

# 性能の異なるコンピュータからなるクラスタによる 流体の移流計算の高速化

生田目 大地<sup>†</sup><sup>†</sup> 早稲田大学森島 繁生<sup>‡</sup><sup>‡</sup> 早稲田大学理工学術院総合研究所

安東 遼一

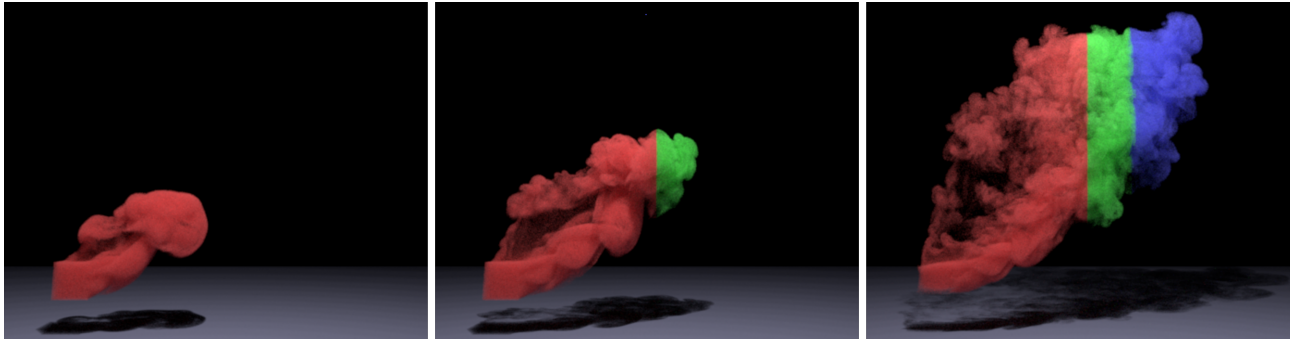


図 1: 上昇する煙。煙の色は分割した領域を表している。領域ごとに異なるコンピュータを使って並列に移流計算を行っており、単一のコンピュータのみを使った場合に比べて、約 1.61 倍の高速化となっている。

## 1. はじめに

流体 CG の移流計算はクラスタを用いた並列分散処理により高速化が可能である。しかし、各ノードに均等な量のタスクを分散した場合、それらの性能が等しくない限り、効率的な高速化が困難なことがある。そこで、各ノードの計算能力・通信速度を考慮して、適切にタスクを分散する操作が必要になる。

本研究は、性能が一樣でないノードから構成されるクラスタを計算環境とし、流体の移流計算を効率的に高速化する手法を提案する。まず、プログラム実行時に各ノードで通信速度のテストとサンプル問題の計算時間の測定を行う。次に、測定した情報をもとに、各ノードの計算時間と通信コストの合計が一樣になるようなタスク分散の割合を求める。提案手法をセミラグランジュ法や MacCormack 法による移流計算に適用し、有効性を評価した。

## 2. 関連研究

### 2.1 流体シミュレーション

流体の移流項の計算に有効な手法として、セミラグランジュ法がある [1]。この手法は、ある位置における次の時刻の物理量を、そこに仮想的な粒子を考え、粒子が速度場を介してタイムステップ  $\Delta t$  だけ遡ったときの位置における物理量として求める。セミラグランジュ法の応用として、移流を 2 回行って誤差を求め、その誤差を打ち消す手法である MacCormack 法がある [2]。この方法はセミラグランジュ法よりも精度が高い。これらの移流計算法では、遡った位置における物理量は補間により求められる。補間にはトリリニア補間や WENO 補間 [3] が用いられる。WENO 補間は精度が高いが、計算コストが高い。移流項の計算はグリッドの各セルで独立しているため、並列化が可能である。

### 2.2 ヘテロジニアス・クラスタ

性能の異なるコンピュータからなるクラスタは「ヘテロジニアス・クラスタ」と呼ばれる。先行研究として、クラスタを構成する各ノードの性能比を求め、ループのイテレーションを分割する研究 [4] や、通信コストを高い精度で予測する研究 [5] がある。

