OpenCLによるN体計算の高速化

中里直人 (会津大学)

天文学会 2012年春季年会
2012年3月20日@龍谷大学

Agenda

- GPUによるHPCの利点
- 統一的な並列プログラミング手法としてのOpenCL
- OpenCLによるN体計算

粒子シミュレーションでの利点

- 天文の粒子シミュレーションの現状
  - MPP向けのコードが広く使われている
    - あまりスケーラブルではない
- GPUがあるとノードの演算性能が上がる
  - 与えられた問題を少ないノード数で実行できる
  - GRAPE clusterで実証済み
    - PPPM and TreePM Methods on GRAPE Systems (Yoshikawa&Fukushige 2005)
  - CPU+GPU向けのアルゴリズムの検討が必要

HPC Cluster Configuration

ノードの演算性能

CPU+GPUシステム
for Modest N problems
TSUBAME2, HA-PACS

ノード数(MPIプロセス数)を少なくできる

Big MPP cluster
for Large N problems
K computer

ノードの数

OpenCLとは

- 並列計算APIの標準規格
  - AMD, Apple, IBM, Intel, NVIDIA etc.がサポート
  - CPU, GPU, CellBE, DSP, FPGA(?) etc.で動作
- アクセラレータのプログラミングモデル
  - ホスト計算機から「デバイス」の機能を呼び出す
    - デバイスで動作するコードを「カーネル」と呼ぶ
    - CUDAと本質的な差はない
  - SIMD的な動作するスレッド群の並列化
  - 演算の明示的なベクトル化

