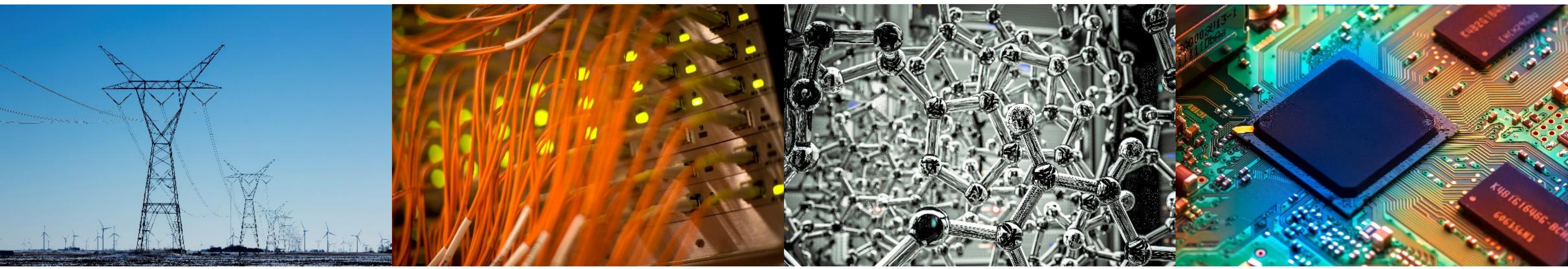
Comparative Performance Evaluation of Multi-GPU MLFMM Implementation for 2-D VIE Problems

Carl Pearson, Mert Hidayetoglu, Wei Ren, Weng Cho Chew, Wen-Mei Hwu
University of Illinois Urbana-Champaign



Outline

This work: compare MLFMM performance on two systems

Brief introduction to **M**ultilevel **F**ast **M**ultipole **M**ethod

Some Implementation Notes

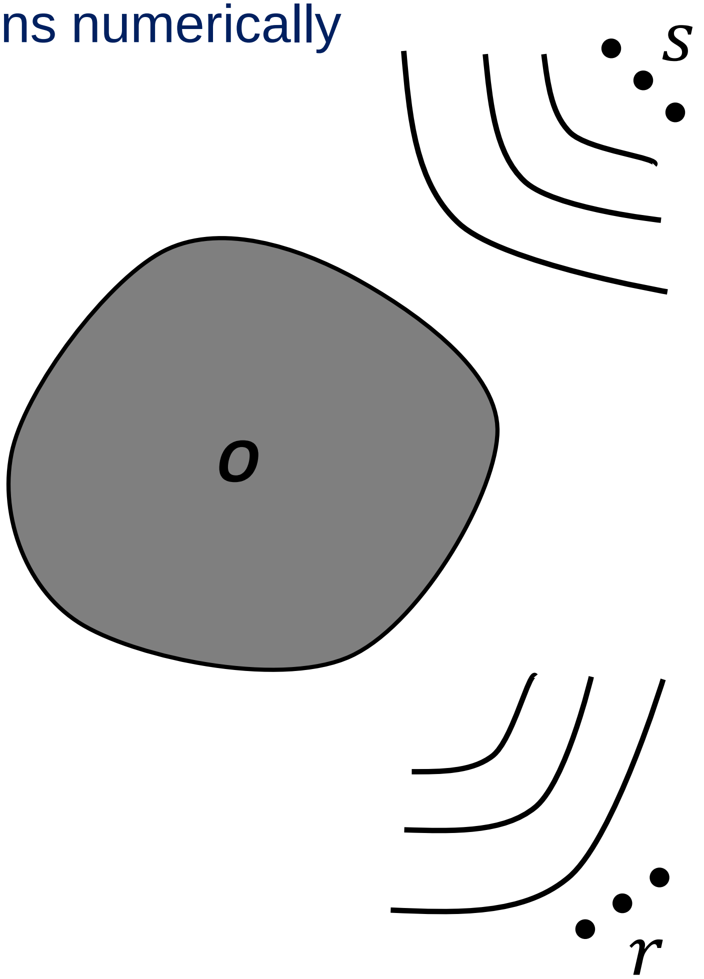
- Matrix Representation of MLFMM Operations
- Kernels and Optimizations
- MPI Parallelization
- Overlapping Communications with Computations

Results

- Blue Waters and IBM S822LC

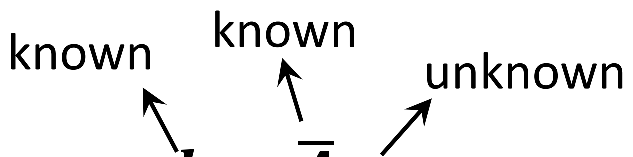
MLFMM

- Solve large scale electromagnetic wave equations numerically
  - Electromagnetics
  - Acoustics
  - Geophysics
  - Radar Cross Section Calculations
  - Medical Imaging
  - Radar Imaging
  - Antenna Design



