Performance Analysis of MPI Over Cluster

Presented by:

Achuth P V, Aniruddh Rao K, Avishkar P

Introduction

- Cluster: Widely used in HPC to scale computing
- MPI: Message Passing Interface (Standard)
  - OPENMPI
  - MVAPICH
  - MPICH
- Communication methods
  - Ethernet
  - InfiniBand
  - Shared Memory
  - RDMA
- CUDA and CUDA aware MPI
  - CPU-GPU
  - GPU-GPU (GPU Direct)

Tools and Methods for MPI

- Cluster of 2 machines each with 16 CPU cores connected by
  - IP over InfiniBand
  - Ethernet
- MPI Implementations used:
  - OPENMPI v 1.8.4
  - MVAPICH2 v 2.1
- Modes of communication:
  - InfiniBand
  - Ethernet
  - Shared Memory
- Matrix Multiplication test code
  - Matrix size from 2000x2000 to 10000x10000
  - With number of processes ranging from 8 to 32

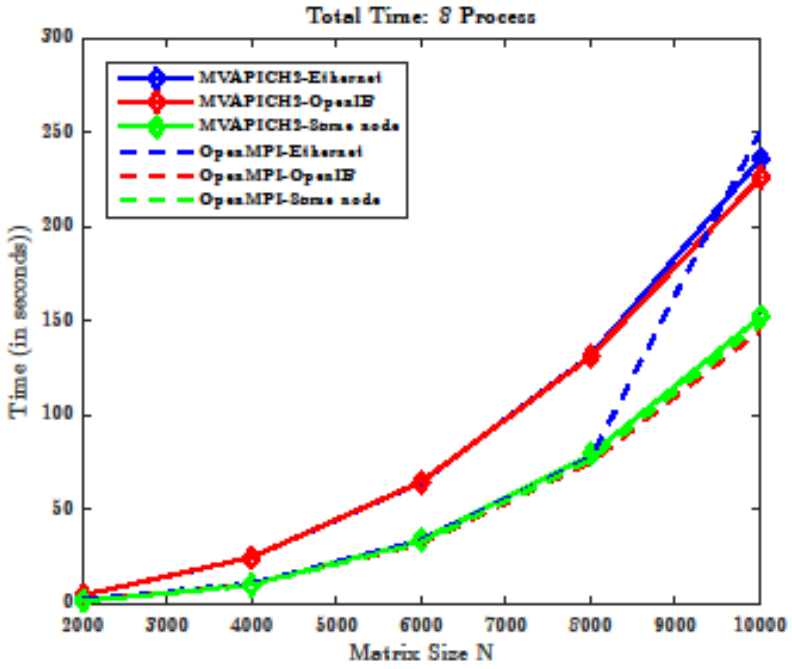
Tools and Methods for CUDA aware MPI

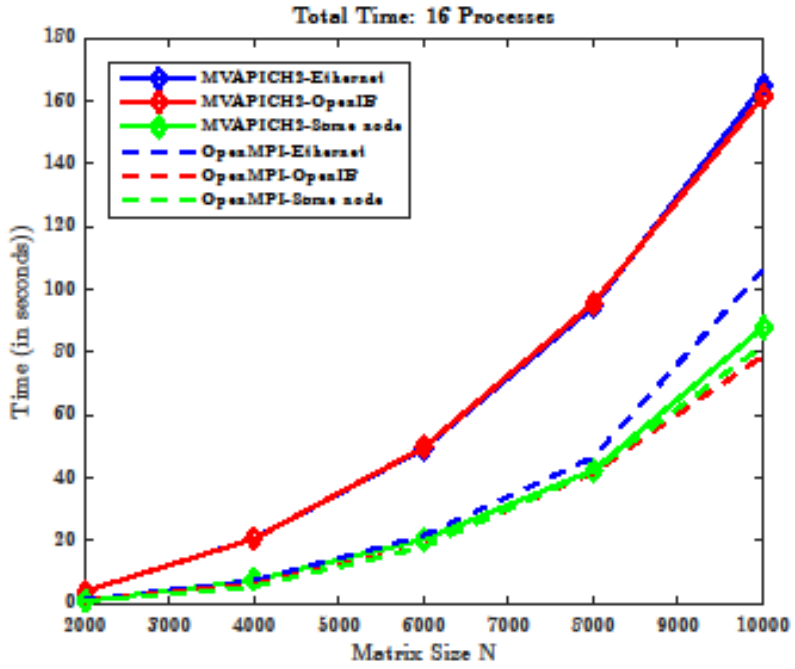
- Setup cluster of 2 machines each with 4 CPU cores, GT 640 NVIDIA CUDA card
  - CUDA Aware MPI
  - CUDA Over MPI
- MPI Implementations used:
  - OPENMPI v 1.8.8
  - MVAPICH2 v 2.1
- Modes of communication:
  - Ethernet
  - Shared Memory
- Test code
  - Message sending code and Matrix multiplication
  - Message size varying from 1 Byte to 4MB