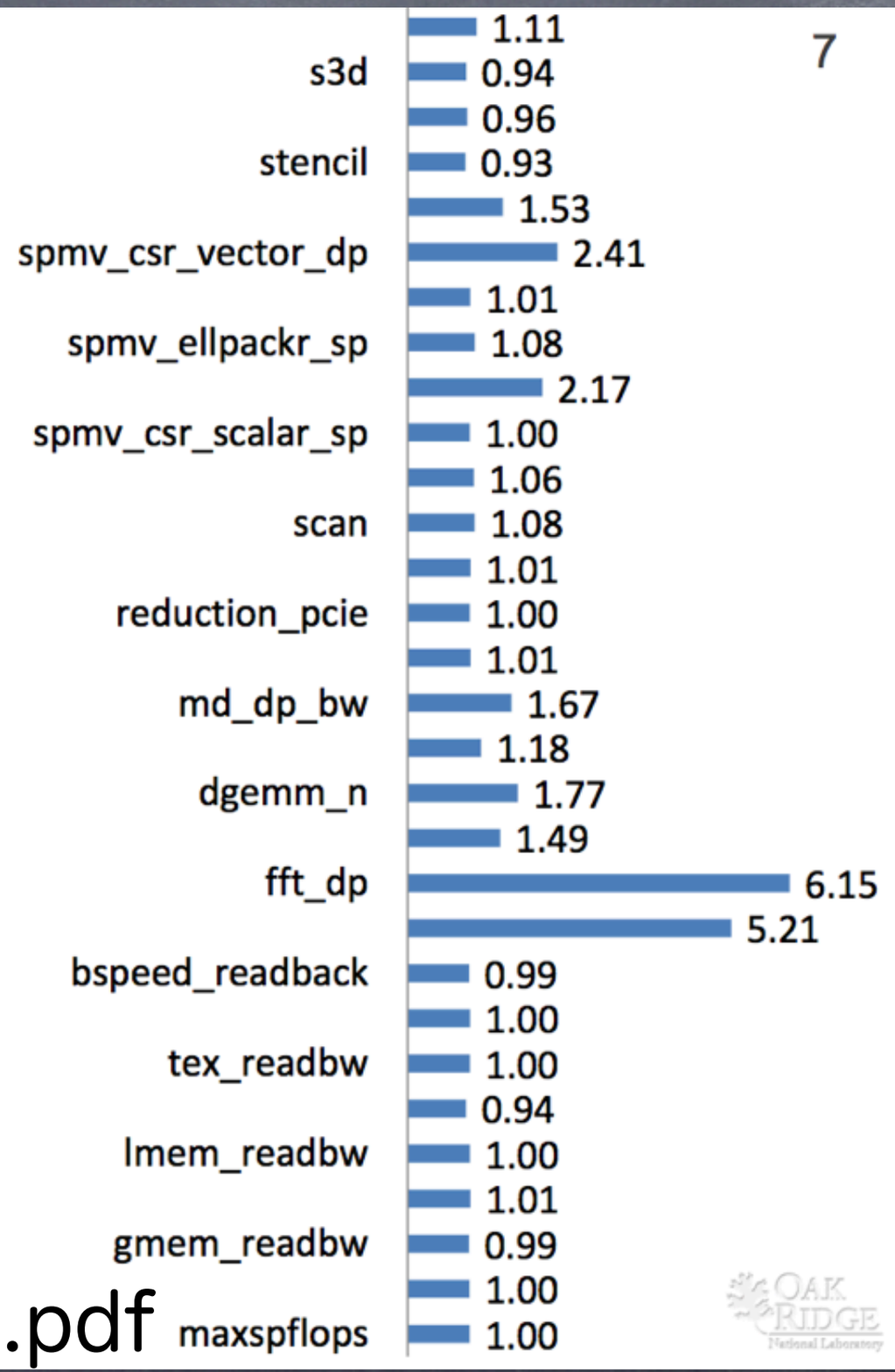
OpenCLの利点と欠点

- 利点
  - 各社のデバイスで動作する
  - CPUとGPUを同時扱うことが可能
    - 従来: SSEベクトル化 + OpenMP/pthread + MPI
    - OpenCL時代:
      - OpenCL(ベクトル変数 + スレッド) + MPI
      - (OpenCL+OpenMP) + MPI
- 欠点
  - より煩雑なプログラミング: C++ APIは簡単
  - NVIDIAのデバイスでは性能が落ちる場合あり

OpenCLとCUDAの性能比較

CUDA and OpenCL

- What does performance look like today?
- This chart shows the speedup of CUDA over OpenCL on a single Tesla M2070 on KIDS (CUDA 4.0)
- Note that performance is (in most cases, close to equivalent)
- Cases where it's not tend to be related to texture memory or transcendental intrinsics



