

GPUを用いた天文数値シミュレーション

筑波大学 宇宙物理理論研究室
博士前期課程 1年 大野純

1 Introduction

1.1 N体シミュレーション

銀河系のシミュレーションでは、その中の星を質点とみなし質点同士の重力相互作用をみるN体シミュレーションが非常に有効な手段となっている。

天体	質点(星)の数
散開星団 (open clusters)	$10^2 \sim 4$
球状星団 (globular clusters)	$10^4 \sim 6$
銀河 (galaxy)	10^{11}
銀河団 (clusters of galaxies)	10^{15}

表 1.1 天体の質点の数

近年、演算機性能の爆発的向上により扱える質点の数は増加してきた。しかし、このN体シミュレーションにおいて質点同士の重力相互作用計算量は直接計算で $O(N^2)$ であり、粒子数が多くなるにつれてその量は増大していく。

$$F_i = -G \sum_{j \neq i}^N m_i m_j \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|^3} \quad (1)$$

また、相互作用の計算は「各質点のデータ(初期位置など)が異なるが同じ計算」であるため、高速化には並列化は非常に有効な手段と推測される。

1.2 GPUによる数値計算

これまで数値計算にはCPU(Central Processing Unit)が使われてきたが、ここ最近にGPU(Graphic Processing Unit)も数値計算に広く用いられるようになった。GPUはもともとPCゲームなどでグラフィック処理を担うグラフィックボードと呼ばれる専用外部デバイスに搭載されていたが、この演算性能を数値計算など他の分野にも活かそうというGPGPU(General-Purpose computing on GPU)の考えが出てき

た。その黎明期ではグラフィック関連の言語を用いるなど開発は非常に難しかったが、最近ではグラフィックボードのメーカーにより GPU プログラミングの開発環境が無償で提供されているため敷居は低くなっている。

	CPU(Intel Corei7 2600K)	GPU(NVIDIA GeForce GTX580)
コア数 (演算器数)[GHz]	4	512
クロック周波数	3.4	1.5
単精度演算性能 [GFLOPs]	110.8	1581.1
メモリバンド幅 [GB/sec]	21	192.4
専用スロット	有	無
価格	26,000 円	50000±10000 円

表 1.2 現在市販されている CPU と GPU のスペックの比較

GPU は CPU とは異なりそれ単体で計算をすることは出来ない外部デバイス (アクセラレータ) であり、必ず CPU とデータ通信が必要となる。

GPU はチップの内部構造のほとんどをコア (演算器) に割り当てているため、大量のコアを活かした並列計算が可能である。一方で CPU はコア数こそ少ないものの、それ以外のスペースを演算回路に用いているため分岐命令などに非常に強く、GPU は分岐命令に弱い。

またピーク性能あたりの価格が CPU に比べて安く、並列計算を非常に手軽に行うことができるといえる。

1.3 CUDA

今回の GPU による数値計算には NVIDIA 社より提供されている GPGPU プログラミング統合開発環境 CUDA(Compute Unified Device Architecture) を用いた。これは C/C++ の拡張言語であるため、既存のシングルコア用ソースコードの並列化への拡張が容易にできる。

2 実行

今回計算したのは直接法による collisionless な自己重力系の N 体シミュレーションである。

重力相互作用の計算部分を CPU と GPU でそれぞれ行い、粒子数ごとの計算時間のと演算性能を計測・比較を行った。

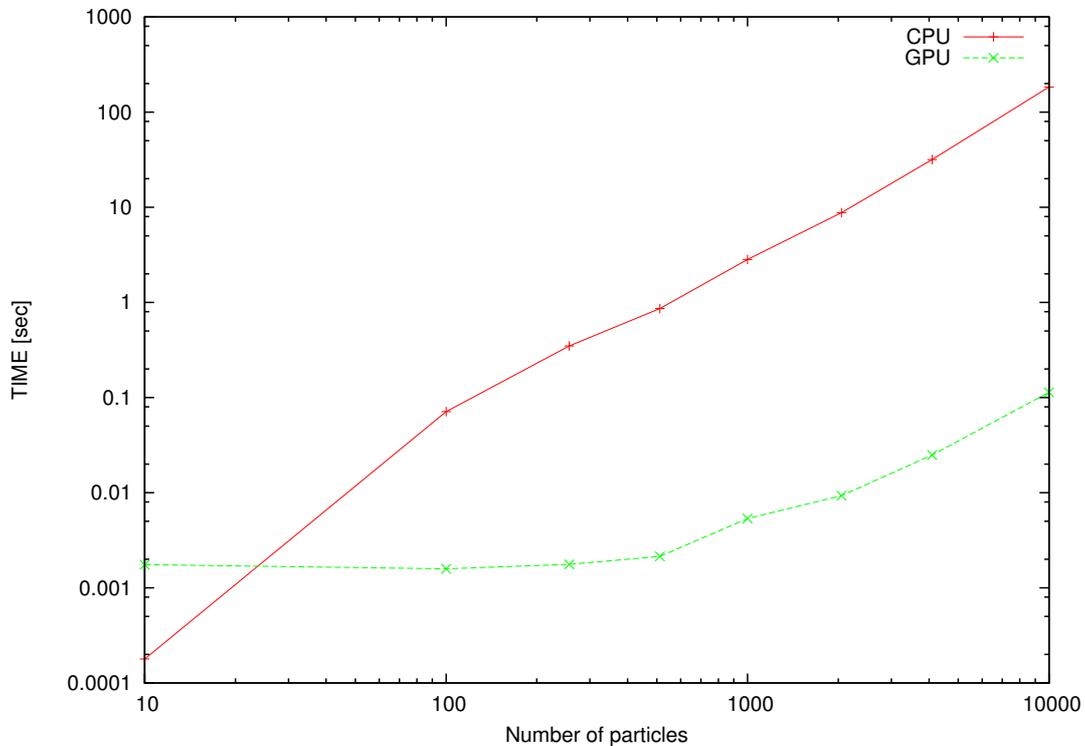
今回の数値計算で用いた CPU と GPU は以下のものである。

	Intel Xeon X5550	NVIDIA Tesla C2070
コア数 (演算器数)	4	448
クロック周波数 [GHz]	2.66	1.15
単精度演算性能 [GFLOPs]	42.56	1030
メモリバンド幅 [GB/sec]	32	144

