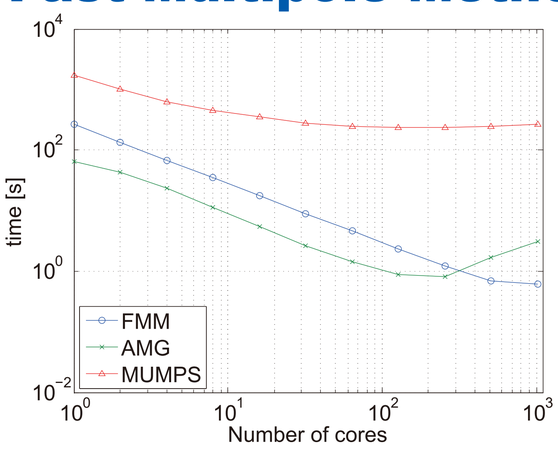
Hierarhical Algorithms and Mesh-Based CFD for High Performance Computing

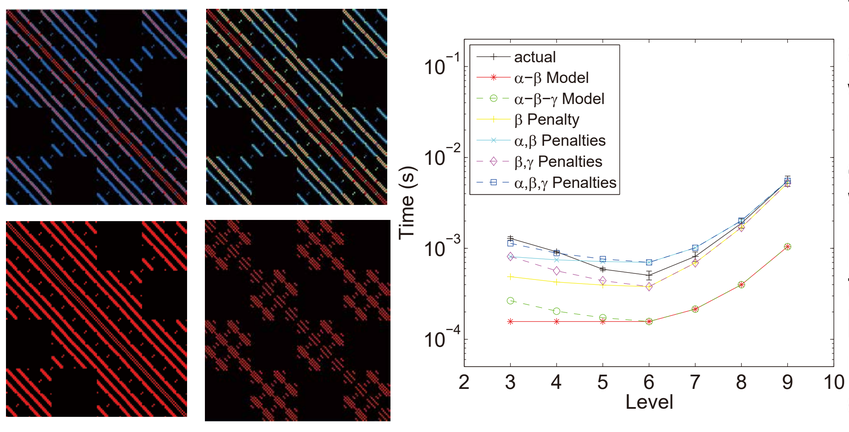
Scalable Hierarchical Algorithms for Scientific Computing

Fast Multipole Method Preconditioner



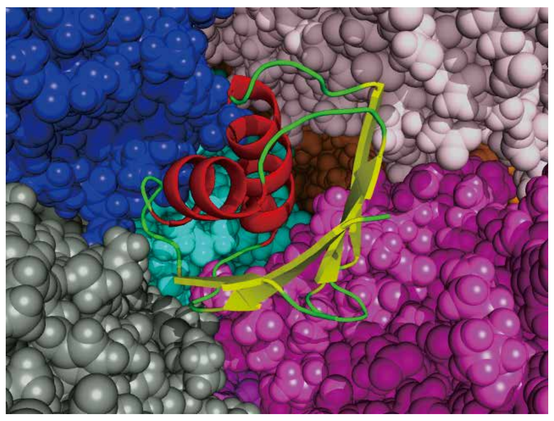
The fast multipole method (FMM) was originally developed as an O(N) algorithm for solving N-body problems. However, it can be used as a direct solver or preconditioner for solving linear systems that arise from elliptic partial differntial equations that have a Green's function solution such as Poisson's equation. The FMM by itself can only handle far field boundary conditions, but it can be combined with boundary element methods to handle Dirichlet and Neumann boundary conditions. Comparsions against the algebraic multigrid code BoomerAMG and sparse direct solver MUMPS, shows that the FMM preconditioner becomes competitive as the degree of parallelization increases.

Performance Model for FMM Communication



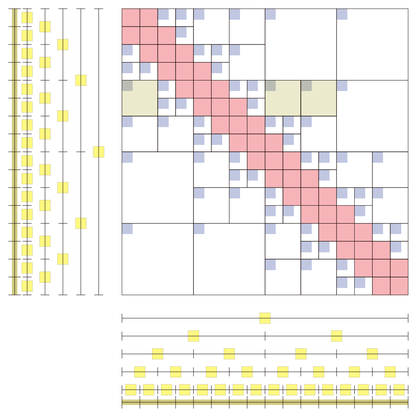
The FMM has a complex communication pattern, which results from its hierarchical and global data dependency.
We have developed a performance model for the communication in FMM, which accurately predicts the communication time for each level in the tree structure. The model considers latency, bandwidth, hops, and multicore penalties in the network. The left 4 figures show the colormap of the FMM communication for different levels of the tree structure. The right figure shows the communication time at each level along with the model prediction on Shaheen2 (KAUST).

FMM for Molecular Dynamics



The scalability of FMM makes it an interesting alternative to long range force calculation methods in classical molecular dynamics such as PME. For problems that have a high contrast in the distribution of atoms, such as phase transition calculations, the FMM has an advantage over FFT-based methods. The challenge for FMM is to incorporate techniques to smooth the transition between the short range direct calcualtion and the long range approximation, which will allow the FMM to conserve energy over long time integrations.

