Enabling High Performance Computational Dynamics in a Heterogeneous Hardware Ecosystem

Assistant Prof. **Dan Negrut**
Simulation-Based Engineering Lab
Department of Mechanical Engineering
University of Wisconsin – Madison

Toward High-Performance Computing Support
for the Simulation and Planning of Robot Contact Tasks
June 27, 2011

Acknowledgements

- People:
  - Alessandro Tasora – University of Parma, Italy
  - Mihai Anitescu – Argonne National Lab
  - Lab Students:
    - Hammad Mazhar
    - Toby Heyn
    - Andrew Seidl
    - Arman Pazouki
    - Hamid Ansari
    - Justin Madsen
    - Naresh Khude
    - Makarand Datar
    - Dan Melanz
    - Markus Schmid
- Financial support
  - National Science Foundation, Young Investigator Career Award
  - FunctionBay, S. Korea
  - NVIDIA
  - Microsoft
  - Caterpillar

GOAL OF OUR EXERCISE…

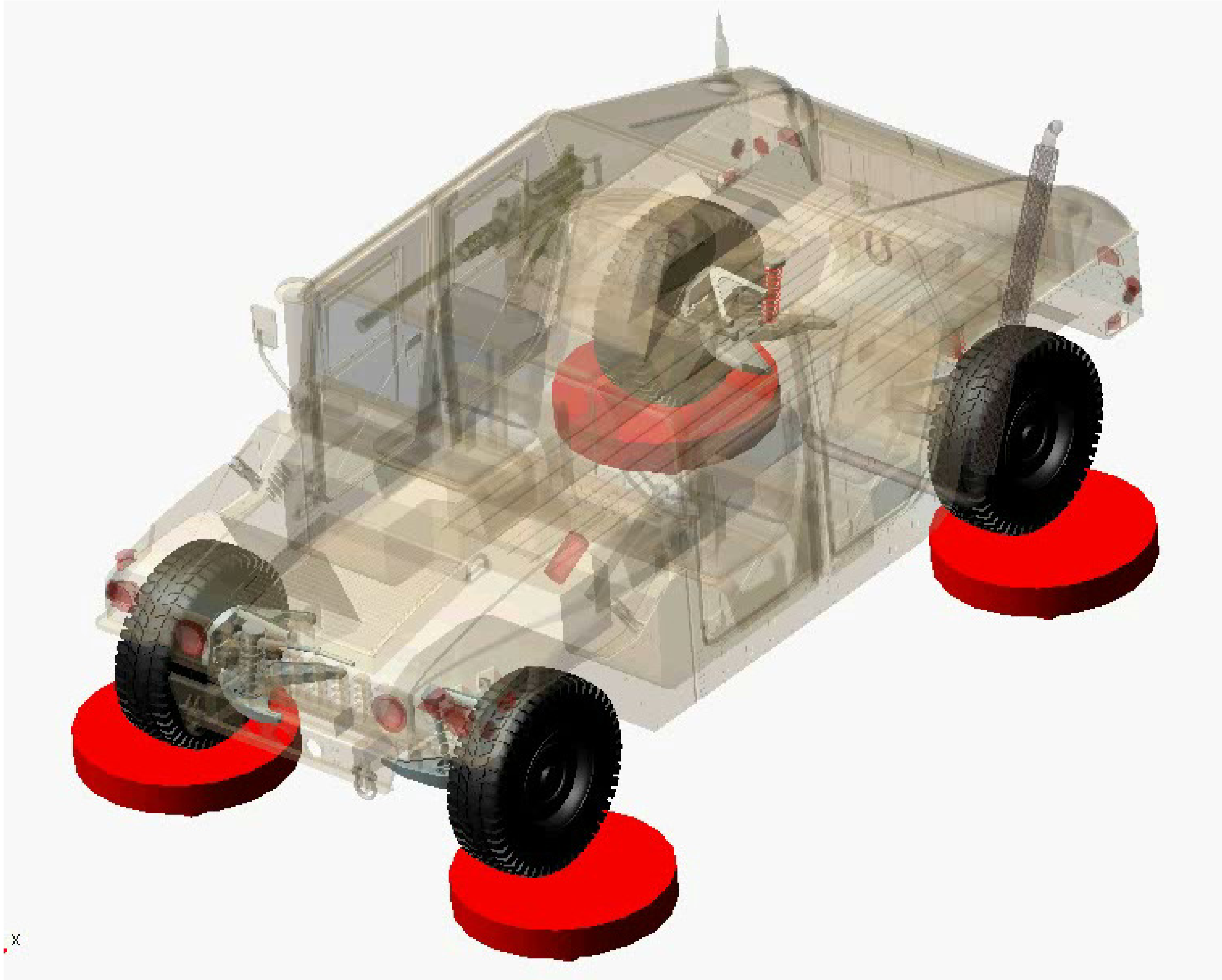
- Use HPC to simulate the dynamics of real-life engineering mechanical systems at unprecedented levels of accuracy

- HPC hardware targeted:
  - Cluster of CPUs and GPUs (accelerators)
    - More than 100 CPU cores, tens of GPU cards

Talk Overview

- Overview of the engineering problems of interest
- Large-scale Multibody Dynamics
  - Problem formulation, solution method, and parallel implementation
- Overview of Heterogeneous Computing Template (HCT)
- Numerical Experiments
- Conclusions

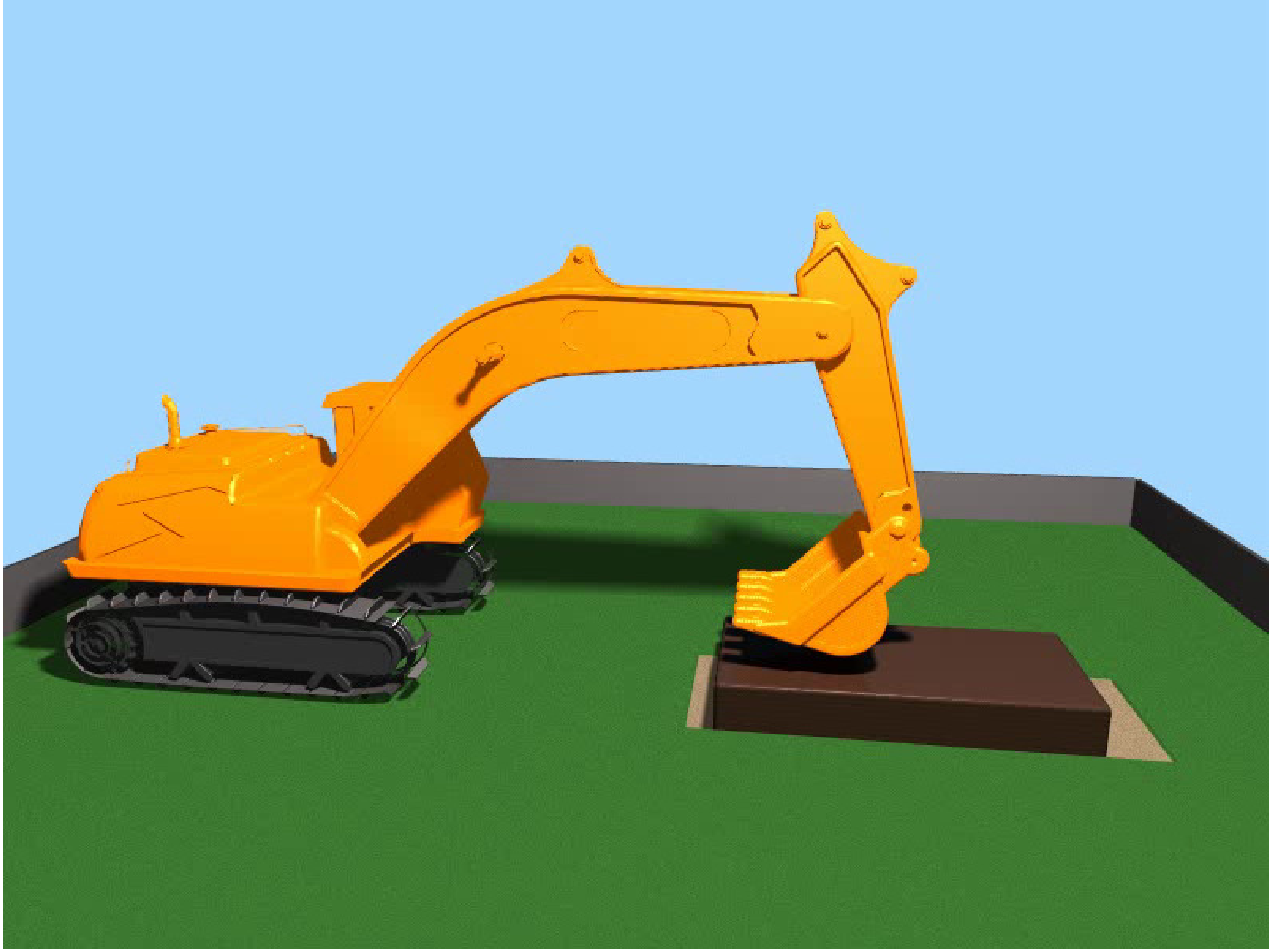
Computational Multibody Dynamics



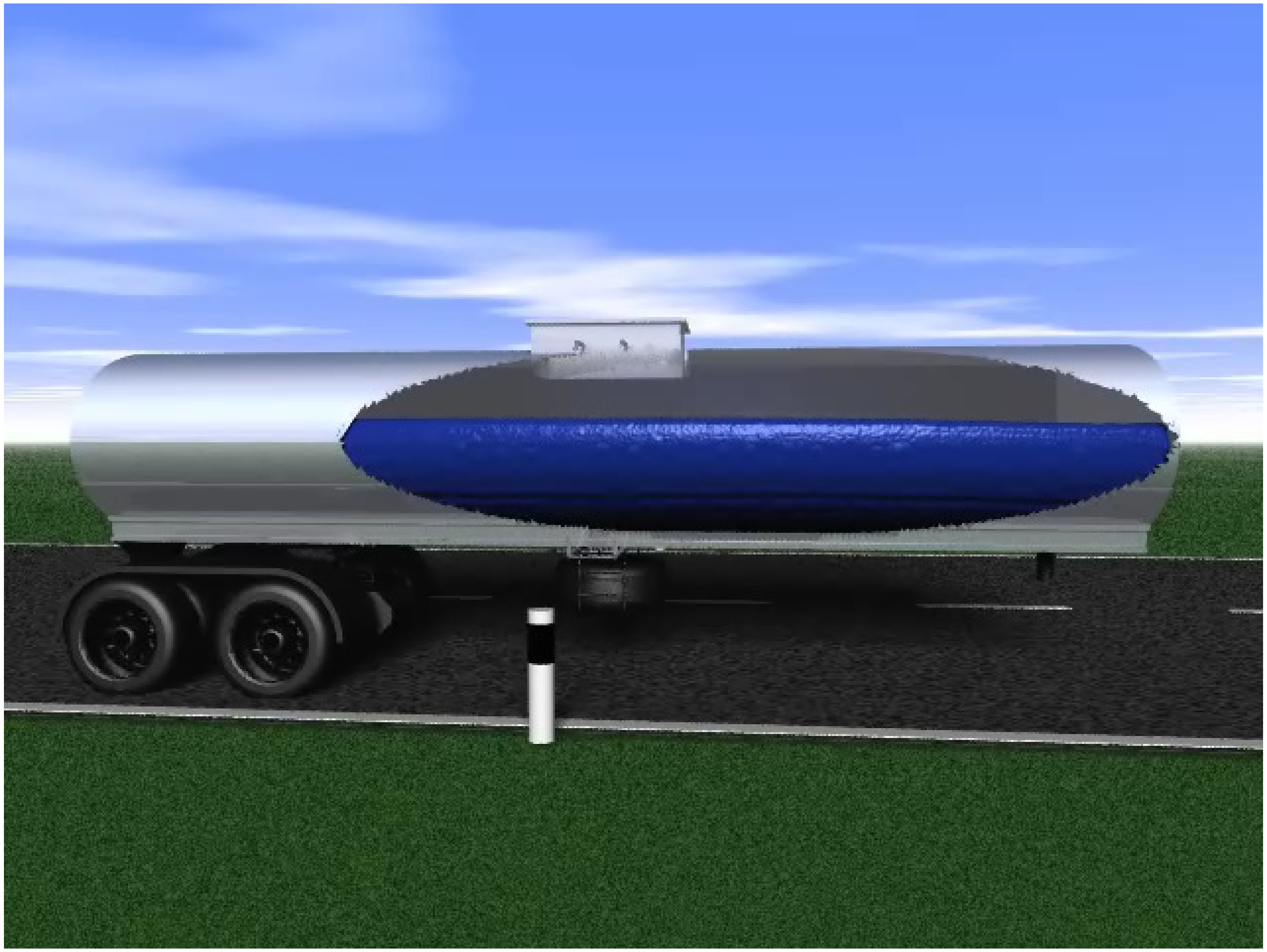
Simulation generated in ADAMS

Multi-Physics…

Fluid-Solid Interaction: Navier-Stokes + Newton-Euler.

