On Distributed Gravitational N-Body Simulations

Alex Brandt

Department of Computer Science
University of Western Ontario, Canada

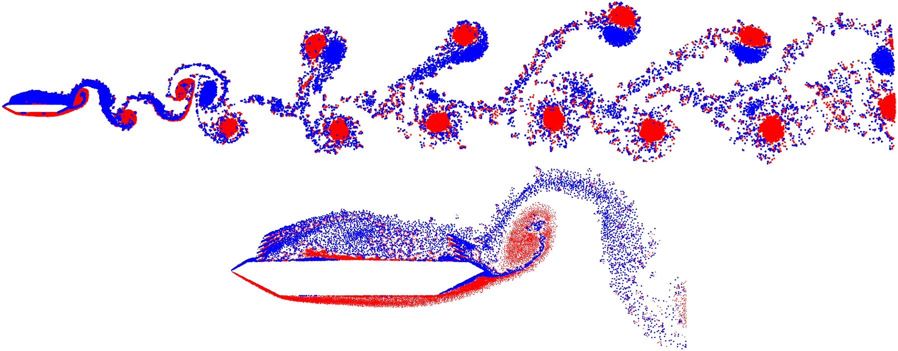
UWORCS 2021

Wednesday May 19, 2021

N-Body Simulations

Many discrete bodies moving by physical forces

- Molecular dynamics (van der Walls forces, electrostatic forces, etc.)
- Astrophysics (gravity, dark matter, general relativity)
- Fluid dynamics (Navier-Stokes equations)



Vortex Particle Method for Fluid dynamics, Ryatina and Lagno (2021) [6]

Gravitational N-Body

- Bodies moving under the force of gravity
- Exact solutions only known for $N \leqslant 3$
- Yet, $N = 10^4$–10^{11} is of practical importance
- Simulation is needed!

Three-Body Problem [9]

Solar System Simulation [2]

Need for Performance

- For N bodies, gravity simulation requires computing up to N^2 forces *at each time step*
- Hierarchical approximation methods introduced in the 1980s (Barnes-Hut [1], Fast Multipole [5]) reduce num. forces to $N \log N$
- Even with approximation methods, long-term simulations still run for days, weeks, or even **months**
- Parallel shared memory methods, and eventually, distributed computing methods are needed to obtain better performance

ACM/IEEE Super Computing Conference 1993

One conference, two pioneering parallel hierarchical N-body methods:

1. **Costzones** by Singh, Holt, Hennessy, and Gupta [7]
   ↳ Each process builds a local octree and merges them into a global one in a shared-memory system
2. **Hashed Octree** by Warren and Salmon [8]
   ↳ Each process builds a *locally-essential* octree in a distributed-memory system

Problems:

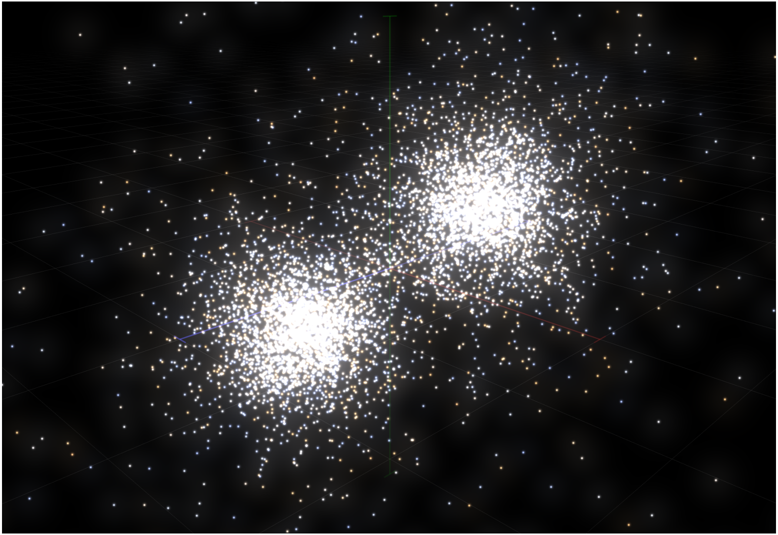
1. Code is unpublished
2. Details are missing to implement and reproduce

Goals of This Work

1. **Pedagogical Resource**
   - ↳ Consolidate information from SC '93, their follow-up papers, and continuing research, to create a coherent text
   - ↳ Include previously missing algorithmic and implementation details
2. **Reproduce**
   - ↳ Provide a well-documented, open-source code
3. **Adapt to Modern Hardware**
   - ↳ How well do these methods adapt to modern hardware, communication protocols, and hybrid distributed-shared memory systems?

