Scalable Fast Multipole Methods on Distributed Heterogeneous Architecture

Qi Hu, Nail A. Gumerov, Ramani Duraiswami

Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, MD

• Fast Multipole Methods

1.Problem: compute matrix-vector product of some kernels

$$\phi(\mathbf{y}_j)=\sum_{i=1}^{N} q_i \Phi(\mathbf{y}_j-\mathbf{x}_i), \quad j=1,\ldots,M$$

1. Linear computation and memory cost with any accuracy

2. Divide the sum to the far field and near field terms as

$$\phi(\mathbf{y}_j)=\sum_{\mathbf{x}_i \notin \Omega(\mathbf{y}_j)} q_i \Phi(\mathbf{y}_j-\mathbf{x}_i)+\sum_{\mathbf{x}_i \in \Omega(\mathbf{y}_j)} q_i \Phi(\mathbf{y}_j-\mathbf{x}_i)$$

3. Direct kernel evaluations for the near field

4. Approximations of the far field sum via the *multipole expansions* of the kernel function and spatial data structures (octree for 3D, hierarchical data structures)

5. The local and multipole expansions of the Laplace kernel at the center $\mathbf{X}_*$ with the truncation number p

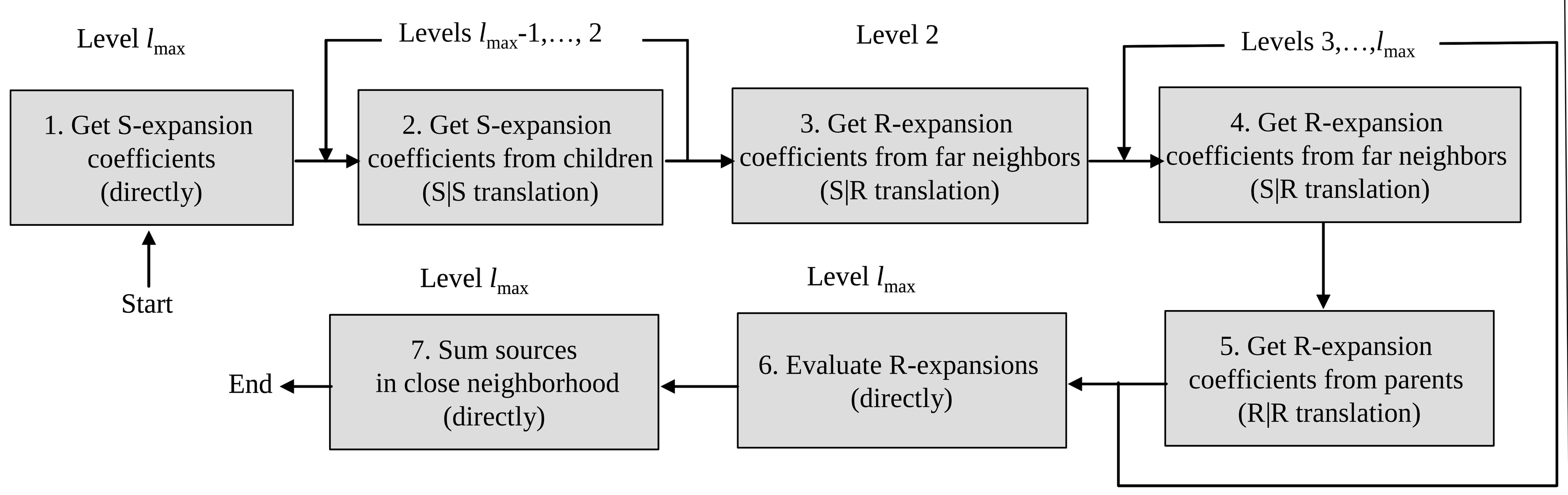
$$\phi(\mathbf{y},\mathbf{x})=\frac{1}{|\mathbf{y}-\mathbf{x}|}=\sum_{n=0}^{p-1}\sum_{m=-n}^{n} D_n^m r^n Y_n^m(\mathbf{y}-\mathbf{x}_*)+\varepsilon$$

$$\phi(\mathbf{y},\mathbf{x})=\frac{1}{|\mathbf{y}-\mathbf{x}|}=\sum_{n=0}^{p-1}\sum_{m=-n}^{n} C_n^m \frac{1}{r^{n+1}} Y_n^m(\mathbf{y}-\mathbf{x}_*)+\varepsilon$$

