

# 宇宙物理アプリケーションのためのFPGA演算オフローディングの検討

埴 敏博<sup>1,a)</sup> 三木 洋平<sup>1</sup>

**概要:** 近年, 単精度浮動小数点の演算器を多数内蔵したFPGA(Field Programmable Gate Array)が登場している. 加えて三次元積層による高バンド幅メモリHBM2を搭載したFPGAが利用可能になっている. 本研究では宇宙物理アプリケーションを対象に, FPGAの特性を活かし, 可変精度も考慮に入れた演算オフローディングについて検討する. それに先立って, HBM2を使用したカーネルの記述方法やメモリバンド幅性能について確認した.

## 1. はじめに

高性能計算やデータ解析, 機械学習などにおいて, 高速かつ大量の演算処理が求められる中で, 今日では数多くのコアを備えたメニーコアプロセッサや, GPUなどに代表されるアクセラレータなど, 多数の演算コアを用いることによりチップ内部の並列度を向上させていくことで性能を高めている. しかし, 半導体プロセスの微細化が限界を迎える「ポストムーア」時代においては, チップ内ではコア単体性能の向上は見込めず, チップ面積の制約から, コア数を増加させることも困難になると考えられている.

そのような中で, 再構成可能なハードウェアとしてFPGA(Field Programmable Gate Array)が注目されている. アプリケーションに特化し, カスタマイズされたハードウェアを用いることで汎用プロセッサに比べて効率よく高い性能を発揮しうると考えられる.

そのため様々な用途に対するFPGAの活用が模索されており. 近年では回路設計技術に精通していなくてもFPGA上のハードウェアを設計する方法として, OpenCLが利用されるようになってきた. OpenCLは, マルチコアプロセッサやGPUなど, 様々な異なるプラットフォーム間での並列処理を容易にプログラムするためのプログラミング言語である[1]. 実際いくつかのFPGA製品においては, Verilog HDLなどを用いることなくOpenCLのみを用いて汎用のプログラムを作成することが可能であり, HPC分野におけるOpenCLプログラミングについても検討が進められている[2], [3], [4], [5], [6].

さらに, C++言語などのプログラミング言語での記述

から直接回路を生成できるような高位合成(High-Level Synthesis: HLS)コンパイラもFPGAベンダから提供されるようになってきた[7], [8].

しかしこれまでFPGAを高性能計算向けに用いるには, メモリバンド幅の不足が問題になってきた. 性能不足を補うために様々な工夫も行われてきたが, 本質的に適用範囲が限られ, また最適化の労力が大きかった. 近年, 三次元積層による高バンド幅メモリHBM2(High Bandwidth Memory Gen.2)を搭載することで, GPUに匹敵するメモリバンド幅が得られるFPGAが登場している.

このようなFPGAを用いて, 実アプリケーションに適用し, FPGAの特性を活かしたOpenCLによる平易なオフローディングの実現が望まれる. 本研究では, 宇宙物理アプリケーションとして $N$ 体シミュレーションを対象に, 可変精度も考慮に入れたアルゴリズムの演算オフローディングについて検討する.

## 2. OpenCLによるFPGAプログラミングと性能最適化

### 2.1 OpenCLを用いたFPGAプログラミング

FPGA内部の論理を設計するためには, 従来はVerilog HDLやVHDLといったハードウェア記述言語を用いて記述するのが一般的であり, 求められるアルゴリズムにあわせて手で論理回路レベルに変換する必要があった. そのため, 例えばC言語やFortranを用いれば数行で実装できるような単純な処理を行うだけでも, FPGA上に実装するためには多大な時間と労力が必要であり, 様々なHPCアプリケーションにFPGAを活用することは現実的ではなかった.