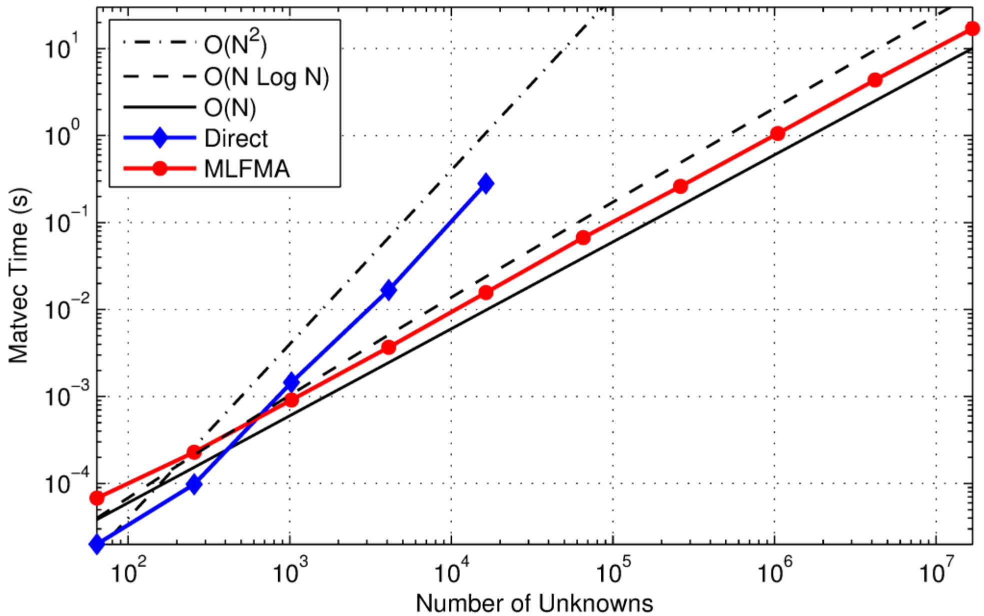
Algorithmic Speedup with MLFMM

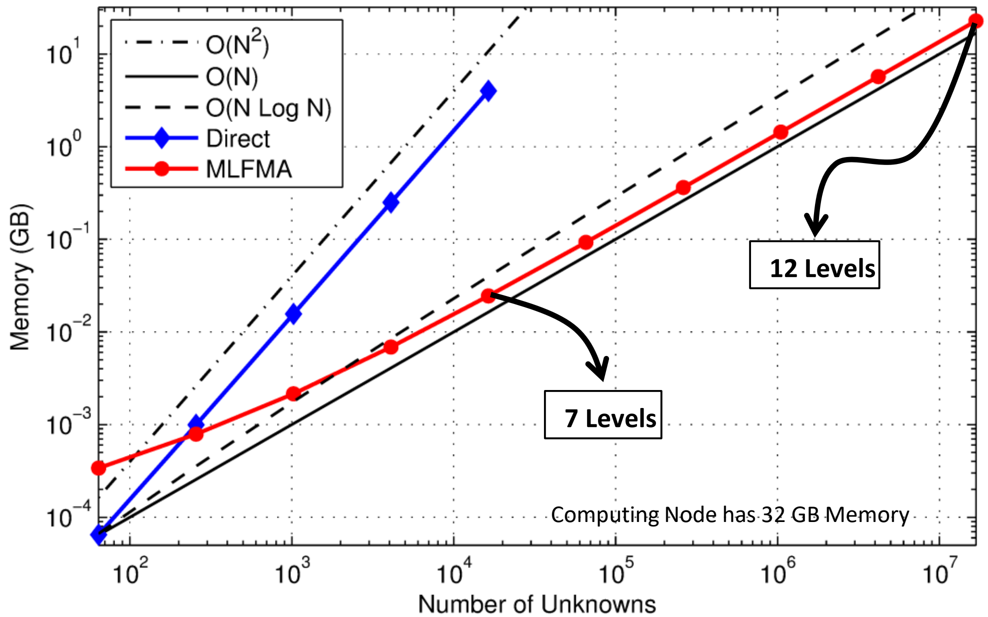
- Direct Methods: $\mathcal{O}(N^3)$
- Iterative Methods: $\mathcal{O}(N^2)$
- Fast Multipole Method: $\mathcal{O}(N^{1.4}) - \mathcal{O}(N^{1.5})$
- Multilevel Fast Multipole Algorithm: $\mathcal{O}(N) - \mathcal{O}(N \log N)$



$$\boldsymbol{b} = \bar{\boldsymbol{A}}\boldsymbol{x}$$

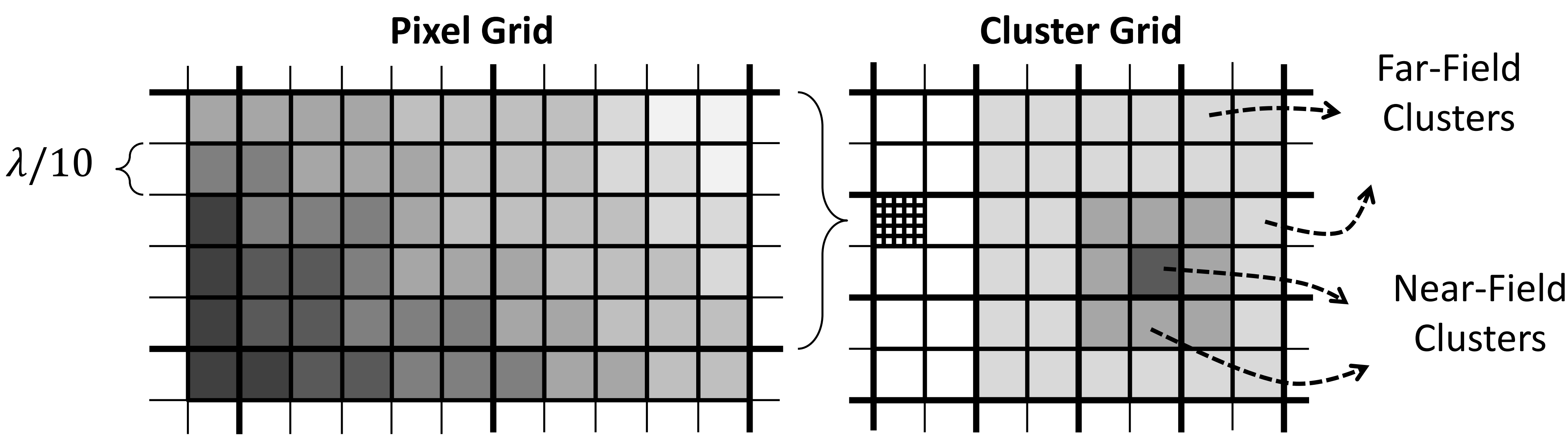
$$\boldsymbol{x} = \bar{\boldsymbol{A}}^{-1}\boldsymbol{b}$$