表 2.1 使用した CPU と GPU

3 結果

3.1 計算時間

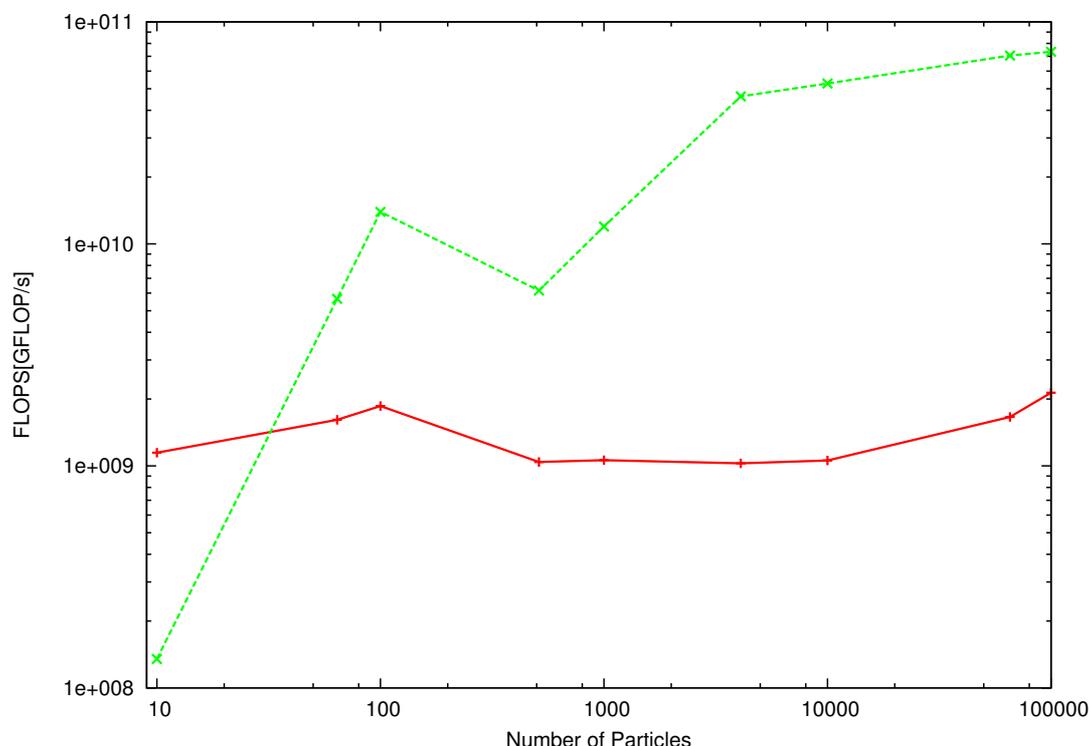


計算時間は最終的に粒子数 10 万において CPU と GPU で二倍以上の差が出た。

また、粒子数が極端に少ないと GPU の方が遅いが、粒子数 100 を超えると CPU の方が時間がかかるようになる。さらに、GPU の計算時間は粒子数が少ないところだと $O(N^2)$ の線にフィットしていない。

これは粒子数が少ないと、GPU のデータ通信のロスが露わに出てきてしまうためであり、粒子数が増加すると計算時間の増加オーダーとロスの増加オーダーが異なるためロスの時間が隠蔽されてくる。逆にデータ通信のロスと計算時間のオーダーが近い場合、即ち計算量に比べてデータ通信が多い場合には、隠蔽されにくくなるため並列化による恩恵はあまり受けられないといえる。

3.2 演算性能



演算性能は最終的に CPU が約 2GFLOPs、GPU が 70GFLOPs 程度となった。

まず、CPU についてはシングルコアで計算を行っているため、粒子数によって演算性能は特に変化せず一定の値である。

一方 GPU では、粒子数が増加するとその演算性能も合わせて上昇している。GPU は大量のコアで並列計算を行うため、粒子数が少ないとその演算性能を出し切れずにいるが、粒子数が増えると並列計算がフルに働くようになり演算性能が上がっていく。もちろん無限に上昇していくわけではなく、ある程度行くと saturation してしまう。

4 まとめと今後

N 体シミュレーションの高速化には GPU による並列計算が非常に有効であり、特に粒子数が多い場合にはそれが顕著となる。

今回の計算では CPU で作成した N 体シミュレーションのコードにただ CUDA を実装しただけであるため、その理論演算性能の 1/10 以下しか出せていない。CUDA に限らず GPU プログラミングにおいて、並列化による高速化を十分に受けるには搭載されている複数のメモリの効率的使用が不可欠である。すなわち、CUDA コードの最適化が高速化のカギとなる。

今後はこの GPU を用いて多次元輻射輸送方程式を数値計算していく予定である。