## 3. 手法

各ノードに割り当てるタスクの割合を決定する手順を説明する。この計算はシミュレーションの事前に行う。以下では、コンピュータの数を  $n$ 、ノードを表すインデックスを  $i=1, 2, \dots, n$  とし、ホストコンピュータのインデックスは 1 とする。まず、各ノードの 1MB 分の情報の通信時間  $t_i$  を、実際に情報転送のテストを実行することで計測する。ここで、ホストコンピュータの通信時間は 0 であることに注意する。次に、各ノードにおいて、グリッド解像度が  $200^3$ 、密度場の解像度が  $400^3$  の流体でサンプルの移流問題を解き、グリッドの情報量が 1MB の場合の計算時間  $c_i$  を求める。このサンプルの問題は、分散処理を行いたい問題と、スケールの差異を除いて同じ問題にする。最後に、各ノードの計算時間と通信時間が一樣になることを表す以下の連立一次方程式を解き、各ノードのタスク割合  $w_i$  を求める。

$$c_1 w_1 = \sum_{k=2}^i t_k w_k + c_i w_i + \sum_{k=i}^n t_k w_k \quad (i=2, 3, \dots, n) \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i = 1. \quad (2)$$

式 (1) は、図 2 のように各ノードに順にデータを送り、計算を行い、ホストがデータを受信するまでの時間が一樣になることを表す。通信チャンネルの競合が起きると、通信速度が低下するため、データの送受信を逐次的に行うことにしている。式 (2) は、すべてのノードのタスク

Accelerating fluid advection by using a cluster of computers with different performance:

Daichi Namatame<sup>†</sup>, Shigeo Morishima<sup>‡</sup>, and Ryoichi Ando (<sup>†</sup>Waseda University, <sup>‡</sup>Waseda Research Institute for Science and Engineering)

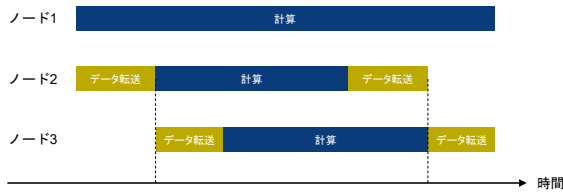


図 2: 各ノードの処理時間のイメージ ( $n=3$  の場合)

割合は足すと 1 になるという条件である。

タスクの割合が求まったら、シミュレーションの空間領域を分割し、それぞれの領域の移流計算を各ノードに割り当てる。領域分割は特定の座標軸に垂直な面により行い、分割した領域の体積の比はタスクの割合に等しくなるようにする。分割の際は、各領域にオーバーラップを作る。それは、セミラグランジュ法にて遡った位置が、領域外にはみ出してしまふことを防止するためである。領域のオーバーラップ分の長さは、次の式で表される。

$$\max_{\mathbf{x} \in \Omega} |u(\mathbf{x})| \cdot \Delta t + \frac{l}{2} \cdot \Delta x. \quad (3)$$

ここで、 $\mathbf{x}$  は位置、 $\Omega$  は全領域、 $u(\mathbf{x})$  は領域の切断面に垂直な成分の速度、 $\Delta t$  はタイムステップ、 $\Delta x$  は格子間隔、 $l$  は補間のステンシルサイズで、トリリニア補間の場合は 2、WENO 補間の場合は 6 である。

## 4. 実験

### 4.1 実験概要

計算環境は 3 台のノードが 1Gbps のハブでつながっているヘテロニアス・クラスタである。ノード 1, 2, 3 に搭載されている CPU はそれぞれ、AMD Ryzen 9 3900X, AMD Ryzen 5 3400G, AMD Ryzen 7 3700X であり、ノード 1 はホストコンピュータである。グリッド解像度を  $300^3$ 、密度場の解像度を  $600^3$  として、煙のシミュレーションを行う。移流計算はヘテロニアス・クラスタを用いて行い、各ノードの計算時間、データ送信・受信時間を測定する。時間の計測は 5 タイムステップに対して行い、結果は平均を取ったものとする。

### 4.2 結果

本手法を用いて行った煙のシミュレーション映像を図 1 に示す。移流の計算には MacCormack 法と WENO 補間を用いている。煙の色は分割した領域を表していて、赤色の領域はノード 1、緑色の領域はノード 2、青色の領域はノード 3 により計算されている。