Goals of This Work

4. Have fun!



Plan

1 Background: Gravity, Barnes-Hut

2 Octrees Encodings

3 Spacial Decomposition for Parallel Processing

4 Experimental Results

Newtonian Gravity

In Newtonian dynamics, a point mass m_i at $\boldsymbol{r}_i$ evolves as:

$$\frac{d^2}{dt^2}\boldsymbol{r}_i = \boldsymbol{a}_i$$

Let $\boldsymbol{F}_{ij}$ be the force acting on mass m_i by m_j. From $\boldsymbol{F} = m\boldsymbol{a}$:

$$\frac{d^2}{dt^2}\boldsymbol{r}_i = \frac{1}{m_i}\sum_{\substack{j=1 \\ j\neq i}}^{N} \boldsymbol{F}_{ij} = \frac{-1}{m_i}\sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{Gm_im_j(\boldsymbol{r}_i - \boldsymbol{r}_j)}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^3}$$

The force acting on m_i by a collection of bodies J can be approximated by their center of mass $\boldsymbol{r}_J$:

$$\boldsymbol{F}_{iJ} = \sum_{j\in J} \frac{-Gm_im_j(\boldsymbol{r}_i - \boldsymbol{r}_j)}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^3} \approx \frac{-Gm_im_J(\boldsymbol{r}_i - \boldsymbol{r}_J)}{\|\boldsymbol{r}_i - \boldsymbol{r}_J\|^3}$$

The Barnes-Hut Method

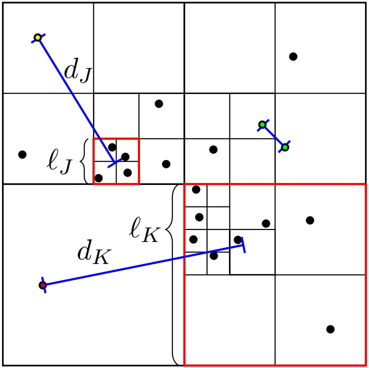
The Barnes-Hut method relies on two key observations:

1. The centre of mass approximation approaches equality as $d_J = ||\boldsymbol{r}_i - \boldsymbol{r}_J||$ increases
2. An octree (quadtree) hierarchically groups bodies to easily determine groups J for which computing $\boldsymbol{F}_{iJ}$ is sufficient

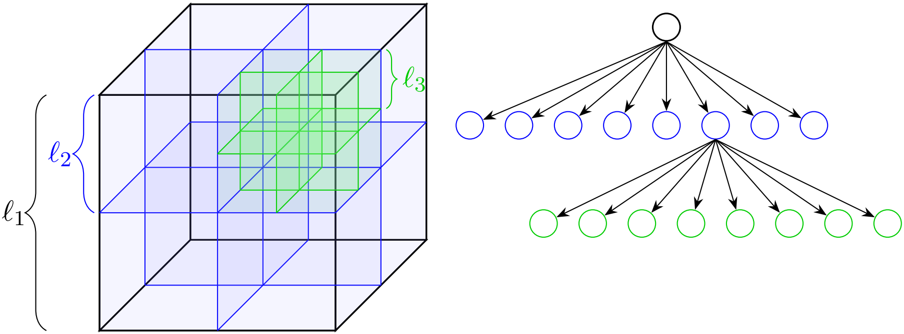
For each m_i :

If $\frac{\ell_J}{d_J} < \theta$, use F_{iJ}. Otherwise, recurse.

- Yellow is *well-separated* from J
- Purple might be well-separated from K
- Force between greens cannot be approximated



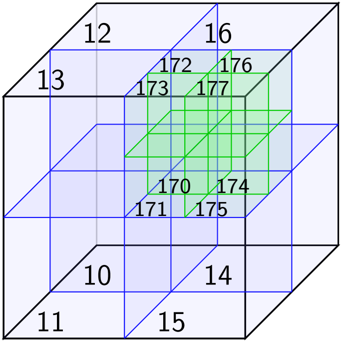
Octrees

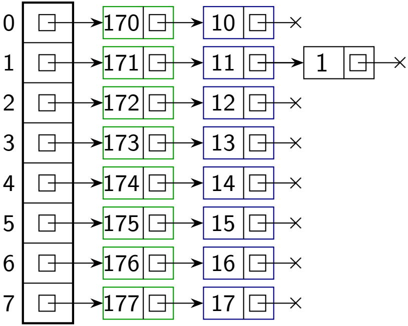


A binary tree based on pointers can easily be extended to an octree:

- 8 children per node
- Augment each node with its spatial information:
  its center and side length ℓ_i
- Divide the space until each body resides in a unique leaf node

Linear Octree (Hashed Octree)

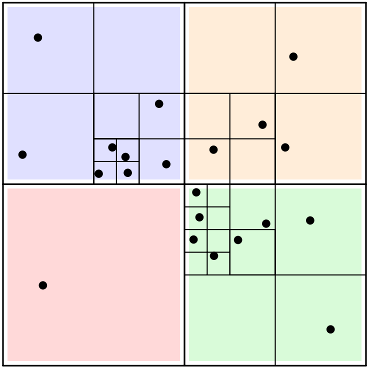




A **linear octree** (introduced by Gargantini, UWO Professor emeritus [4]) allows for direct indexing of any node via a hash table encoding.