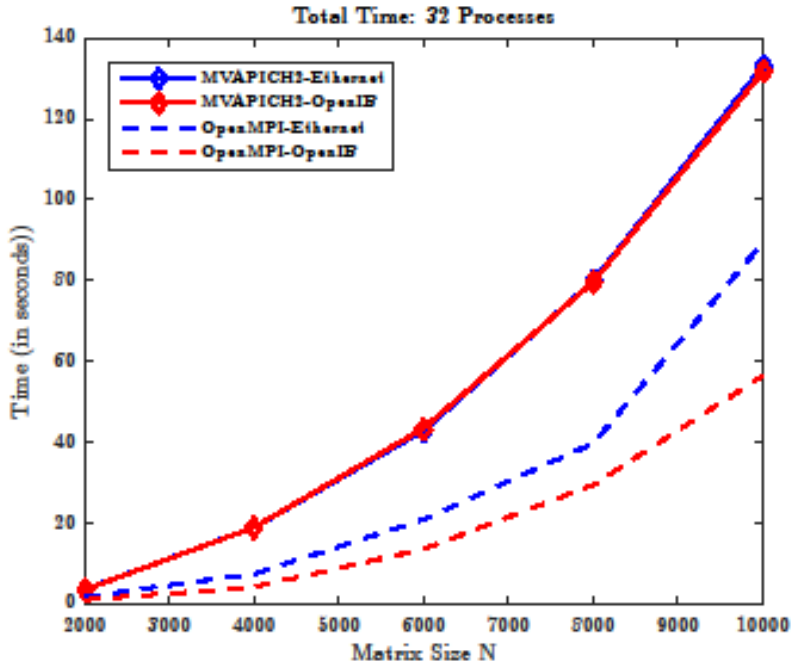
Code flow and Plots for MPI

- Code flow : At Each Node
  - Broadcast B
  - Transfer part of A from PE 0 to others
  - Compute part of C
  - Transfer part of C from all processes to PE 0
- Plotting : (with fixed number of processes: 8, 16 and 32 for matrix size from 2000x2000 to 10000x10000 and different modes of communication Viz InfiniBand, Ethernet and Shared Memory)
  - Plot for Full timing of code
  - Plot for time taken to transfer A
  - Plot for time taken to broadcast B
  - Plot for time taken to transfer C