6. Expansions regions are validated by the *well separated pairs* realized by using spatial boxes of octree

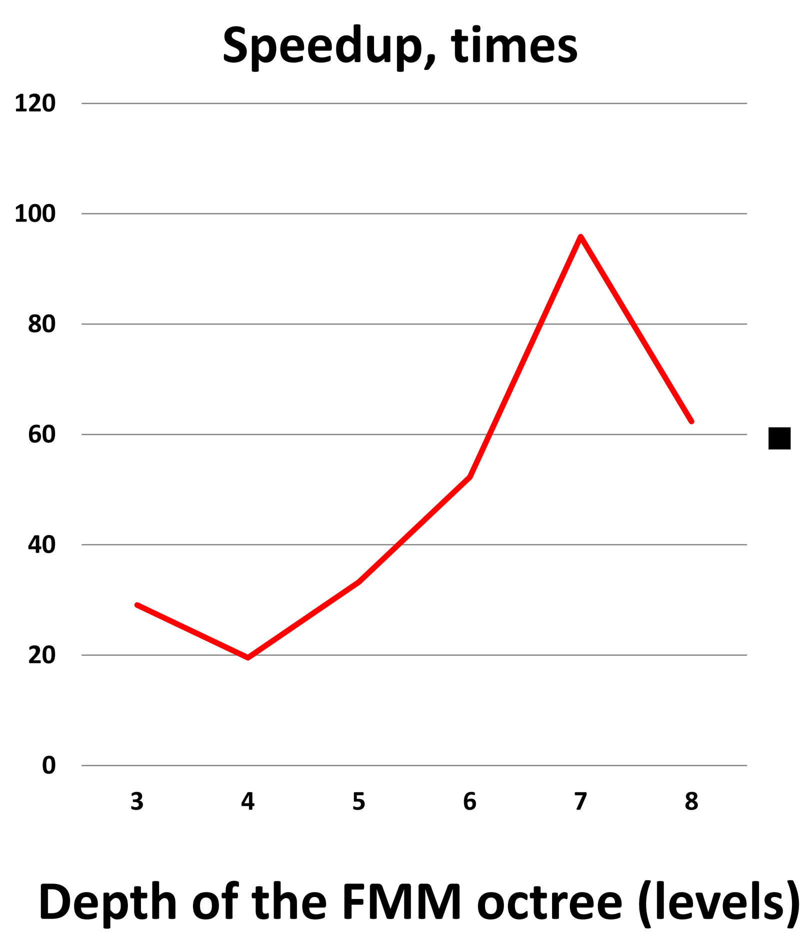
7. Translations of expansion coefficients