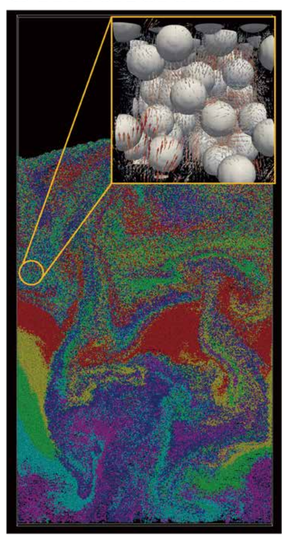
Algebraic FMM



The matrix representation of the FMM algorithm yields algebraic variants of FMM such as H-matrices and hierarchical semi-seperable (HSS) matrices. These methods can approximate matrix-matrix multiplications and LU decompositions in near-linear complexity. These algebraic varients of FMM lie on the opposite end of the Byte/Flop spectrum from the analytical FMM because they store more to compute less. When the cost of data movement increases faster than arithmetic opera- tions on future architectures, it is important to consider the whole spectrum of hierarchical low-rank approximation methods.

Large-scale Mesh-based CFD Simulations

Large-scale Fluid-structure Interaction Simulation

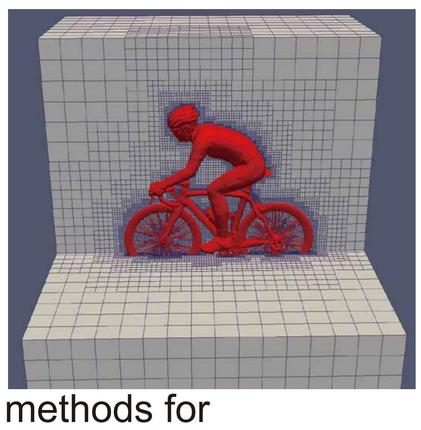


Direct numerical simulations of interaction between fluid and solid particles require huge amount of computational resources. We have developed a large scale simulation of a fluid-particle system using coupled a lattice Boltzmann method (LBM) with a discrete element method (DEM) by means of the GPU performance. We demonstrate a direct numerical simulation of fluidized bed with 562,500 DEM particles using 128 GPUs. The left figure shows a result of a simulation for falling leaves using 2.1 billion mesh system and 128 GPUs.

Tsunami Simulation

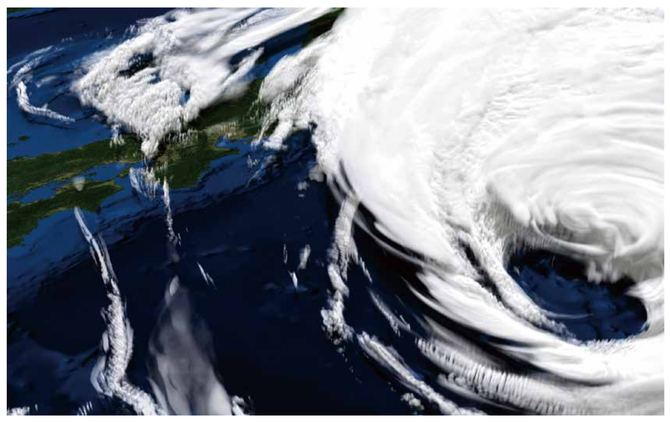
Tsunami impact pressure analysis for coast area buildings is conducted with a goal of high accurate safety estimation. Volume of Fluid based finite difference method is applied to calculate the incompressible two phase flow. AMG-BiCGSTAB is applied to solve the pressure Poisson equation with is the most time consuming part of the simulation. The picture shows building structures are hit by a 10 m height tsunami. 18 GPUs on TSUBAME2.5 are used to calculate 768x768x128 grids simulation.

Aerodynamics Simulation using Locally Mesh-refined Lattice Boltzmann Method



A lattice Boltzmann method is one of suitable methods for aerodynamics simulation especially for GPU computing. The area near a complex body is locally refined to assign high resolution to calculate the turbulent flow in the way of large eddy simulation. An octree-based mesh refinement has been introduced and we optimize the implementation by using the C++ template for code-expansion. Turbulent flows with Reynolds number of more than 100,000 have been studied on a GPU computation.

Framework-based Weather Prediction code



Numerical weather prediction is one of the major applications in high-performance computing and is accelerated on GPU supercomputers. Obtaining high-performance using thousands of GPUs often needs skillful programming. We have implemened the weather prediction code ASUCA on a multi-GPU platform by using a C++ high-productivity framework. Our framework automatically translates user-written stencil functions that update grid points and generates both GPU and CPU codes. User-written codes are parallelized by MPI. These codes can easily utilize optimizations such as overlapping technique to hide communication overhead by computation.

Reference: T. Shimokawabe, T. Aoki and N. Onodera **"High-productivity Framework on GPU-rich Supercomputers for Operational Weather Prediction Code ASUCA,"** *in Proceedings of the 2014 ACM/IEEE conference on Supercomputing (SC'14) ,* New Orleans, LA, USA, Nov 2014.