- Each node gets a unique key: its parent's key concatenated with (0-8), in octal
- Root node gets key 1

The Need For Spatial Decomposition



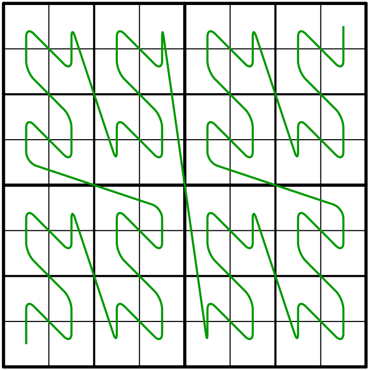
Goal: partition the simulation domain and assign one partition to each process (thread, processor)

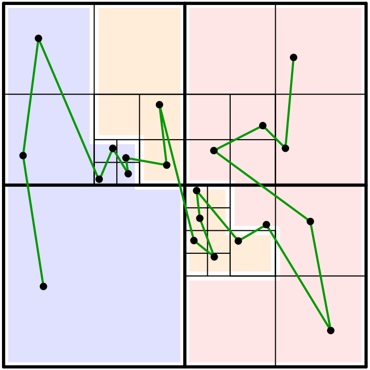
Naively dividing the simulation domain into equal-sized chunks is insufficient for *load-balancing*.

N-order, Z-order, Morton Order

A space-filling curve gives a linear order to 2D or 3D space.

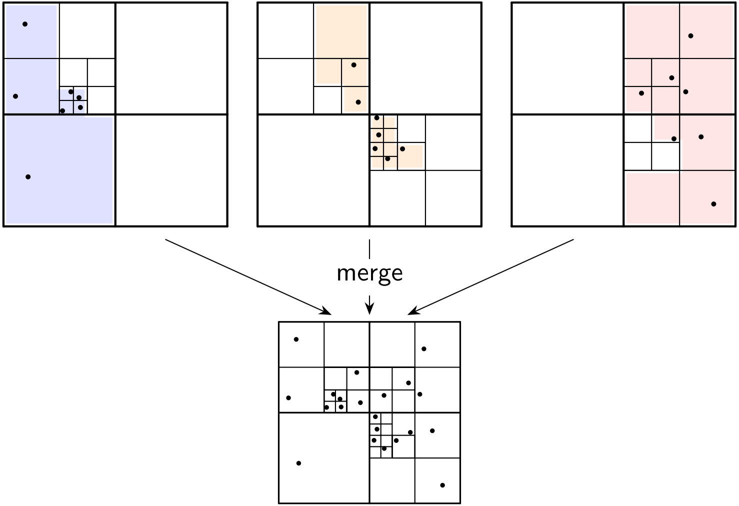
Linearize the bodies, partition space so that each partition has an equal number of bodies.





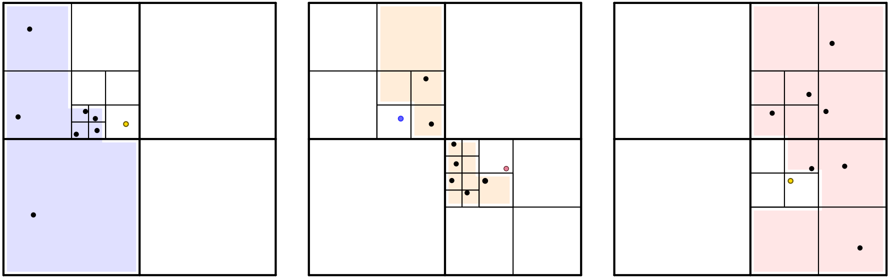
Costzones Method

- Each processor builds a local, pointer-based octree for its partition.
- Merge local trees into a single global, shared tree.



Hashed Octree Method

- Each processor builds a local, hashed octree for its partition.
- **Neighbour bodies** are shared between adjacent domains to act as entry points to remote trees.
- During force calculation, remote data is dynamically requested as needed during recursive calls.