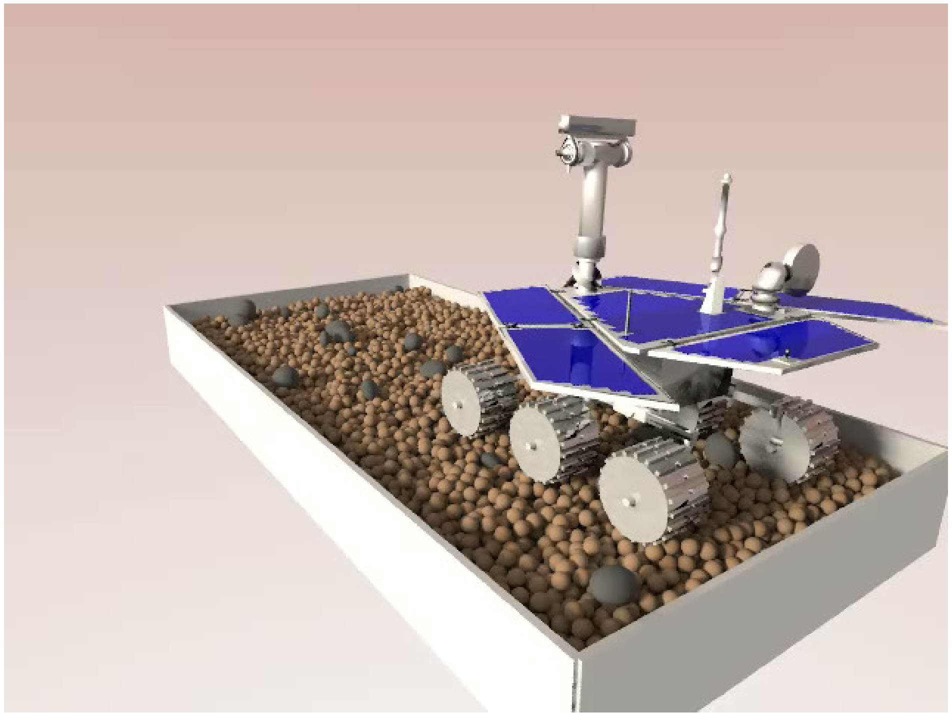


Computational Dynamics

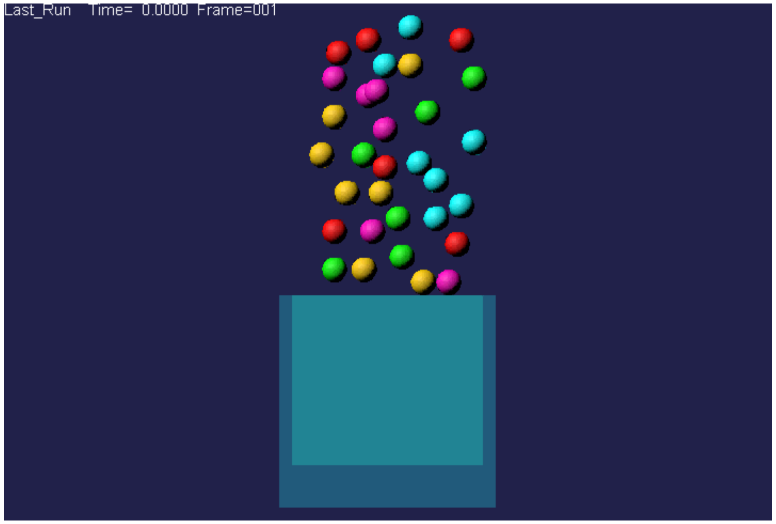


Rover Mobility on Granular Terrain

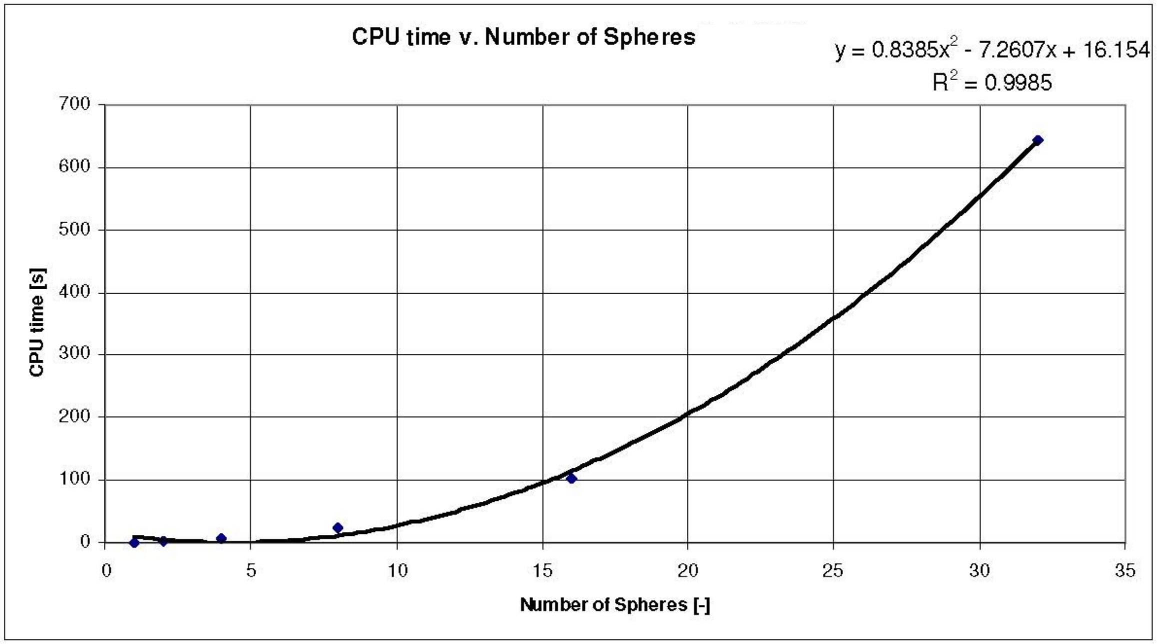
- Wheeled/tracked vehicle mobility on granular terrain
- Also interested in scooping and loading granular material



Frictional Contact Simulation
[Commercial Solution]



- Model Parameters:
  - Spheres: 60 mm diameter and mass 0.882 kg
  - Forces: smoothing with stiffness of 1E5, force exponent of 2.2, damping coefficient of 10.0, and a penetration depth of 0.1
  - Simulation length: 3 seconds



Frictional Contact: Two Different Approaches Considered

- Discrete Element Method (DEM) - draws on a “smoothing” (penalty) approach
  - Lots of heuristics
  - Slow
  - General purpose
  - Used in ADAMS

- DVI-based (Differential Variational Inequalities)
  - A set of differential equations combined with inequality constraints
  - Fast (stable for significantly larger integration step-sizes)
  - Less general purpose
  - Used widely in computer games

The Modeling Component

Equations of Motion: Multibody Dynamics

Generalized Positions

Generalized Mass Matrix

Velocity Transformation Matrix

Generalized Velocities

Kinematic Differential Equations → $\dot{\mathbf{q}} = \mathbf{T}(\mathbf{q})\mathbf{v}$

Force Balance Equations → $\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{f}(t,\mathbf{q},\mathbf{v}) - \mathbf{g}_{\mathbf{q}}^{\mathrm{T}}(\mathbf{q},t)\boldsymbol{\lambda} + \sum_{i=1}^{N_c}(\gamma_n^i \mathbf{D}_n^{T,i} + \gamma_u^i \mathbf{D}_u^{T,i} + \gamma_w^i \mathbf{D}_w^{T,i})$

Reaction Force

Frictional Contact Force

Applied Force

Holonomic Kinematic Constraints → $\mathbf{g}(\mathbf{q},t) = \mathbf{0}$

Contact Impulse, for Contact "*i*"

Contact Complementarity Conditions → $0 \le \Phi^i(\mathbf{q},t) \perp \gamma_n^i \ge 0 \qquad i = 1,2,\ldots,N_c$

Total Number of Contacts

Coulomb Friction Model → $(\gamma_u^i, \gamma_w^i) = \underset{\mu^i \gamma_n^i \ge \sqrt{(\gamma_u^i)^2 + (\gamma_w^i)^2}}{\arg\min} (\gamma_u^i \mathbf{v}^T \mathbf{D}_u^i + \gamma_w^i \mathbf{v}^T \mathbf{D}_w^i)$

Friction Dissipation Energy

Gap Function, for Contact "*i*"

Friction Impulse Components, for Contact "*i*"

Traditional Discretization Scheme

positions

time step index

$$\mathbf{q}^{(l+1)} = \mathbf{q}^{(l)} + h\mathbf{L}(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)}$$

Mass Mat.

speeds

Applied Forces

Reaction impulses

$$\mathbf{M}(\mathbf{v}^{(l+1)} - \mathbf{v}^{l}) = h\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) + \sum_{i \in \mathcal{A}(q^{(l)},\delta)} (\gamma_{i,n}\,\mathbf{D}_{i,n} + \gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w})$$

$$i \in \mathcal{A}(q^{(l)}, \delta): \quad 0 \leq \frac{1}{h}\Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^{T}\mathbf{v}^{(l+1)} \perp \gamma_n^i \geq 0,$$

Complementarity Condition

$$(\gamma_{i,u}, \gamma_{i,w}) = \underset{\mu_i \gamma_{i,n} \geq \sqrt{\gamma_{i,u}^2 + \gamma_{i,w}^2}}{\operatorname{argmin}} \mathbf{v}^T (\gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w}) \, .$$

Stabilization term

Coulomb 3D fricion model

(Stewart & Trinkle, 1996)

Relaxed Discretization Scheme Used

$$\mathbf{q}^{(l+1)} = \mathbf{q}^{(l)} + h\mathbf{L}(\mathbf{q}^{(l)})\mathbf{v}^{(l+1)}$$

$$\mathbf{M}(\mathbf{v}^{(l+1)} - \mathbf{v}^{l}) = h\mathbf{f}(t^{(l)}, \mathbf{q}^{(l)}, \mathbf{v}^{(l)}) + \sum_{i\in\mathcal{A}(q^{(l)},\delta)} \left(\gamma_{i,n}\,\mathbf{D}_{i,n} + \gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w}\right)$$

$$i \in \mathcal{A}(q^{(l)},\delta): \quad 0 \leq \frac{1}{h}\Phi_i(\mathbf{q}^{(l)}) + \mathbf{D}_{i,n}^{T}\mathbf{v}^{(l+1)} - \mu^{i}\sqrt{(\mathbf{v}^{T}\,\mathbf{D}_{i,u})^2 + \mathbf{v}^{T}\,\mathbf{D}_{i,w})^2} \perp \gamma_n^i \geq 0,$$

Relaxation Term

$$(\gamma_{i,u}, \gamma_{i,w}) = \underset{\mu_i\gamma_{i,n}\geq\sqrt{\gamma_{i,u}^2+\gamma_{i,w}^2}}{\operatorname{argmin}} \mathbf{v}^{T}\left(\gamma_{i,u}\,\mathbf{D}_{i,u} + \gamma_{i,w}\,\mathbf{D}_{i,w}\right).$$

(Anitescu & Tasora, 2008)

The Cone Complementarity Problem (CCP)

- First order optimality conditions lead to Cone Complementarity Problem

- Introduce the convex hypercone...

$$\Upsilon = \left(\bigoplus_{i \in \mathcal{A}(\mathbf{q}^l, \epsilon)} \mathcal{FC}^i \right)$$

... and its polar hypercone:

$$\Upsilon^\circ = \left(\bigoplus_{i \in \mathcal{A}(\mathbf{q}^l, \epsilon)} \mathcal{FC}^{i\circ} \right)$$

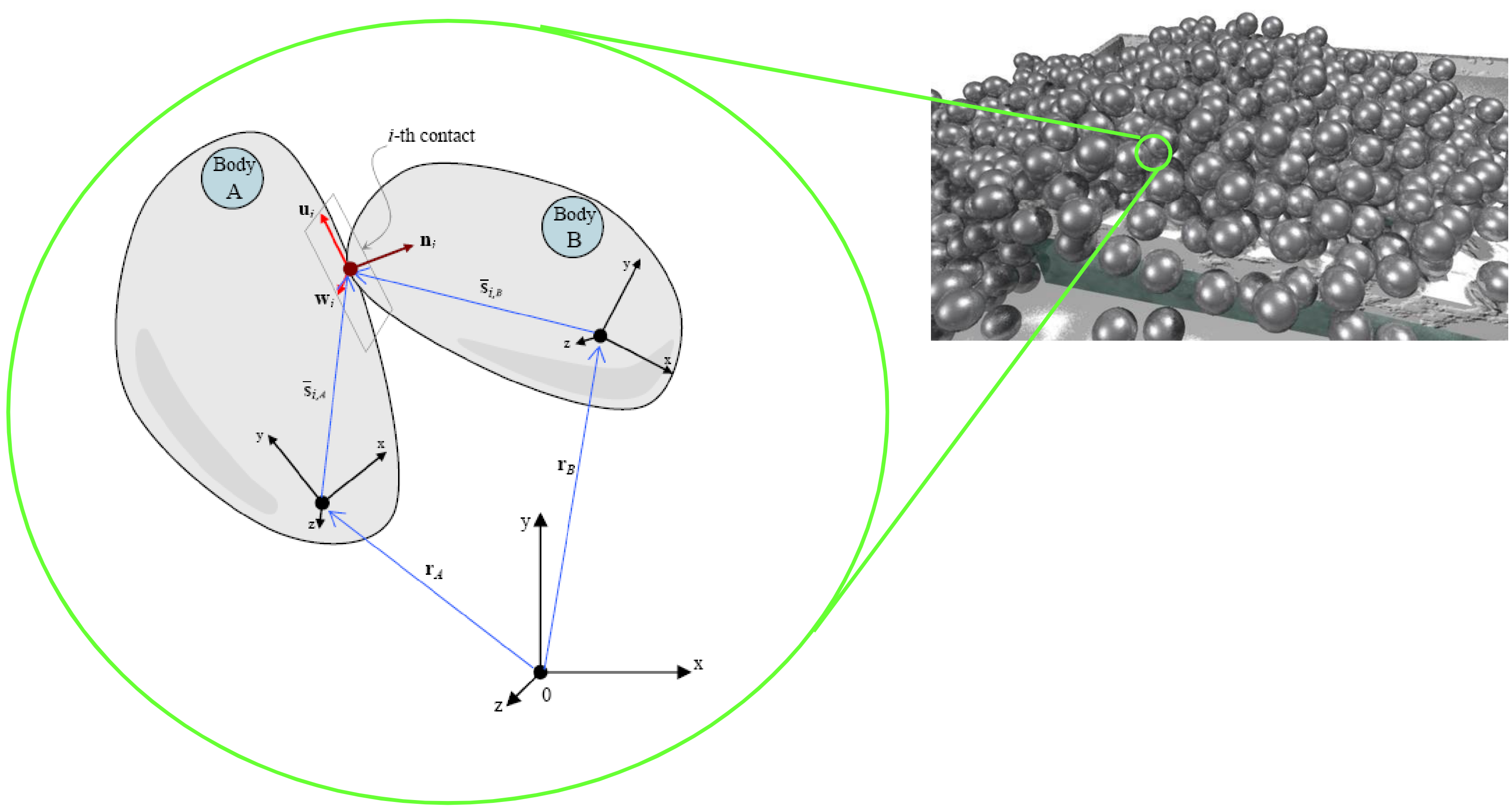
$\mathcal{FC}^i \in \mathbb{R}^3$ represents friction cone associated with i^{th} contact