<sup>1</sup> 東京大学 情報基盤センター

<sup>a)</sup> hanawa@cc.u-tokyo.ac.jp

しかし近年では、OpenCL, C++などを用いた設計ツールがFPGAベンダーによって提供されるようになってきた。

OpenCL は Khronos グループによって標準化されている並列化プログラミング環境であり、GPU などのアクセラレータ向けに仕様策定や開発が進められたものである [1]. FPGA 向けのコンパイラでは、OpenCL での記述から内部で Verilog HDL や IP ライブラリを出力し、論理合成を行う。

Intel 社では、自社の FPGA に向けてホスト CPU の演算の一部をオフロードするためのアクセラレータとして FPGA を利用するための開発環境を提供している。この場合、FPGA は PCI Express 拡張ボードの形でホストに装着されるのが一般的である。

本研究では、FPGA として Intel 社の最新 FPGA である Stratix10 と、三次元積層されたメモリである HBM2 (High Bandwidth Memory) を 1 パッケージにした Stratix 10 MX を用いる。実際には Stratix 10 MX が搭載された、Intel 社の PCI Express 評価ボード Stratix 10 MX FPGA Development Kit (DK-DEV-1SMX-H-0ES および DK-DEV-1SMX-H-A) (図 1) を用いる。

本FPGAは、14nmプロセスで製造され、表1に示すように、2,073 K ロジックエレメント相当のAdaptive Logic Module (ALM) と呼ばれる論理モジュールで構成されている。また、FPGA チップ内部には、ハードウェアECCを備える、eSRAMと20KbitからなるM20Kの2種のSRAMメモリブロック、640 bitからなるMemory Logic Array Block (MLAB) が利用可能である。

さらに、固定小数点および浮動小数点算術演算をサポートする、可変精度 DSP (Digital Signal Processor) を備える。DSP 単体で、単精度浮動小数点の乗算、加減算、積和演算などをサポートしている。チップ全体として、理論ピーク性能では、単精度で 6.5 TFLOPS, 80 GFLOPS/W の演算能力を持つ。倍精度演算を行うためには、DSP を 4 つと付加回路を用いることで実現できる。以前の Stratix V 世代に比べ、Arria 10 世代では、浮動小数点演算をハードウェアで直接サポートする DSP が搭載されたことで、より多くの浮動小数点演算を実現することができるようになった。Stratix 10 世代では、さらに多くの DSP が搭載され、計算能力を高めている。

今回使用する Stratix 10 MX では、図 2 に示すように、HBM2 が 2 スタック搭載され、それぞれ 4 GB、計 8 GB の容量を持つ。また、理論ピークメモリバンド幅は 512 GB/sec となり、最新の HBM2 搭載 GPU のメモリバンド幅 1 TB/sec には及ばないが、従来の一般的な FPGA 評価ボードと比較すると、20 倍程度のバンド幅を持つ。

## 2.2 OpenCL における HBM2 の使用

これまでも FPGA に向けた OpenCL による記述, 処理

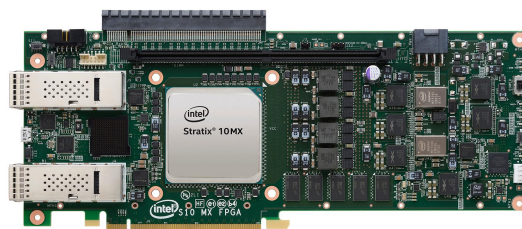


図 1 Intel 社 Stratix 10 MX FPGA Development Kit (Intel 社提供)

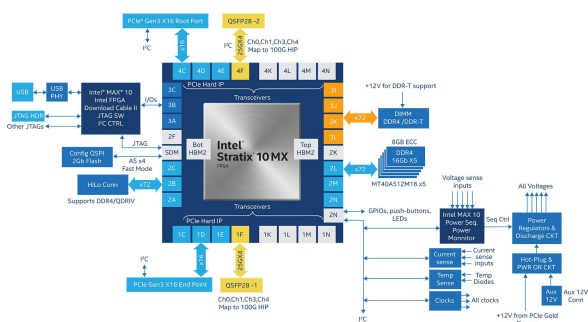


図 2 Stratix 10 MX FPGA Development Kit のブロック図 (Intel 社提供)

```
1 kernel void カーネル名 (
2   __global __attribute__((buffer_location("HBM0")))
      int *x,
3   __global __attribute__((buffer_location("HBM1")))
      int *v)
```

