Solving N-body Problem via Barnes-Hut Algorithm

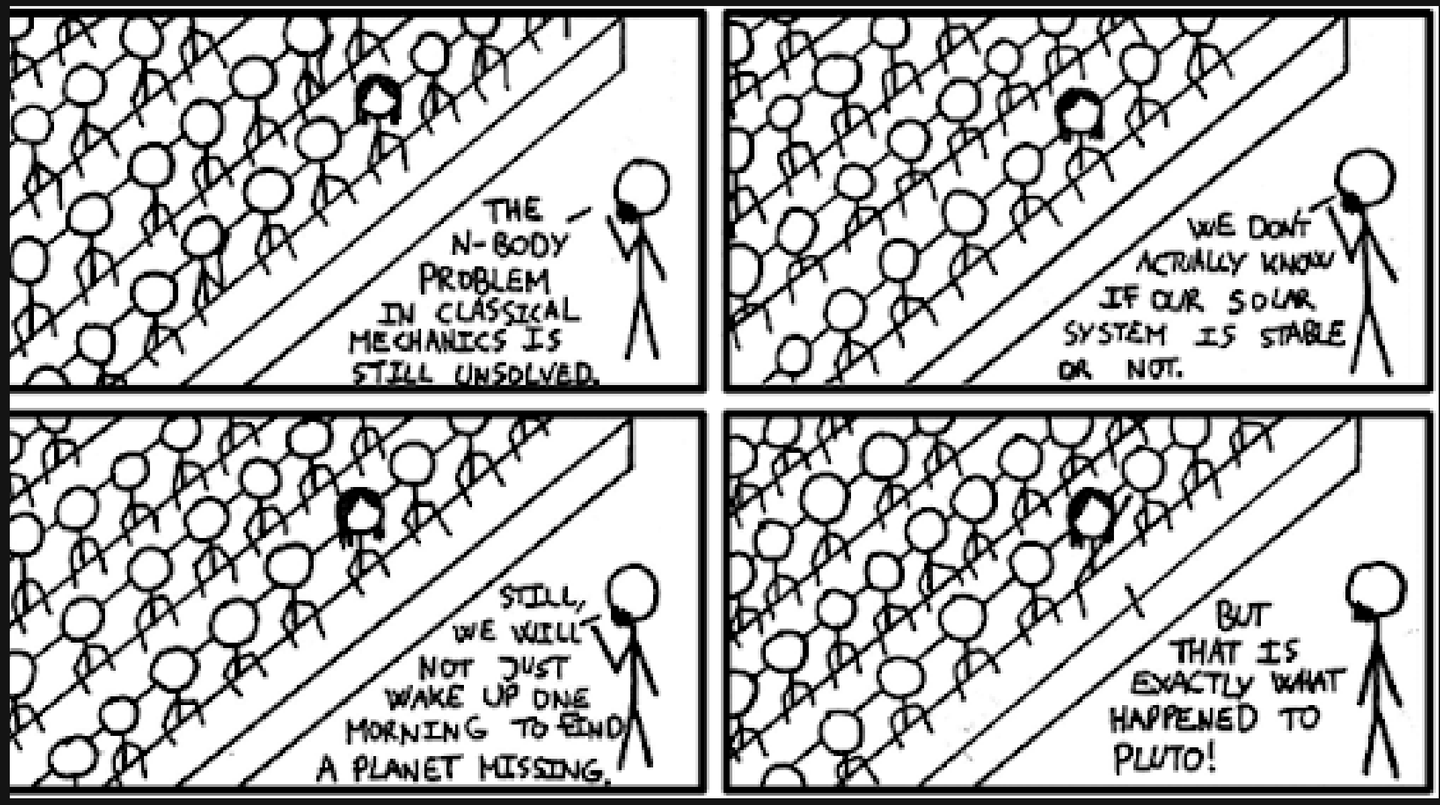
GPU ACCELERATED COMPUTING

SYED MUHAMMAD SUFFWAN

SYED MUHAMMAD HASAN HAIDER

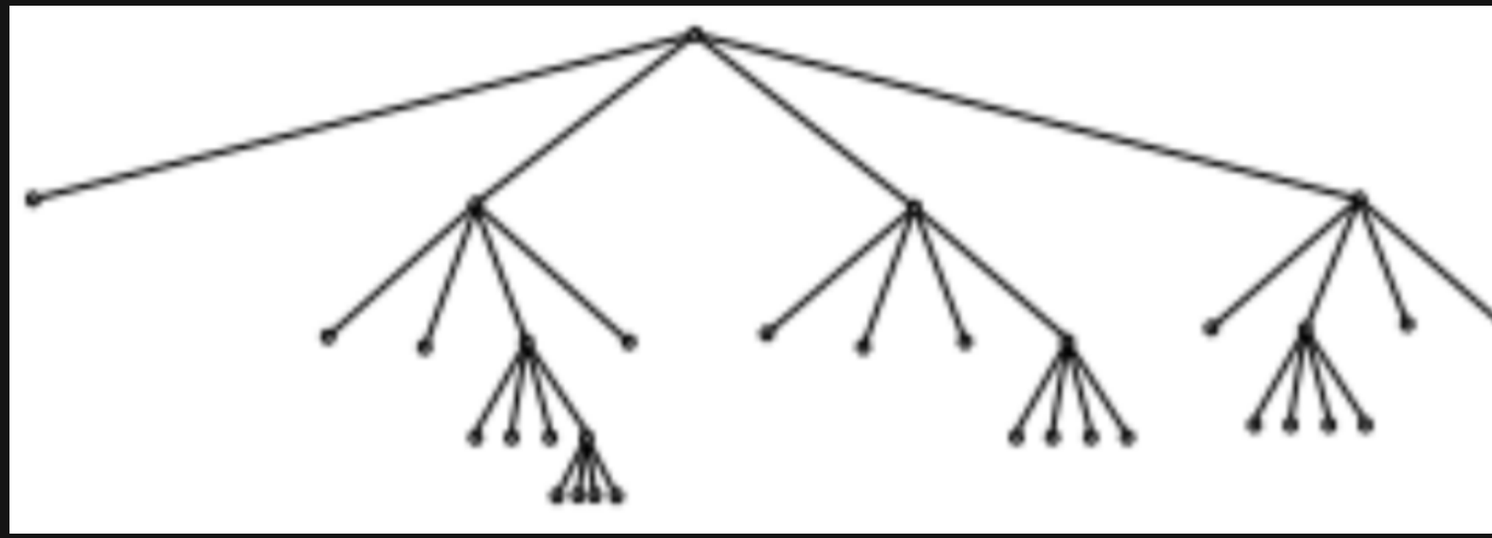
Abstract

- N-body problem is predicting the behaviour of a mass in presence of the other masses and the masses could range.
- The current computational cost of simulating of all-pairs interactions is O(n^2).
- We parallelised Barnes-hut algorithm for N-body problem using CUDA
- Computational cost brought down to O(n log n).
- All these approaches have been implemented out on NVIDIA Tesla K80