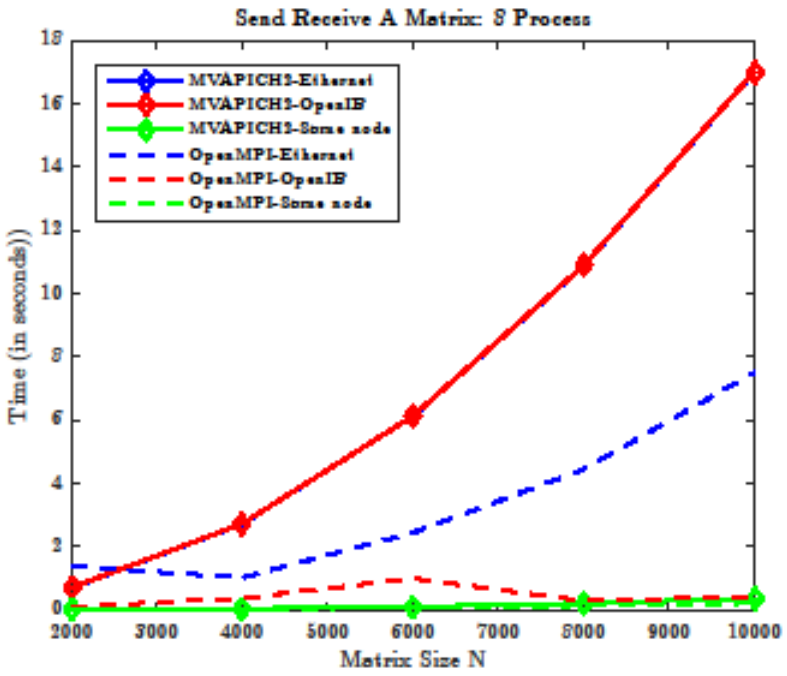
Plots for full timing of Code

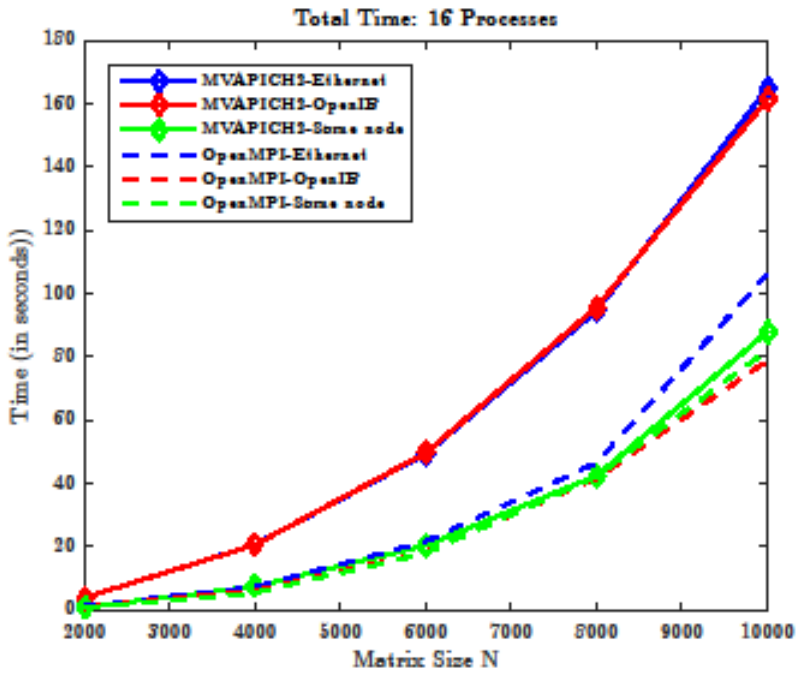


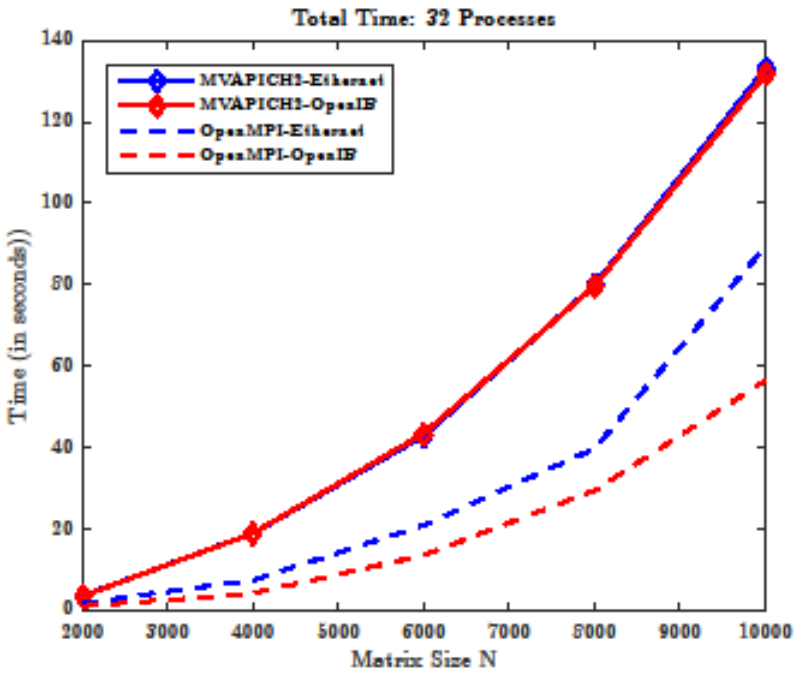




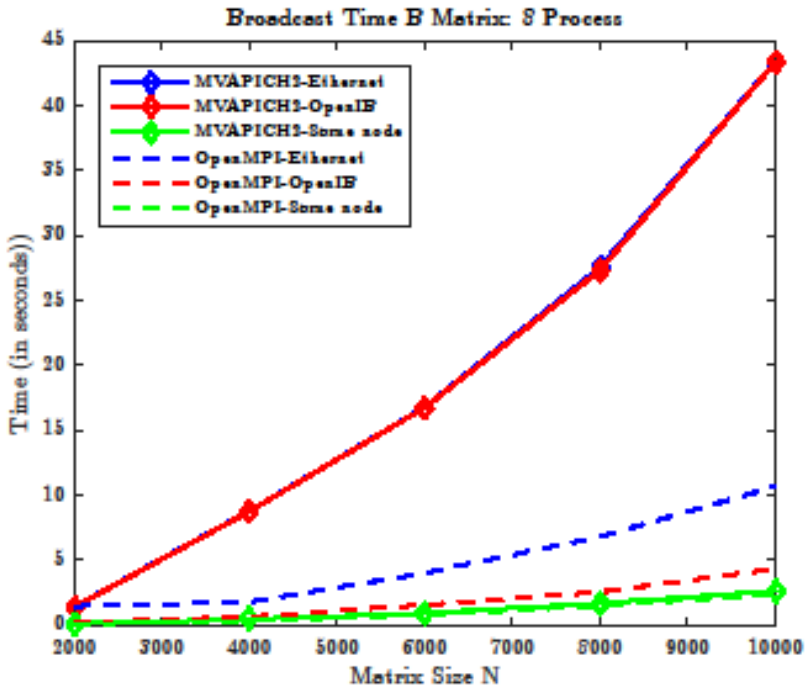
Plots for time taken to transfer A

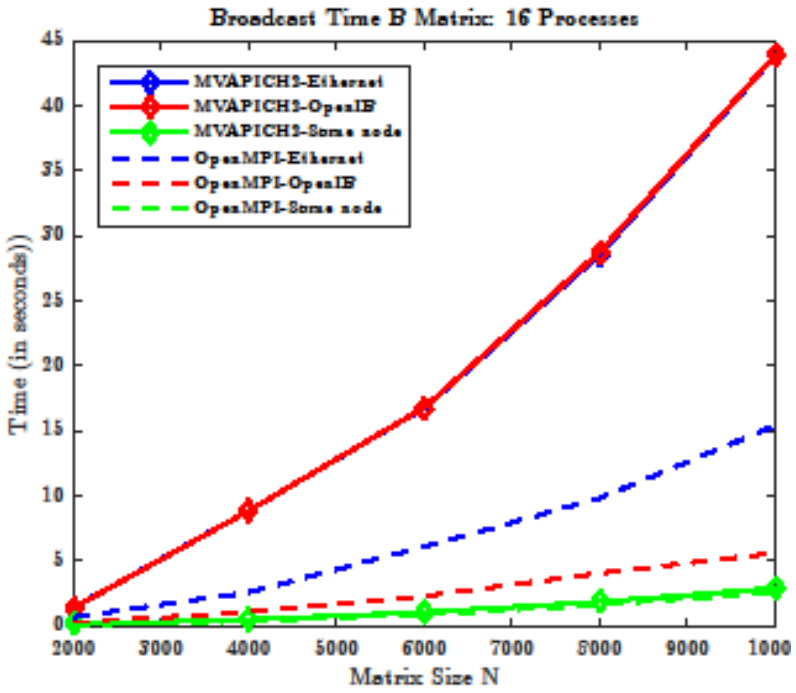


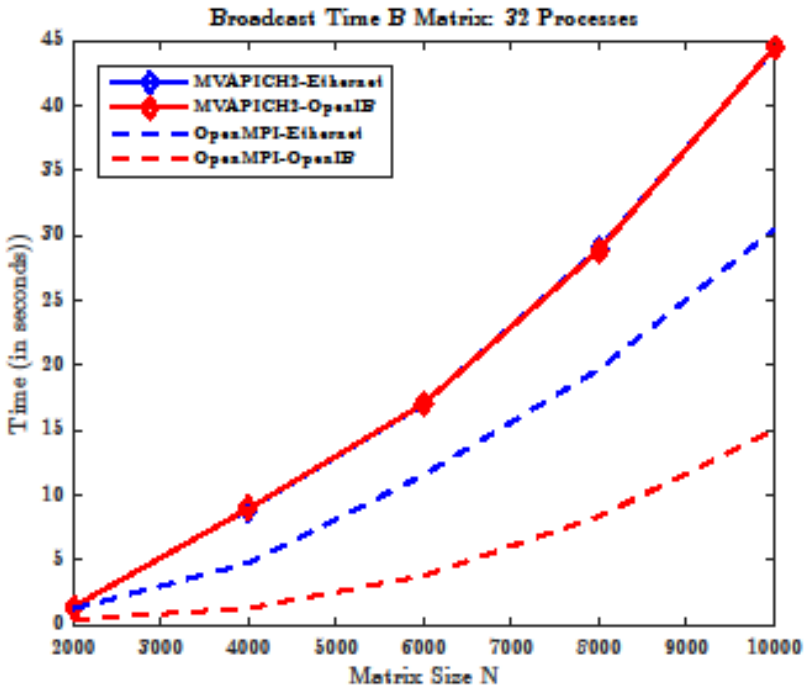