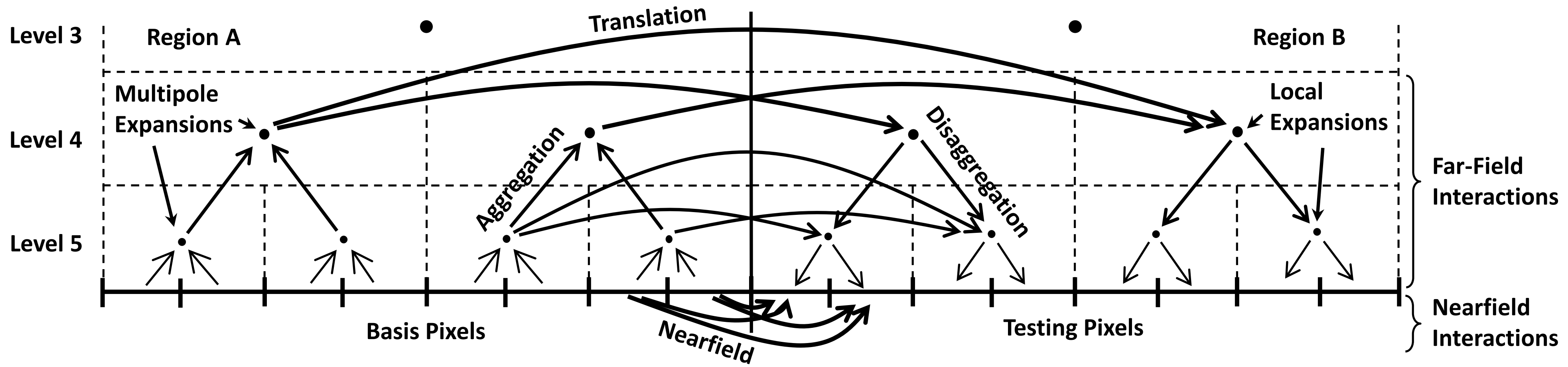


MLFMM has desirable time and memory scaling characteristics

From Approximate to Full-Wave

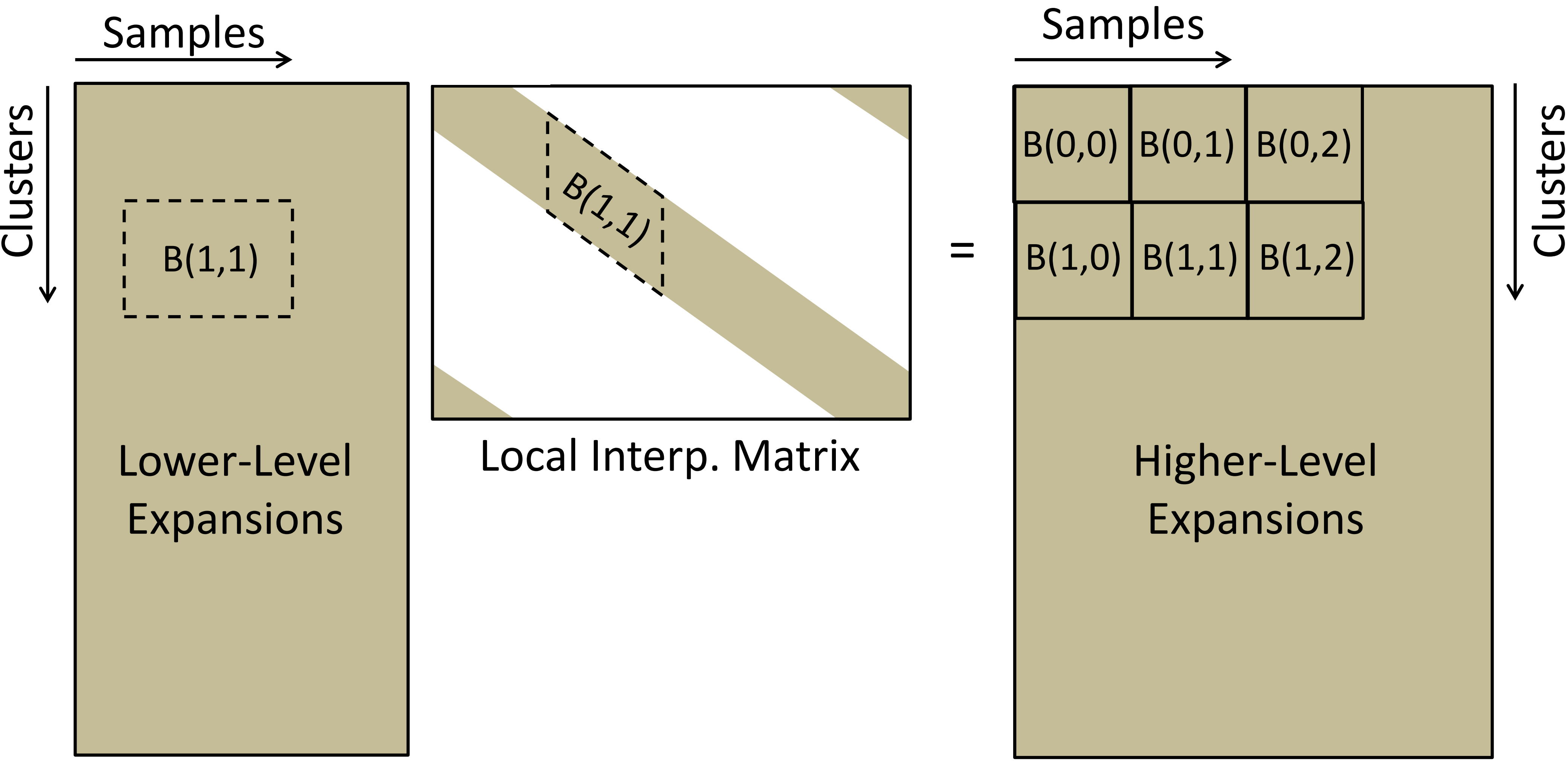


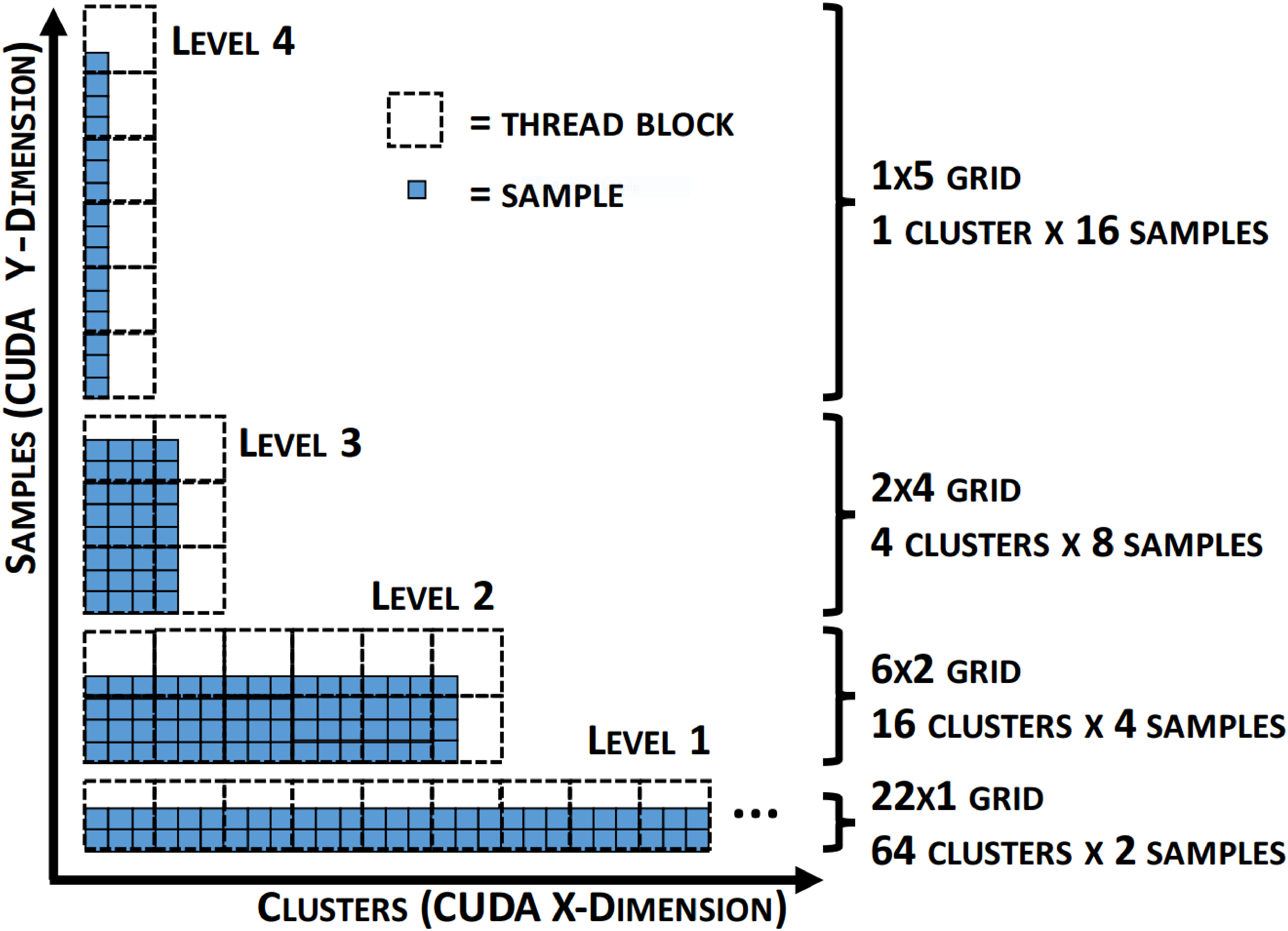
MLFMM Schematic



MLFMA Operation	Structure
Multipole & Local Expansions	Dense
Interpolations & Anterpolations	Band-Diagonal
Multipole & Local Shiftings	Diagonal
Translations	Diagonal
Near-Field Interactions	Sparse

Matrix Formulation of Interpolation




Level 4
Level 3
Level 2
Level 1
= Thread Block
= Sample
Samples (CUDA Y-Dimension)
Clusters (CUDA X-Dimension)
...
1x5 grid
1 cluster x 16 samples
2x4 grid
4 clusters x 8 samples
6x2 grid
16 clusters x 4 samples
22x1 grid
64 clusters x 2 samples

Kernels and Optimizations

Kernel	Mnemonic	MLFMM Operation
P2M	Particle-to-multipole	Aggregation
M2M	Multipole-to-multipole	
M2L	Multipole-to-local	Translation
L2L	Local-to-local	Disaggregation
L2P	Local-to-particle	
P2P	Particle-to-particle	Nearfield

Traditional GPU optimizations:

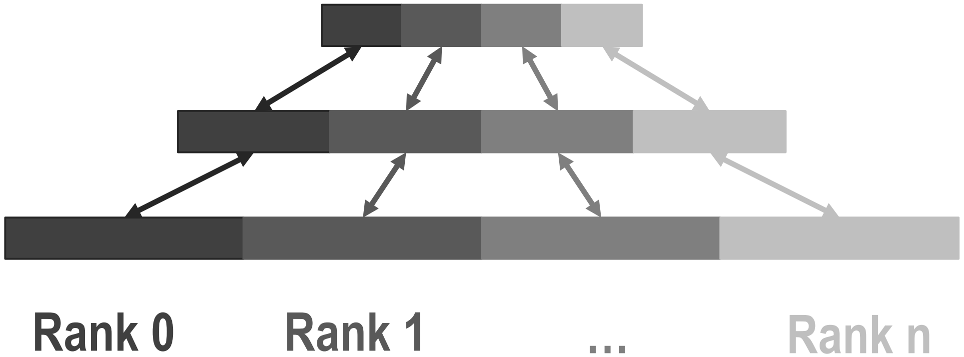
Shared-memory / register tiling

Thread coarsening

MLFMM Tree Parallelization

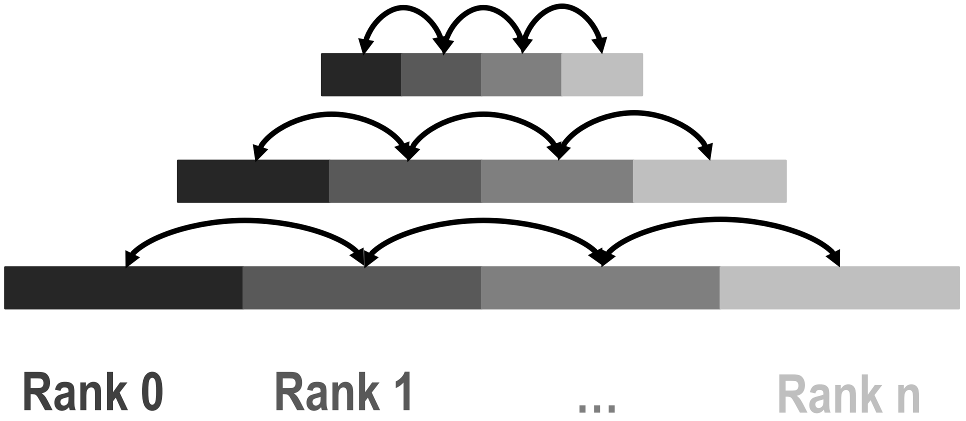
Aggregation / Disaggregation

- No communication



Translation

- Inter-rank communication



Execution Environments

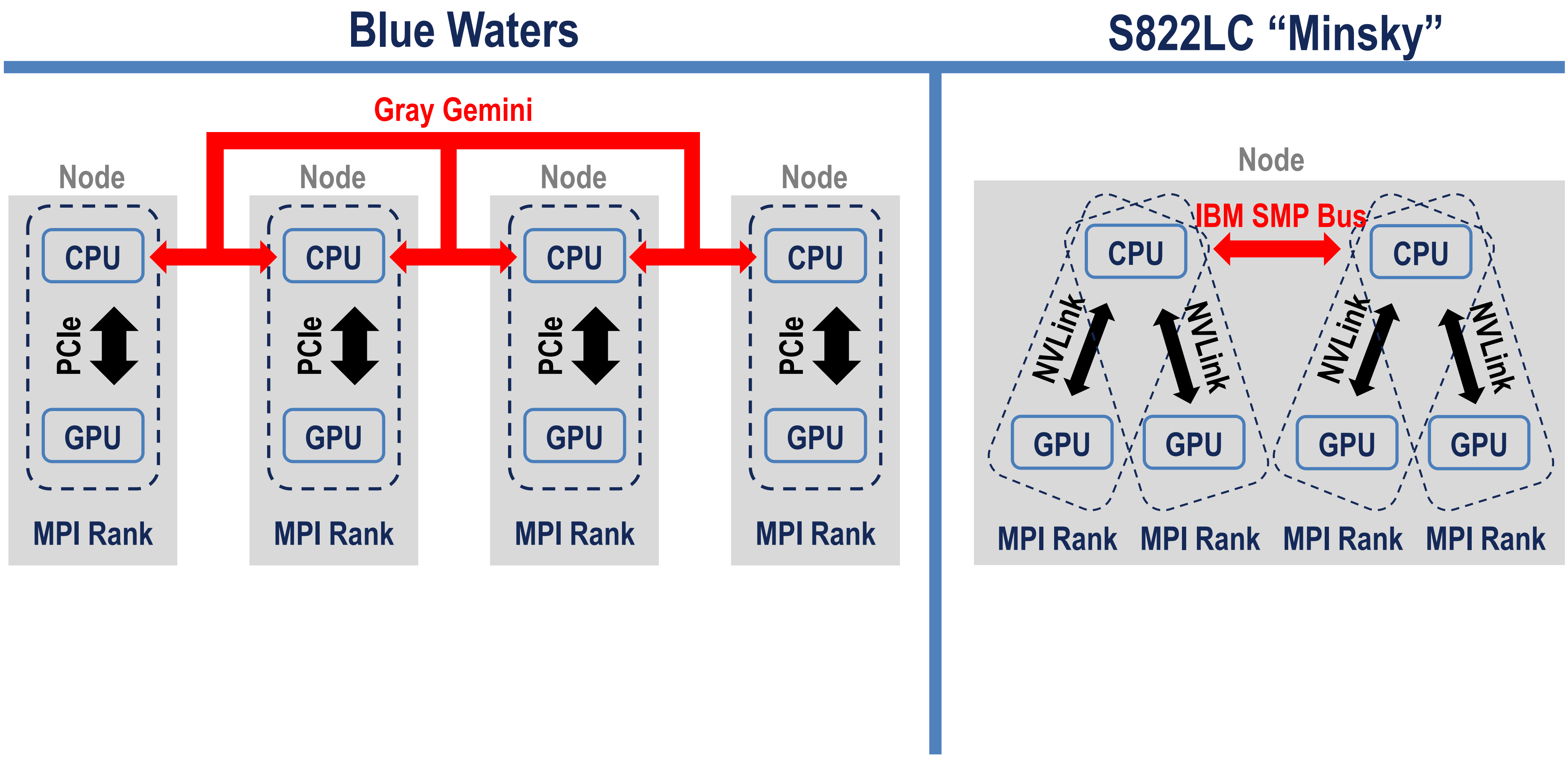
Blue Waters, National Petascale Computing Facility, University of Illinois

- 22,500 XE CPU and 4,200 XK GPU nodes
- 1.5 PB RAM, 13.34 PF peak performance

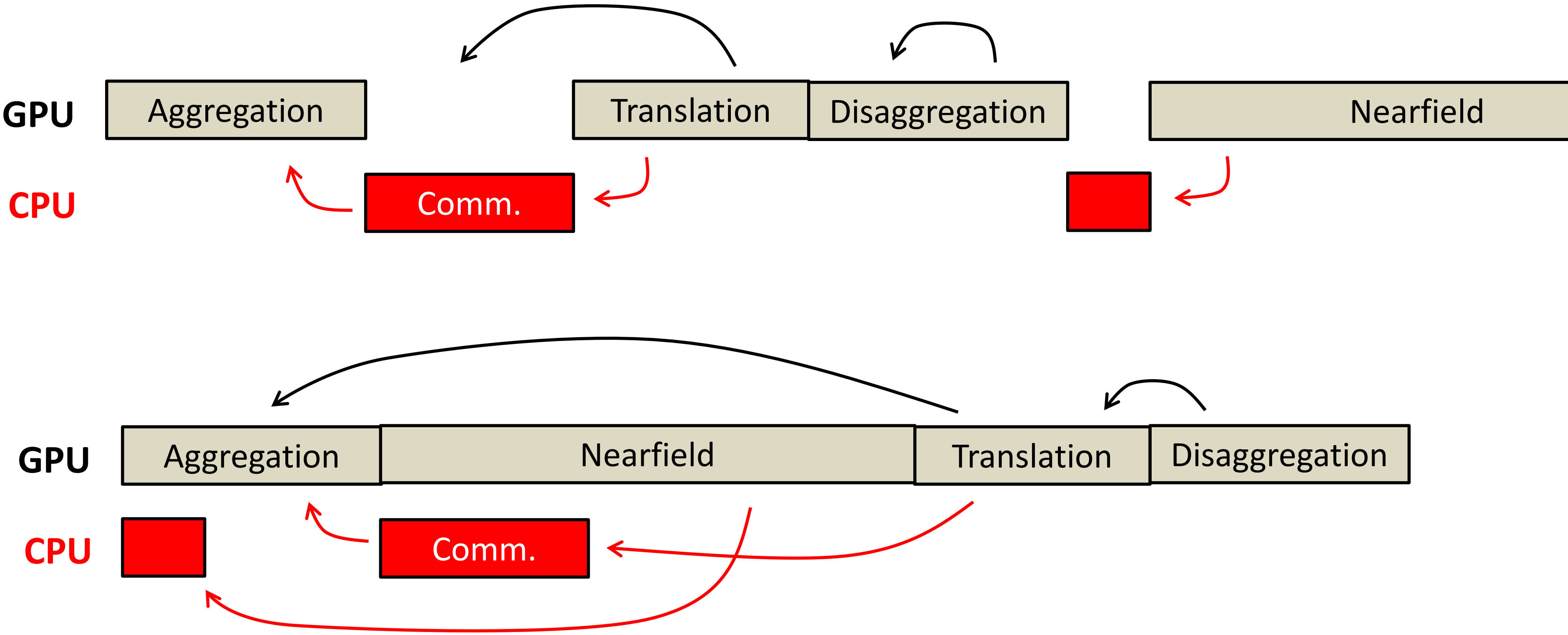
IBM S822LC "Minsky" system

	Blue Waters XK	Blue Waters XE	S822LC "Minsky"
CPU 0	Opteron 6276	Opteron 6276	Power8
CPU 1	--	Opteron 6276	Power8
GPU 0	NVIDIA K20X (Kepler, 6GB)	--	NVIDIA P100 (Pascal, 16 GB)
GPU 1	--	--	NVIDIA P100 (Pascal, 16 GB)
GPU 2	--	--	NVIDIA P100 (Pascal, 16 GB)
GPU 3	--	--	NVIDIA P100 (Pascal, 16 GB)
RAM	32 GB	64 GB	512 GB