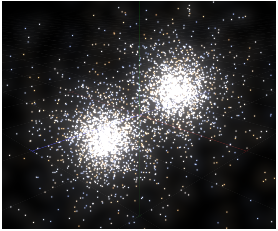


Implementation and Experimental Setup

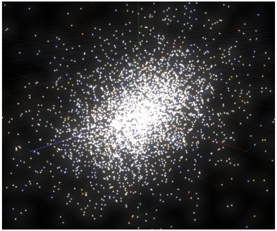
Code, technical report: https://github.com/alexgbrandt/Parallel-NBody

- Code written in C, visualization implemented in OpenGL 3.3
- Parallelization and distributed computing by OpenMPI [3]
- 10-node LAN compute cluster, each with 2x 6-core Intel Xeon X5650

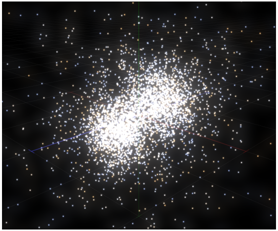
Experiment: collision of two globular clusters



$t = 0.0$

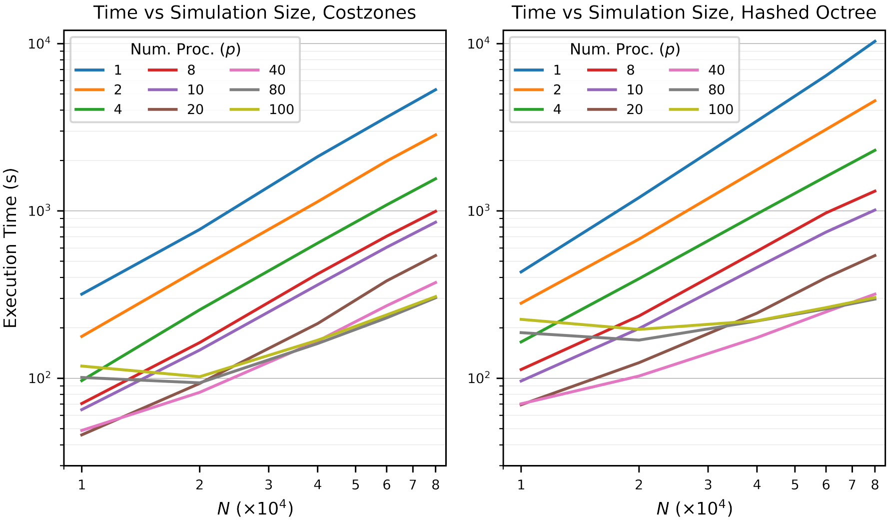


$t = 2.0$

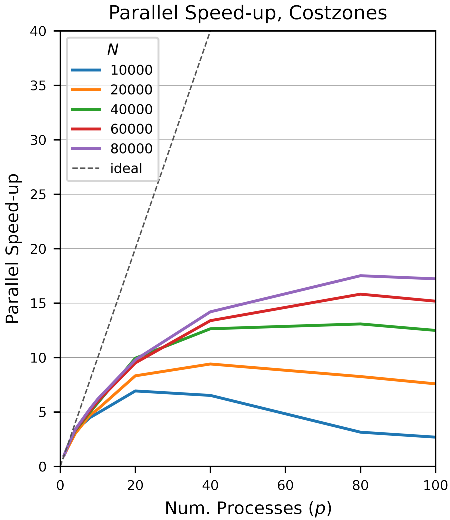


$t = 4.0$

Execution Time



Parallel Speedup



References

[1] J. Barnes and P. Hut. "A hierarchical O(N log N) force-calculation algorithm". In: *Nature* 324.6096 (Dec. 1986), pp. 446–449.

[2] P. Connolly. *Solar System Simulation in MATLAB*. 2015. URL: https://personalpages.manchester.ac.uk/staff/paul.connolly/teaching/practicals/solar%5C_system.html.

[3] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation". In: *Proceedings, 11th European PVM/MPI Users' Group Meeting*. Budapest, Hungary, Sept. 2004, pp. 97–104.

[4] I. Gargantini. "An Effective Way to Represent Quadtrees". In: *Commun. ACM* 25.12 (1982), pp. 905–910. DOI: 10.1145/358728.358741. URL: https://doi.org/10.1145/358728.358741.

[5] L. Greengard and V. Rokhlin. "A fast algorithm for particle simulations". In: *Journal of Computational Physics* 73.2 (1987), pp. 325–348.

[6] E. Ryatina and A. Lagno. "The Barnes—Hut-type algorithm in 2D Lagrangian vortex particle methods". In: *Journal of Physics: Conference Series*. Vol. 1715. 1. IOP Publishing. 2021, p. 012069.

[7] J. P. Singh, C. Holt, J. L. Hennessy, and A. Gupta. "A parallel adaptive fast multipole method". In: *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*. 1993, pp. 54–65.

[8] M. S. Warren and J. K. Salmon. "A Parallel Hashed Oct-Tree N-Body Algorithm". In: *Proceedings of the 1993 ACM/IEEE Conference on Supercomputing*. Supercomputing '93. Portland, Oregon, USA: Association for Computing Machinery, 1993, pp. 12–21.

[9] Wikipedia. *Three-Body Problem*. 2021. URL: https://en.wikipedia.org/wiki/Three-body_problem.

Thank you!

Questions?