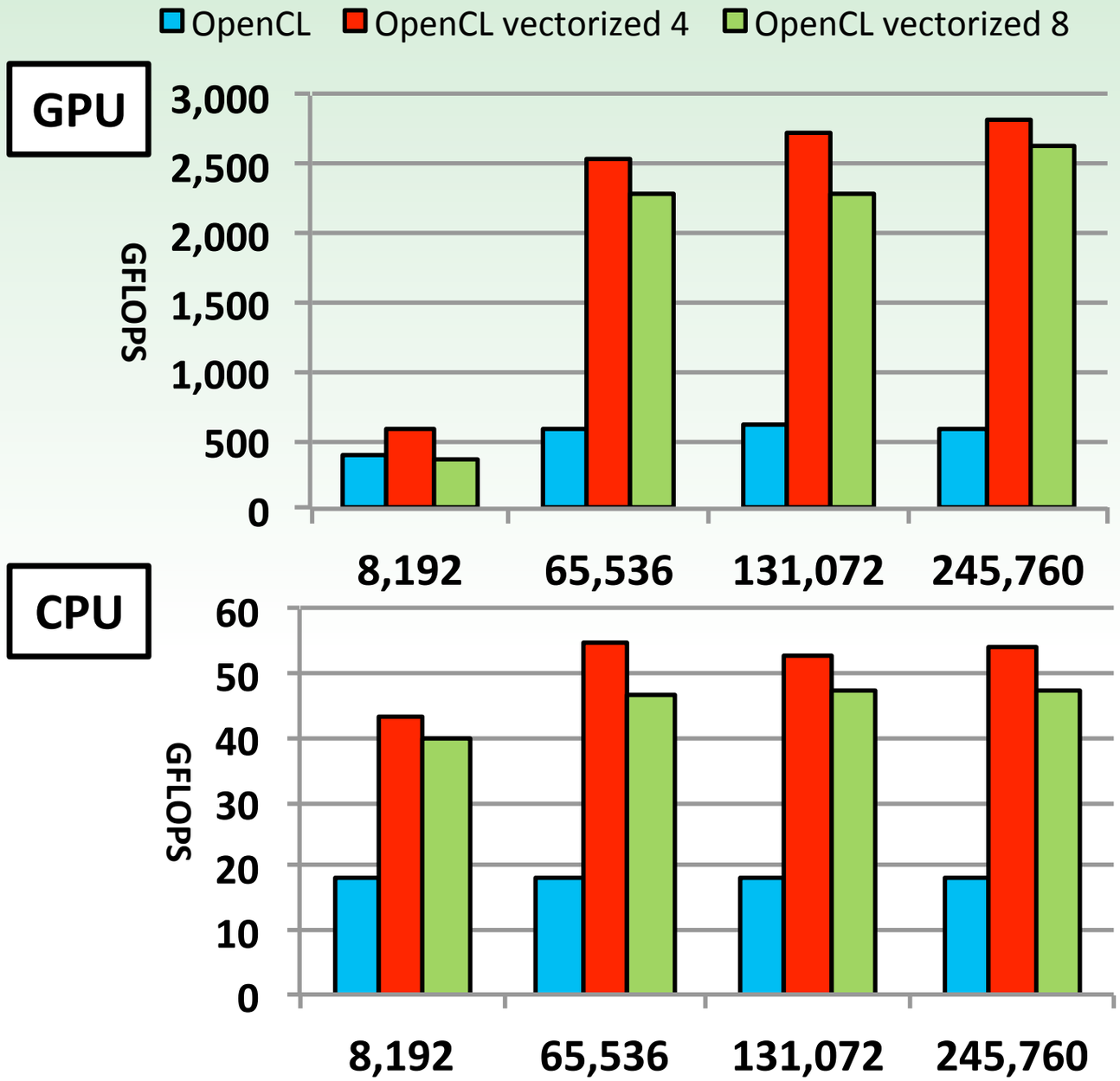
from Spafford 13-shoc.pdf

http://keeneland.gatech.edu/2012-02-20-workshop

OpenCLの評価:N-Body $O(N^2)$ (1)

