GPU accelerated heterogeneous computing for Particle/FMM Approaches and for Acoustic Imaging

Ramani Duraiswami
University of Maryland, College Park

http://www.umiacs.umd.edu/~ramani

With Nail A. Gumerov, Qi Hu, Dmitry Zotkin, Adam O’Donovan

Work supported by NSF, AFOSR, ONR, DARPA and UMIACS
Fast Multipole Methods – Particle Methods

• Fast summation of radial functions
  – Green’s functions $\Phi(x,y)$ for classical operators $L$
  – Solution $y$ to forcing at points $x$

• Applications
  – Electrostatics, Molecular/Stellar dynamics, Vortex
  – Boundary Integral Methods
  – Particle discretization of field equations
  – Machine Learning

$$s(x_j) = \sum_{i=1}^{N} \alpha_i \phi(x_j - x_i), \quad \{s_j\} = [\Phi_{ji}]\{\alpha_i\}.$$
What is the FMM?

- Decompose singular sum into
  - “local” or sparse part +
  - “far-field” and dense part

**N-body problem**

- Source points: \( \{x_i\}, i = 1, \ldots, N \)
- Receiver points: \( \{y_j\}, j = 1, \ldots, M \)
- The matrix-vector product:
  \[
  \phi(y_j) = \sum_{i=1}^{N} q_i \Phi(y_j - x_i), \quad j = 1, 2, \ldots, M
  \]
  \[x_i, y_j \in \mathbb{R}^d.\]
- Complexity of direct method: \( O(NM) \)

\[
\phi(y_j) = \sum_{x_i \notin \Omega(y_j)} q_i \Phi(y_j - x_i) + \sum_{x_i \in \Omega(y_j)} q_i \Phi(y_j - x_i)
\]
A few pretty pictures from papers
FMM for stellar/molecular dynamics

\[ G(r) = \frac{1}{r}, \quad r = |r| = \sqrt{x^2 + y^2 + z^2}, \]

then

\[ G(r - r_0) = \frac{1}{|r - r_0|} = \frac{1}{\sqrt{(r - r_0) \cdot (r - r_0)}} = \frac{1}{\sqrt{r^2 - 2r \cdot r_0 + r_0^2}} \]

\[ = \frac{1}{\sqrt{r^2 - 2rr_0 \cos \theta + r_0^2}} = \frac{1}{\sqrt{r^2 - 2\mu rr_0 + r_0^2}} \]

\[ = \begin{cases} 
  r_0^{-1} \sum_{n=0}^{\infty} P_n(\mu) (r/r_0)^n, & r < r_0, \\
  r^{-1} \sum_{n=0}^{\infty} P_n(\mu) (r_0/r)^n = r_0^{-1} \sum_{n=0}^{\infty} P_n(\mu) (r/r_0)^{-n-1}, & r > r_0. 
\end{cases} \]

At \( r = r_0 \) the series also converges, if \( \cos \theta \neq 1 \) \((r \neq r_0)\).
Addition Theorem for Spherical Harmonics

Spherical Harmonics

order

degree

\[ P_n(\cos \theta) = \frac{4\pi}{2n+1} \sum_{m=-n}^{n} Y_n^{-m}(\theta', \phi') Y_n^{m}(\hat{\theta}, \hat{\phi}), \]

\[ Y_n^{m}(\theta, \phi) = (-1)^m \sqrt{\frac{2n+1}{4\pi}} \frac{(n-|m|)!}{(n+|m|)!} P_{n}^{m}(\mu)e^{im\phi}, \quad \mu = \cos \theta. \]

where \( \theta \) is the angle between two points on a sphere with spherical angles \((\theta', \phi')\) and \((\hat{\theta}, \hat{\phi})\).

Vector form of the addition theorem

\[ P_n(s_1 \cdot s_2) = \frac{4\pi}{2n+1} \sum_{|m|=-n}^{n} Y_n^{-m}(s_1) Y_n^{m}(s_2) = \frac{4\pi}{2n+1} \sum_{|m|=-n}^{n} Y_n^{m}(s_1) Y_n^{-m}(s_2). \]
Data structures for the FMM

- **Separation:** Cells satisfy local separation and WSPD properties across all levels.

- **Indexing:** Cell indices satisfy some order relation in memory.

- **Spatial addressing:** Cell centers are computable from addresses and each particle can find its bounding cell.

- **Hierarchical addressing:** Cell children, parent, and neighbor indices are computable from addresses.

- All this must be done in $O(N \log N)$ and must be parallelizable


Moved to implement FMM on GPU almost with the introduction of CUDA (2007)

FMM translation based on rotation translation, and showing that it is competitive or superior to asymptotically faster diagonal translations

Real valued formulations

Algorithm splitting

Two vortex rings interaction demo

- Two vortex rings move at the same direction
- Two vortex rings collision
- Visualization as computation proceeds!
Heterogeneous architectures

• GPUs and MIC live on a bus which sits inside a CPU box
• Insight --- why do gymnastics to fit tree algorithms with irregular access on the GPU
• CPU is multicore
• Let the GPU do stuff it is good at and let CPU do what it is good at
Motivation -- Helicopter Brownout

- Complicated phenomena involving interaction between rotorcraft wake, ground, and dust particles
- Causes accidents to poor visibility and damage to helicopters
- Understanding can lead to mitigation strategies
Single node algorithm

- ODE solver: source receiver update

- GPU work:
  - particle positions, source strength
  - data structure (octree and neighbors)

