

# ビルディングキューブ法に基づく非圧縮性流体計算の専用計算機アーキテクチャに関する一考察

千葉 諒太郎<sup>†</sup> 高野 芳彰<sup>†</sup> 佐野 健太郎<sup>†</sup> 山本 悟<sup>†</sup> 佐々木 大輔<sup>‡</sup> 中橋 和博<sup>\*</sup>

東北大学<sup>†</sup> 金沢工業大学<sup>‡</sup> 宇宙航空研究開発機構<sup>\*</sup>

## 1. 緒論

今日、実用的な問題への適用が可能となった数値流体力学(CFD)は、自動車、航空機、エネルギーなどの基幹産業において、コストや労力の点から必要不可欠な技術となっている。その一方で、大規模化、高精度化した今日の CFD は計算時間が膨大なものとなりつつあり、計算の高速化が求められている。

しかし、汎用計算機を用いて大規模な CFD 計算を行う場合、演算器を十分に稼働できないという効率の問題がある[1]。また、大規模な計算を行うには計算機を多数接続した並列計算機が必要となるが、この場合演算器稼働率はさらに低下する傾向にある。高い演算器稼働率が得られない原因は、外部メモリ間や計算ノード間のバンド幅が演算性能に比べて低いためである。バンド幅とは単位時間に転送されるデータ量のことであり、計算機の性能を左右する重要な要素である。バンド幅の不足により、サイクルあたりの入力データ量が制限される場合、演算器稼働率は低下せざるを得ない。

次世代の並列 CFD 計算手法であるビルディングキューブ法(BCM)[2]においても、演算器稼働率低下が問題となるとの報告がなされている。BCM は直交格子による CFD の一つであり、複雑な計算格子が不要かつ計算が比較的簡単という利点がある。また、全計算領域をキューブと呼ぶ立方体に分割し、並列に計算を行う。

BCM の演算器稼働率を低下させる原因の一つは、その低い演算密度である。ここでいう演算密度とは、外部メモリを参照したデータ 1 バイトあたりの演算回数である。非圧縮性流体の BCM 計算はステンシル計算の集合であり、演算数に対し必要なデータ数が比較的多いため、メモリバンド幅により計算性能が制限されやすい。また、ネットワークのバンド幅も演算器稼働率低下の大きな原因となっている。BCM は負荷分散が均等であるため並列化に適しているが、並列化が大規模になるほどデータ通信回数は増加する。ネットワークのバンド幅が不十分な場合には、大量のデータ通信に必要な時間が増大し、演算器稼働率の低下を引き起こす。このため、計算機の台数を増やしても、それに比例した性能向上は得られない[5]。

このようなバンド幅問題の解決策として、専用計算機の構築があげられる。専用計算機は、対象となる計算問題に適した演算器やデータパス、メモリシステム、およびネットワークを構築することにより、演算器稼働率と台数効果の高い効率の良い計算を実現可能と言われている。BCM に対しても、特にキューブの計算やキューブ間のデータ交換に適したアーキテクチャを用いることにより、台数効果の高い大規模計算が期待される。

近年では専用計算機を実装する手段として、回路再構

“Custom Accelerator Architecture for Incompressible Fluid Computation based on Building Cube Method”

<sup>†</sup>Ryota Chiba, Yoshiaki Kono, Kentaro Sano, Satoru Yamamoto, Tohoku University

<sup>‡</sup>Daisuke Sasaki, Kanazawa Institute of Technology

<sup>\*</sup>Kazuhiro Nakahashi, JAXA

成可能デバイスである FPGA (Field Programmable Gate Array)が注目されている。汎用デバイスとして 1 チップから入手可能な FPGA は、少量多品種の専用アクセラレータを低コストにより構築できる。以上の背景の下、本研究では高い実効演算性能と低消費電力を両立する BCM 専用計算機の FPGA による実現を目指している。専用ネットワークとストリーム計算は実効性能向上に効果的であると考えられるため、本稿ではそれらを想定した BCM 専用計算機的设计を行う。