CCP assumes following form: Find γ such that

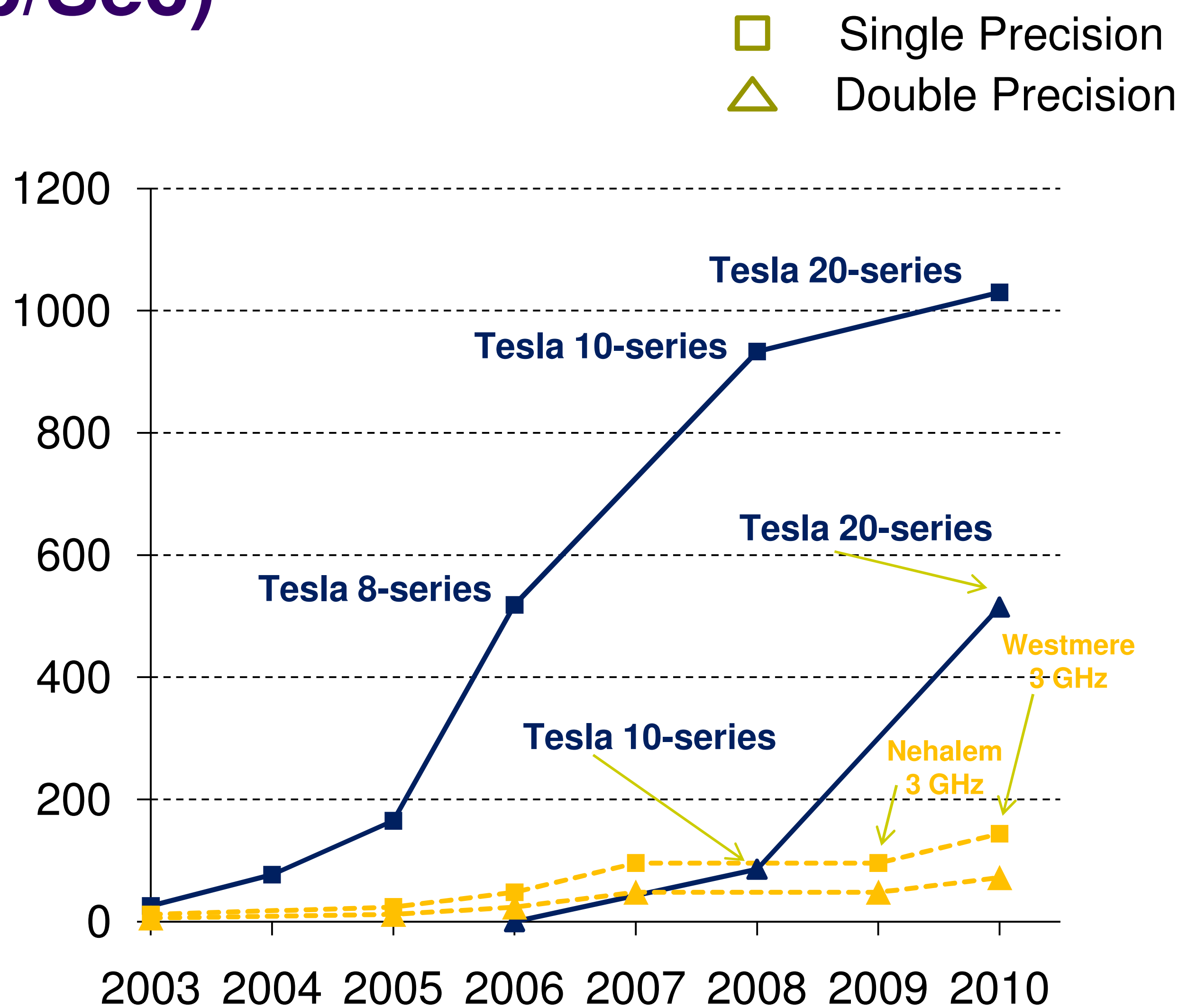
$$\gamma \in \Upsilon \perp -(\mathbf{N}\gamma + \mathbf{d}) \in \Upsilon^\circ$$

Large Scale Granular Dynamics

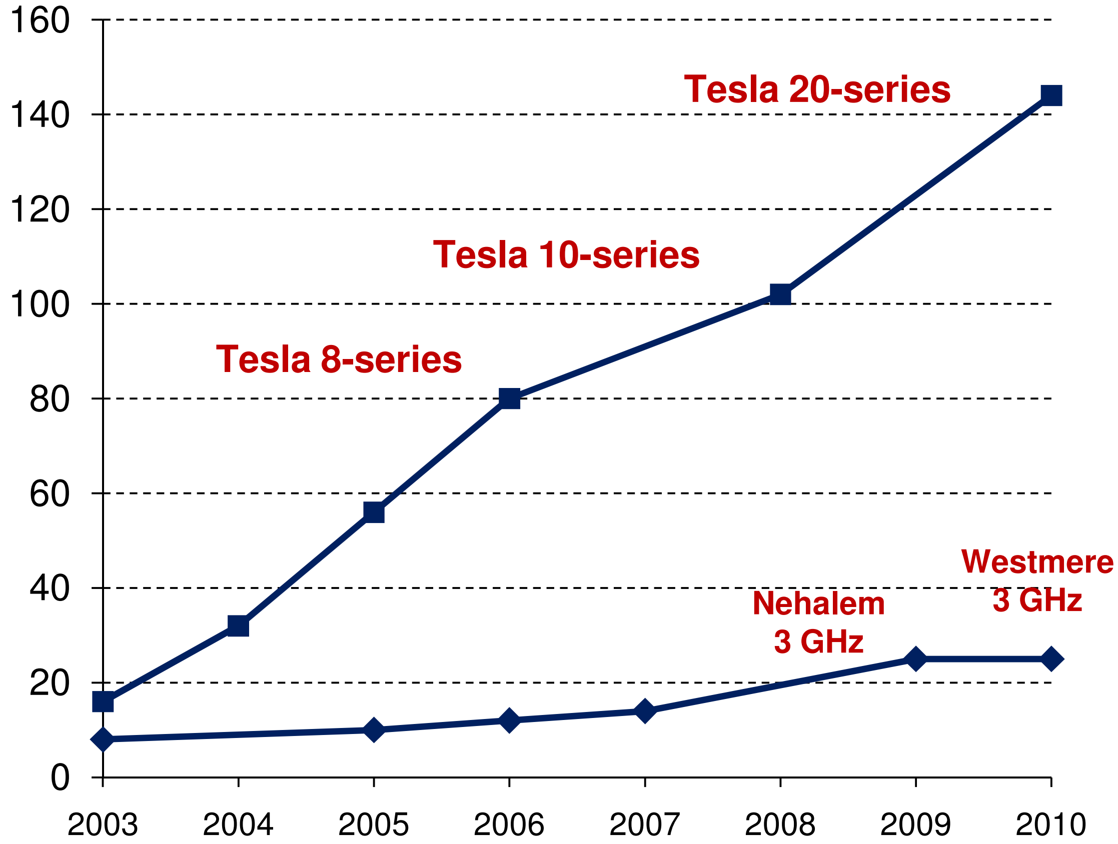
- Numerical solution can leverage parallel computing



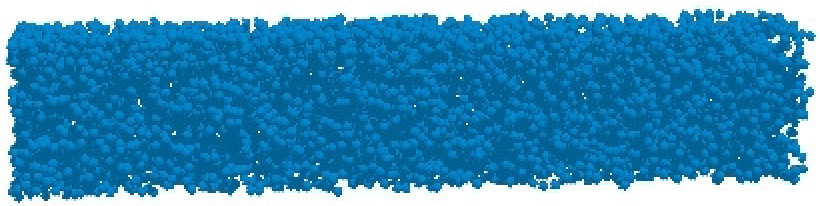
CPU vs. GPU – Flop Rate (GFlop/Sec)



CPU vs. GPU– Memory Bandwidth
[GB/sec]

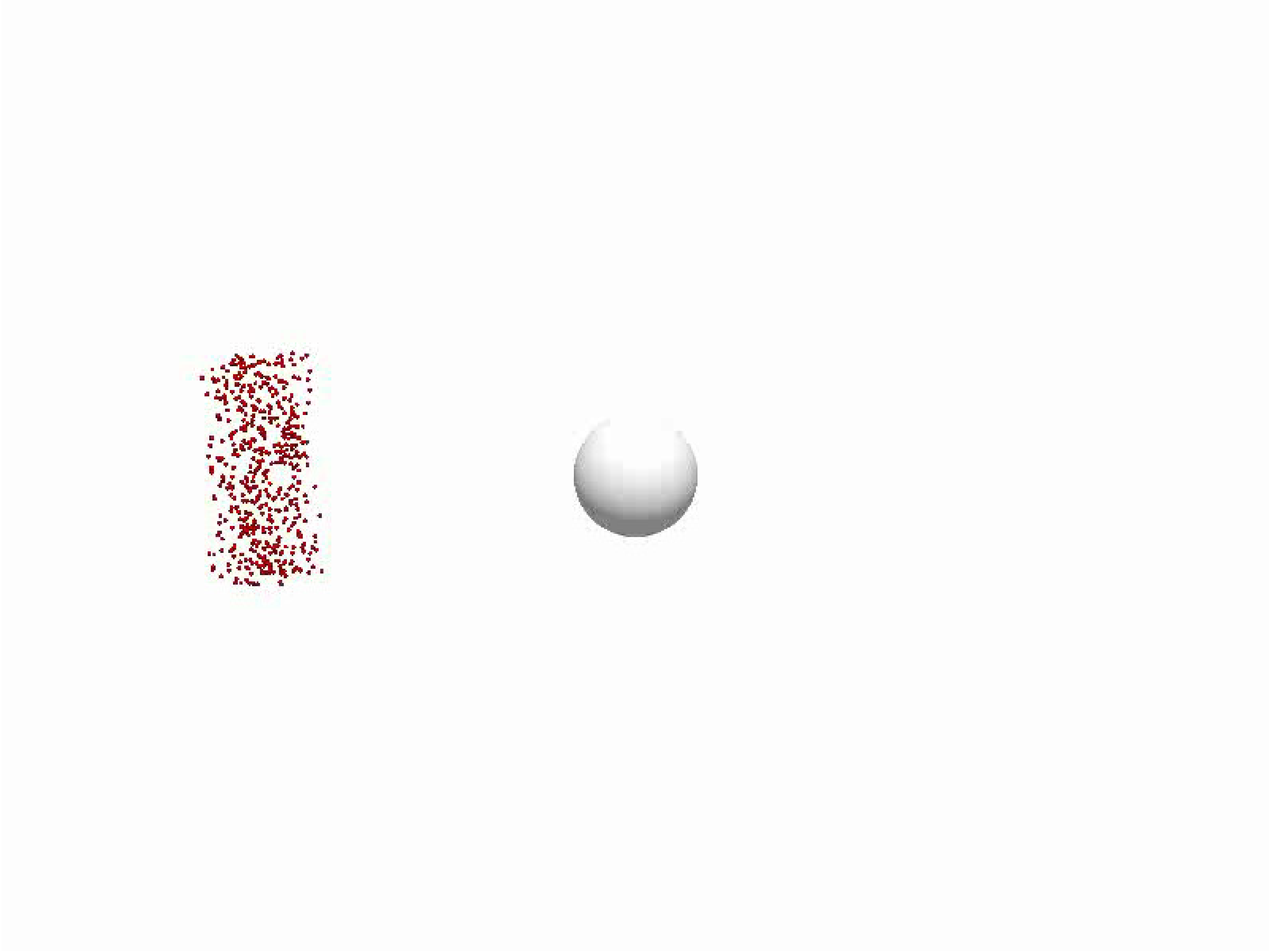


Mixing 40,000 Spheres on the GPU



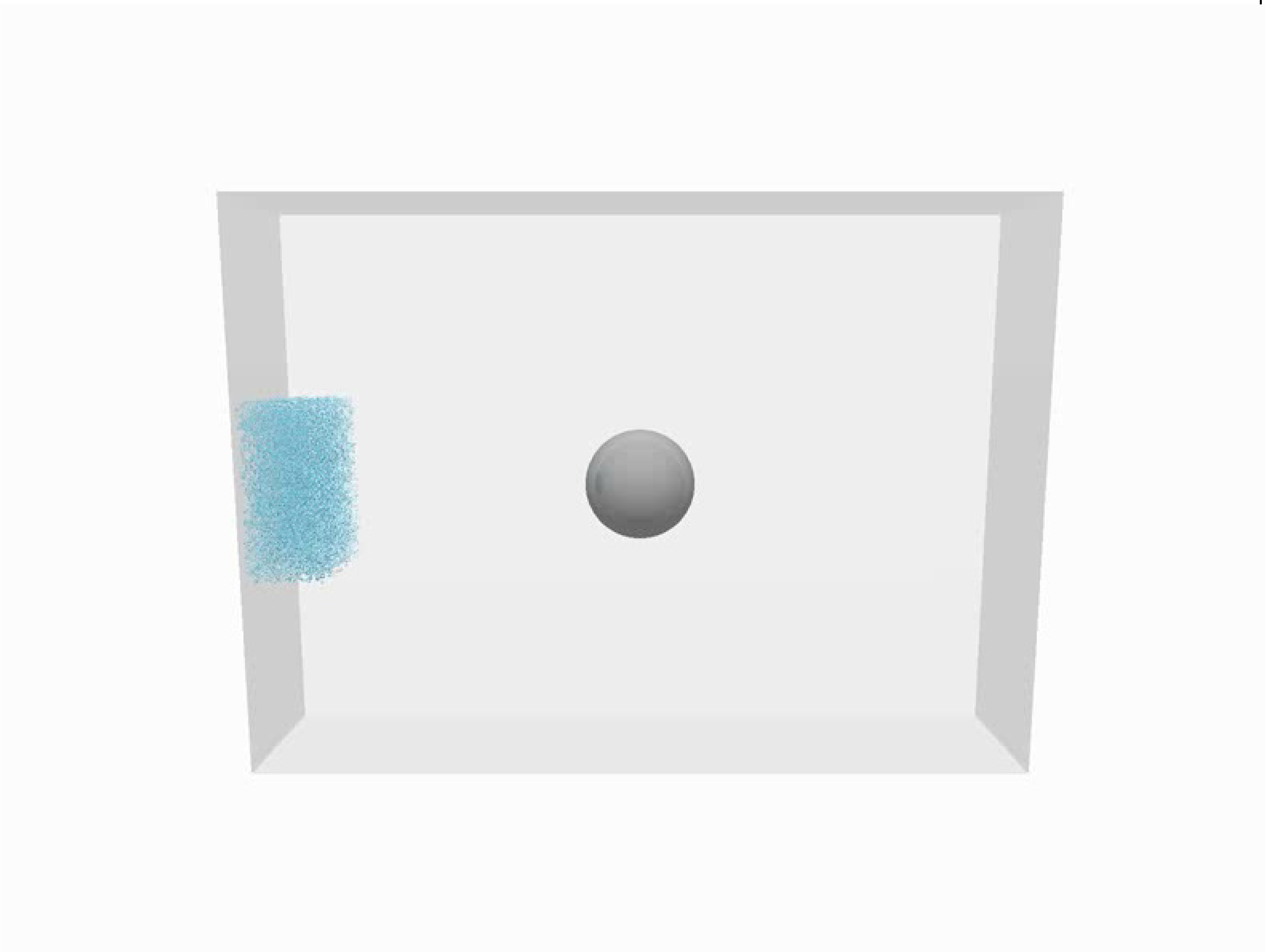
300K Spheres in Tank

[parallel on the GPU]