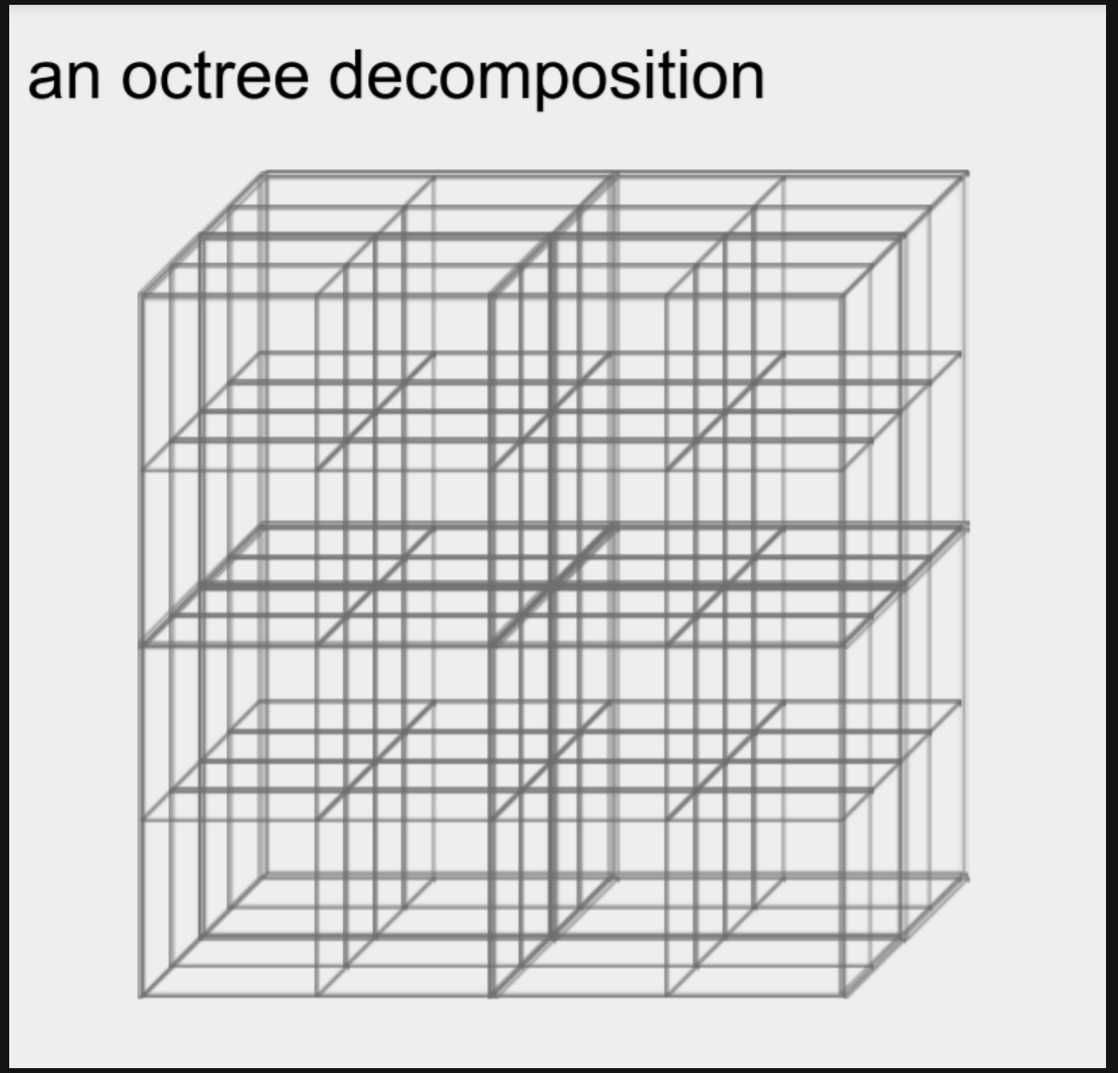
Application - Barnes-Hut Algorithm

- **Approximation algorithm for n-body simulation**
- **Approximate a particle's dependence on distant data**
- **We use approximations on their centre of mass and total mass with mass of each body is assumed to be constant**
- **It uses a oct-tree which is similar to a binary tree,except that each node has 8 children (some of which may be empty).**



Application - Barnes-Hut Algorithm

- **Recursively divides the n bodies into groups while storing in an octree (3D simulation)**
- **Each node in the tree is 3D space.**
- **Breakdown the space untill each division has a 1 or zero bodies.**
- **Internal Node: Represents group of bodies beneat it. Stores centre of mass of all its children**
- **External Node: No childern (at consist one body at max)**
- **We have implmented this algorithm in three different ways; Serial, Naive and Shared Memory.**