## 2. BCM に基づく非圧縮性流体計算

### 2. 1 ビルディングキューブ法

BCM は、計算領域を異なる大きさの小計算領域(キューブ)に分割し並列数値計算を行う手法である。複数のキューブに分割された計算格子の例を図 1 に示す。各キューブは同じ数の格子点(セル)からなる等間隔直交格子であり、小さなキューブを用いた部分では高解像度の計算ができる。これら大きさの異なるキューブを配置することにより、必要最小限の格子点数で計算を行うことが可能である。また、各セルは流体セルと物体セルに分類され、基本的な BCM の場合、物体形状は階段状に表現される。流体セルでは、その中心において流速や圧力などの物理量が定義される。BCM はキューブあたりの演算数と同じことから、負荷分散に優れた並列計算が行えるという特長がある。加えて、計算アルゴリズムが単純、格子生成が容易、といった等間隔直交格子の利点をそのまま引き継いでおり、非常に有用な流体解析手法である[2]。

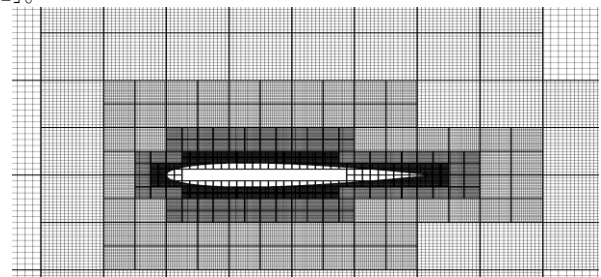


図 1 BCM 計算格子例

### 2. 2 非圧縮性流体計算

本研究では、計算対象として非圧縮性流体計算を取り扱う。時間積分にはフラクショナルステップ法を採用している。フラクショナルステップ法は仮速度計算、定数項計算、圧力計算、真速度計算の 4 つのステップを時間発展的に繰り返すことにより各時刻の速度、圧力を順次計算する。圧力計算には反復解法の一つであるレッドブランク SOR 法を用いる。各計算ステップは、隣接格子点データから自格子点データを更新する、ステンシル計算を行う。BCM において Cube 単位の並列計算を行うためには、各計算ステップ間で格子点データを同期する必要がある。

### 3. BCM 専用計算機アーキテクチャ

#### 3.1 全体構成と Accelerator Element

高効率な大規模並列計算を実現する専用計算機を設計するにあたっては、計算要素を増加しても稼働率が低下しないように、不要なデータ移動を抑えつつデータ参照に十分なバンド幅を確保する必要がある。このためには、計算の局所性に基づきキューブデータや境界データを配置し、大域的なデータ移動をできる限り行わないような構造が適している。本節では、そのような要件を満たす図2の専用計算機アーキテクチャを提案する。

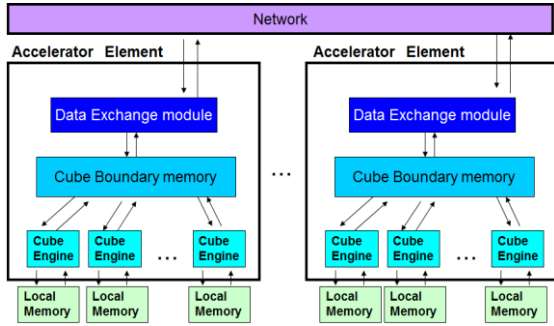


図2 BCM 専用計算機アーキテクチャ

本アーキテクチャは、計算領域を Accelerator Element (AE) と呼ばれる複数の計算要素に割り当て、さらに Cube Engine (CE) と呼ばれる AE 内部の演算器群によってキューブ単位でのストリーム並列計算を行う構造を持つ。複数の AE がネットワークにより接続されており、AE 数を自由に増加させることができる。AE 間通信が必要な場合は、データ交換モジュール (DEM) が自律的に他の AE との通信を制御する。キューブのデータは CE それぞれの Local Memory (LM) に格納されている。LM には担当キューブの全セルの速度、圧力等の物理量が記憶されており、CE の計算の進行に応じてデータの読み書きが行われる。

また、キューブ境界データを各 CE ができるだけ容易に参照できるようにするため、Cube Boundary Memory (CBM) と呼ばれるメモリを備えている。フラクショナルステップ法ではキューブ境界セルの計算の際に隣接キューブの値を参照するが、この時に発生する遅延が性能の低下を招く可能性がある。本アーキテクチャでは、各ステップにおいて計算結果が得られると、境界データを CBM に書き込む。各 CE は CBM を共有しており、書き込まれているキューブ境界データを参照することができる。これにより CE が隣接キューブの計算を待つ時間を削減できる他、データ読み込み中に別の部分の計算を行うことにより遅延の隠蔽が可能となる。