2011年度卒業研究 鈴木Y

GPUとCPUの性能比較・及びベクトル化の有効性



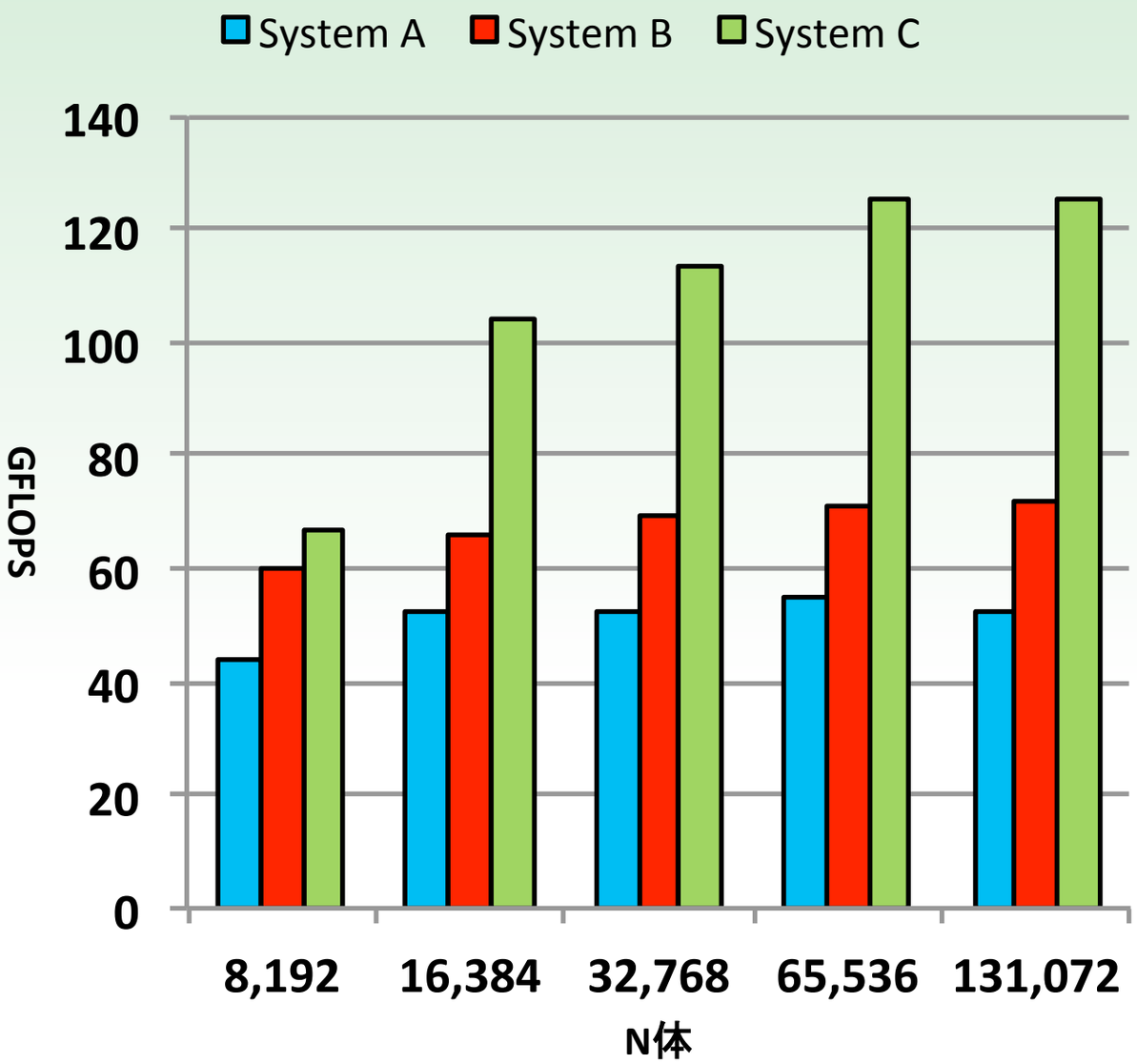
	System A
Maker	AMD
CPU	Phenom II X6 1090T
N_{core}	6
N_{thread}	6
clock	3.2 GHz
memory	4GB
SP(SSE)	153.6 GFLOPS
GPU	HD5870
SSE of GPU	2.72 TFLOPS

理論性能と出力性能の比較:

	OpenCL	OpenCL vectorized 4	OpenCL vectorized 8
GPU	0.21	0.99	0.97
CPU	0.12	0.36	0.31

OpenCLの評価:N-Body $O(N^2)$ (2)

3つの異なるマシンにおいての性能比較



	System A	System B	System C
Maker	AMD	Intel	AMD
CPU	Phenom II X6 1090T	Intel Core i7-2600K	Opteron Processor 6168 x 2
N_{core}	6	4	24
N_{thread}	6	8	24
clock	3.2 GHz	3.4 GHz	1.9 GHz
memory	4GB	16GB	32GB
SP(SSE)	153.6 GFLOPS	108.8/217.6 GFLOPS	364.8 GFLOPS