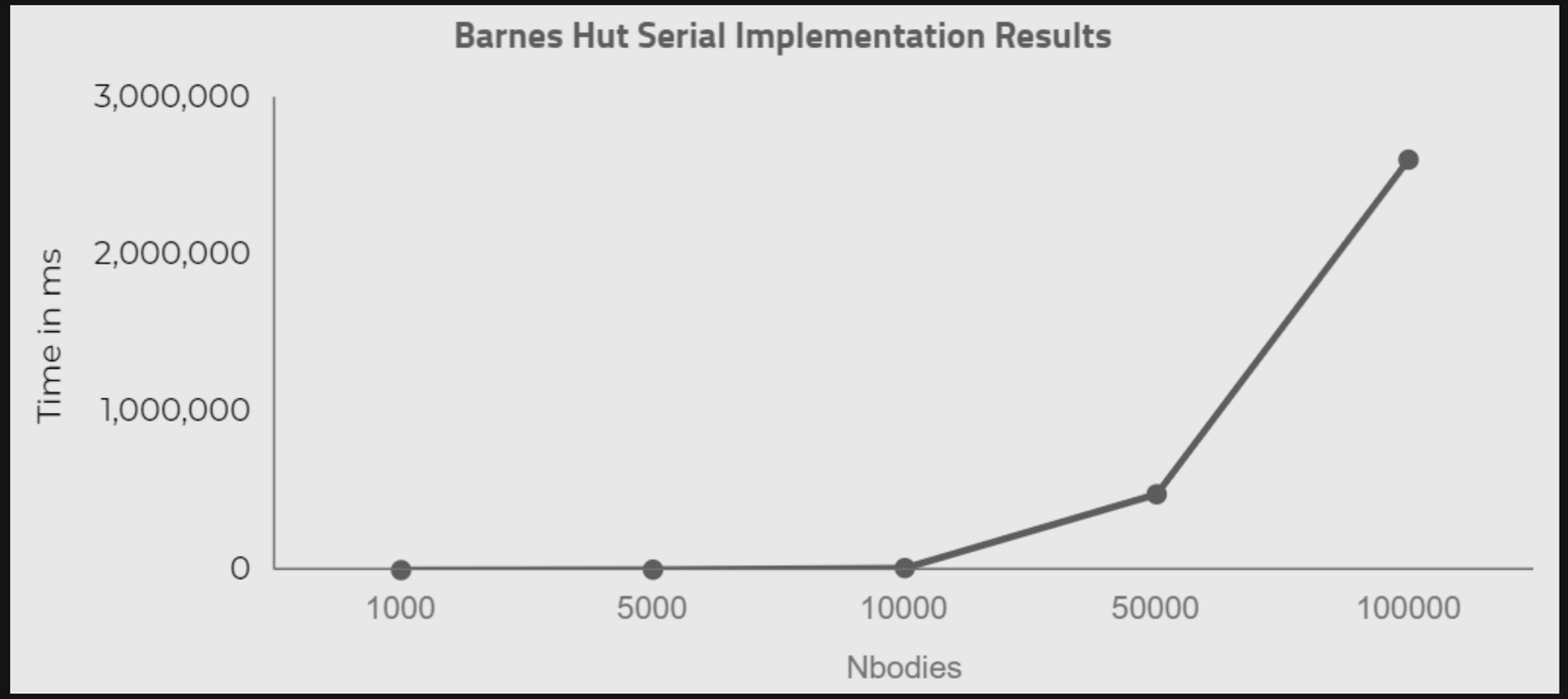


Serial Implementation Bottlenecks

BUILDING OCT TREE

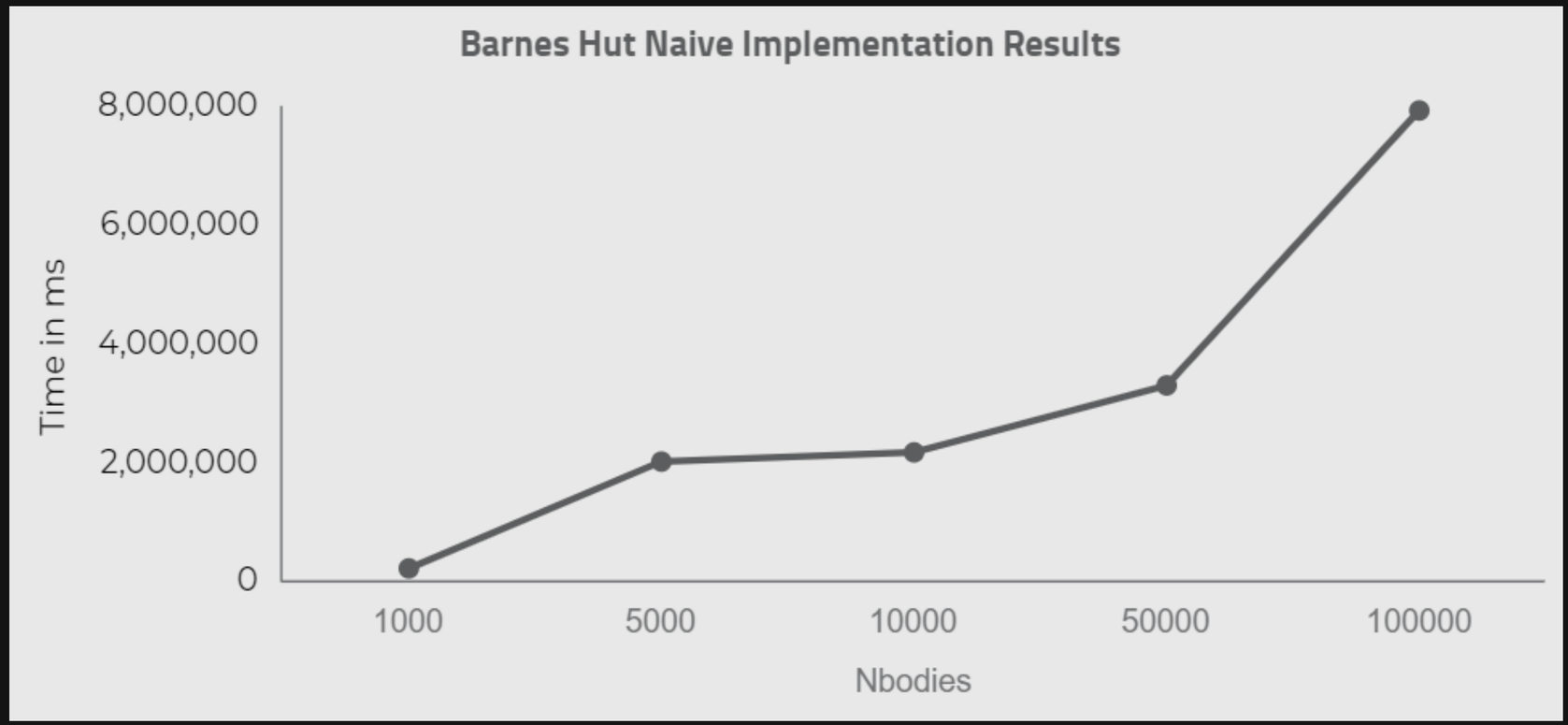
TREE TRAVERSAL IN FORCE COMPUTATION