ノード 1 のコンピュータのみを用いて計算を行った場合と、クラスタによる分散処理を行った場合の、1 タイムステップあたりの総計算時間を図 3 に示す。トリリニア補間を用いる場合、領域分割の計算コストにより、高速化に失敗しているが、WENO 補間を用いる場合、高速化に成功しており、特に MacCormack 法と WENO 補間を用いた場合は約 1.61 倍の高速化となっている。各ノードの計算時間と通信時間を図 4 に示す。領域分割の計算コストを除いた移流の計算時間はノード 1 からノード 3 までの時間の最大値となるため、各ノードの計算・通信時間が揃っていることが望ましい。図 4 から、各ノードの計算時間の最大値と最小値の差は最大で 4.93% に収まっていることが分かる。ノード間の計算・通信時

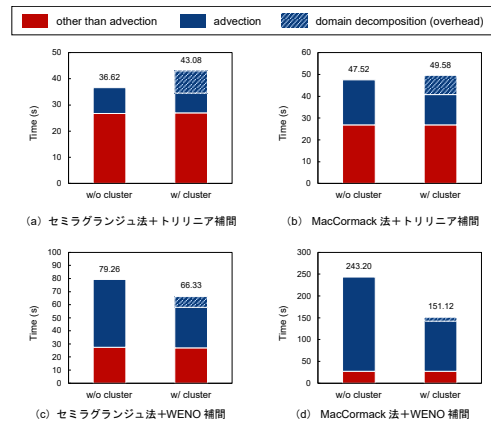


図 3: 計算時間の比較

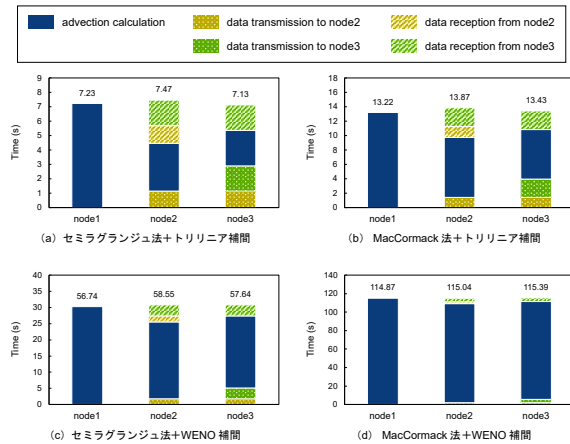


図 4: 各ノードの計算・通信時間

間に差が生じる原因の 1 つとして、領域分割の際に取ったオーバーラップが考えられる。

## 5. おわりに

本稿では CG 流体シミュレーションの移流計算にヘテロニアス・クラスタを用いた並列分散処理を適用した。今後の展望として、各ノードにおいて GPU による並列化を行い、計算を高速化することが挙げられる。

**謝辞** この研究は、JST 未来社会創造事業 (JPMJMI19B2) および JSPS 科研費 (19H01129, 19H04137, 21H05054) の補助を受けた。

## 参考文献

- [1] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, 1999.
- [2] Andrew Selle, Ronald Fedkiw, Byungmoon Kim, Yingjie Liu, and Jarek Rossignac. An unconditionally stable maccormack method. *Journal of Scientific Computing*, 35(2):350–371, 2008.
- [3] Colin B Macdonald and Steven J Ruuth. Level set equations on surfaces via the closest point method. *Journal of Scientific Computing*, 35(2):219–240, 2008.
- [4] Chao-Tung Yang, Wen-Chung Shih, and Shian-Shyong Tseng. Dynamic partitioning of loop iterations on heterogeneous pc clusters. *The Journal of Supercomputing*, 44(1):1–23, 2008.
- [5] Juan-Antonio Rico-Gallego, Alexey L Lastovetsky, and Juan-Carlos Diaz-Martin. Model-based estimation of the communication cost of hybrid data-parallel applications on heterogeneous clusters. *IEEE Transactions on Parallel and Distributed Systems*, 28(11):3215–3228, 2017.