

陽的・半陰的 SPH 法の多数 GPU 環境下での動的負荷分散並列による高速化 Boosting an explicit and semi-implicit SPH code with dynamic load balancing parallelization on many GPUs

大崎 春輝 (九州大・工) 森川 ダニエル 茂夫 (九州大・工) 浅井 光輝 (九州大・工)

Haruki OSAKI, Kyushu University
Daniel S. MORIKAWA, Kyushu University
Mitsuteru ASAI, Kyushu University
E-mail: h-osaki@doc.kyushu-u.ac.jp

The SPH method are attracting attention as an efficient method for solving free-surface flows in complex geometries such as tsunami run-up phenomena. In the case of urban-scale disaster analysis, the particle method must be applied to a large area with evenly spaced particles, which results in a large computational model. However, the memory size of GPUs is smaller than that of CPUs, and therefore, multiple GPUs are required to perform large-scale analysis. In this study, the parallel SPH particle computation using many GPUs including communication across computational nodes is developed.

1. はじめに

我が国では、2011 年の東日本大震災、あるいは甚大化している豪雨災害など、大規模災害が頻発している。今後危惧される大規模災害に備えるため、事前の被害予測が求められ、数値解析による高精度な災害影響評価は重要性を増している。なかでも、対象とする物体の大変形・分裂・結合を伴う形状変化の激しい問題を効率よく計算できる SPH 法などの粒子法は、津波遡上現象などの複雑形状内の自由表面流れを効率的に解く手法として注目されている。本研究では、効率が期待できる GPU を用いた並列計算¹⁾を拡張し、計算ノードを跨ぐ環境での通信を含む多数 GPU の利用を検討した。本研究が目的とする、都市規模の災害解析へと適用する場合、広域な対象領域に対して、一般的な粒子法は均等な粒子間隔で解くことが要求されるため、必然的に計算モデルが大規模になる。しかし、GPU のメモリサイズは CPU と比較すると小さく、大規模な解析を行うには複数 GPU を用いた計算が必要となる。本研究では、大規模津波遡上解析を高速に実施する SPH 法を開発することを目的とし、MPI 通信を含む動的負荷分散コードを GPU 環境下で実装し、性能分析を行った。

2. 解析手法

2.1 安定化 ISPH 法

本研究で用いた解析手法¹⁾である SPH 法は、解析領域を有限個の粒子に離散化し、対象粒子の物理量を一定の領域内に存在する近傍粒子の物理量を重み付けした平均として算出する。本研究で用いたのは、SPH 法の中でも非圧縮性流れに適した安定化 ISPH 法である。安定化 ISPH 法では、ISPH 法の解析精度と安定性の両者の向上を目的とし、微圧縮性を一部許容した下記の圧力ポアソン方程式により圧力を求める。

$$\langle \nabla^2 P \rangle_i^{n+1} = \frac{\rho_0 \nabla \cdot \mathbf{v}^*}{\Delta t} + \alpha \frac{\rho_0 - \langle \rho \rangle_i^n}{\Delta t^2} \quad (1)$$

ここで、 $P, \rho, \mathbf{v}^*, \alpha$ はそれぞれ、圧力、流体密度、流体速度、予測ステップにおける速度、安定化パラメータである ($\langle \cdot \rangle$ は SPH 近似による物理量を指す)。(1)式を離散化したのが次に示す(2)式である。

$$\sum_j A_{ij} (P_i^{n+1} - P_j^{n+1}) = b_i \quad (2)$$

ここで、

$$A_{ij} = \frac{2}{\rho_0} m_j \frac{\mathbf{r}_{ij} \nabla W(\mathbf{r}_{ij}, h)}{r_{ij}^2} \quad (3)$$

$$b_i = \frac{\rho_0 \nabla \cdot \mathbf{v}^*}{\Delta t} + \alpha \frac{\rho_0 - \langle \rho \rangle_i^n}{\Delta t^2} \quad (4)$$

である。(2)式において、時間ステップ n から $n+1$ までの微小な時間間隔では圧力の変動が小さいと仮定し、右辺の近傍粒子の圧力を前ステップで近似すると、次式のように圧力の陽的な更新も可能である。

$$P_i^{n+1} = \frac{b_i + \sum_j A_{ij} P_j^n}{\sum_j A_{ij}} \quad (5)$$

(2)式は、圧力に関して連立 1 次方程式を解く必要があるため半陰解法と呼び、一方、(5)式は陽解法と呼ぶ

2.2 マルチ GPU による並列計算

MPI 通信による複数の GPU を用いたハイブリッド並列化の概念図を Fig.1 に示す。粒子法の MPI による並列化では、背面に用意したバックグラウンドセルに粒子を事前に登録し、セル単位で領域分割を行う。MPI プロセスを 1 台の GPU に割り当てて計算し、GPU のスレッドを用い、粒子毎の計算を並列処理する。この際、SPH 近似計算を行う計算で毎回近傍粒子探索を行い、近傍粒子のリンクリストを保存しないことでメモリを節約している。各 GPU のメモリ領域は独立しており、隣接するプロセスにおける計算で参照される袖部領域の粒子データは、隣接プロセス間で MPI 通信を行うことで整合性を担保した。また、SPH では時間経過に伴い粒子配置が大きく変動するため、前ステップからの計算時間を計算コストとして、均等化するように MPI プロセスを再構成する動的負荷分散を行った。また、毎ステップにおいて粒子番号のリナンバリング及びバックグラウンドセルの再登録を行った。

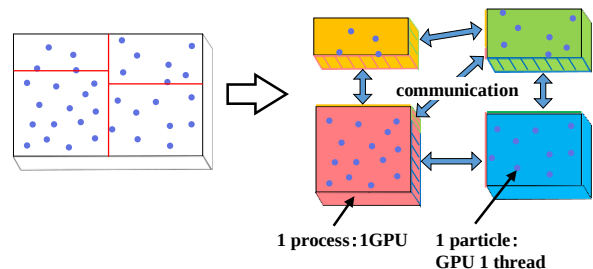


Fig. 1 Multi-GPUs calculation by MPI Communication

3. ITO を用いた性能分析

九州大学・情報基盤開発センターのスパコン ITO のうち、GPU を搭載しているサブシステム B は、16 ノードから構

成され、1 ノードは2 ソケットの CPU (18-cores Intel Xeon Gold 6140 2.3 GHz) と 4 台の GPU (NVIDIA Tesla P100) から構成される。ITO-B を用いてダムブレイク問題の計算を行い、本コードの実行性能を計算時間により計測した。計算は倍精度浮動小数点を用いて行い、陽解法・半陰解法それぞれにより計算を行った。

まず、1 プロセス (1GPU) 当たりの問題サイズを固定して並列数を増加させる weak スケーリングの評価を行った結果を Fig.2 に示す。1 台の GPU 当たりの水粒子数を約 256 万粒子 (壁粒子約 126 万粒子, 総粒子約 382 万粒子) とし、1 台の GPU から 64 台の GPU までの実行性能を計測した。64 台の GPU の条件で、水粒子約 1 億 6227 万粒子 (壁粒子約 2222 万粒子, 総粒子数 1 億 8449 万粒子) である。計測結果をプロット点で、理想的な実行性能を点線で示している。64 台の GPU の場合で、陽解法では 94.0% の並列化効率を得られ、60.2 倍の性能向上であり、ほぼ理想に近い良好な weak スケーリングが得られた。また、陽解法でのプロファイラを Fig.3 に示す。GPU 数を増加させても演算時間及び通信時間の割合をほぼ一定にキープできている。一方で、半陰解法では 58.5% の並列化効率で 37.5 倍の性能向上に留まった。半陰解法では、圧力を求める際に反復解法である共役勾配法を用いており、計算時間の大半を占めている。問題規模の増加に伴い、共役勾配法による連立 1 次方程式ソルバーの反復回数が増加しており、演算数に対する並列化効率を計測すれば、58.5% → 95.8% と、陽解法と同程度の良好なスケーリングが得られた。

次に、全体の問題サイズを固定して並列数を増加させる strong スケーリングを半陰解法により測定したときのプロファイラを Fig.4 に示す。水粒子数を約 256 万粒子 (壁粒子約 126 万粒子, 総粒子約 382 万粒子) として計測した。この図からも半陰解法では連立一次方程式ソルバーの実行時間 (= 演算時間 + 通信時間) が大半を占めることが確認できる。また、GPU 数を増やすほど、演算時間は単調に減少していることが確認できるが、通信時間はほぼ一定となっていた。

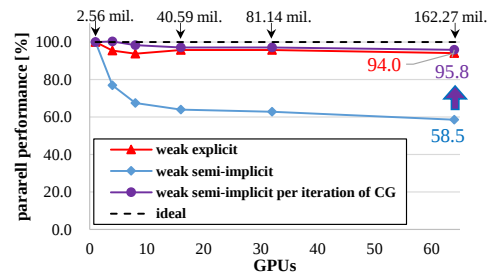


Fig. 2 weak scaling measurement results

4. 結論

本研究では、多数 GPU による SPH 解析の大規模化を図るために MPI による並列化を行い、性能を計測した。その結果、圧力を陽的に更新する陽解法ではほぼ理想通りの良好な weak スケーリングを達成できた。一方、半陰解法では、連立 1 次方程式ソルバーが計算時間の大半を占め、問題規模の増加に伴い反復回数が増加するため、陽解法ほどの効率性は得られなかった。ただし、反復回数の増加に伴い計算量が増えたため、計算時間に関する並列化効率は良好なものであった。

今後、特に半陰解法の更なる改良が必要と考えている。たとえば、連立 1 次方程式ソルバーの反復解法において、自由度の増加に対し反復回数が増加しない前処理方法の開発があげられる。また、多階層構造をなす GPU のメモリにおける、小容量ではあるが高速な演算が可能なキャッシュメモリのヒット率を上げることが望ましい。現状のコードでは、セルを一方向に走査しながら粒子番号をリナンバリングしているが、これを空間充填曲線による粒子番号のリナンバリングに変更することで、メモリの効果的な利用を検討していく。

参考文献

- 1) D. S. Morikawa, H. Senadheera and M. Asai, Explicit Incompressible Smoothed Particle Hydrodynamics in a multi-GPU environment for large scale simulations. *Comput. Part. Mech.*, Vol. 8(3), pp. 493-510, 2020

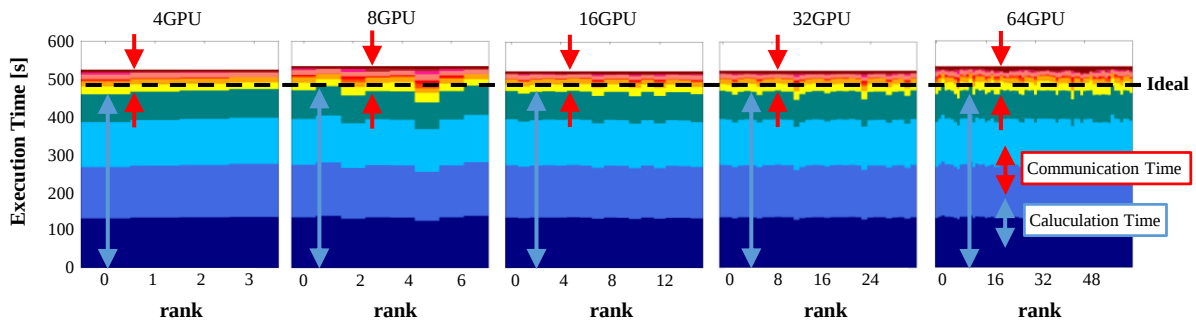


Fig. 3 Profiler for weak scaling (explicit method) (each subroutine is displayed by color)

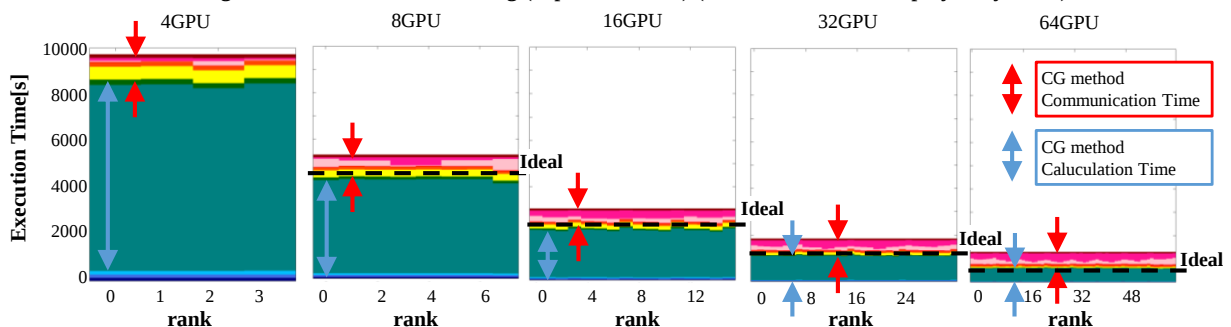


Fig. 4 Profiler for strong scaling (semi-implicit method) (each subroutine is displayed by color)