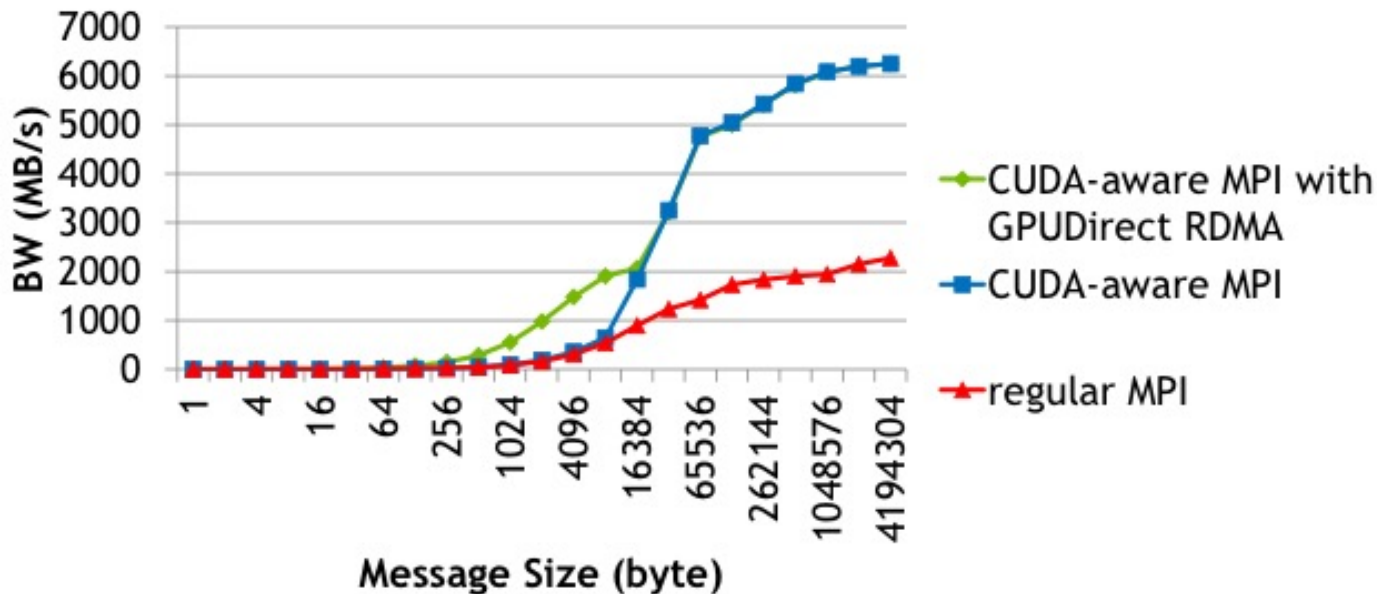


CUDA-aware MPI

```
if( 0 == rank ) {  
    MPI_Send(buf_dev, size, MPI_CHAR, 1, tag, MPI_COMM_WORLD);  
} else { // assume MPI rank 1  
    MPI_Recv(buf_dev, size, MPI_CHAR, 0, tag, MPI_COMM_WORLD, &status);  
}
```

OpenMPI 1.7.4 MLNX FDR IB (4X) Tesla K40

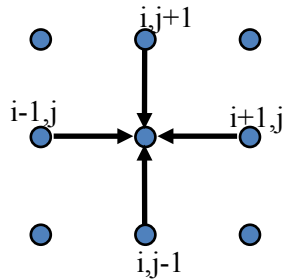


Latency (1 byte) 19.04 us 16.91 us 5.52 us

[MVAPICH2 1.8/1.9b](#)
[OpenMPI 1.7 \(beta\)](#)
[CRAY MPI \(MPT 5.6.2\)](#)
[IBM Platform MPI \(8.3\)](#)
[SGI MPI \(1.08\)](#)

$$U_{i,j}^{n+1} = \frac{1}{4} (U_{i-1,j}^n + U_{i+1,j}^n + U_{i,j-1}^n + U_{i,j+1}^n)$$

Laplace' s equation – MPI + CUDA



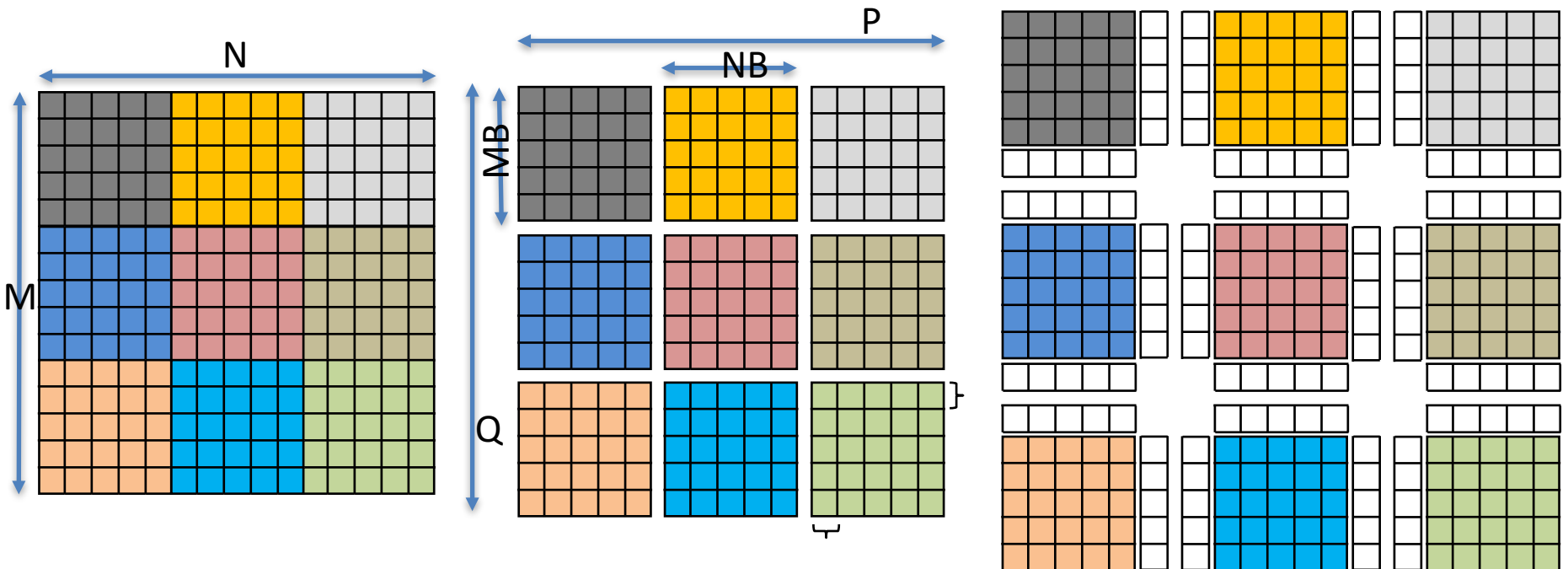
for j = 1 to jmax

for i = 1 to imax

$$U_{\text{new}}(i,j) = 0.25 * (U(i-1,j) + U(i+1,j) \\ + U(i,j-1) + U(i,j+1))$$

end for

end for



Laplace's equation – MPI + CUDA

Data distribution

Create datatypes

Exchange data with neighbors
(north, south, east, west)

Do local computation

- Lets first assume we don't use datatypes (instead we will manually pack/unpack)