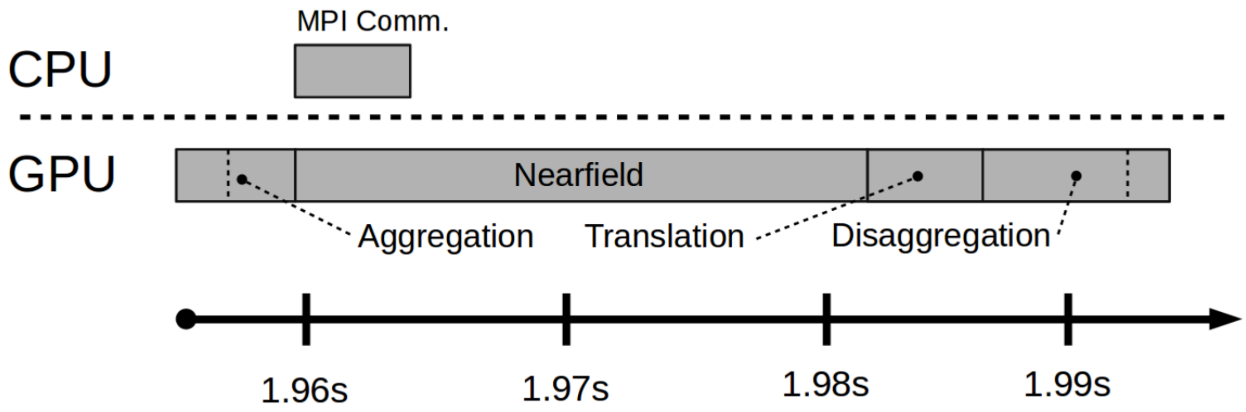
MPI Rank Arrangement



Eliminating Communication Overhead



Multi-Rank MLFMM GPU Scaling

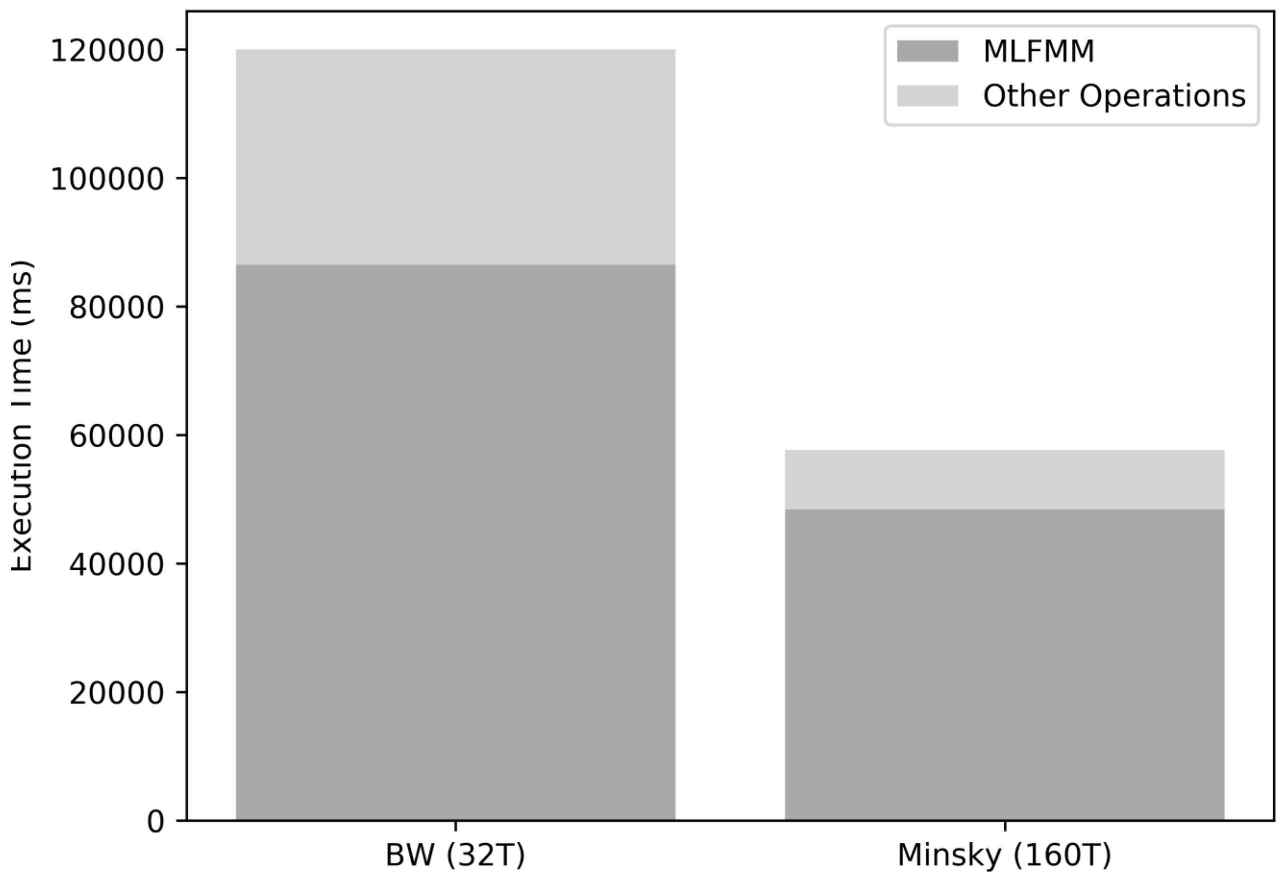


Execution Setup	MLFMM Time (ms)
Blue Waters (1 MPI Rank)	619
Blue Waters (4 MPI Rank)	156 (3.96x)
Blue Waters (16 MPI Rank)	40 (15.58x)
Minsky (1 MPI Rank)	118.92
Minsky (4 MPI Rank)	30.54 (3.89x)

Scalability limits not reached at 16 nodes for us…

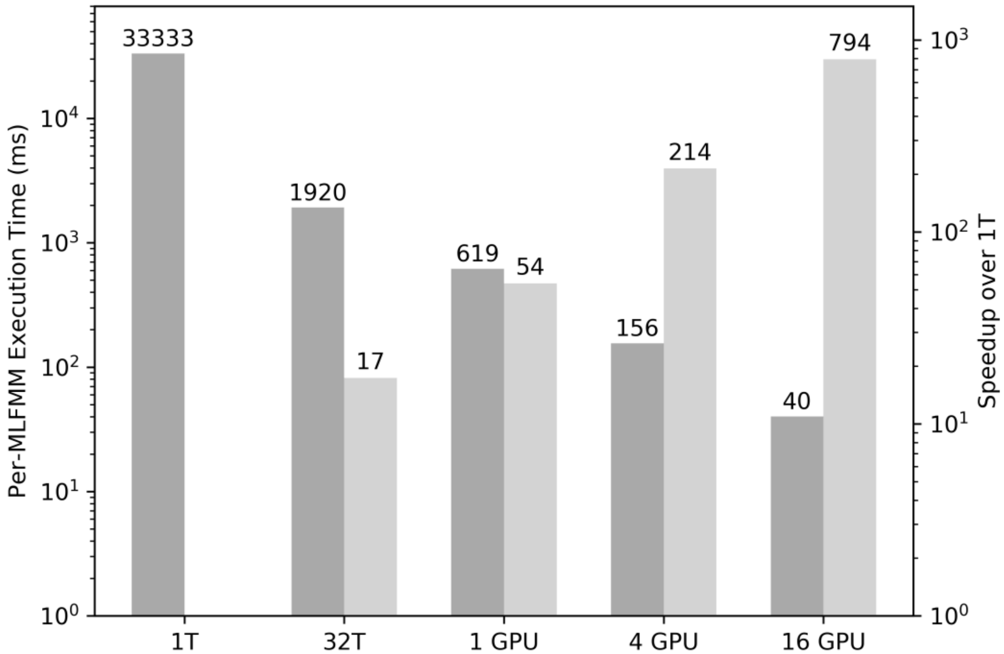
…but in the future, fat nodes can reduce communication costs and improve scalability