図 3 HBM メモリのチャネル指定方法

表 1 Stratix 10 MX FPGA の仕様

|                                 |                                                                   |
|---------------------------------|-------------------------------------------------------------------|
| FPGA                            | Intel Stratix 10 MX<br>1SM21BHU2F53E1VG                           |
| Logic Elements                  | 2,073 K                                                           |
| eSRAM                           | 94.5 Mbits                                                        |
| M20K                            | 6,847<br>134 Mbits                                                |
| MLAB                            | 17,568<br>11 Mbits                                                |
| DSP                             | 7,920<br>3,960 (FP32 乗算相当)                                        |
| HBM2<br>容量<br>バンド幅<br>スタックあたり接続 | 2 スタック<br>8 GB<br>512 GB/sec<br>128 bit× 8 ch<br>or 64 bit× 16 ch |
| PCI Express                     | Gen3 x16<br>(OpenCL で使えるのは x8)                                    |

方法についてはたびたび言及してきた [9], [10], [11]. Intel による開発環境 Intel FPGA SDK for OpenCL の最新版は 19.3 になり, レポート生成機能が充実したことや, Best Practice Guide[12] による順を追った最適化プロセスの解説などにより, 性能最適化は以前に比べると容易になってきた. 特に, Stratix 10 に向けた最適化の方針については個別に章を設けて記述されている.

ここでは, 今回利用した Stratix 10 MX FPGA Development Kit 特有の HBM2 の使用方法について説明する. OpenCL のための Board Support Package (BSP) は 2019 年 5 月に Intel から 19.1 向けに提供されたものを使用している.

表 1 に示すように, HBM2 は スタック当たり 8 または 16 チャンネルで管理されており, この BSP においては, 16 チャンネル, チャンネル当たり 256 MB の容量を持つ. 従って, FPGA 全体では 32 チャンネルのメモリを扱う.

実際に OpenCL のカーネルでこれらのチャンネルを使うためには, 明示的に OpenCL のカーネル引数で “buffer\_location” 属性を使って指定する必要がある. 例を図 3 に示す. この例では, ポインタ  $x$  を HBM0 チャンネルに, ポインタ  $y$  を HBM1 チャンネルに指定している.

### 3. 宇宙物理アプリケーション: $N$ 体シミュレーション

宇宙物理学の研究においては, 銀河や球状星団といった重力多体系の形成・進化過程を詳細に調べるために  $N$  体シミュレーションが用いられている.  $N$  体シミュレーションでは, 系を構成する粒子間にはたらく重力を計算し, 各粒子の軌道進化を追うことで系の時間発展を求める.  $N$  体

シミュレーションにおいては, 軌道積分に必要な加速度は Newton の運動方程式

$$\mathbf{a}_i = \sum_{j=0, j \neq i}^{N-1} \frac{Gm_j(\mathbf{r}_j - \mathbf{r}_i)}{(|\mathbf{r}_j - \mathbf{r}_i|^2 + \epsilon^2)^{3/2}} \quad (1)$$

によって与えられる. ここで  $m_i$ ,  $\mathbf{r}_i$ ,  $\mathbf{a}_i$  はそれぞれ  $i$  番目の粒子の質量, 位置, 加速度であり, また  $G$  は重力定数である. 重力ソフトニング  $\epsilon$  は,  $N$  体シミュレーションで用いる粒子数  $N$  が現実の系の粒子数よりも少ないために生じる人工的な効果を低減するために導入されており, ゼロ割による発散や自己相互作用を取り除くという役割も果たす.