#### 3.2 Cube Engine

図3に、Cube Engine のブロックダイアグラムを示す。CE は LM から格子点データを読み出し、ストリーム計算を行う。ストリーム計算においては、データを規則的かつ連続的に演算回路に流すことにより、連続して計算結果が出力される。これにより、外部メモリへの参照回数を抑えることができる他、規則的参照によりその帯域を有効に用いることが可能となり、演算器稼働率の向上が期待できる。本計算における入力データは各方向の速度  $u, v$ , 圧力  $p$ , および格子点属性情報  $CelAtb$  の4つである。これらを図3の CE に流すと、計算ユニットである仮速度計算ユニット、定数項計算ユニット、圧力計算ユニット、真速度計算ユニットによる複数パイプラインステージの

計算の後、更新後の  $u, v, p$  が出力される。

ただし、ステンシル計算においては、隣接格子点データを参照するため、同一格子点のデータを複数回参照する必要がある。このステンシル計算をストリーム化するために、図3に示すバッファモジュールを各計算ユニットの前に配置する。2次元キューブの辺の長さを  $N$  とすると、このモジュールは長さ  $(2N+2)$  のシフトレジスタである。シフトレジスタにおいて  $N+1$  番目のレジスタにあるデータを座標  $(i, j)$  とした場合、1番目には  $(i, j-1)$ ,  $2N+1$  番目には  $(i, j+1)$  の座標のデータが格納されている。これらを読み出せば、ステンシル計算に必要な隣接格子点のデータを得ることができる。以上により、外部メモリへの規則的かつ連続的なストリーム参照が可能となる。圧力計算ユニットを25段とした場合、図3の CE 1つあたりに必要なバッファサイズは  $340(2N+2)$  [Kbyte] となる。  $N=32$  のときは  $22.4$  [Kbyte] である。

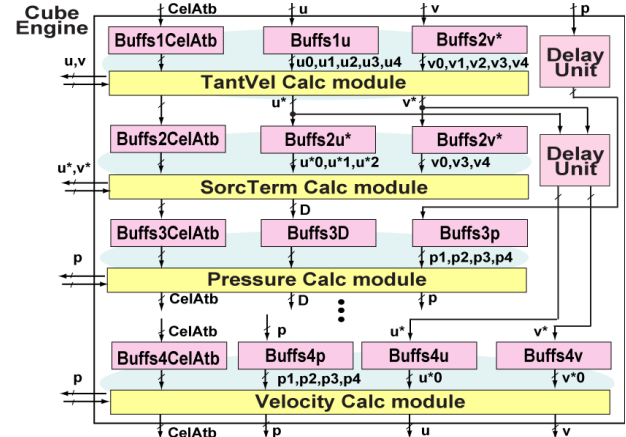


図3 CE のブロックダイアグラム

### 4. 結論

次世代の並列 CFD 計算手法である BCM の演算器稼働率の問題を解決するため、本研究では BCM 専用計算機アーキテクチャを提案している。本アーキテクチャは、メモリシステムのバンド幅に対し、BCM の演算密度を考慮した性能の計算回路を有する。また、キューブ間のデータ通信がボトルネックとならないようなデータ交換モジュールを持つ。本稿では、キューブを計算する回路である Cube Engine (CE) について、そのストリーム計算方法と内部構造の説明を行った。CE では、入力格子データをバッファに蓄えることにより、ステンシル計算のストリーム化を実現する。

今後は、CE における境界計算の検討や、キューブ間のデータ通信を行うためのモジュールの設計を行い、複数の FPGA による実装により提案アーキテクチャの有効性を評価する予定である。

#### 参考文献

[1] 佐野健太郎, 王陸州, 初田義明, 山本悟, “FPGA による数値流体力学専用計算機の設計と評価”, 第21回数値流体力学シンポジウム, G6-2, 2008.  
 [2] Nakahashi, K., “Building-Cube Method for Flow Problems with Broadband Characteristic Length,” Computational Fluid Dynamics 2002, edited by Armfield S., Morgan R., Srinivas K., Springer, pp. 77-81, 2003.  
 [3] 中橋和博, 佐々木大輔, 小林広明, 江川隆輔, 高橋俊, 新井紀夫, 東田学, 石井克哉, “次世代メタスケール CFD のアルゴリズム研究”, 学際大規模情報基盤共同利用・共同研究拠点 平成23年度共同研究 中間報告書, 2011