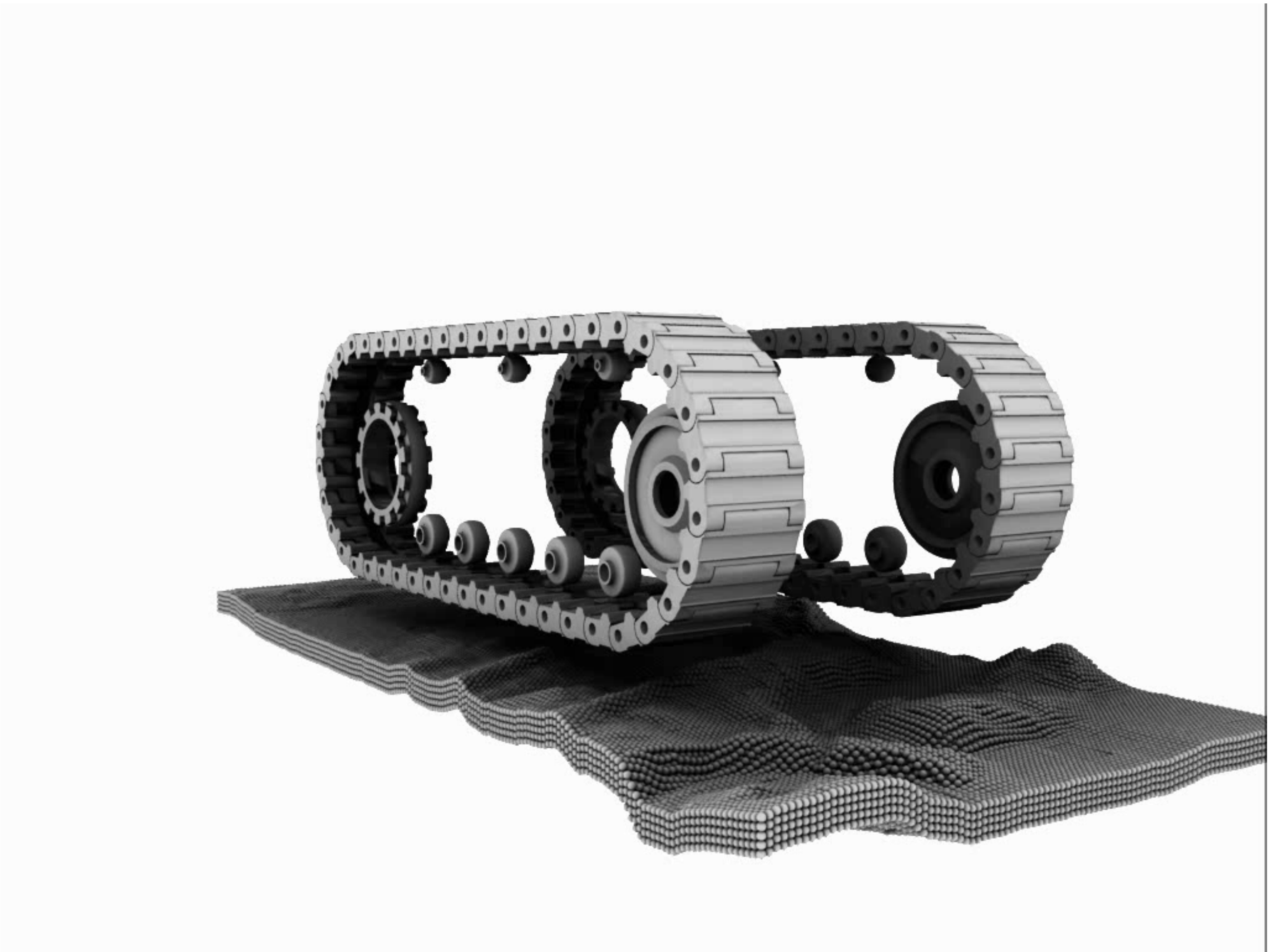
1.1 Million Rigid Spheres

[parallel on the GPU]



Computational dynamics

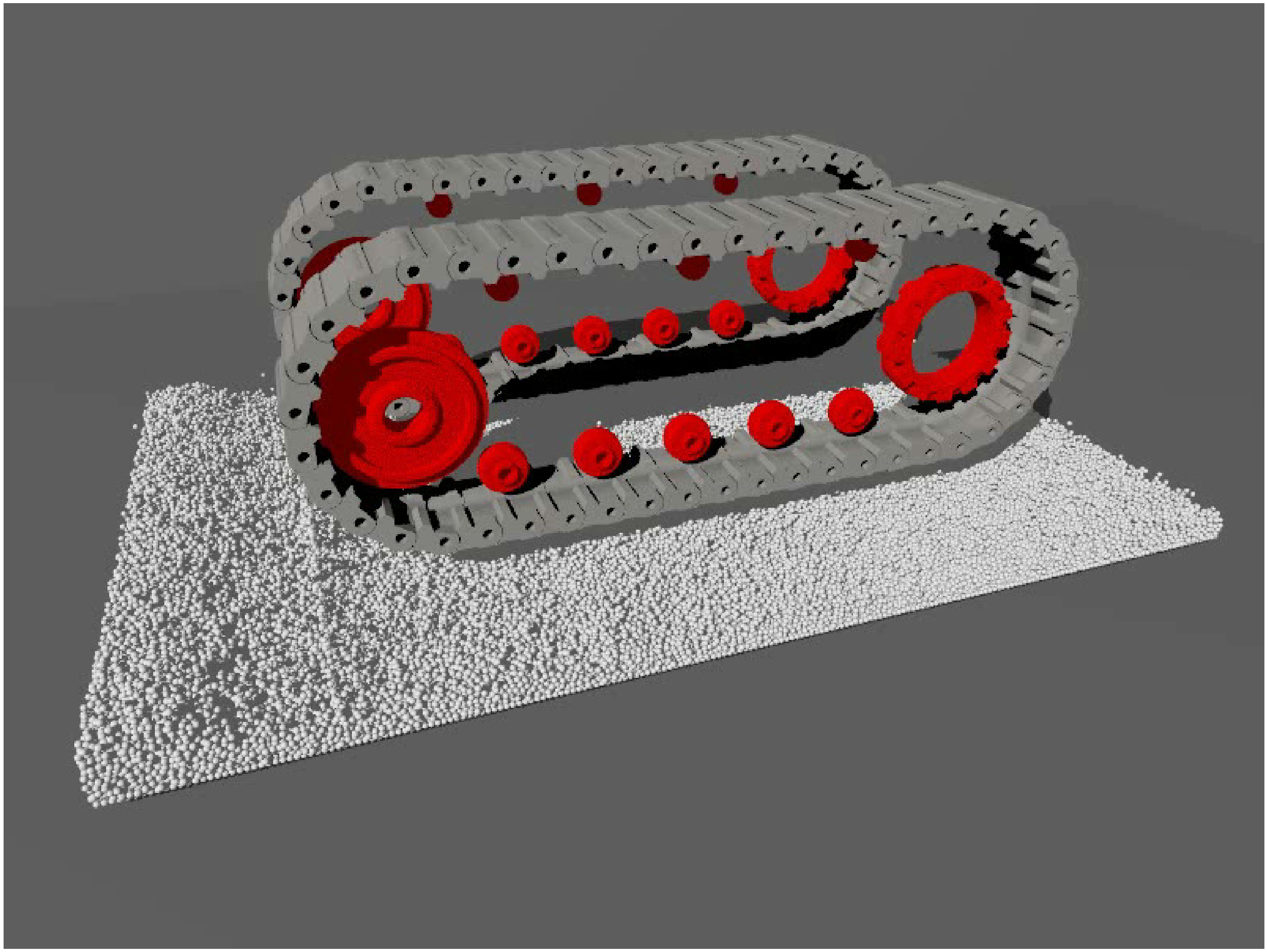
Tracked vehicle mobility



Simulation Setup:

- Driving speed: 1.0 rad/sec
- Length: 12 seconds
- Time step: 0.005 sec
- Computation time: 18.5 hours
- Particle radius: .027273 m
- Terrain: 284,715 particles

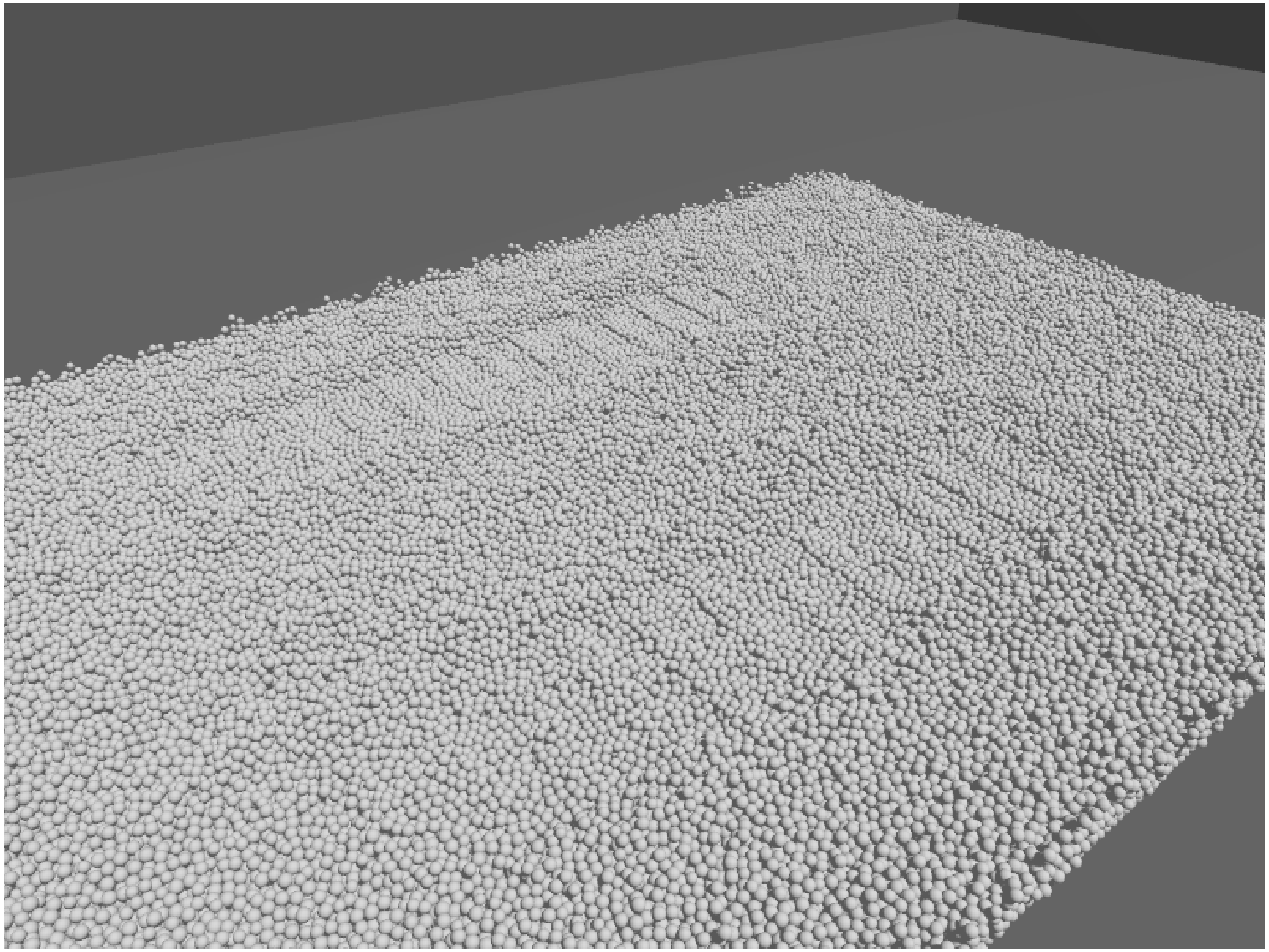
Track Simulation



Parameters:

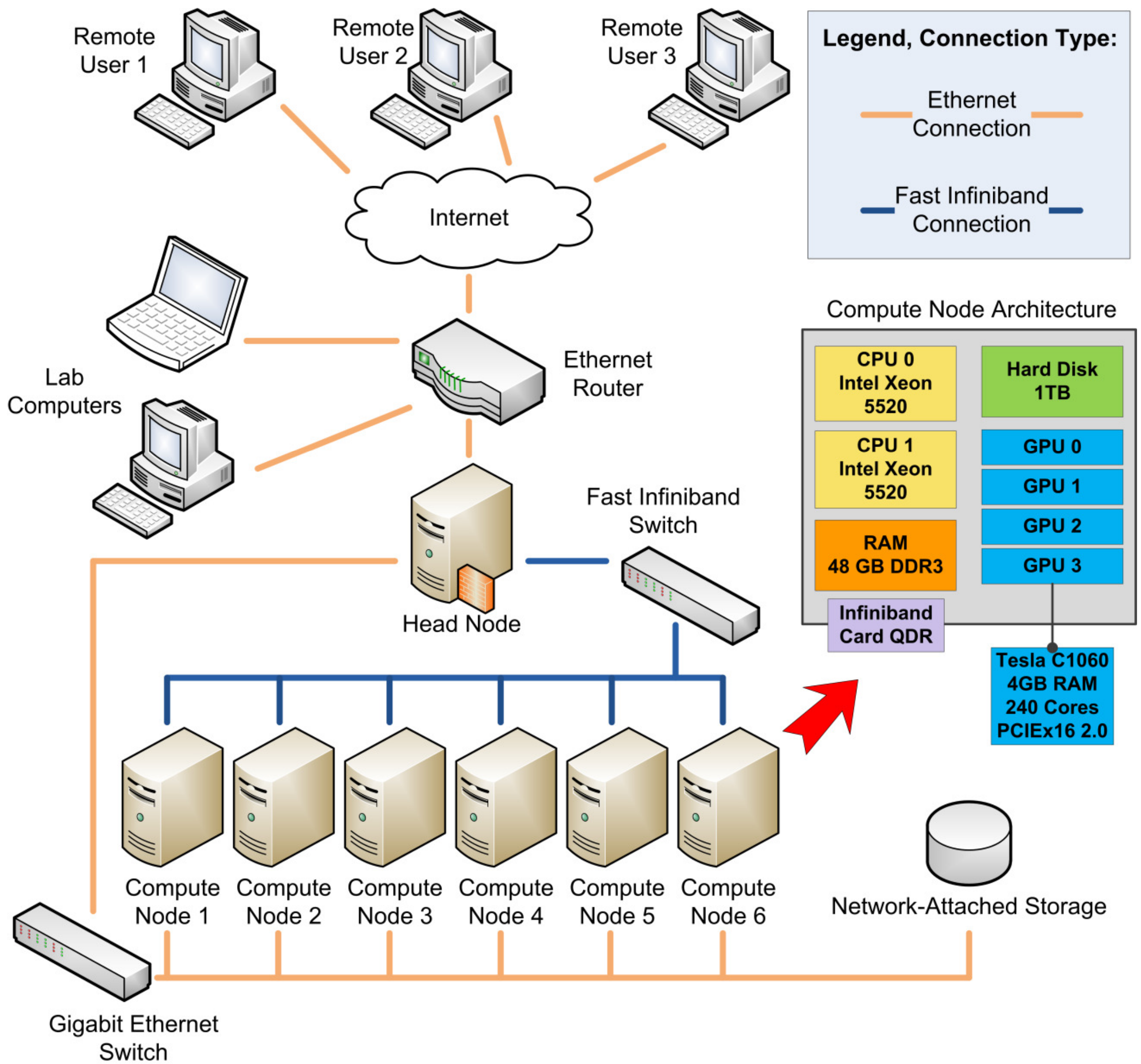
- Driving speed: 1.0 rad/sec
- Length: 10 seconds
- Time step: 0.005 sec
- Computation time: 17.8 hours
- Particle radius: .025±.0025 m
- Terrain: 467,100 particles

Track Footprint



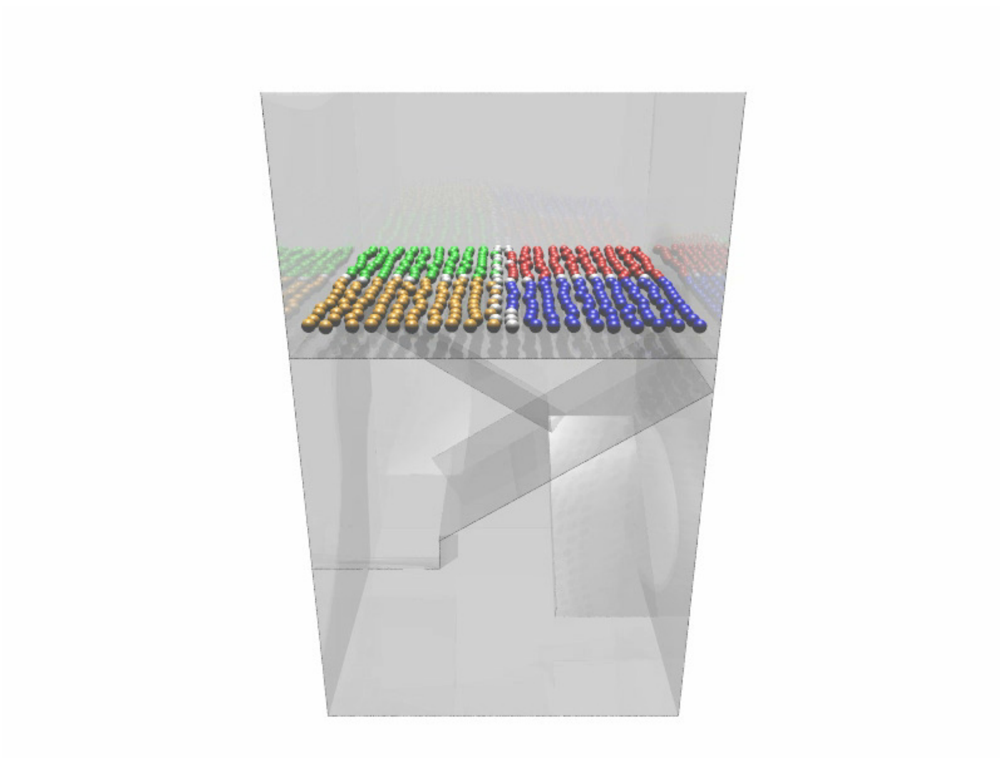
A Heterogeneous Computing Template for Computational Dynamics

Heterogeneous Cluster



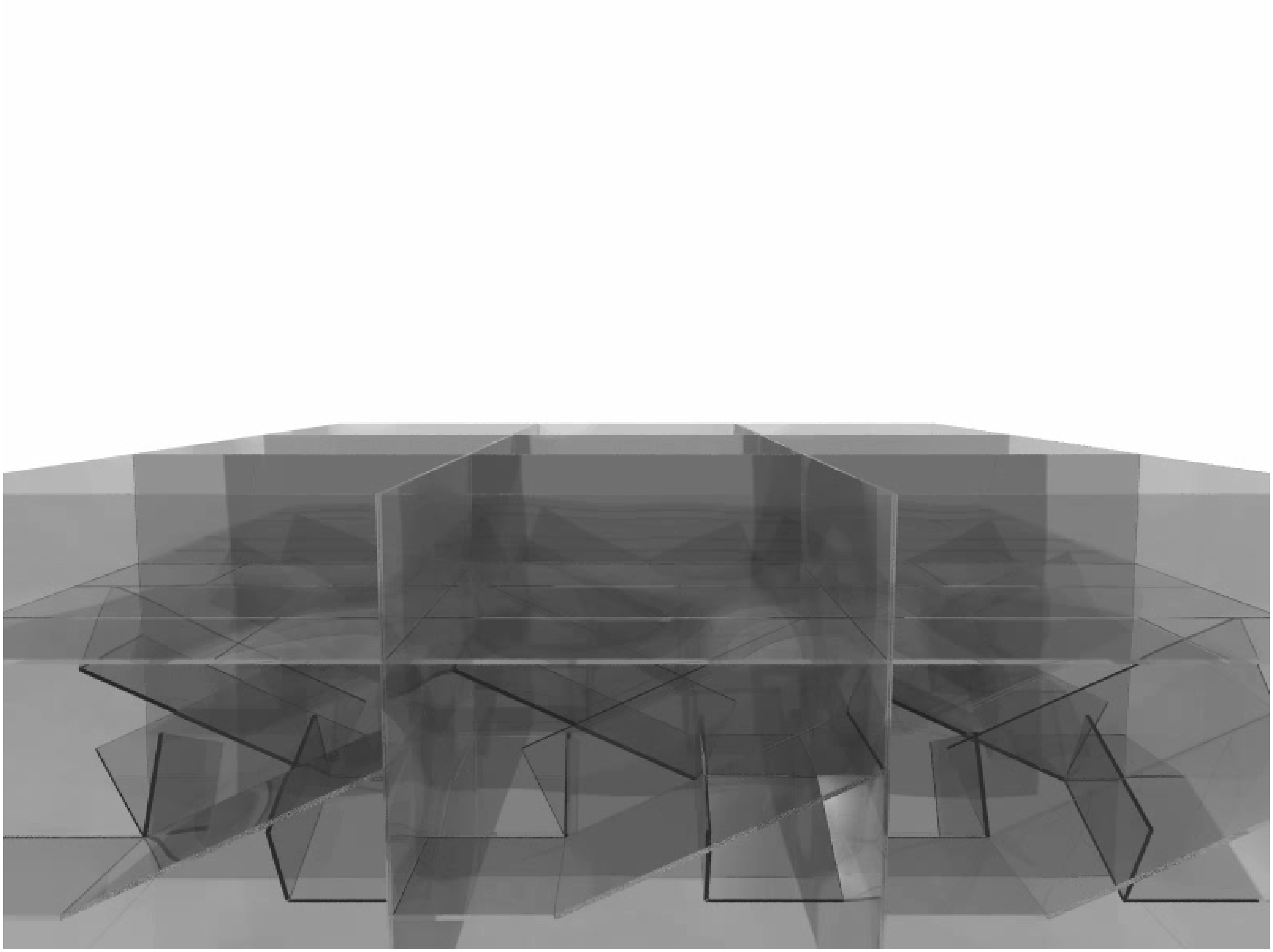
Second fastest cluster at University of Wisconsin-Madison

Computation Using Multiple CPUs
[DEM solution]



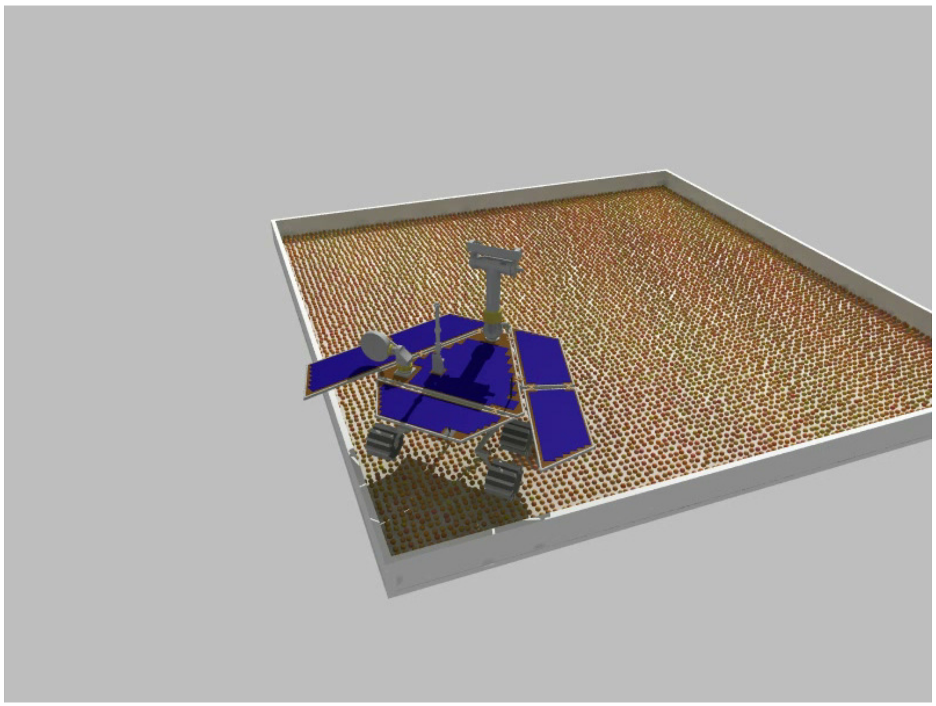
Computation Using Multiple CPUs

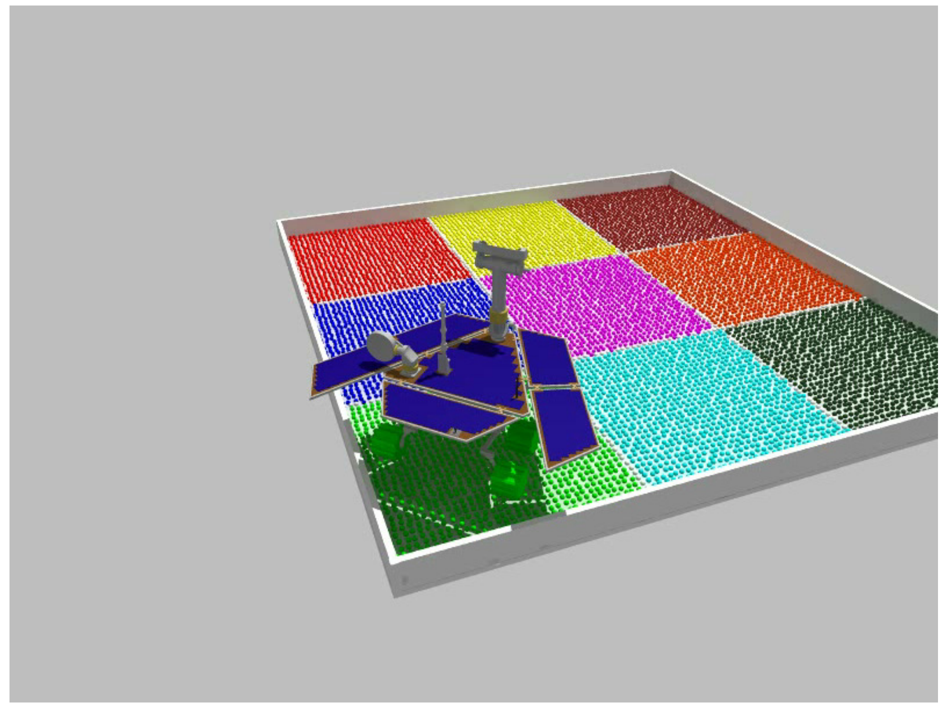
[DEM solution]