$N$  体シミュレーションの高速化手法としては, ツリー法 [13] などの多重極展開に基づいて重力計算の演算量を削減する近似手法が一般的であるが, 一方で独立時間刻み法 [14] や階層化時間刻み法 [15] のように軌道積分の対象となる粒子数を削減するという手法もある. 多くの宇宙物理学の問題においては, 粒子ごとに軌道進化の早さが数桁に渡って異なるという状況が実現されるため, 後者の手法は宇宙物理学のアプリケーションに適したアルゴリズムである. こうした手法を同時に採用することも可能であり, 階層化時間刻み法を採用した GPU 向けの重力ツリーコード GOTHIC といった例もある [16], [17].

#### 3.1 対象コードの概要

我々は, 浮動小数点演算精度を適切に調節しながら重力計算・軌道積分を実行する高精度  $N$  体シミュレーションコードの開発に取り組んでいる. 軌道積分については階層化時間刻み法を用い, また衝突系向け  $N$  体シミュレーションにおいて広く採用されている時間 4 次精度の Hermite 法 [18] を採用した. 重力計算については, 直接法を用いて加速度と加速度の時間 1 階微分を計算しており, 倍精度・単精度演算を組み合わせた実装 [19] となっている.

#### 3.2 $N$ 体シミュレーションにおける粒子データのソート

$N$  体シミュレーションにおいて最も時間を要する部分は重力計算であるが, その他の関数の最適化も実行時間の短縮に寄与する. 独立時間刻み法や階層化時間刻み法を採用した場合には粒子時刻で, ツリー法を採用した場合には Morton 曲線や Peano-Hilbert 曲線といった空間充填曲線を用いて粒子データをソートするという処理が必要となる.

例えば GPU 向けの重力ツリーコード GOTHIC においては, 重力計算関数とツリー構造を再構築する関数の実行時間の総和が最小化されるように, ツリー構造の再構築頻度を自動調整している [16], [17]. これは, 粒子データの論理構造を表現したツリー構造と粒子の空間分布とのずれが小さいうちにはツリー構造の構築コストを削減できるという利点があるが, ずれが大きくなってくると重力計算関数の

実行時間が増大するために、ツリー構造の構築頻度を調整することでコード全体の実行時間を最小化するという設計思想に基づいている。ツリー構造の構築コストとしては空間充填曲線をソートするコストが支配的であるため、ソート関数の高速化はツリー構造の高頻度な再構築を実現することとなり、その結果として重力計算関数の実行時間も短縮されることが期待される。

## 4. 予備評価

本節では、Stratix 10 MX Development Kit を用いて、 $N$  体シミュレーションにおける演算オフローディングのための準備として、いくつかの予備評価を実施した。

本予備評価では、上記ボードの Early Silicon 版を用いた。この版は、HBM2 メモリ部分のクロックが製品版よりも 20%遅く、従って理論ピークメモリバンド幅は 409.6 GB/sec となる。Intel SDK for OpenCL は 19.3 を用い、OpenCL BSP は、2.2 節で述べた通り、Intel SDK for OpenCL 19.1 向けの 2019 年 5 月版を用いた。併せて古い版向けの BSP を使用するために必要な `QUARTUS_ROOTDIR_OVERRIDE` 環境変数を指定した。

### 4.1 メモリバンド幅とカーネル引数

BSP と同時に配布されている `hbm.boardtest.cl` を元にして、メモリバンド幅を測定した。`hbm.boardtest.cl` は、図 3 節での HBM メモリのチャンネル指定を用いて、チャンネルごとのメモリバンド幅や、HBM2 全体のメモリバンド幅を測定する。実際には stream ベンチマークの copy に相当する。

オリジナルでは、32 個のカーネルを起動し、各カーネルがそれぞれ HBM2 の各チャンネルを担当し、配列の copy を繰り返す。その結果、32 個のカーネルを個別に起動した結果は、平均でカーネルごとに 11.07 GB/sec のバンド幅が得られた。また、32 個のカーネルを同時に動作させた場合には、353.66 GB/sec のバンド幅が得られた。これは、理論ピークの 86.3%に相当する高い性能である。