Application Time in MLFMM

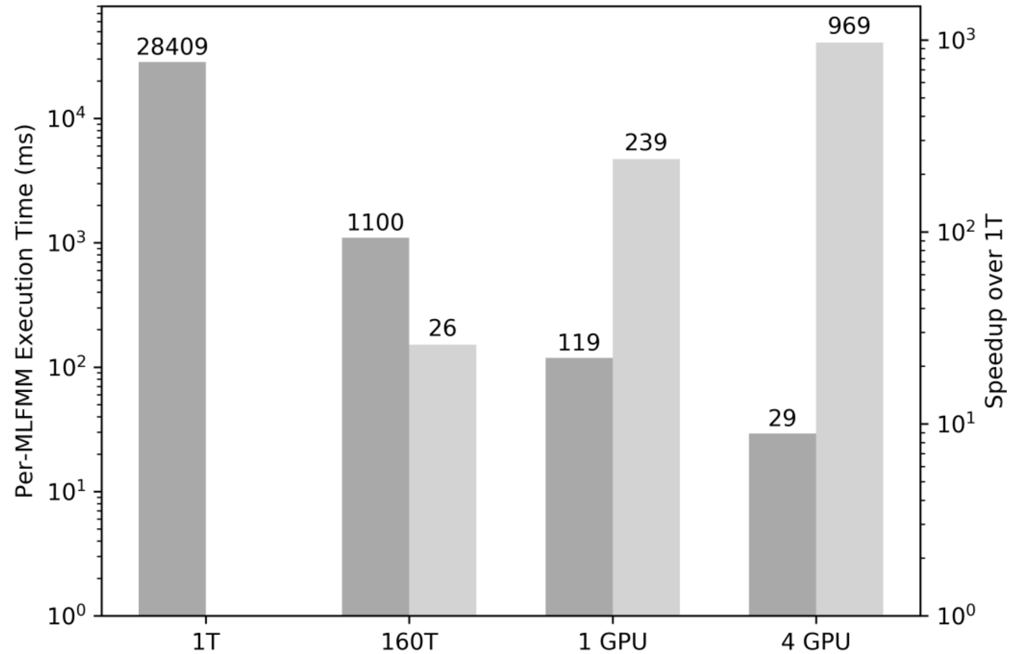


MLFMM is the majority of the execution time

MLFMM CPU/GPU Scaling



Blue Waters Scaling



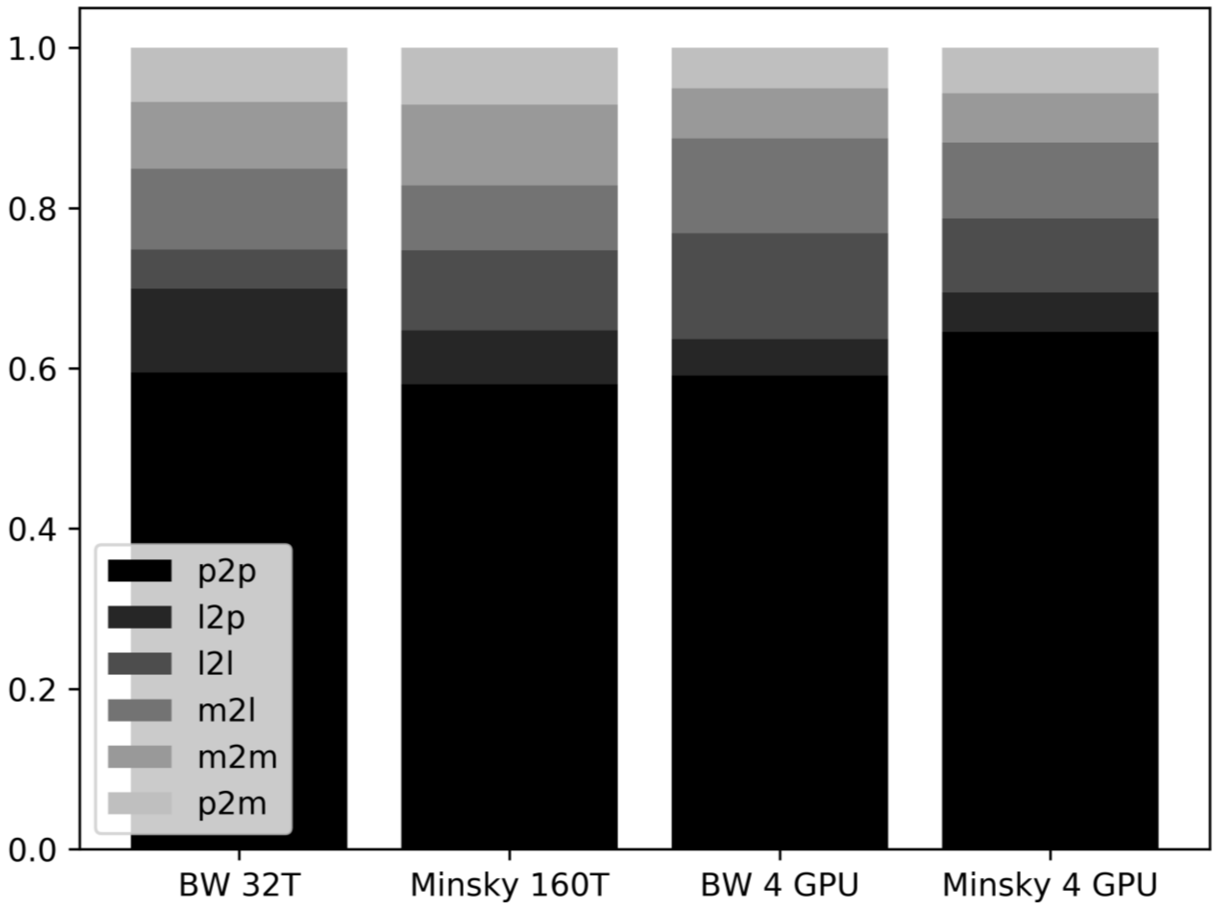
Minsky Scaling

1T → 32T: 17x (16 fp units)
32T → 1 GPU: 3.1x

1T → 160T: 25x (20 fp units)
160T → 1 GPU: 9.5x

Minsky CPU Speedup over BW: 1.8x
Minsky 4-GPU Speedup over BW: 5.3x
Minsky Node Speedup over BW: 21x

MLFMM Kernel Breakdown



P2P is most of the execution time

L2L has largest speedup from K20x to P100

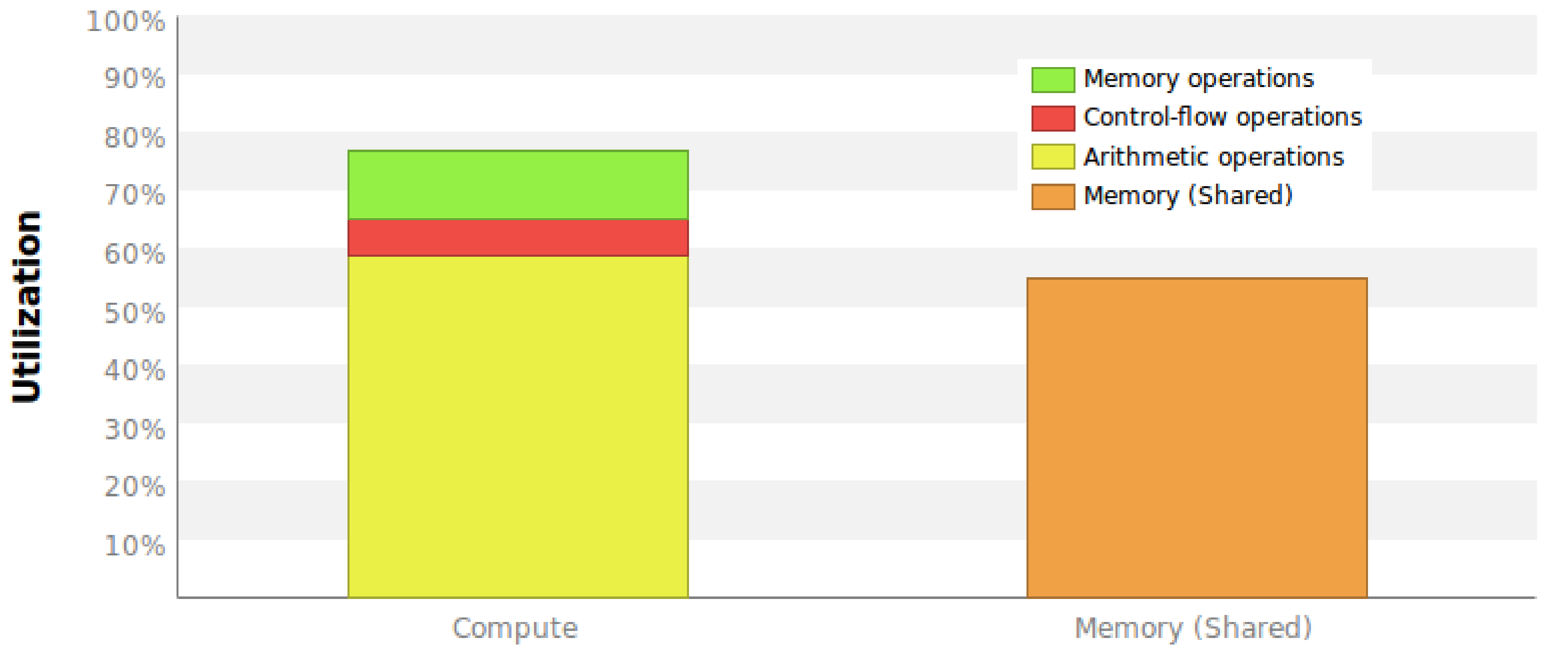
Kepler vs. Pascal

Feature	K20x Kepler(GK110)	P100 Pascal (GP100)
Core Clock	732 MHz	1328 MHz
Global Memory Bandwidth	250 GB/s	720 GB/s
Peak GFLOPs(single / double)	3935 /1312	9519 /4760
L2	1.5 MB	4 MB
# of SMs	14	56
Register File	256 KB	256 KB
L1	48 / 32 / 16 KB	0 KB
Shared Memory	16 / 32 / 48 KB	64 KB
"CUDACores"	192	64
Max Resident Blocks	16	32

Pascal: More registers and shared-memory per thread, more warps per SM.

Lessons from Nearfield Kernel

Longest-running kernel (60% of MLFMM time)



87% threads inactive in inner loop

mod/div on 2^x: immediate 1.3x speedup

Important to give the compiler information and understand profiling results

Lessons from Disaggregation Kernel

	BW	Minsky	Speedup
L2L Time(ms)	78.5	9.9	8.0
MLFMM Time (ms)	633	118.9	5.3

L2L Kernel	BW	Minsky
Theoretical Occupancy	43.8	56.2
Achieved Occupancy	30.7	42.1

Occupancy limited by shared memory in both cases
Relative performance improves due to increased shared memory size

Conclusion

- Low-effort port of GPU MLFMM to fat nodes yields good speedup
- Fat nodes will improve scalability for massively parallel MFLMM
- GPU architectures seem to be moving in a beneficial direction

Thank you

pearson@illinois.edu

MLFMM GPU Performance Data

Kernel	BW 32T (ms)	BW K20x (ms)	BW Speedup (GPU /32T)	Minsky 160T (ms)	Minsky P100 (ms)	Minsky Speedup (GPU / 160T)	Speedup (P100 /K20x)
P2M	127.1	30.9	4.1	72.1	6.4	11.3	4.8
M2M	156.3	37.3	4.2	102.6	6.6	15.6	5.7
M2L	189.6	72.3	2.6	82.7	10.2	8.1	7.1
L2L	91.6	78.5	1.2	101.6	9.9	10.3	8.0
L2P	196.2	28.0	7.0	68.4	5.5	12.4	5.0
P2P	1117.4	361.9	3.1	590.5	72.5	8.1	5.0
Iteration Time	1962.1	633.0	3.1	1074.8	118.9	9.0	5.3

MLFMM CPU Performance Data

Step	Blue Waters XE 32T(ms)	Minsky 160T (ms)	Speedup BW 32T / 1T	Speedup Minsky 160T / 1T	Speedup Minsky / BW
P2M	127.1	72.1	17.9	25.2	1.8
M2M	156.3	102.6	20.2	24.5	1.5
M2L	189.6	82.7	12.5	19.2	2.3
L2L	91.6	101.6	34.6	28.4	0.9
L2P	196.2	68.4	11.4	26.6	2.9
P2P	1117.4	590.5	18.5	29.0	1.9
MLFMM	1962.1	1074.8	17.3	25.8	1.8