Serial Implementation Performance



Naive Implementation Performance

UPDATING POSITIONS



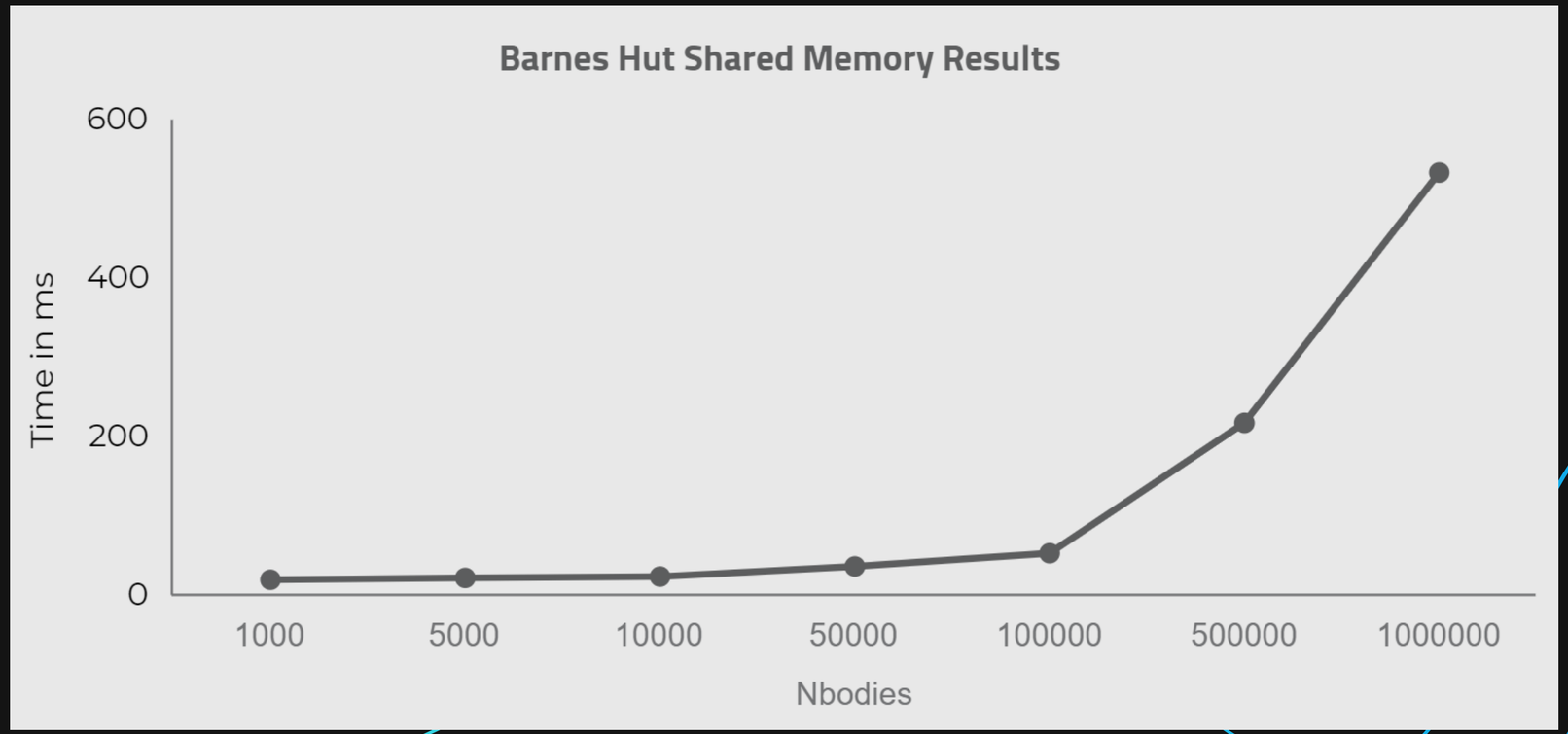
BARNES-HUT SHARED MEMORY EXPLANATION

EACH PROCESSOR FIRST BUILDS A LOCAL TREE; THESE ARE MERGED INTO A GLOBAL TREE STORED IN SHARED MEMORY. WORK IS EVENLY DISTRIBUTED AMONG PROCESSORS BY PARTITIONING THE BODIES.