続いて、プログラムを修正し、1 チャンネル 1 カーネルに分割せず、複数チャンネルを集約した場合についても調べた。バンド幅測定結果を表 2 に示す。この結果から、カーネルに使用チャンネル数を集約しても全体のメモリバンド幅はほとんど差がないことがわかった。

問題は、カーネル数を 1 にした場合で、このとき引数には、32 チャンネル分のソース、ディスティネーションの配列へのポインタを指定し、さらに 2 つのフラグを引数として指定しており、全体で 66 個の引数を必要とする。しかしながら、本 SDK でカーネル引数として使用できるバッファサイズが 256 バイトであり、64 個までしか指定することができない。

実アプリケーションでは、8 GB のメモリ容量を全て使

いたい場合や、物理量ごとにカーネルの引数として与えたい場合など、問題になると考えられる。その場合の解決策としては、

- 複数チャンネルをインターリーブするようなカーネルを入力段、出力段に置き、OpenCL channel で接続する
- チャンネル数を集約した BSP を用意する

の 2 つが考えられる。前者は制御が複雑になる可能性があり、リソースも余分に消費してしまう。後者は、どの程度の修正が必要か未知数である。実用に供することを考えると得失を含めた今後の検討が必要である。

### 4.2 演算精度の違いによる回路規模の比較

次に、演算精度を変えた場合に、回路規模がどのように変化するか、について調べた。Stratix 10 では、単精度 (FP32) の演算が可能な DSP を備えているが、OpenCL の言語仕様として倍精度 (FP64) と半精度 (FP16) も扱うことができる。

Intel が公開している行列演算のサンプル [20] を用いて、`aoc -c` を用いた場合の使用リソースの違いを表 3 示す。なお、いずれも配布時のデフォルトのパラメータを使用している。FP32 については、専用のハードウェアを持っているため、DSP 以外のリソースもあまり使用していないことがわかる。一方、FP64 の場合、DSP や Memory blocks については妥当な増加量であるが、その他のリソース、特に ALUTs, Logic utilization については 10 倍を超えている。また、FP16 に関しても、FP32 に比べて DSP や Memory blocks の使用量は減っているものの、他のリソースの使用量がかなり多い。FP16 に関しては、無駄な実装になっている可能性があり、調査の必要がある。必要に応じてコアの部分は Verilog HDL で記述し、ラッパーを用意することで、OpenCL から利用可能になる。詳細は後述する。

### 4.3 基数ソートの実現

階層化時間刻み法を採用した  $N$  体シミュレーションにおける粒子時刻 (浮動小数点数) をキーとした際のソート、及びツリー法における Peano-Hilbert key (符号なし整数) をキーとした際のソートを想定し、基数ソートの実装について検討した。

ソートのキーになる値は、整数同士の場合と浮動小数点数同士の場合とがある。しかし浮動小数点の場合には、符号ビットを取り除き、負の数だった場合には 2 の補数表現に置き換えるだけで、整数とみなして比較することが可能になる。従って、基数ソート自体は整数のみについて考えればよい。

まず、浮動小数点数の変換について、ビット演算は Verilog HDL の方が容易に記述できるため、Verilog HDL の記述を OpenCL 用のライブラリとして実現した。これを用いた FP32 での基数ソートについて試験実装が一通り完成し

表 2 カーネル数を変えたときのメモリバンド幅 (GB/sec)

| カーネル数          | 32 (オリジナル) | 16     | 8      | 4      | 2      | 1  |
|----------------|------------|--------|--------|--------|--------|----|
| カーネル毎の使用チャンネル数 | 1          | 2      | 4      | 8      | 16     | 32 |
| メモリバンド幅        | 353.66     | 354.59 | 355.97 | 355.85 | 354.31 | 不可 |

たが、性能測定、および最適化を実施する予定である。実際には、実アプリケーションの粒子データ構造を反映した形に変更する必要がある。また、メンバー数や精度に応じた基数やソートアルゴリズム自体の検討も必要だと考えられる。