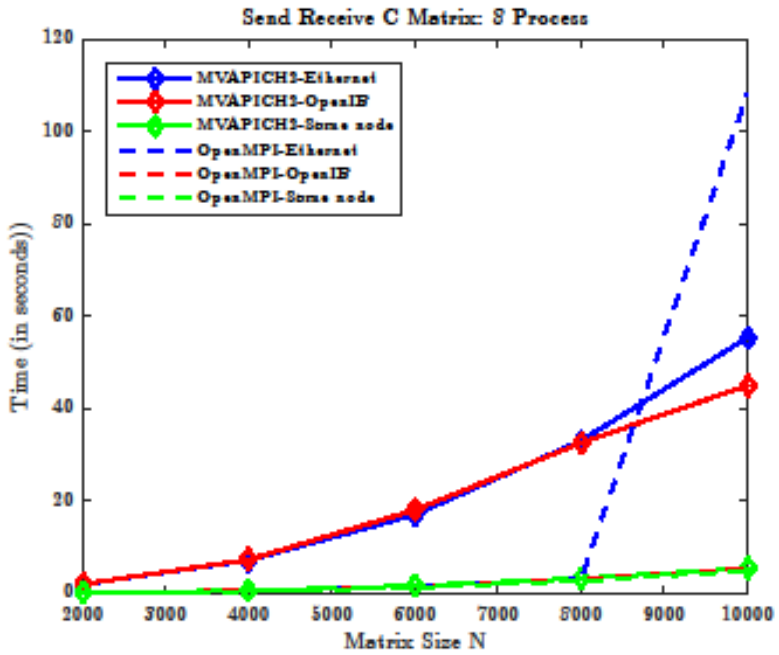
Plots for time taken to Broadcast B

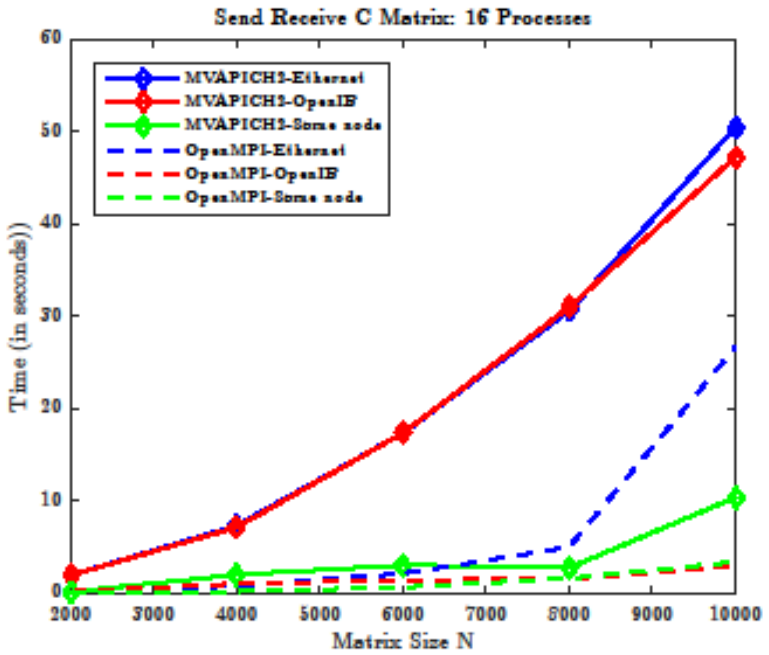


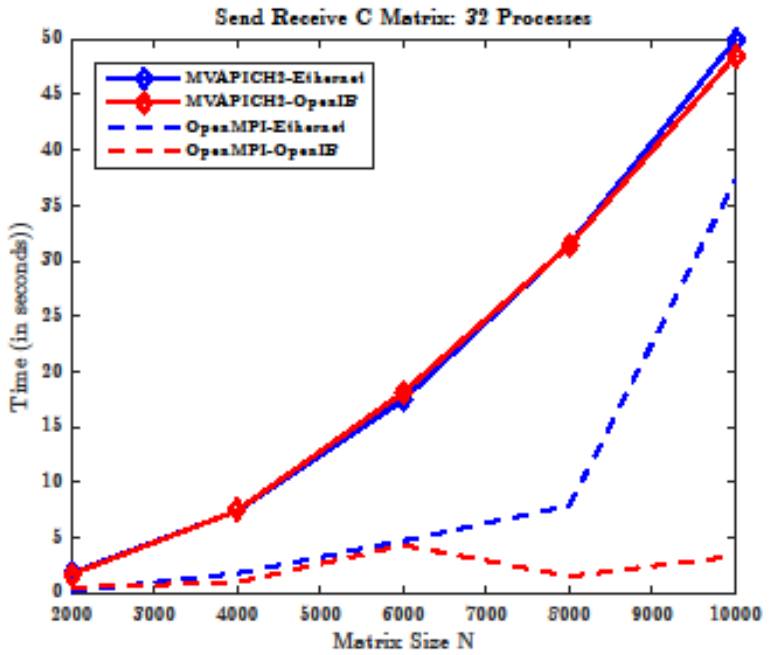




Plots for time taken to transfer C







Observations and Inference

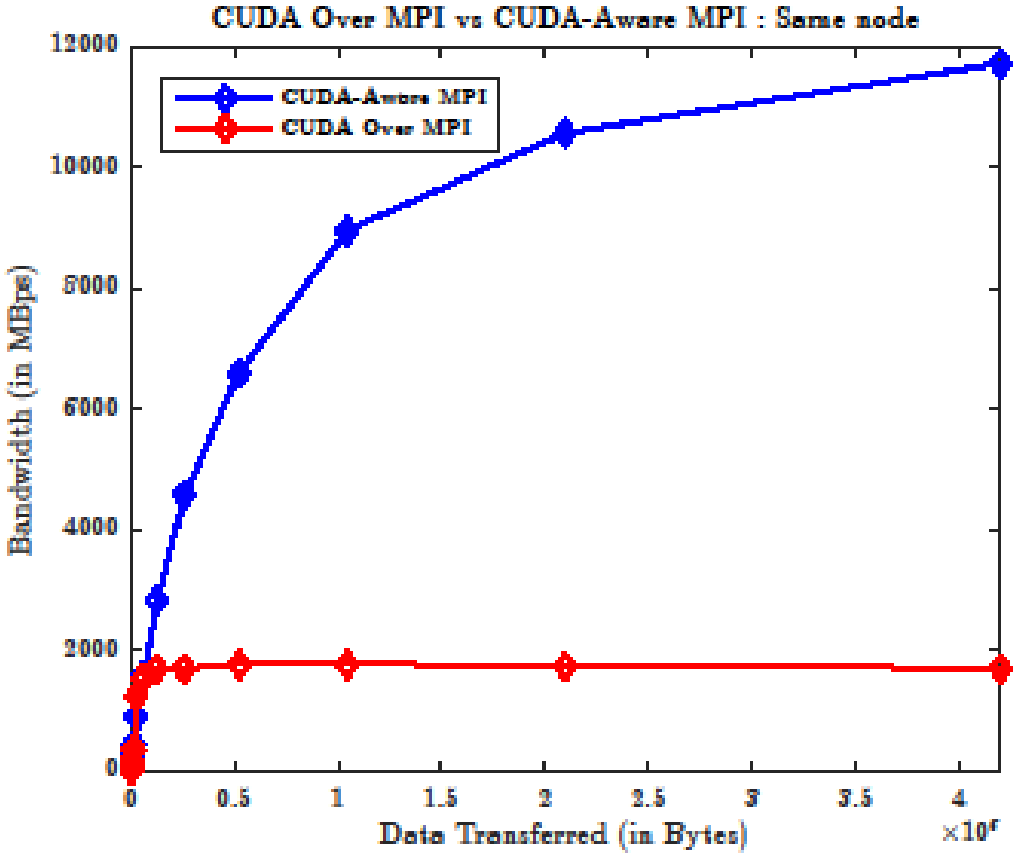
- InfiniBand Communication faster than Ethernet
  - Clear in OpenMPI
  - Not so clear in MVAPICH2
- Intra process Communication faster than both InfiniBand and Ethernet
- For Small N (matrix size), InfiniBand comparable to Intra Process
- With increased np, computation time per node decreases, but communication time remains almost same
  - Each node gets lesser number of computations
  - Time to Bcast B is largest among all communications and remains same
- OpenMPI has better performance than MVAPICH
  - May be due to configuration of MVAPICH (was setup by us –without much knowledge of configuration)
- Non Blocking vs Blocking
  - Non-blocking is faster
  - But Non-blocking is inconsistent