Computation Using Multiple CPUs

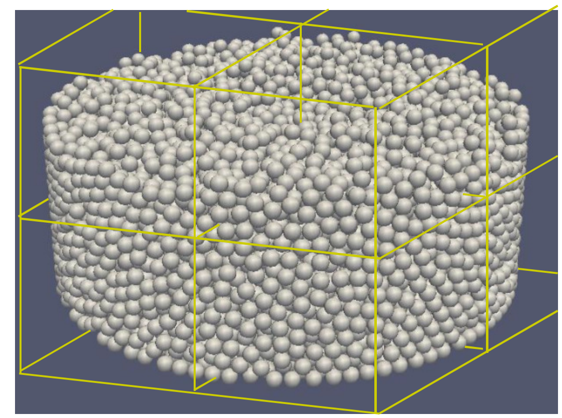
[DEM solution]





Heterogeneous Computing Template
Five Major Components

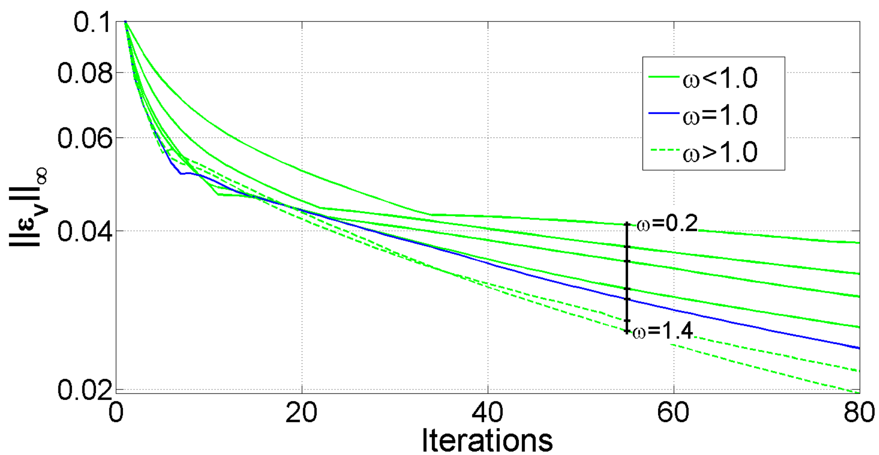
- Computational Dynamics requires
  - Domain decomposition
  - Proximity computation
  - Inter-domain data exchange
  - Numerical algorithm support
  - Post-processing (visualization)

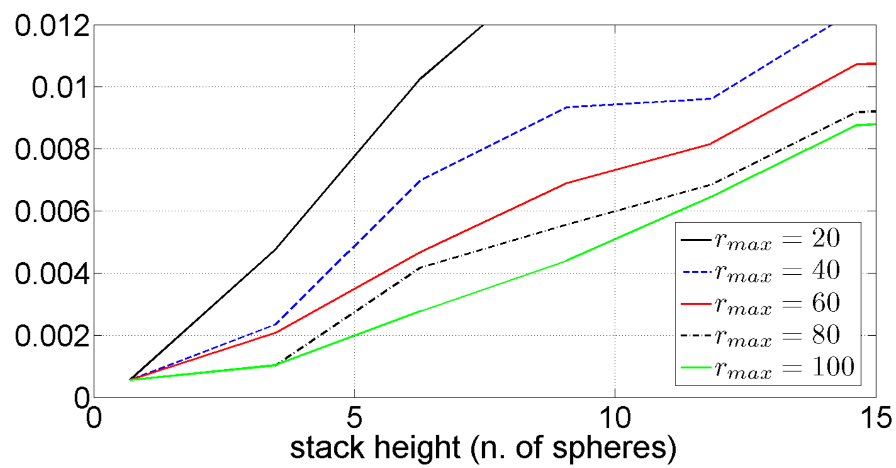


- HCT represents the library support and associated API that capture this five component abstraction

Typical Simulation Results…

- LEFT: Infinity norm of the residual vs. iteration index in the CCP solution
  - Convergence rate (slope of curve) becomes smaller as the iteration index increases.



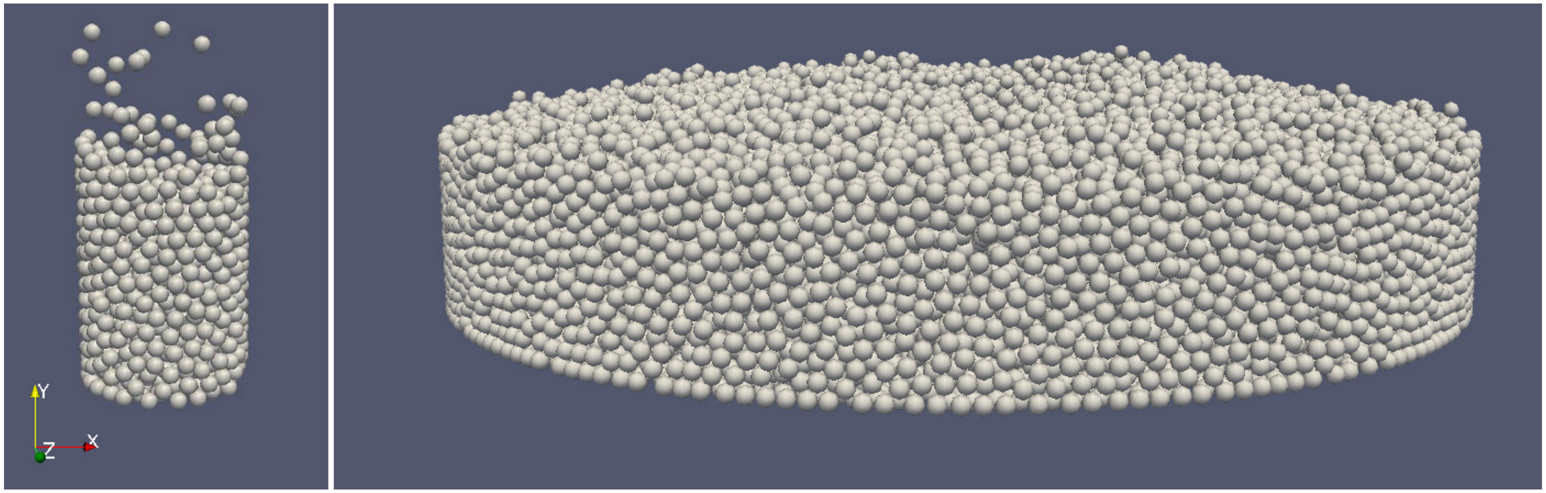


- RIGHT: Infinity norm of the CCP residual after r_{max} iterations as function of granular material depth (number of spheres stacked on each other).

Searching for Better Methods

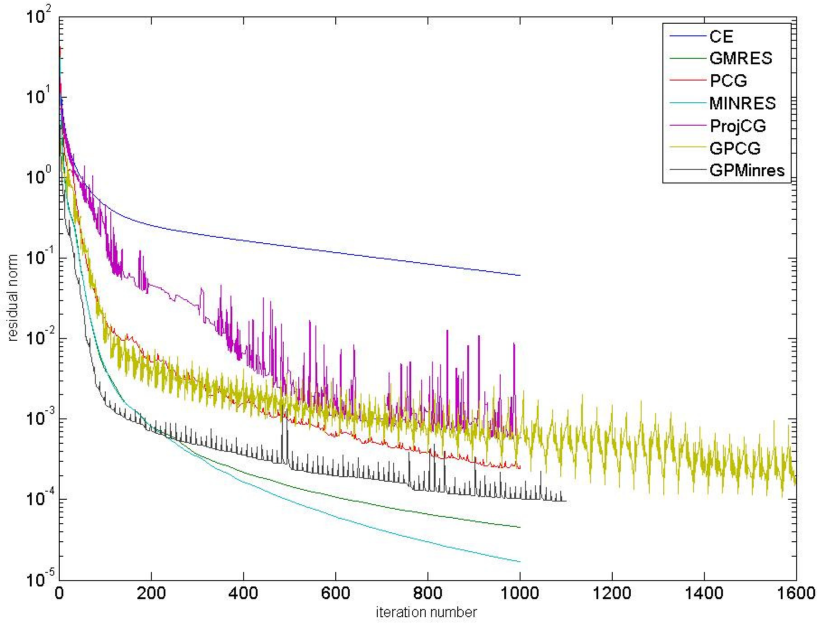
- Frictionless case (bound constraints in place)
  - Gauss-Jacobi (CE)
  - Projected conjugate gradient (ProjCG)
  - Gradient projected conjugate gradient (GPCG)
  - Gradient projected MINRES (GPMINRES)
- Friction case (cone constraints - ongoing)
  - Newton's Method for large bound-constrained problems
    - Uses re-parameterization to handle friction cones (replace with bound constraints)

Numerical Experiments



- Test Problem: 40,000 bodies $\Rightarrow$ 157,520 contacts
- Frictionless

Test Problem (MATLAB)



Method	Iterations	Final Residual Norm	γ_{min}	γ_{max}	Time [sec]
CE	1000	6.11×10^{-2}	0.0	2.0598	1849.5
ProjCG	1002	5.6344×10^{-4}	0.0	2.2286	1235.6
GPCG	1600	1.0675×10^{-4}	0.0	2.6349	382.3644
GPMinres	1100	9.5239×10^{-5}	0.0	2.3090	238.0744
PCG	1000	2.4053×10^{-4}	-1.1116	2.5254	27.9686
GMRES	1000	4.5315×10^{-5}	-1.1635	2.5227	736.3007
MINRES	1000	1.6979×10^{-5}	-1.1316	2.5253	41.5790

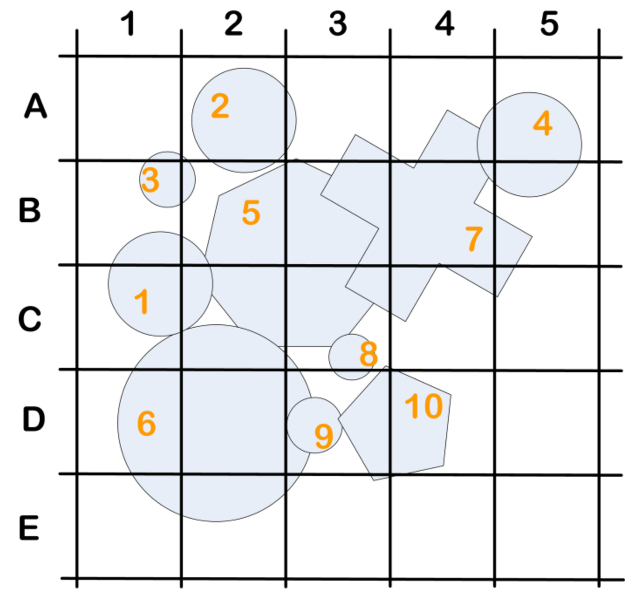
Proximity Computation

GPU Collision Detection (CD)

- 30,000 feet perspective:
  - Carry out spatial partitioning of the volume occupied by the bodies
    - Place bodies in bins (cubes, for instance)
  - Follow up by brute force search for all bodies touching each bin
    - Embarrassingly parallel

Basic Idea:
Search for Contacts in Different Bins in Parallel

- Example: 2D collision detection, bins are squares



Example: Ellipsoid-Ellipsoid CD

$$\mathbf{d} = \mathbf{P}_1 - \mathbf{P}_2 = (\frac{1}{2\lambda_1}\mathbf{M}_1 + \frac{1}{2\lambda_2}\mathbf{M}_2)\mathbf{c} + (\mathbf{b}_1 - \mathbf{b}_2)$$

$$\frac{\partial \mathbf{d}}{\partial \alpha_i} = \frac{\partial \mathbf{P}_1}{\partial \alpha_i} - \frac{\partial \mathbf{P}_2}{\partial \alpha_i} \quad , \quad \frac{\partial^2 \mathbf{d}}{\partial \alpha_i \partial \alpha_j} = \frac{\partial^2 \mathbf{P}_1}{\partial \alpha_i \partial \alpha_j} - \frac{\partial^2 \mathbf{P}_2}{\partial \alpha_i \partial \alpha_j}$$

$$\frac{\partial \mathbf{P}}{\partial \alpha_i} = (\frac{1}{2\lambda}\mathbf{M} - \frac{1}{8\lambda^3}\mathbf{M}\mathbf{c}\mathbf{c}^T\mathbf{M})\frac{\partial \mathbf{c}}{\partial \alpha_i}$$

$$\frac{\partial^2 \mathbf{P}}{\partial \alpha_i \partial \alpha_j} = (-\frac{1}{8\lambda^3}\mathbf{M} + \frac{3}{32\lambda^5}\mathbf{M}\mathbf{c}\mathbf{c}^T\mathbf{M})\mathbf{c}^T\mathbf{M}\frac{\partial \mathbf{c}}{\partial \alpha_j}\frac{\partial \mathbf{c}}{\partial \alpha_i} - \frac{1}{8\lambda^3}[(\mathbf{c}^T\mathbf{M}\frac{\partial \mathbf{c}}{\partial \alpha_i})\mathbf{M} + \mathbf{M}\mathbf{c}(\frac{\partial \mathbf{c}}{\partial \alpha_i})^T\mathbf{M}]\frac{\partial \mathbf{c}}{\partial \alpha_j} + (\frac{1}{2\lambda}\mathbf{M} - \frac{1}{8\lambda^3}\mathbf{M}\mathbf{c}\mathbf{c}^T\mathbf{M})\frac{\partial^2 \mathbf{c}}{\partial \alpha_i \partial \alpha_j}$$

$$\varepsilon : \frac{x^2}{r_1^2} + \frac{y^2}{r_2^2} + \frac{z^2}{r_3^2} = 1$$

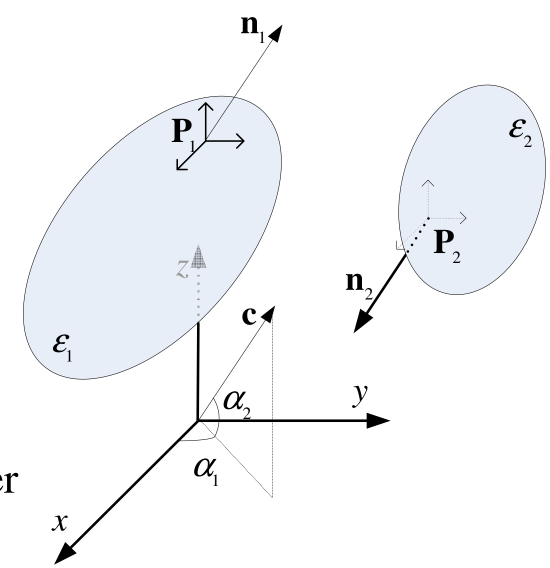
$\mathbf{A}$: Rotation Matrix

$\mathbf{M} = \mathbf{A}\mathbf{R}^2\mathbf{A}^T$

$\mathbf{R} = diag(r_1, r_2, r_3)$

$\mathbf{b}$: Translation of ellipsoids center

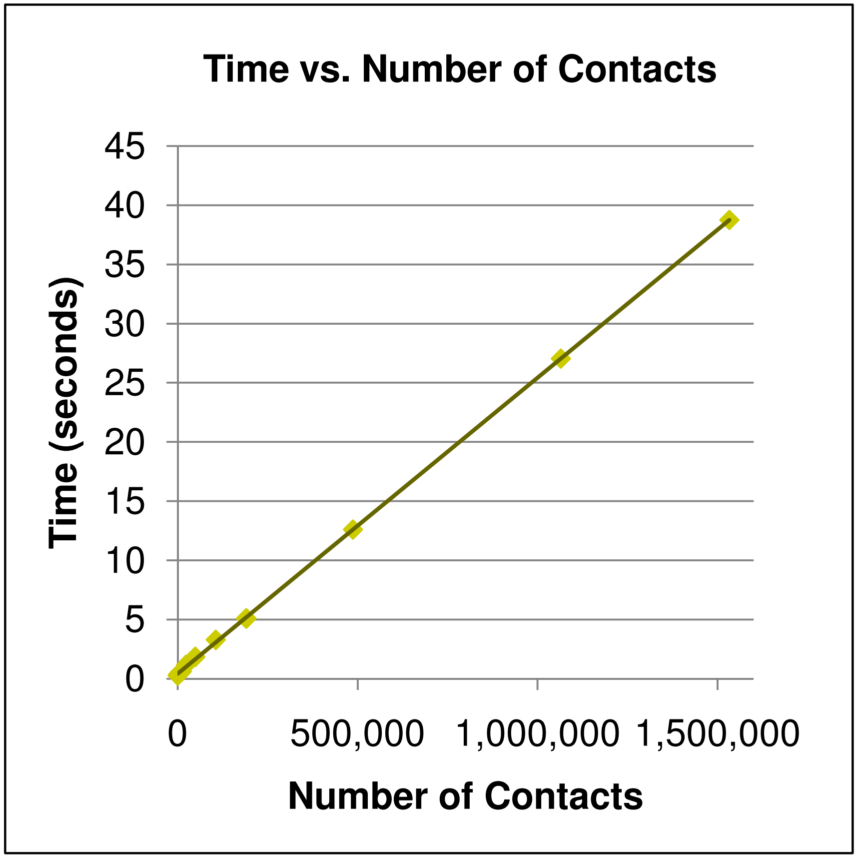
$$\lambda^2 = \frac{1}{4}\mathbf{n}^T\mathbf{M}\mathbf{n}$$

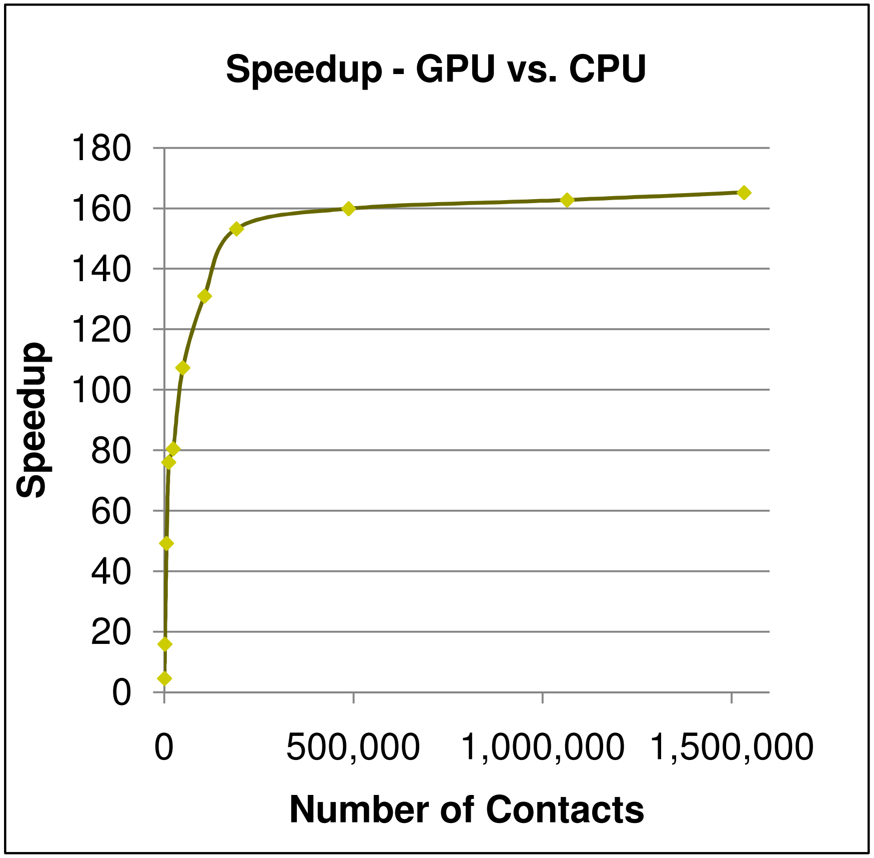


$$\mathbf{d} = \mathbf{P}_1 - \mathbf{P}_2$$

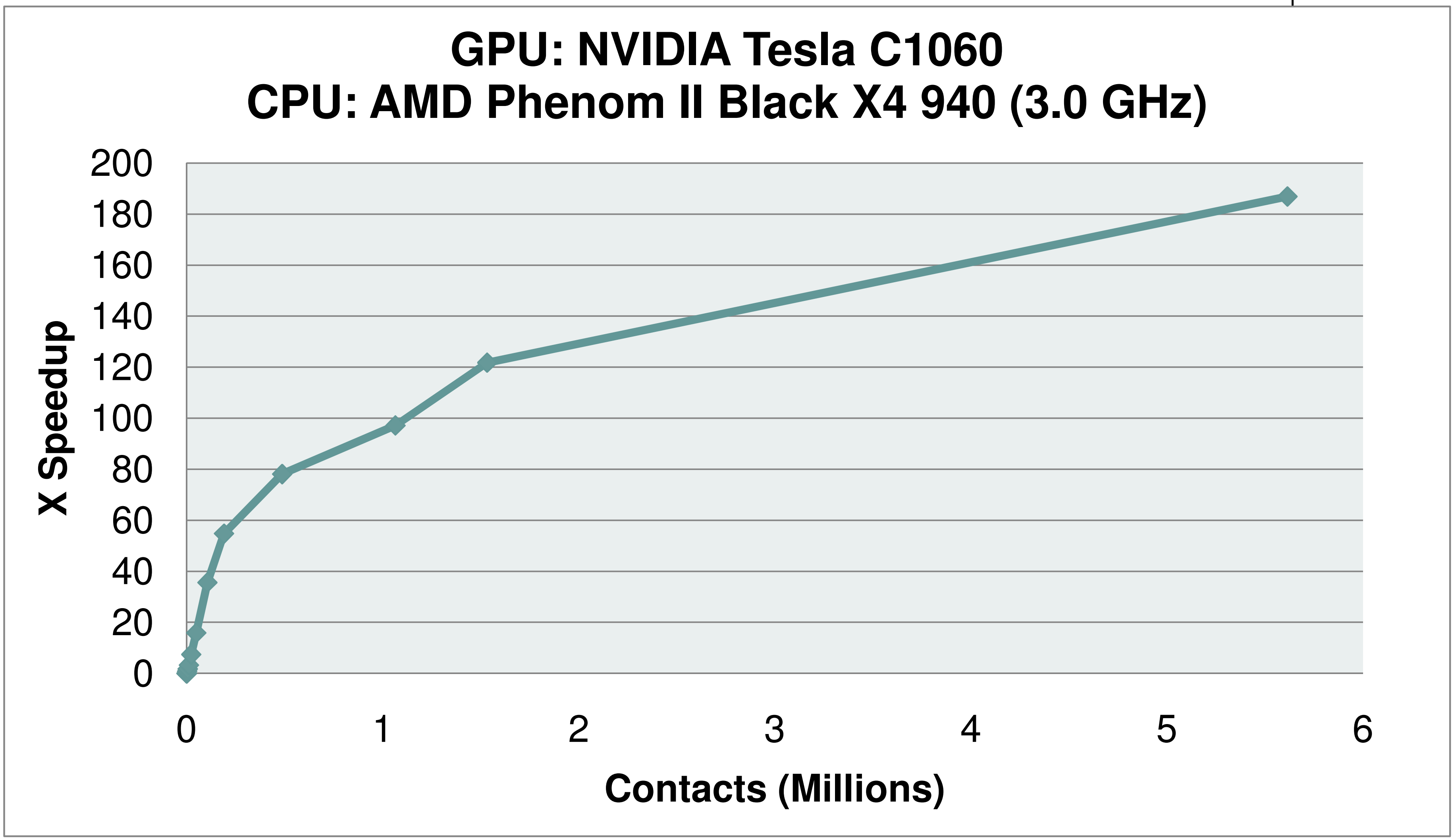
$$\min_{\alpha_1, \alpha_2} \| d(\alpha_1, \alpha_2) \|^2$$

Ellipsoid-Ellipsoid CD: Results

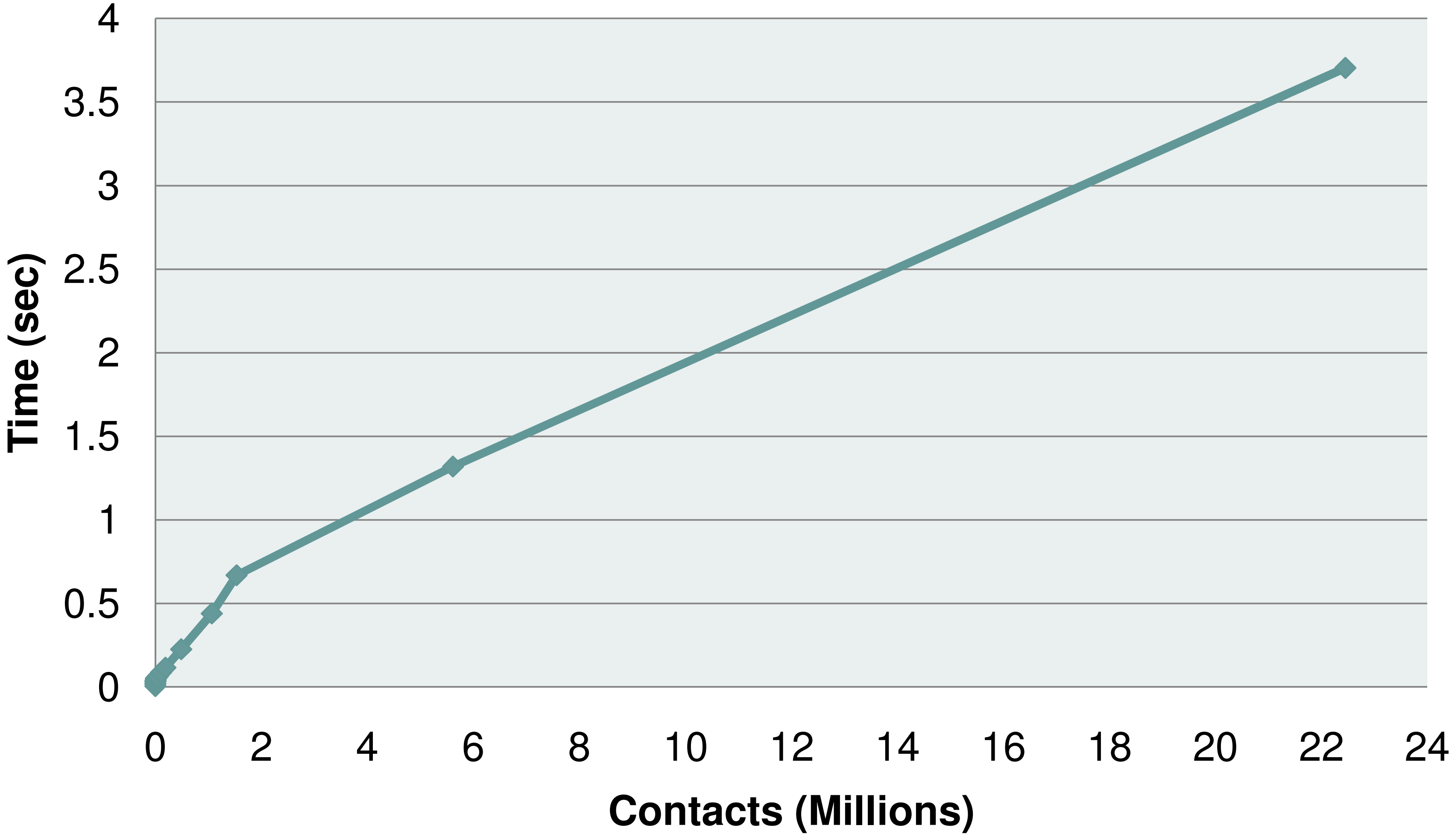




Speedup
GPU vs. CPU (sequential Bullet)
[results reported are for spheres]

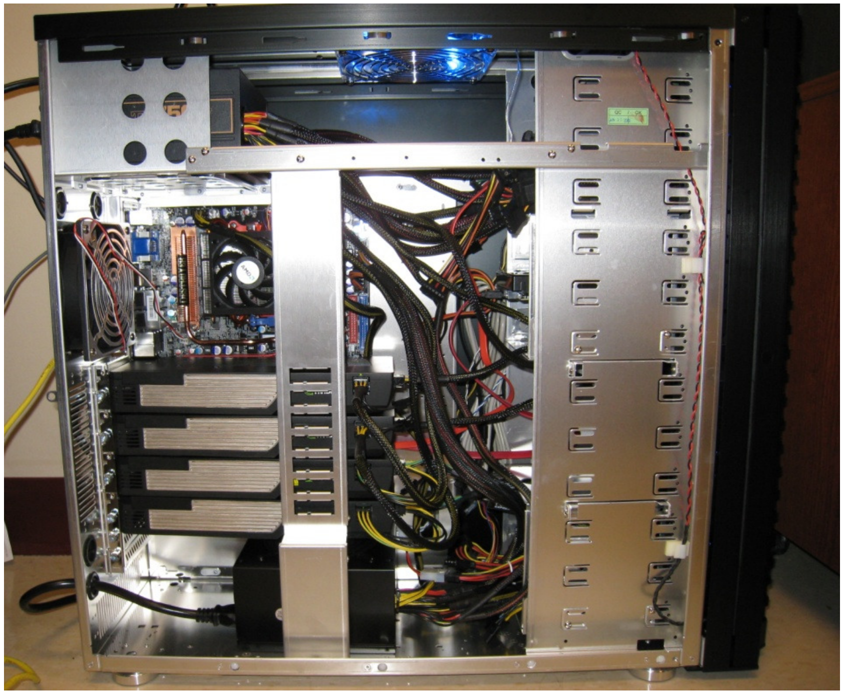


Parallel Implementation: Number of Contacts vs. Detection Time [results reported are for spheres]



Multiple-GPU Collision Detection

Assembled Quad GPU Machine



Processor: AMD Phenom II X4 940 Black

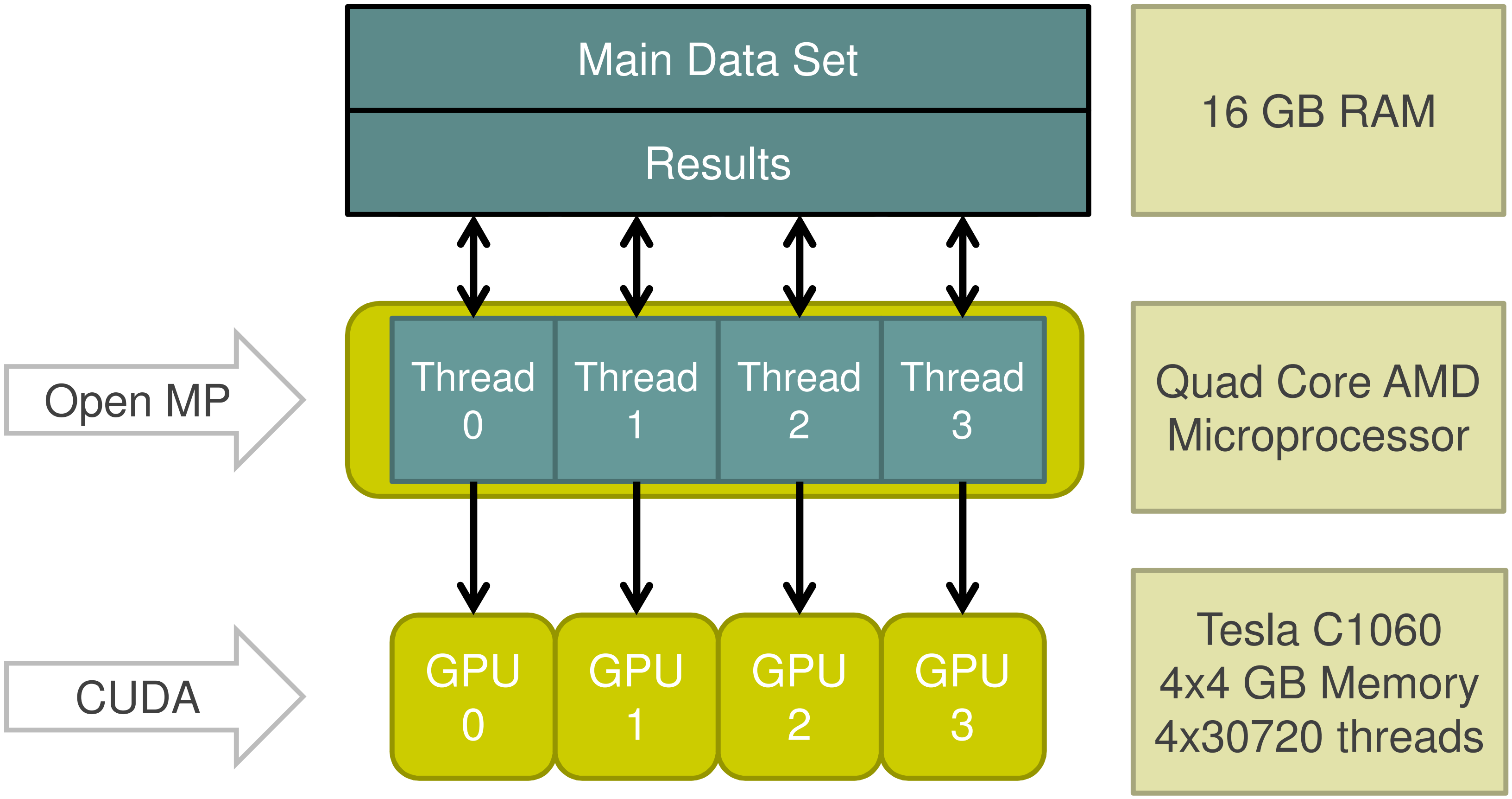
Memory: 16GB DDR2

Graphics: 4x NVIDIA Tesla C1060

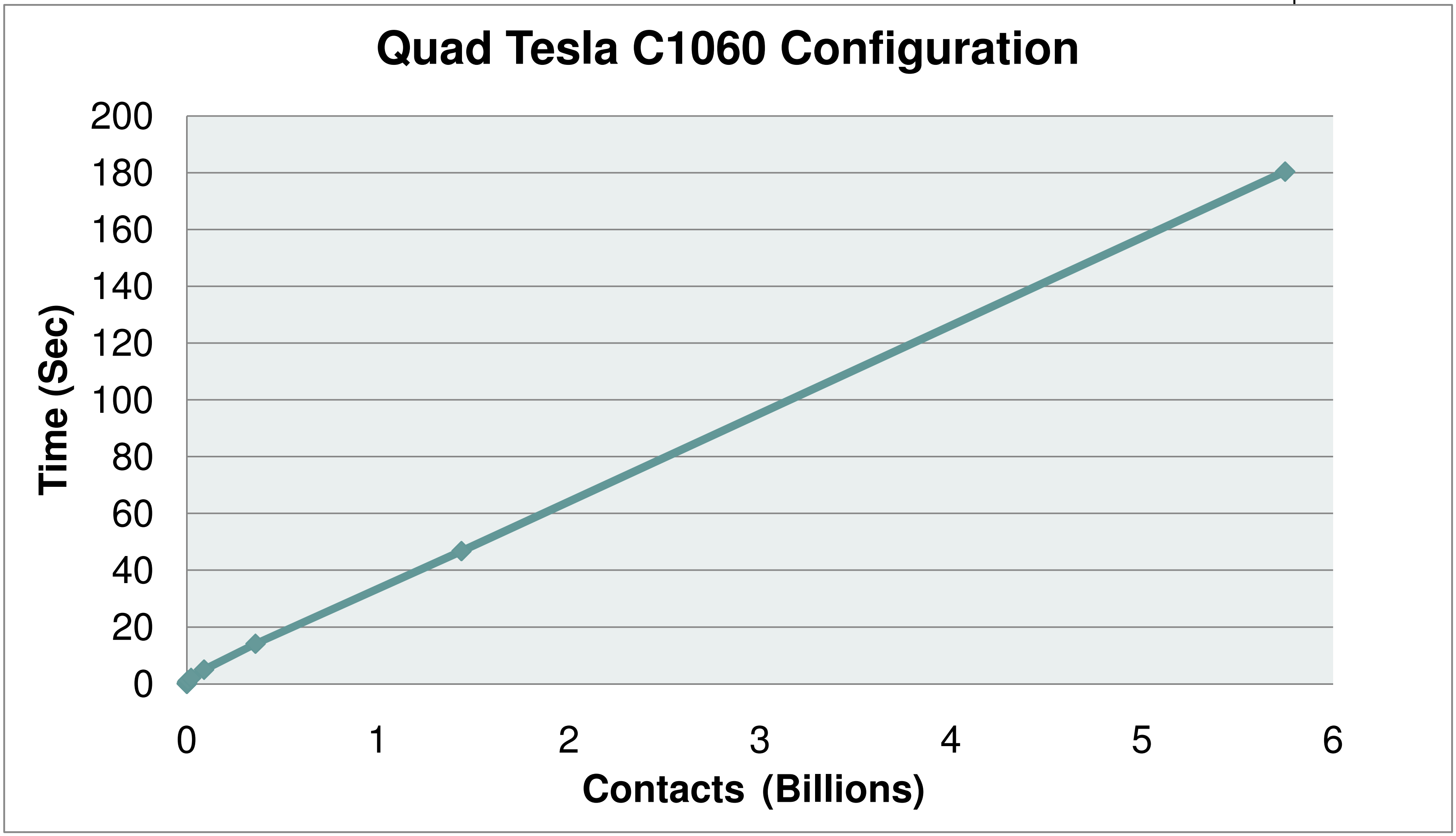
Power supply 1: 1000W

Power supply 2: 750W

SW/HW Setup


Main Data Set
Results
16 GB RAM
Open MP
Thread 0
Thread 1
Thread 2
Thread 3
Quad Core AMD Microprocessor
CUDA
GPU 0
GPU 1
GPU 2
GPU 3
Tesla C1060
4x4 GB Memory
4x30720 threads

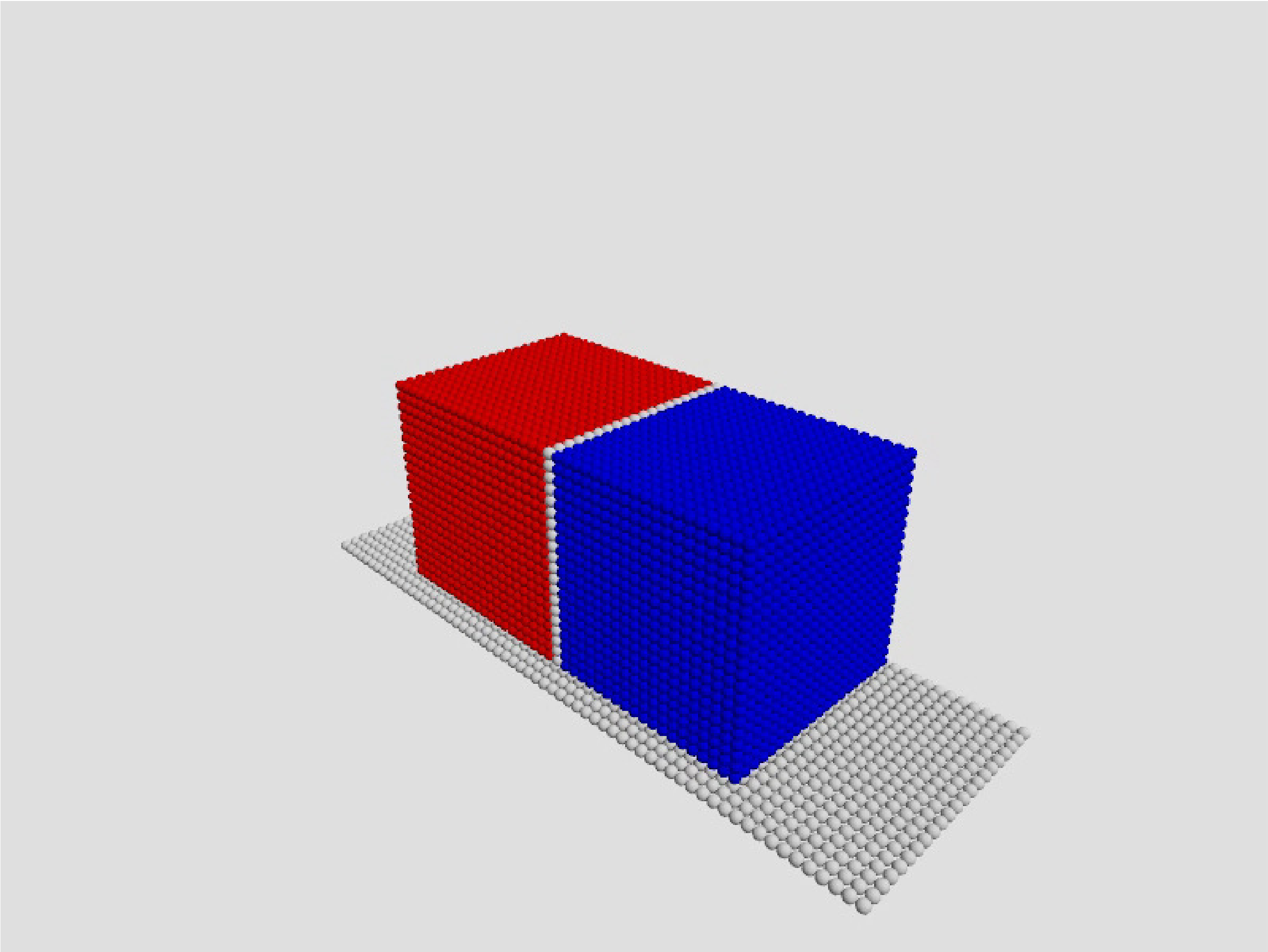
Results – Contacts vs. Time



HCT Demonstration

2 sub-domains, breakeven point is at 16,000 bodies

- CPU only: 9.58 hrs to reach steady-state
- CPU+GPU: 9.43 hrs to reach steady-state



Conclusions

- Work aimed at enabling high-fidelity discrete models using a physics-based approach
- HCT draws on symbiosis of CPU + GPU computing
- Accomplishments to date
  - Billion body parallel collision detection
  - Parallel solution of cone complementarity problem, about 12 million unknowns
  - Early validation results encouraging
- Aiming at billion bodies simulations

Ongoing/Future Work

- Experimental validation (three efforts, at CAT, US Army, and JPL)
- Massively parallel linear algebra for solution of CCP problem
  - Preconditioned gradient projected Krylov method
- Effective parallel collision detection algorithms for complex geometries
- Multiphysics:
  - Fluid-solid interaction
  - Electrostatics

Thank You.