- CPU work:
  - source S-expansions
  - translation stencils
  - local direct sum
  - upward S\mid S
  - downward S\mid R R\mid R

- receiver R-expansions

- final sum of far-field and near-field interactions

- time
- loop

Diagram:
- Connections between nodes: source positions, data structure, expansions, direct sums, translations, and final sum interactions.
The algorithm flow chart

Node A

- positions, source strength
- data structure (octree)
- merge octree
- data structure (neighbors)
- redistributed particle
- single heterogeneous node algorithm
- exchange final R-expansions
- final sum

Node B

- positions, source strength
- data structure (octree)
- merge octree
- data structure (neighbors)
- redistributed particle
- single heterogeneous node algorithm
- exchange final R-expansions
- final sum

ODE solver: source receiver update

ODE solver: source receiver update
The billion size test case

- Using all 32 Chimera nodes
- Timing excludes the partition
- 1 billion runs potential kernel in 21.6 s
Scalar potential formulation of Equations of Mathematical Physics

- Vector valued functions are subject to constraints
  - E.g. Stokes flow, Maxwell
- Represent solutions via “potentials”, and remove degrees of freedom via gauge selection
- Approach not usable with FMM directly
- With Nail Gumerov developed this idea for FMM and showed speedup over best other algorithm proposed
- Biharmonic (2/5); Stokes (3/4); Vortex (2/5); Maxwell (2/6); Elasticity (in progress)
- Boundary element formulations for all these
Electrostatics, Gravity, Potential flows

\[ \nabla^2 \phi = 0 \quad \text{Laplace} \]

\[ \nabla^2 \phi = f \quad \text{Poisson} \]

\[ \nabla^2 u = 0 \quad \text{Electricity} \]

\[ \nabla \cdot u = 0 \quad \text{Elasticity} \]

\[ \nabla^4 u = 0 \]

\[ \nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{J} + \frac{1}{c} \frac{\partial}{\partial t} \mathbf{E} \]

\[ \nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial}{\partial t} \mathbf{B} \]

\[ \nabla \cdot \mathbf{E} = 4\pi \rho \quad \text{Maxwell} \]

\[ \nabla \cdot \mathbf{B} = 0 \]

Waves, Diffusion, Quantum mechanics (spectral domain, \( k \) is complex)

Electrodynamics, Plasma physics

Uniform Helmholtz

Non-uniform Helmholtz

Constrained Vector Poisson

\[ \nabla^2 u = f \]

\[ \nabla \cdot u = q \]

Vector Poisson

Creeping flows, Micro- and nanofluidics

\[ \mu \nabla^2 u - \nabla p = \rho \frac{\partial}{\partial t} u \]

\[ \nabla \cdot u = 0 \quad \text{Unsteady Stokes} \]

Fluid dynamics

Vortical flows, Reconstruction of vector fields

High Strouhal number flows

Unsteady Stokes

Elasticity

Electromagnetism
Teaching the FMM

- Course notes since 2004 online
- [www.umiacs.umd.edu/~ramani/cmsc878R](http://www.umiacs.umd.edu/~ramani/cmsc878R)
- Java Applet that demos the FMM
- 10 papers directly from the course
- enabled solidifying broad themes
A computational camera for spatial sound

VisiSonics Corporation

www.visisonics.com
Spherical Beamforming

Wave scattering from a rigid (sound hard) surface find solution to Helmholtz equation which satisfies: the rigid surface, \( \partial \psi / \partial n = 0 \) radiation condition on \( \psi_{\text{scat}} \)

\[
(\nabla^2 + k^2) \psi(\mathbf{r}) = 0
\]

\[
\psi(\theta_s, \theta_k, k\alpha)
= [\psi_{\text{in}}(\mathbf{r}_s, k) + \psi_{\text{scat}}(\mathbf{r}_s, k)]|_{r_s=a}
= 4\pi \sum_{n=0}^{\infty} i^n b_n(k\alpha) \sum_{m=-n}^{n} Y_{n}^{m}(\theta_k)Y_{n}^{m*}(\theta_s),
\]

\[
b_n(k\alpha) = j_n(k\alpha) - \frac{j_n'(k\alpha)}{h_n'(k\alpha)} h_n(k\alpha),
\]
VisiSonics 5/64 Audio Visual Camera

64 microphones: 20 Hz – 20 kHz, 48 kHz sampling, 24 bit digitization

Collects video with 5 HD cameras

Produces a frame-rate video panorama using proprietary GPU based stitching algorithms

Performs beamforming in over 10,000 directions at frame-rate, to produce “audio images”

Portable – the compact microphone and camera layout is achieved via a novel microphone architecture

Connects to Laptop with GPU, no other expensive hardware

Offline processing of listening spaces enabled by a spectrotemporal analysis tool.
Dekelbaum theater at Clarice Smith Performing arts Center at UMD

• Mercator projection created from 24 snapshots
Studying Reverberation
Surveillance
In Car Example Image
Direct Audio Imaging of Harmonic Orders

- Order Tracking module
Conclusions

• Two successful applications shown where GPUs were used to accelerate computing
• In each case accounting for the fact that GPUs are a component in a heterogeneous system lead to success
• GPUs are important, but the algorithm/system has to be considered in its entirety
• Fast multipole summation in heterogeneous environment has lead to the best scaled code capable of exascale performance
• Combining signal acquisition with cameras and microphones, PC based control and display, GPU based beamforming and video stitching, has lead to a unique tool for audio scene analysis