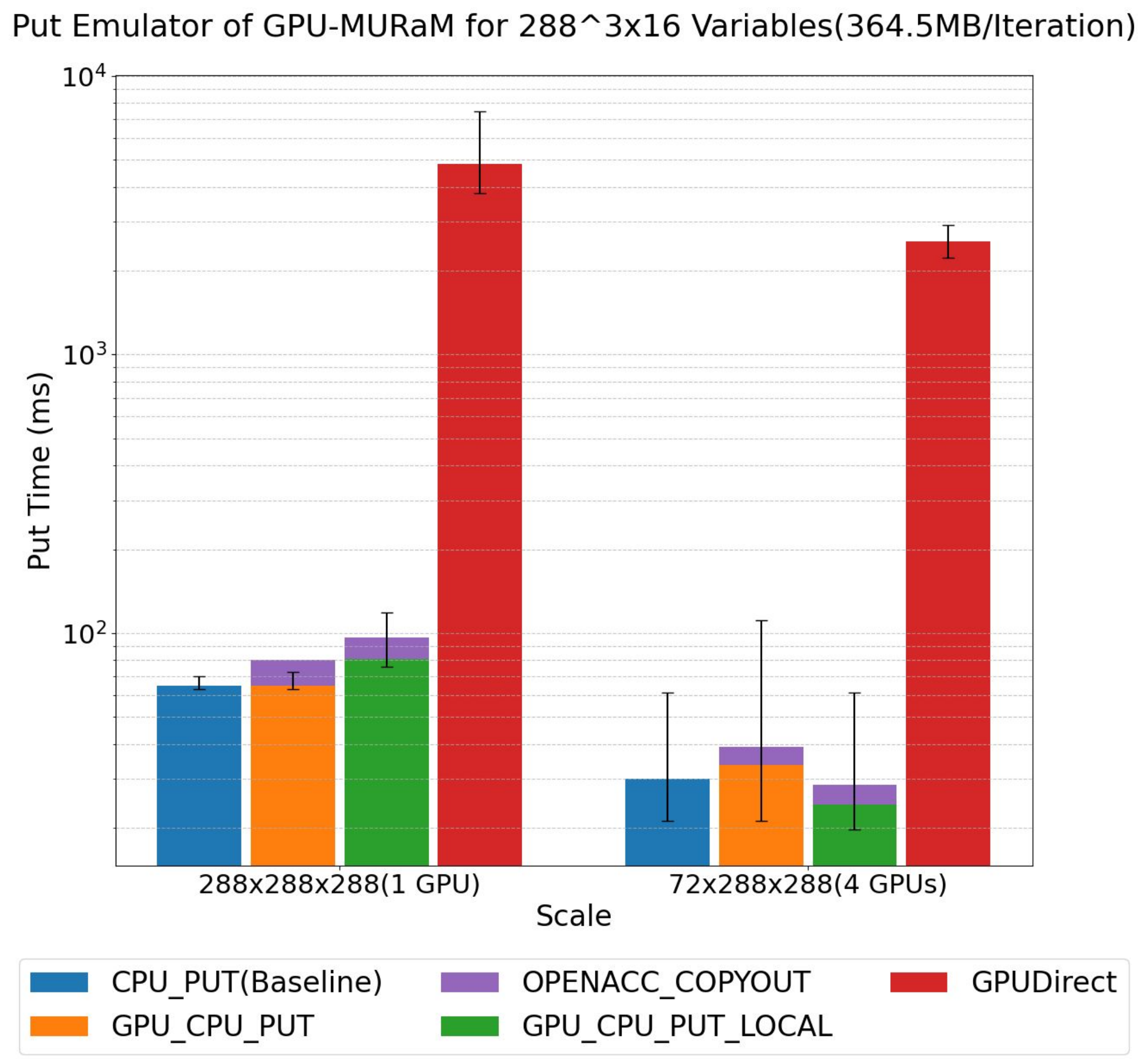

Fig 5 Evaluation Result

Acknowledgements

SIParCS Admins and CODE Assistants: AJ Lauer, Virginia Do, Jerry Cyccone, Max Cordes Galbraith **Research Support:** John Dennis, Matthias Rempel, Haiying Xu **Technical Support:** Brian Vanderwende