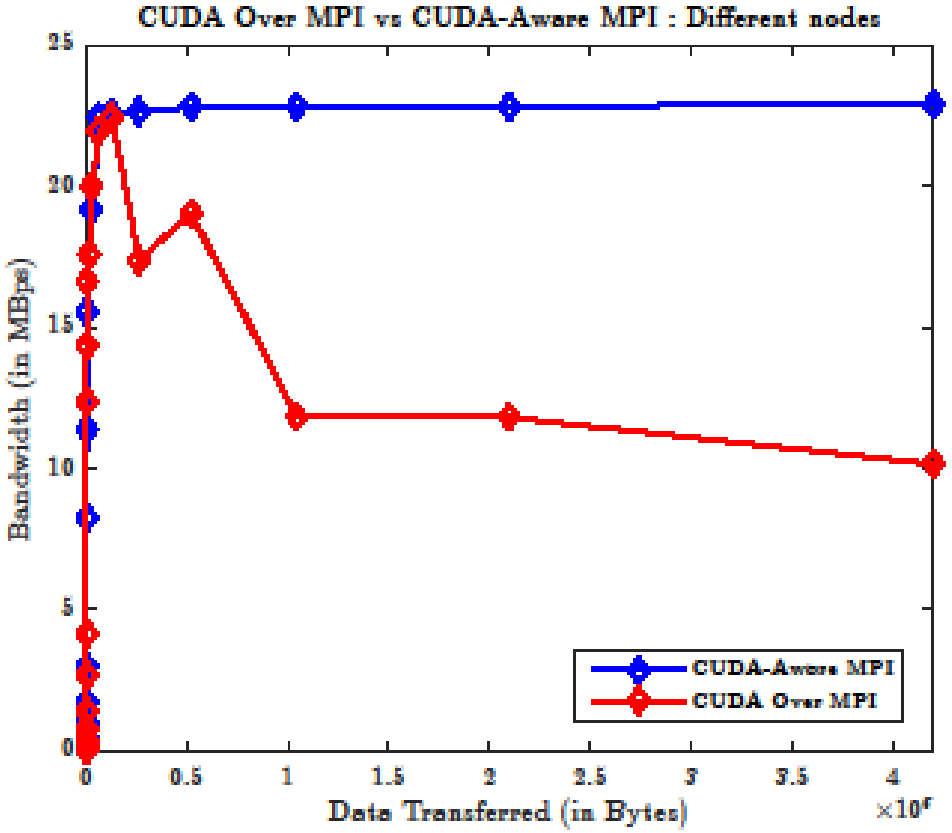
Code flow and Plots for CUDA over MPI

- Code Flow of matrix multiplication
  - Send Parts of A and B to other processes
  - Use GPU to compute
  - Send back Computed part of C to main process

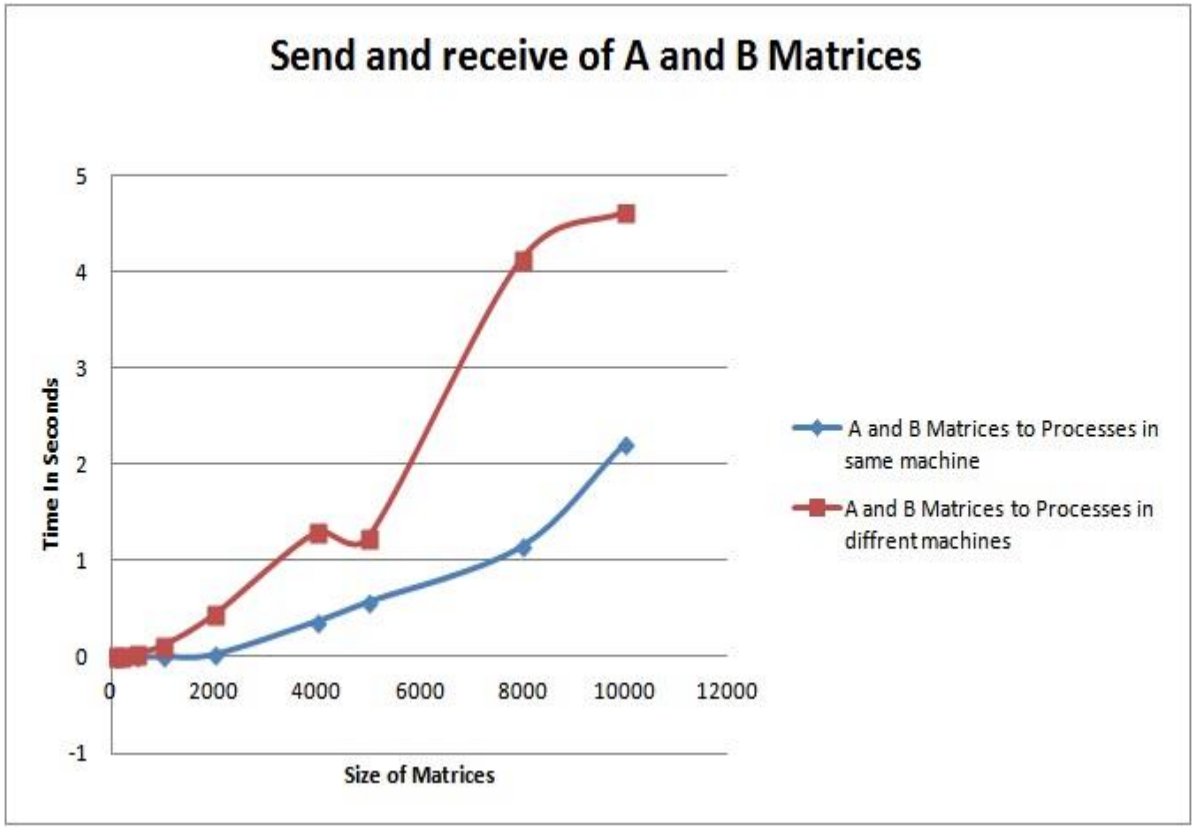
- Plotting (Done for 4 processes: with 2 machines having GPU or only one machine)
  - Plots for BW measurement for CUDA over MPI and CUDA aware MPI
  - Plots for time taken to transfer A and B
  - Plots for time taken to transfer C
  - Plots for time taken for complete code execution