8. FMM algorithm flow chart

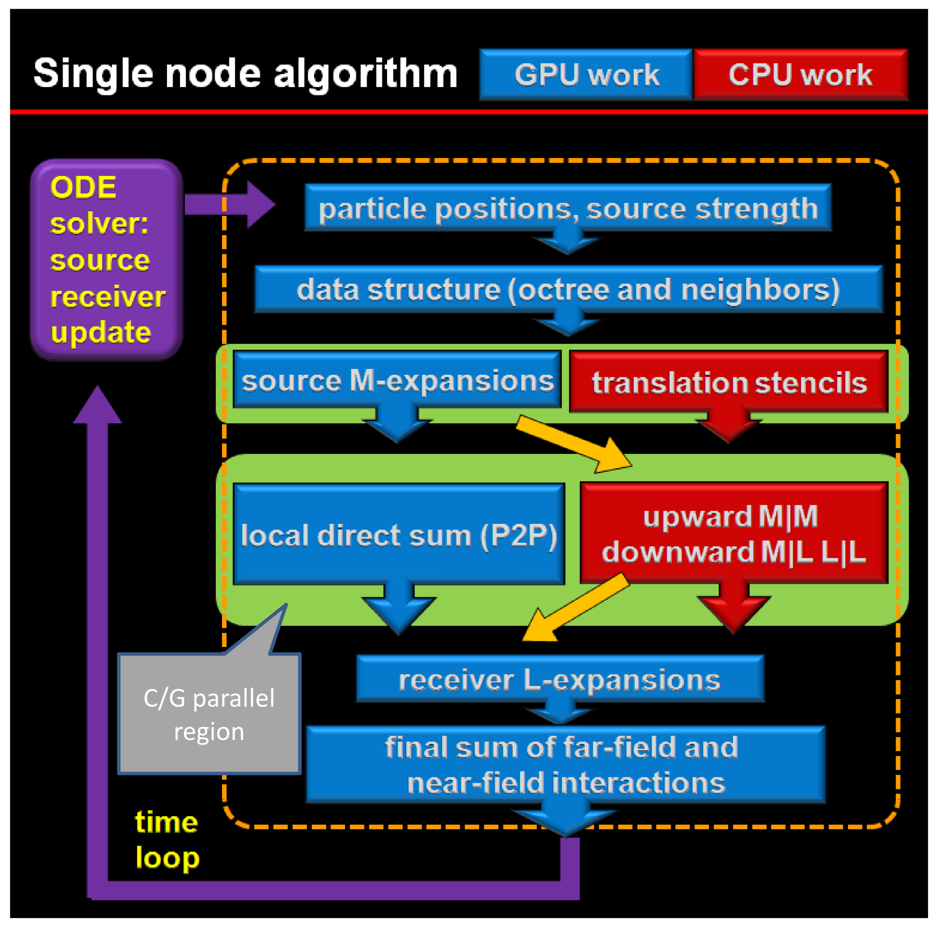


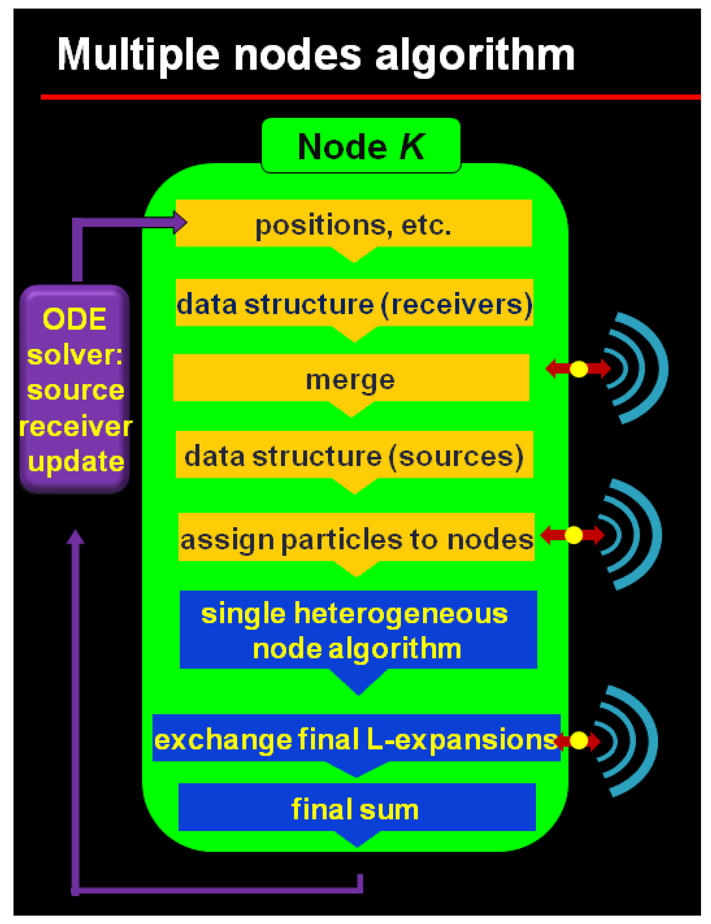
• Novel Parallel Algorithm for FMM Data Structures

1. Data structures for assigning points to boxes, find neighbor lists, retaining only non empty boxes
2. Usual procedures use a sort, and have O(N log N) computation cost
3. Present: parallelizable on the GPU and has O(N) computation cost



- FMM data structures on the GPU for millions of particles in 0.1 s as opposed to 2-10 s required for CPU
- Substantial computational savings for dynamic problems, where particle positions change and data structures need to be regenerated at each time step

