A Distributed-Memory Fast Multipole Method for Volume Potentials

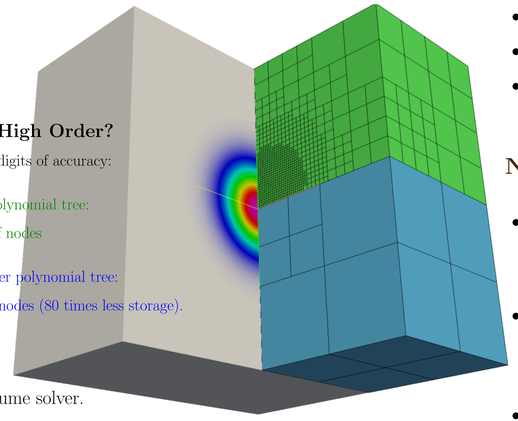
Dhairya Malhotra, George Biros (Advisor)
The University of Texas at Austin

Introduction

The solution of a constant-coefficient elliptic PDE, of the form $\mathcal{L}u = f$, can be computed using an integral transform: convolution of the source density f with the fundamental solution of the PDE. We present a Fast Multipole Method for computing such solutions on an adaptive Chebyshev octree.

$$u(x) = \int_{\mathcal{R}} K(x,y)f(y)dy = \int_{\mathcal{B}(x)} K(x,y)f(y)dy + \int_{\mathcal{R}\backslash\mathcal{B}(x)} K(x,y)f(y)dy$$

Near interactions ($\mathcal{B}(x)$) are evaluated using precomputed quadratures and far ($\mathcal{R}\backslash\mathcal{B}(x)$) interactions using Kernel Independent FMM [1].



Main Features:

- High Order
- Adaptive
- Parallel, Scalable
- Kernel Independent

Why High Order?

For five digits of accuracy:

Cubic polynomial tree:

130k leaf nodes

10th-order polynomial tree:

120 leaf nodes (80 times less storage).

Contributions

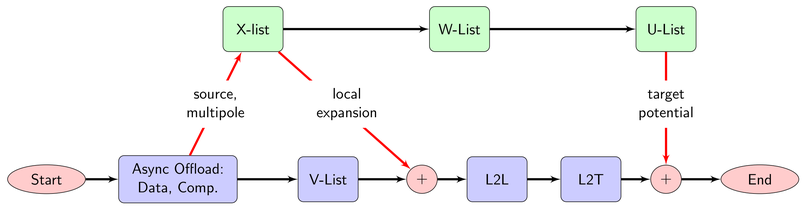
- A distributed-memory KIFMM based volume solver.
- Support for accelerators (Intel Xeon Phi for now).
- Cache optimized schemes for near-field and far-field interactions in KIFMM.

Intra-Node Parallelism

Accelerators and Asynchronous Execution:

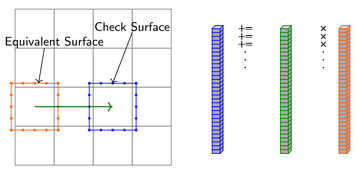
Overlap computation on CPU and Intel Phi during Downward Pass of FMM.

- **CPU:** multipole-to-local (V-list) and down-to-down and down-to-target interactions.
- **Phi:** source-to-local, multipole-to-target and source-to-target (X,W,U-list) interactions.

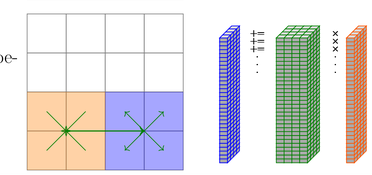


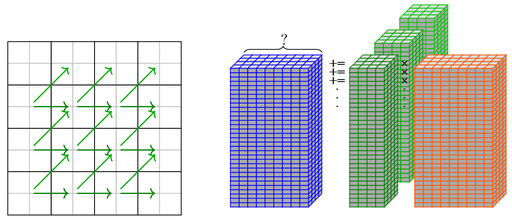
Multipole-to-Local Translation:

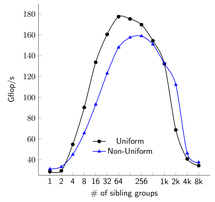
- Check-potential evaluated through convolution.
- In Fourier domain is a Hadamard product.
- Memory-bound computation.



- Interleave data of sibling octants, interaction between sibling groups: 8×8 mat-vec products.
- Vectorization: AVX, SSE intrinsics.



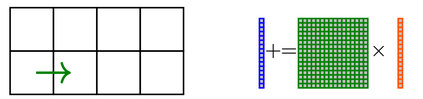




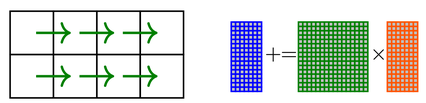
- Multiple interactions between sibling groups (in same direction) can be done with DGEMMs.
- Keep source and target vectors in cache and reuse for interactions in the next direction.
- Optimal matrix width determined experimentaly (results for Intel Xeon E5-2680).

Near Interactions:

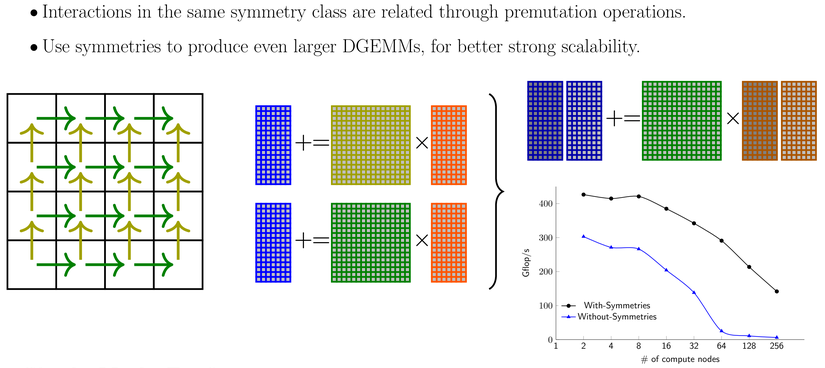
- Near interactions evaluated using precomputed quadratures using DGEMVs.



- Combine several interactions in the same direction to convert DGEMVs to DGEMMs.



- Interactions in the same symmetry class are related through premutation operations.
- Use symmetries to produce even larger DGEMMs, for better strong scalability.



Single Node Performance:

Timing (Gflop/s) results for Laplace kernel, with 14th order Chebyshev tree, 9th order multipole expansion (about 8-digits of accuracy).

One node of **Stampede**:

- $2 \times$ eight core Xeon E5-2680
- $1 \times$ Xeon Phi SE10P Coprocessor
- Peak Performance: 1.34 Tflop/s

	N_{oct}	U,W,X-List	V-List	All	Speedup
Old	904	0.315 (85.0)	0.214 (14.3)	0.570 (55.9)	1.0
CPU	904	0.123 (216.1)	0.037 (101.4)	0.169 (185.9)	3.4
CPU+Phi	904	0.076 (346.9)	0.037 (101.9)	0.128 (245.8)	4.5
Async	904	0.000 (–NA-)	0.037 (101.1)	0.078 (404.0)	7.3
Old	62k	6.919 (234.6)	18.494 (15.7)	25.771 (76.8)	1.0
CPU	62k	5.993 (270.3)	2.784 (122.9)	9.111 (222.9)	2.8
CPU+Phi	62k	3.264 (496.4)	2.875 (119.1)	6.723 (302.0)	3.8
Async	62k	0.000 (–NA-)	2.914 (117.5)	3.409 (595.7)	7.6

Distributed Memory Parallelism

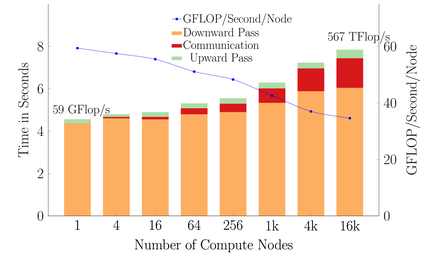
The distributed memory implementation of FMM is discussed in detail in Lashuk et al. [2].

Construct 2:1 balanced linear octree: Local 2:1 balance refinement followed by globally sorting the leaf octants (by Morton Id) and then remove duplicates locally.

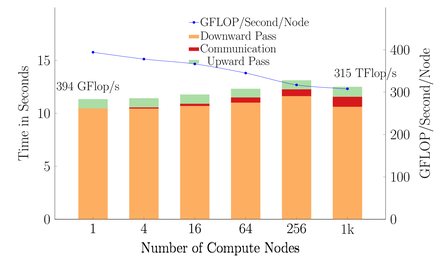
FMM Evaluation: Upward pass followed by reduction of multipole expansions for shared octants. Construct a local essential tree (LET) by broadcasting ghost octants using hypercube communication scheme followed by downward pass.

Weak Scalability Results:

- Weak scalability up to 16k compute nodes on ORNL's Titan supercomputer for Laplace kernel, with 8k octants per node.
- 58% parallel efficiency and 74 billion unknowns in 7.8 seconds at 567 TFlop/s for a highly non-uniform octree with 26-levels.

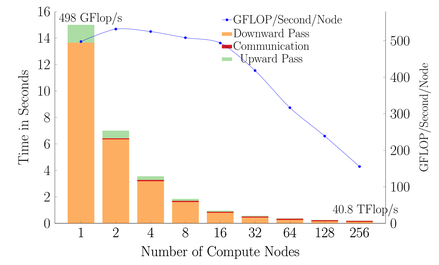


- Weak scalability up to 1k compute nodes on TACC's Stampede supercomputer, for Helmholtz kernel, with 31k octants per node.
- 78% parallel efficiency and 33 billion unknowns in 12.5s at 315 TFlop/s.

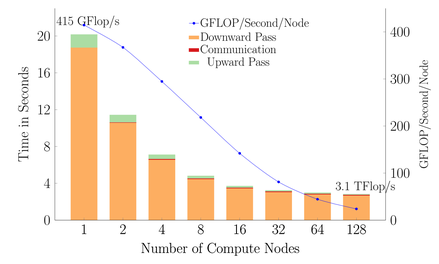


Strong Scalability Results:

- Strong scaling on Stampede for Laplace kernel with 124 million unknowns.
- 30% parallel efficiency on 256 nodes.



- Strong scaling on Stampede for Helmholtz kernel with 70 million unknowns.
- 5.7% parallel efficiency on 128 nodes.
- Low efficiency for non-homogeneous kernels, due to level-by-level evaluation and overhead of transferring large amounts of precomputed data between host memory and the Xeon Phi accelerator.



References

[1] H. Langston, L. Greengard and D. Zorin: A Free-Space Adaptive FMM-Based PDE Solver in Three Dimensions. Communications in Applied Mathematics and Computational Science, vol. 6, no. 1, 2011, pp. 79–122

[2] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. S. Sampath, A. Shringarpure, R. W. Vuduc, L. Ying, D. Zorin, and G. Biros: A Massively Parallel Adaptive Fast-Multipole Method on Heterogeneous Architectures. Proceedings of SC09 ACM/ICEE SCXY Conference Series, pp. 1-11, 2009