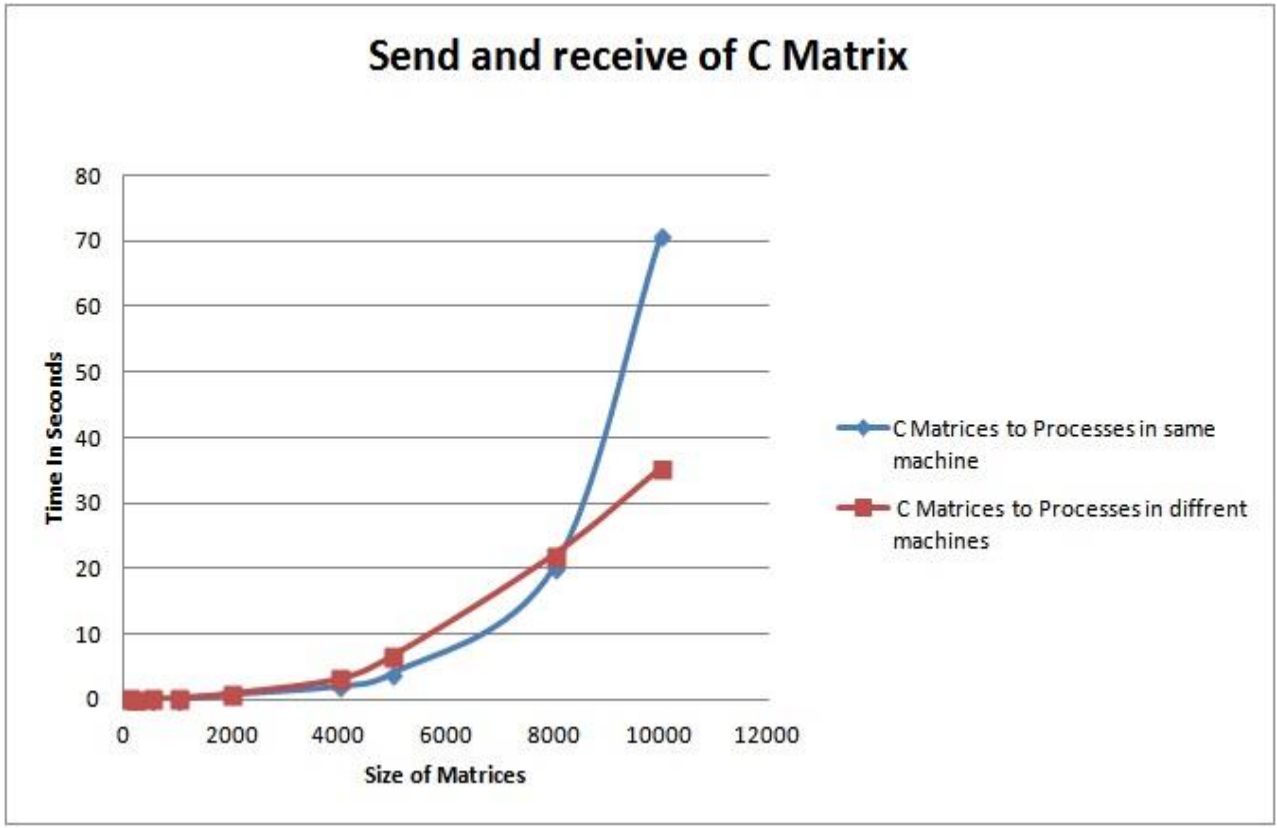
Bandwidth vs Data transferred for CUDA Over and Aware MPI





Plots for time taken to transfer A and B, time taken to transfer C





Plot for complete execution time

Observations and Inference

- Communication was over GbE network
  - Bandwidth in CUDA-Aware-MPI is relatively higher than CUDA-over-MPI for same node.
  - Bandwidth was same for small matrices and then CUDA-over-MPI Bandwidth got deteriorated for big matrices for different nodes
- Transfer over network is slower compared to intra node inter process transfer
  - For larger matrices inter node transfer is getting slower – may be due to shared memory/virtual memory access