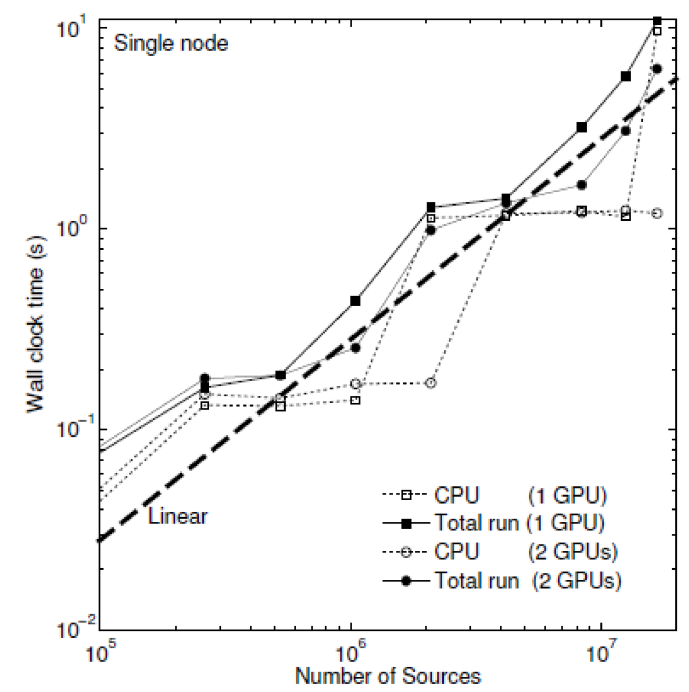


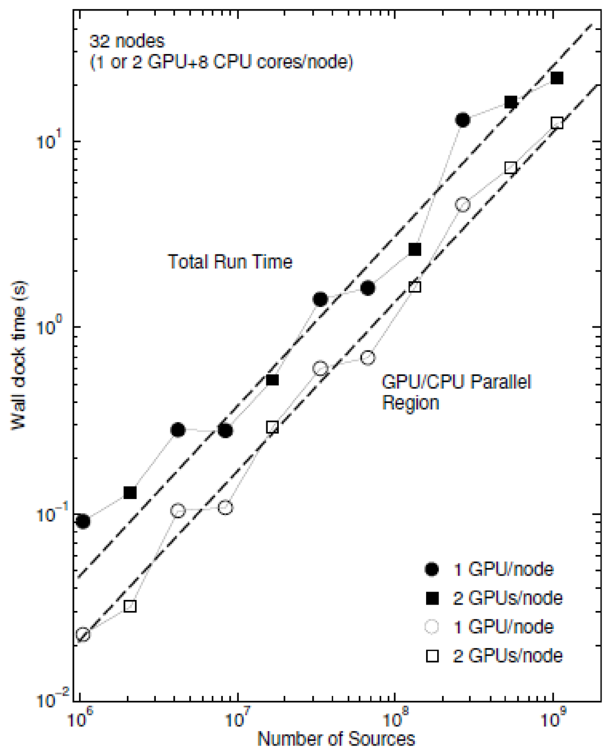


1. Master collects receiver boxes and distributes work regions (to achieve work balance)
2. Assigns particle data according to assigned work regions
3. M|M for local non-empty receiver boxes while M|L and L|L for global non-empty receiver boxes
4. L-coefficients efficiently sent to master node in binary tree order