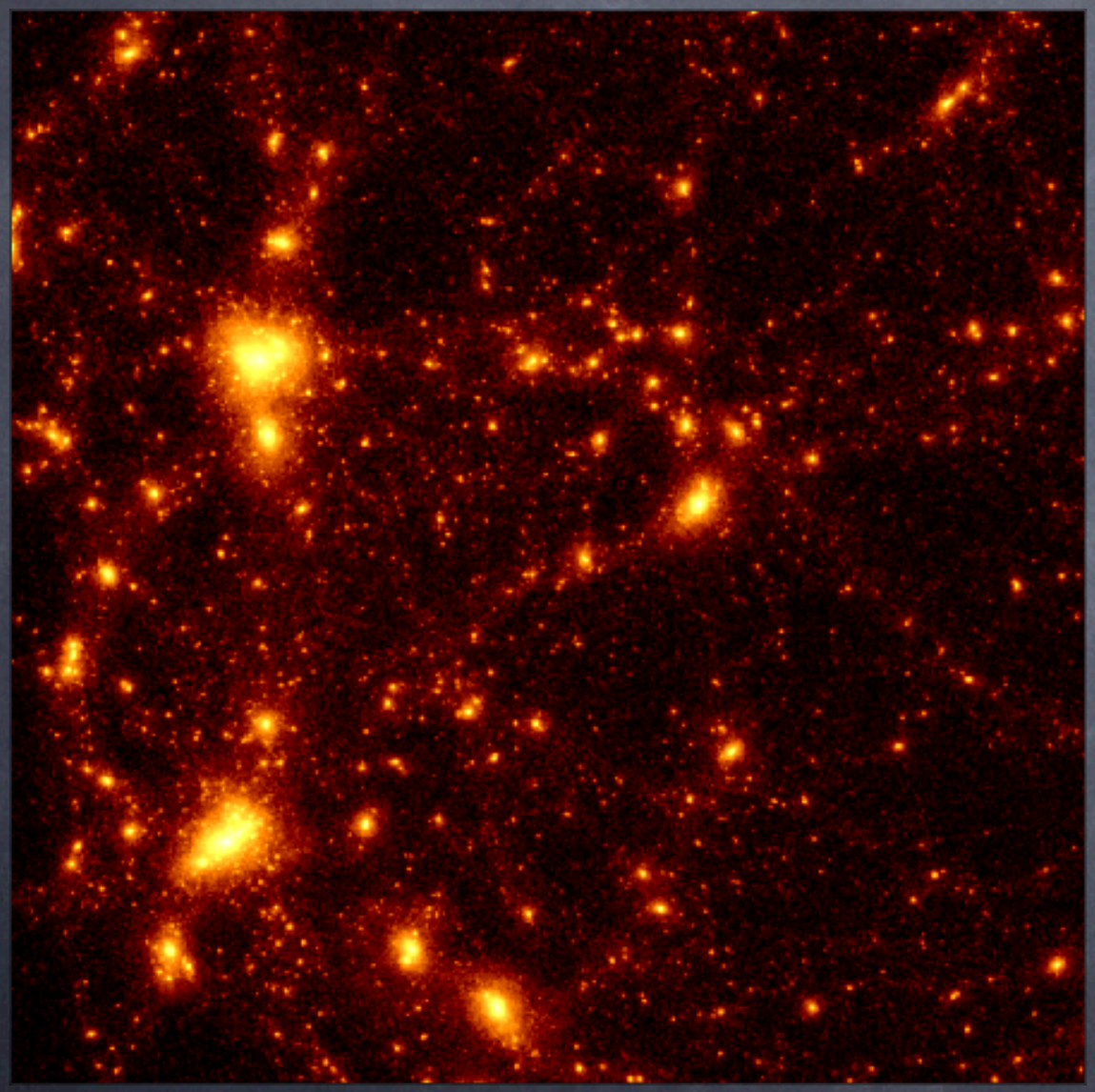
理論性能と出力性能の比較:

System A	System B	System C
0.36	0.34	0.35

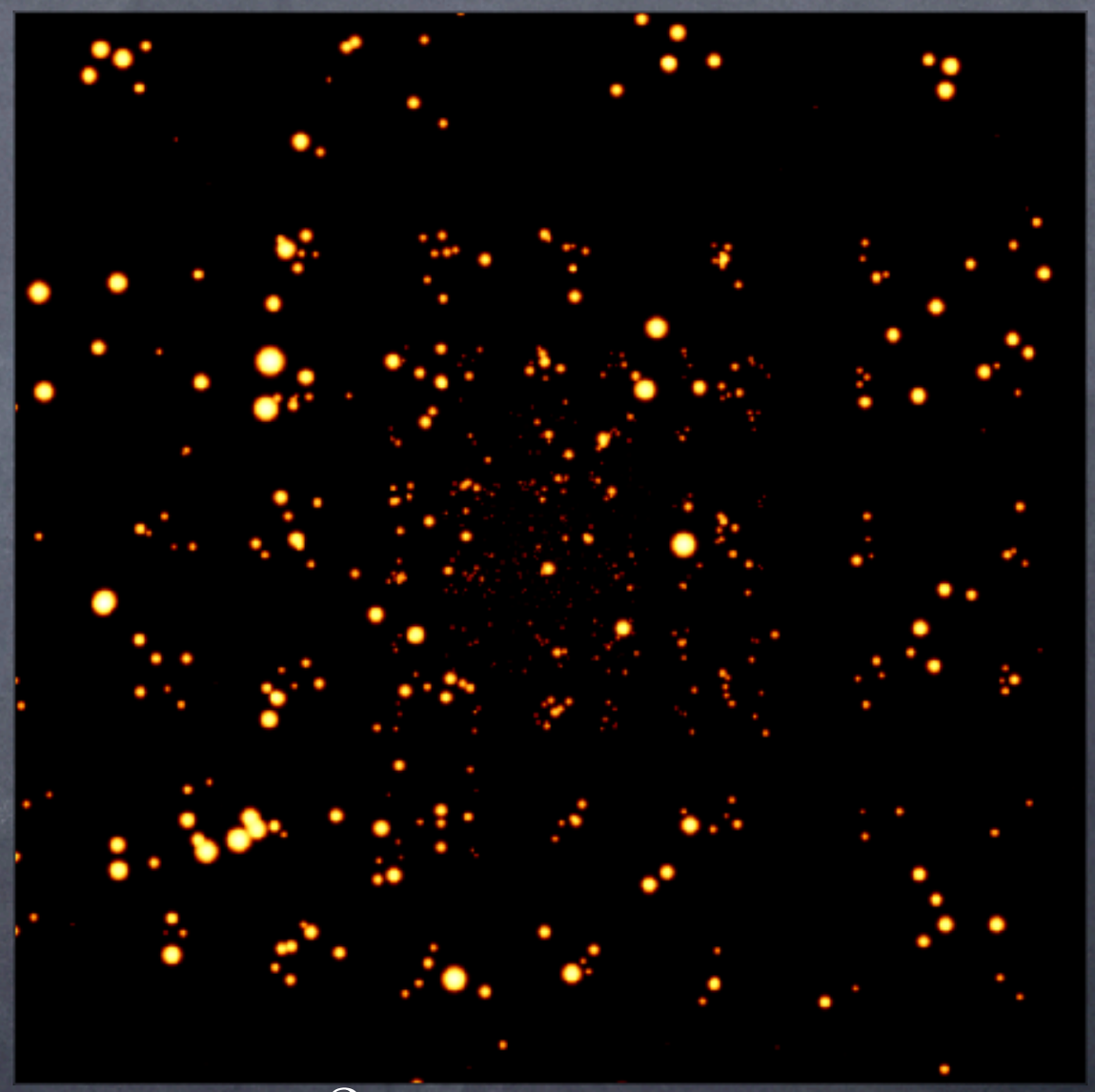
CPUの性能(core数)に比例した性能

Octree法による演算量の削減

- 遠方の粒子群を多重極展開で置き換える



$N \sim 10^6$ distribution

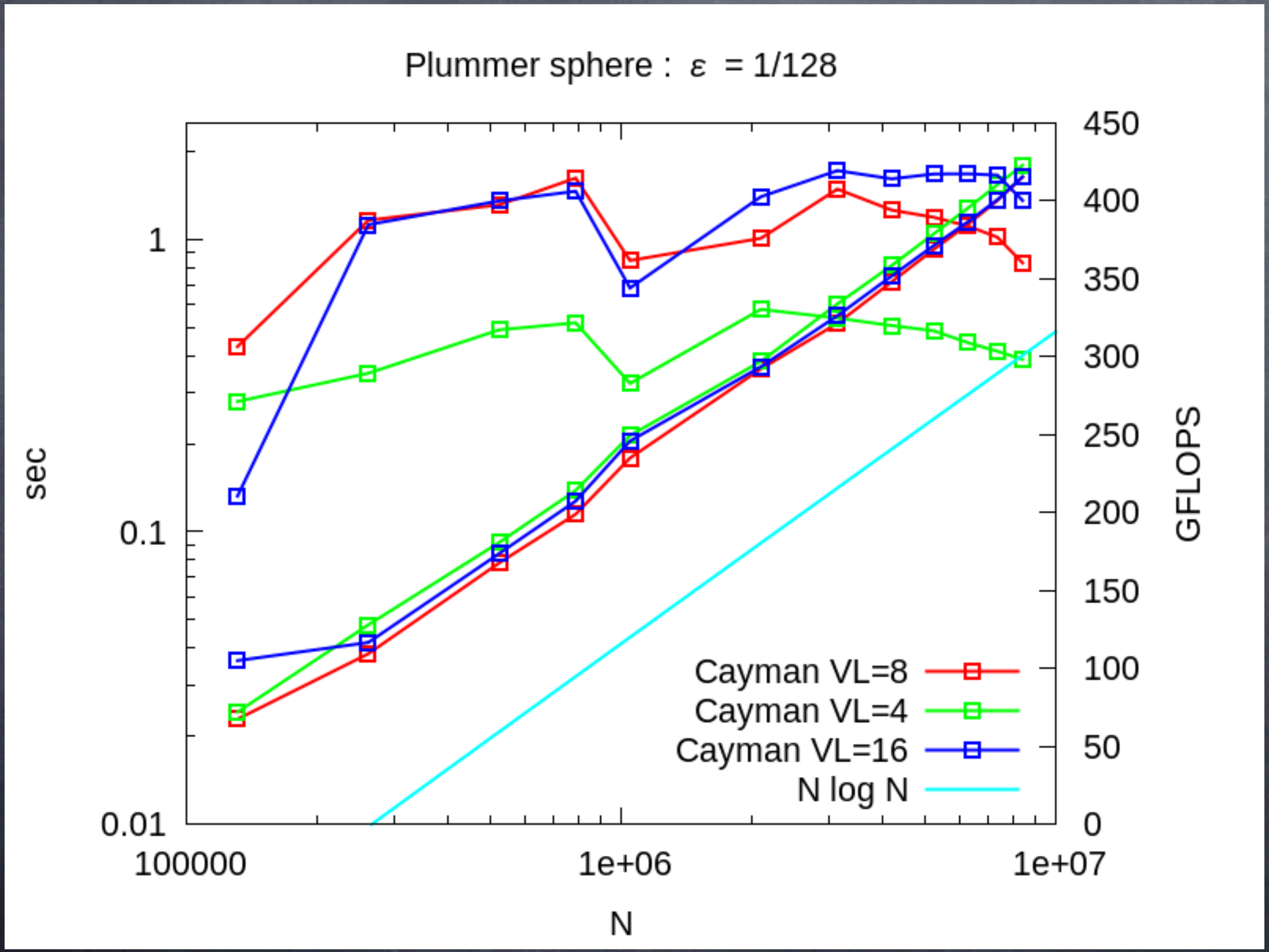


$N \sim 10^3$ particles&nodes

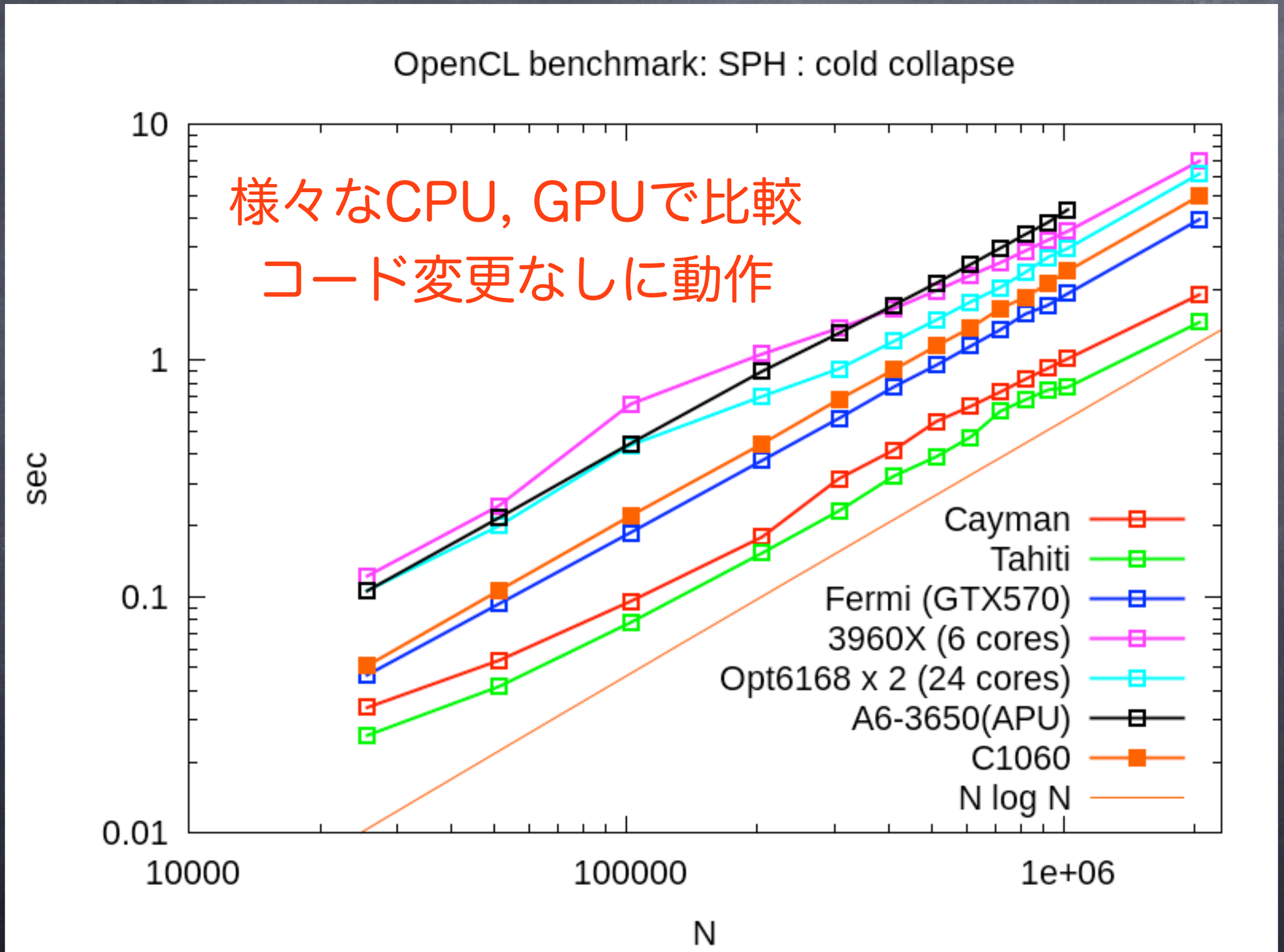
GPUでの最適化Octree

- Octreeで必要なステップ
  - tree構築 (一部並列化難)
  - treeノードの多重極モーメントの計算 (並列化難)
  - tree traversalによる相互作用計算 (並列化易)
- どの部分をGPUで実行するか？
  - 全てをGPUで実行する研究結果もある：遅い
  - O(N)の部分をGPUでやるのはあまり意味がない

ベクトル化されたOctree(重力)



OctreeによるSPH法(SPH+重力)



まとめ

- AMD GPUでのOpenCLは非常に有効
  - 2010年度まではILの利用が必須だった
- OpenCLによるCPUプログラミング
  - 効果的に利用可能
  - 特にSSE/AVXが容易にプログラミングできる
- Octreeでも高速な実装は可能
  - CPUより10 - 20倍高速化
  - OpenCLならCPUでも計算可能