EACH BLOCK HAS EQUAL CHUNK OF DATA

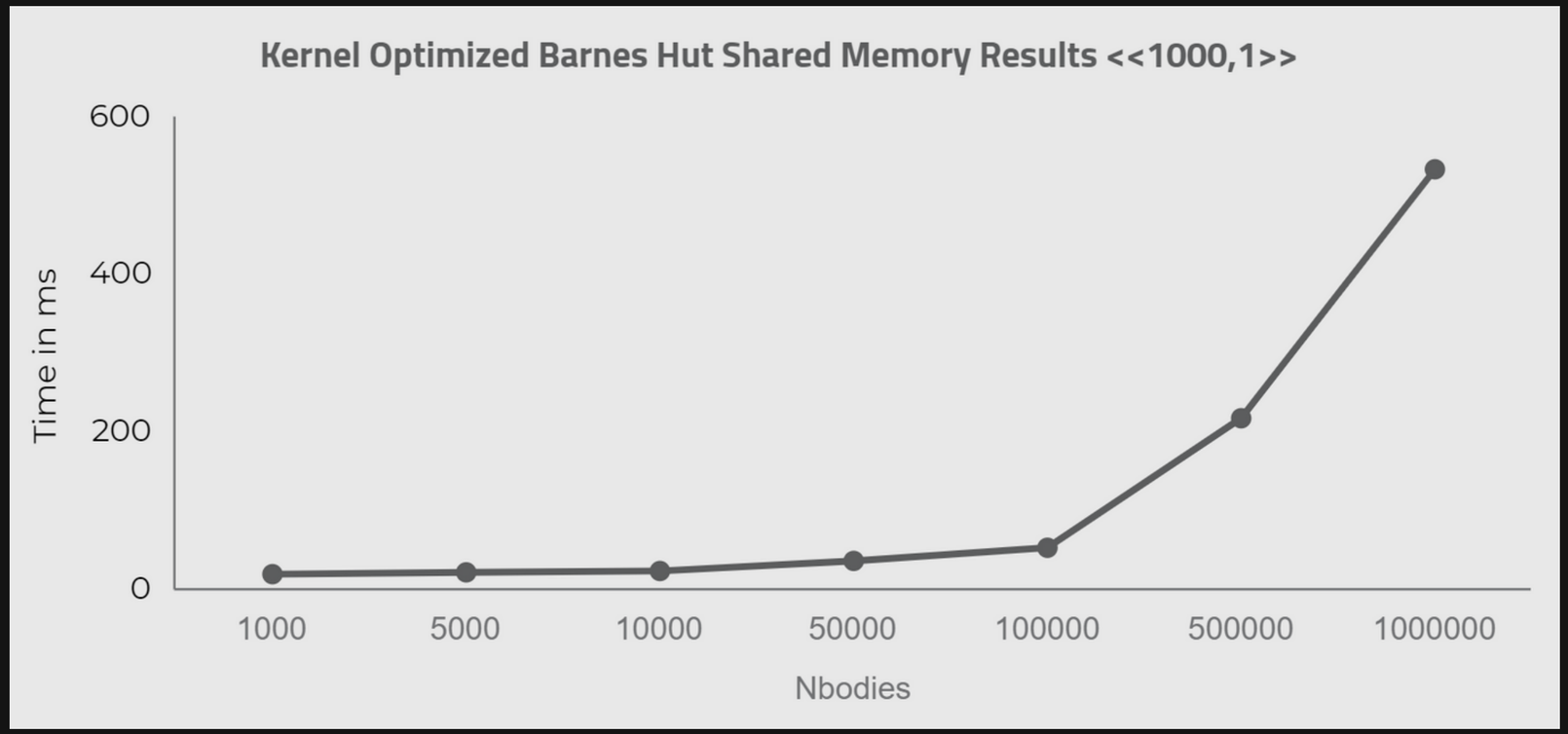
EACH BLOCK FINDS THE LOCAL MINIMA AND MAXIMA.