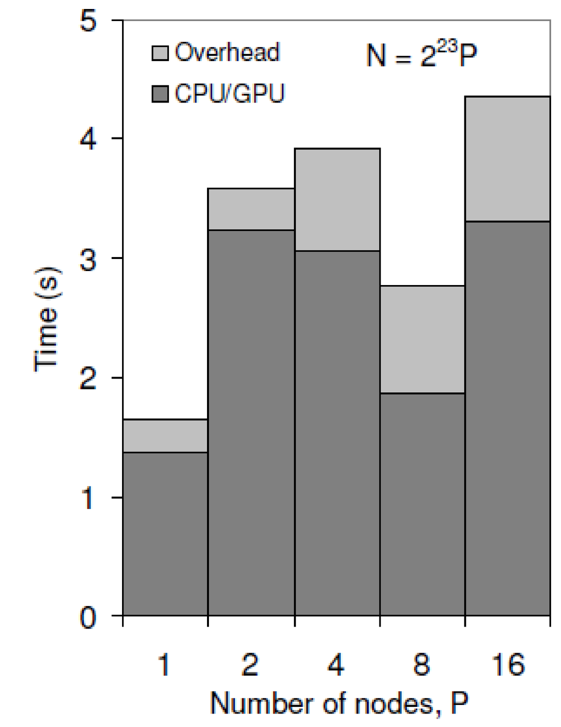
• Performance Tests

○ Single node (Dual quad-core Intel Nehalem 5560 2.8 GHz processors; 24 GB of RAM; Two Tesla C1060 GPU) and multiple node (UMIACS Chimera cluster with 32 nodes) algorithm performance



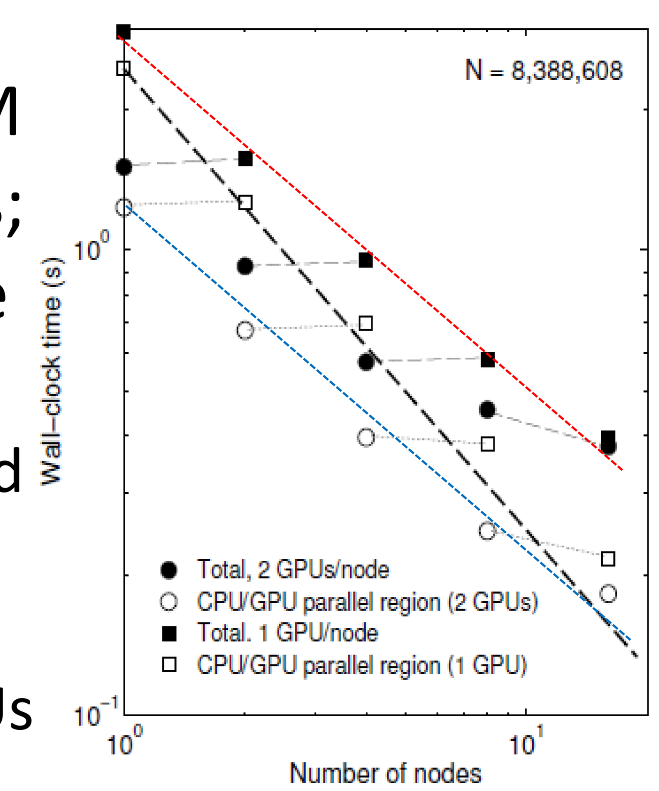


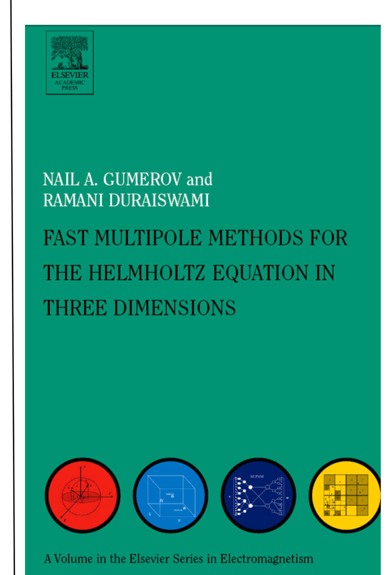
○ Weak scalability test: Fix 8M particles per node; Run tests on 1 ~ 16 nodes; The depth of the octree determines the overhead; The particle density determines the parallel region timing



○ Strong scalability test: Fix the problem size to be 8M Run tests on 1 ~ 16 nodes; Direct sum dominates the computation cost

1. Unless GPU is fully occupied algorithm does not achieve strong scalability
2. Can choose number of GPUs on a node according to size





References:

- L. Greengard & V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys., 73, 1987, 325-348.
- N.A. Gumerov & R. Duraiswami, *Fast multipole methods for the Helmholtz equation in three dimensions*, Elsevier, 2005.
- N.A. Gumerov & R. Duraiswami, Fast multipole methods on graphics processors, J. Comput. Phys.,227, 2008, 8290-8313
- Q. Hu, N.A. Gumerov & R. Duraiswami, Scalable fast multipole methods on distributed heterogeneous architectures, SC'11 Proceedings of the 2011 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis