Speed-up efficiencies of an SPH code with FDPS on GPUs or PEZY-SCs

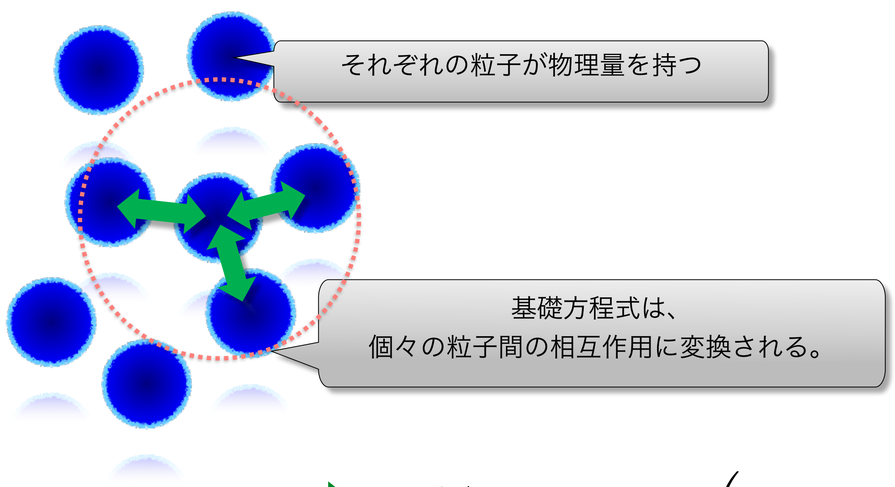
細野 七月[1,2]、岩澤 全規[2]、行方 大輔[2]、谷川 衝[3,2]、似鳥 啓吾[2]、村主 崇行[2]、牧野 淳一郎[4,2]
[1]海洋研究開発機構 [2]理研 計算科学研究センター [3]東京大学 [4]神戸大学

Abstract

Since the many important phenomena in planetary science are difficult to study by means of laboratory experiments, numerical simulations play an important role. Smoothed Particle Hydrodynamics (SPH) is a widely used particle-based numerical hydrodynamic simulation method, which has advantages to deal with large deformation, multi-component and self-gravity. Since the reliability of SPH depends on the number of particles used in each run, high-performance computing can be an important topic. However, compared to mesh-based methods, it requires relatively high computational costs. We have developed a framework, Framework for Developing Particle Simulator (FDPS) which automatically parallelise an arbitrary particle-based numerical code. Thus, recently, it has been popular to apply so-called ``accelerator'', such as GPUs, to SPH. Combining these two techniques, we have developed a massively parallel SPH code which works on either GPUs or PEZY-SCs.

(1)Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH)法とは、
元々天文の分野で開発された粒子的流体数値計算手法であり、
近年では津波などの計算にも応用されている、汎用性の高い手法である。



$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} = -\frac{1}{\rho}\nabla p \quad \Rightarrow \quad \frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij}$$

SPH法の結果の信頼性は、粒子数を上げるほど上昇すると考えられるが、
一方でSPH法の計算時間はO(N^2)で増加するため、
粒子数を増加させると急激に計算量が増える事になる。

(2)High Performance Computing to SPH

そこで、大規模な計算をSPHで行うために、種々の手法が提案されてきた。
その中の一つがTree法(Barnes & Hut, 1986; Hernquist & Katz, 1989)であり、
この手法は計算量をO(N^2)からO(Nlog8N)にする事が出来る。

もう一つが、近年開発された、加速器と呼ばれる外部計算デバイスの使用である。
加速器とは、通常のCPUに比べて性能は劣るが、
その代わり大量のプロセッサを備えた計算装置の事である。
近年はNVIDIA社のGPGPUや、Intel Xeon Phi、PEZY-SCなどが有名である。

一方で、加速器とTree法を両方用いたコードは、コーディング難易度が高い。
そこで、近年我々は、Framework for Developing Particle Simulator (FDPS)
という粒子法の自動大規模並列化ライブラリを開発した(Iwasawa+, 2016)。
FDPSはver. 2.0以降、加速器をサポートしている。

そこで、本研究ではFDPSと加速器を用いたSPHを作成する事を試みた。
現在広く使われている加速器にはいくつかの種類があるが、
ここではNVIDIA P100、PEZY-SCnp及びSC2の3つの加速器を用い、
各デバイス上でSPHの式を計算するのにかかる時間を計測する。

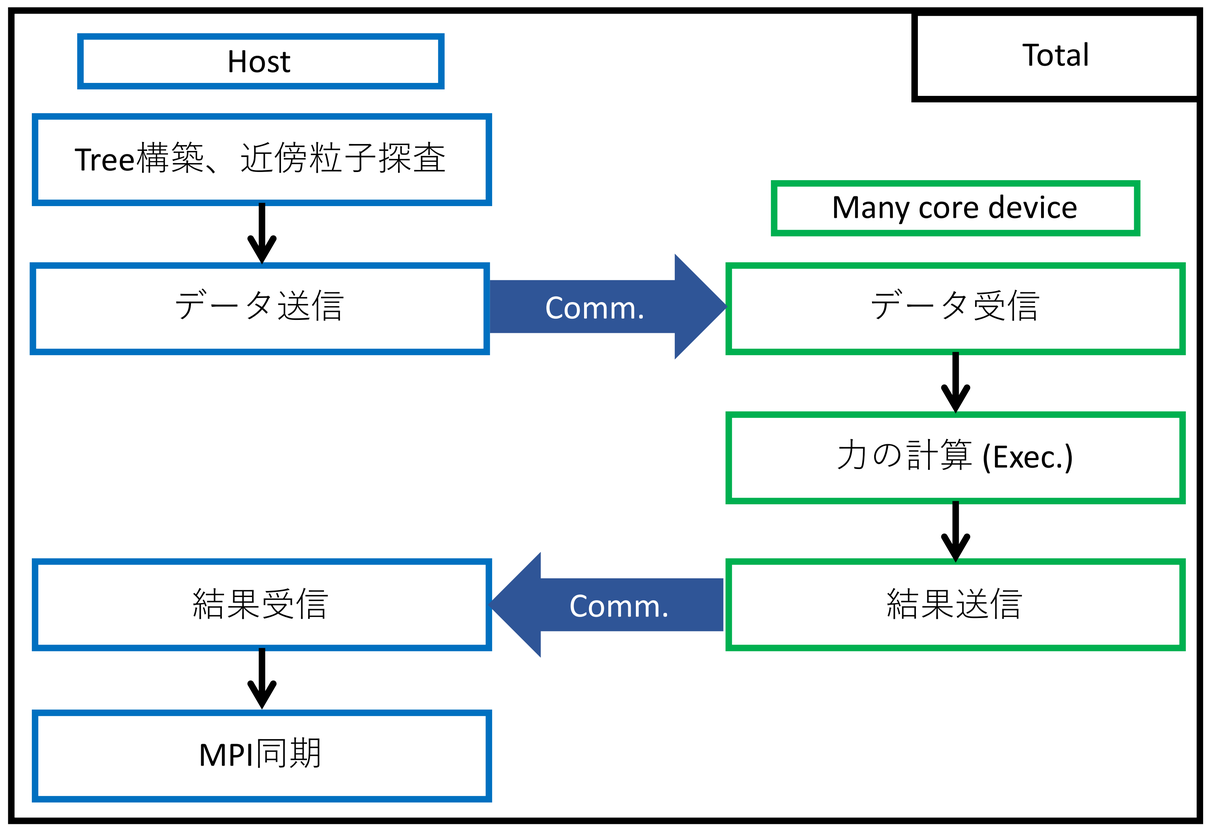
標準的なSPHでは、1stepを計算するのに3つのloopが必要となる。
流体の密度を決定するloopと、人工粘性に必要なdivergence/rotationを
計算するloop、最後に圧力勾配のloopである。
これら各々のloopで、計算に何秒かかるかを測定する。

$$\rho_i = \sum_j m_j W_{ij}$$

$$\nabla \cdot \vec{v} = \sum_j \frac{m_j}{\rho_j} \vec{v}_{ij} \cdot \nabla W_{ij}$$
$$\nabla \times \vec{v} = \sum_j \frac{m_j}{\rho_j} \vec{v}_{ij} \times \nabla W_{ij}$$

$$\frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij}$$
$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{1}{2}\Pi_{ij} \right) \vec{v}_{ij} \cdot \nabla W_{ij}$$

(3)FDPS w/ Accelerators



上図は、FDPSから加速器を用いる際の模式図である。
青はCPU(Host)側が実行する処理で、緑は加速器が実行する処理である。
以降では、この処理全体を1回行うのにかかった時間を、
各デバイス上で測定する事にする。

(4)Timing Results

粒子数は1,000,000粒子とし、各デバイスを1つのみ用いた。
粒子分布は一様球である。測定精度は単精度で、また、FDPSの関数である、
“TimeProfile::getTotalTime()”を用いて測定を行った。
単位は全て、秒である。

デバイス	CPU	CPU SIMD	GPU(P100)	SCnp	SC2
密度	1.4	0.45	0.44	0.56	0.60
div/rot	3.7	0.79	0.75	0.83	0.89
圧力勾配	5.4	1.3	1.6	2.4	1.4

これから分かる通り、最も時間がかかるloopは、圧力勾配である。
これは、圧力勾配を計算するloopが最も演算量が多いためである。
従って、このloopの計算時間に着目すると、
PEZY-SC2は概ねP100と同程度の計算時間でloopの計算を終わらせている。

SIMD化したCPUでの計算は、最も計算時間が短い。
しかし、GPUやSC2は1つのCPUに対して複数個装着できる為、
複数個の加速器を用いて計算する事で、計算時間をより短く出来る。

ただし、今表示されている計算時間は、あくまでもTotalの物である事に留意する。
即ち、CPUでの処理時間が入っており、加速器上で消費された時間や、
ホストと加速器間の通信などにかかった時間などが全て入っている。
そこで、よりフェアにデバイス間の比較をするために、上記の時間を、
CPU上処理時間、ホストと加速器間での通信時間、デバイス上処理時間に分け、
だす事が必要とされる。