CUDA Shared Memory Implementation Performance



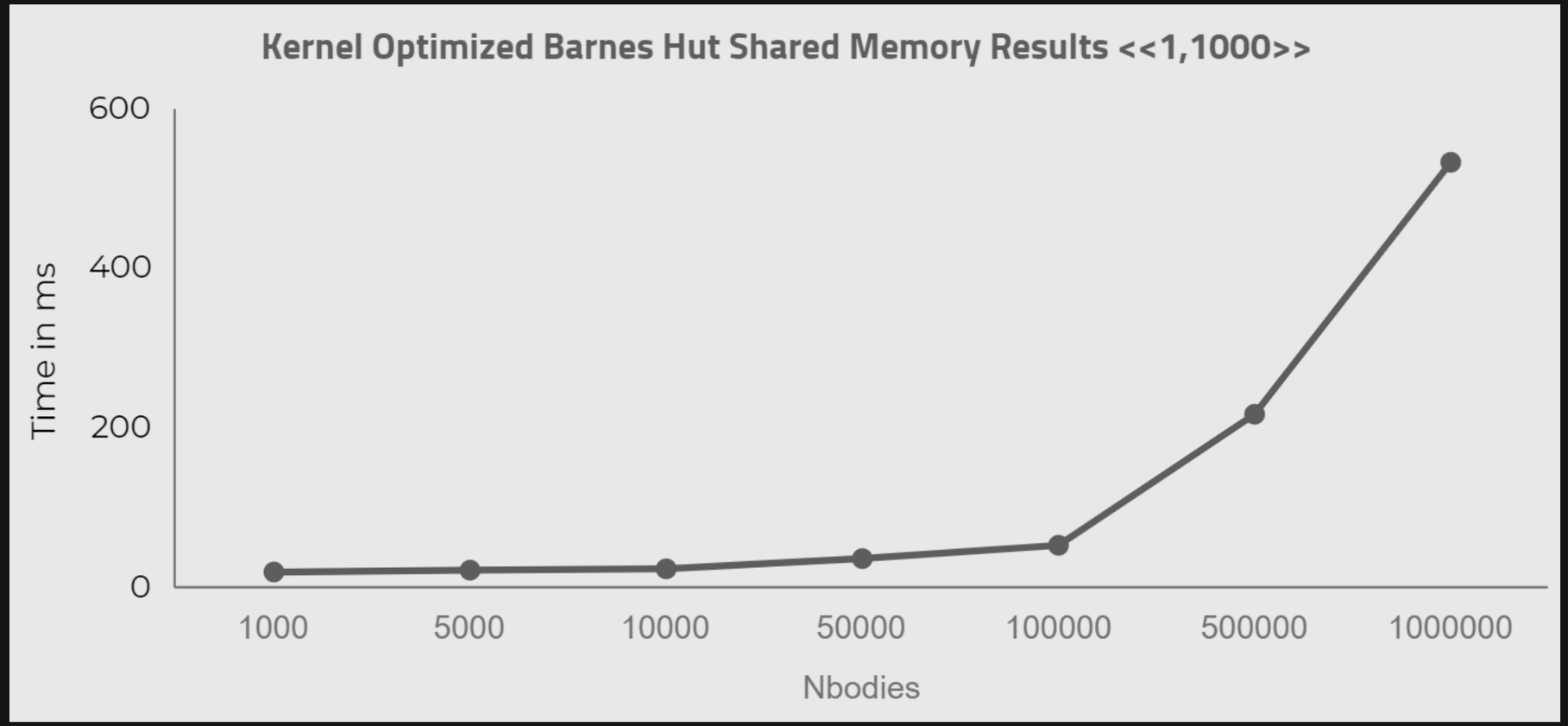
Applied CUDA Optimizations and Performance

INCREASING NUMBER OF BLOCKS FROM 1 TO 1000



Applied CUDA Optimizations and Performance

INCREASING NUMBER OF THREADS FROM 1 TO 1000



Conclusion

Shared Memory

Naive Parallel

Serial

Thank you!!