#### 4.4 粒子データの構成

粒子データのソートに要する実行時間は、ソートのキーに用いられる配列に格納された粒子時刻や Peano-Hilbert key の分布に依存する。したがって、実際の宇宙物理学の研究で用いられる  $N$  体シミュレーション中での分布と近い粒子分布を生成した上で、実際に  $N$  体シミュレーションを実行しながらソート時間の性能評価を実施する必要がある。本研究では、楕円銀河モデル及びアンドロメダ銀河を模したモデルを、銀河向けの初期条件生成コード MAGI[21] を用いて生成する。このとき、楕円銀河モデルにおけるブラックホール粒子以外の粒子質量を等質量とするように設定する。

楕円銀河モデルを表現したモデルは、ダークマターハロー (Einasto モデル [22], [23], [24], 質量  $M = 10^{12} M_{\odot}$ , スケール長  $r_s = 20$  kpc, ベキパラメータ  $\alpha = 0.2$ ), バルジ (de Vaucouleurs 則 [25], 質量  $M = 5 \times 10^{10} M_{\odot}$ , 有効半径  $R_{\text{eff}} = 1.1$  kpc), 中心ブラックホール (質量  $M = 10^8 M_{\odot}$ ) からなるモデルである。アンドロメダ銀河を模したモデルは、Fardal らによって構築された銀河モデル [26], [27] に最近の観測に基づく恒星ハロー成分 [28], [29] を追加したものであり、ダークマターハロー (Navarro-Frenk-White モデル [30], 質量  $M = 8.11 \times 10^{11} M_{\odot}$ , スケール長  $r_s = 7.63$  kpc), 恒星ハロー (Sérsic モデル [31], 質量  $M = 8 \times 10^9 M_{\odot}$ , スケール長  $r_s = 9$  kpc, Sérsic 指数  $n = 2.2$ ), バルジ (Hernquist モデル [32], 質量  $M = 3.24 \times 10^{10} M_{\odot}$ , スケール長  $r_s = 0.61$  kpc), 銀河円盤 (等温・指数関数型 [21], 質量  $M = 3.66 \times 10^{10} M_{\odot}$ , スケール長  $R_s = 5.4$  kpc, 厚み  $z_s = 0.6$  kpc, Toomre の  $Q$  値の最小値  $Q_{\min} = 1.8$  [33]) からなる。

## 5. おわりに

本稿では、宇宙物理アプリケーションとして  $N$  体シミュレーションを対象に、可変精度も考慮に入れたアルゴリズムの演算オフローディングについて検討した。また、HBM2 搭載 FPGA である Stratix 10 MX を用いて、実現に向けた予備評価を行った。予備評価で得られた知見を元に、基

表 3 リソース使用量 (aoc -c による) (%)

|                           | FP32 | FP64 | FP16 |
|---------------------------|------|------|------|
| Logic utilization         | 6    | 67   | 33   |
| ALUTs                     | 3    | 34   | 24   |
| Dedicated logic registers | 4    | 35   | 12   |
| Memory blocks             | 11   | 17   | 9    |
| DSP blocks                | 7    | 26   | 4    |

数ソートから実装を開始している。

可変精度に関する検討については、他のアプリケーションにおいても有効である。可変精度を実現するエミュレータとしての役割と、それを踏まえつつ Stratix 10 FPGA で実行する上で最適な方式の実現との双方で、OpenCL 向けのライブラリ化を検討していく予定である。

さらに、筑波大計算科学研究センターに導入された、GPU + FPGA 混載クラスタ Cygnus 上で、GPU、FPGA 両者の特徴を活かして使い分けつつ、複数ノードを使った高速な処理の実現についても検討していく。

謝辞 本研究の一部は、JSPS 科研費 18H03246、19H04122 の助成を受けたものです。また、本研究の一部は、“Intel University Program”を通じてハードウェアおよびソフトウェアの提供を受けて実施した。ここに Intel の支援に感謝する。

## 参考文献

- [1] OpenCL - The open standard for parallel programming of heterogeneous systems: .
- [2] 丸山 直也, Zohouri, H. R., 松田 元彦, 松岡 聡: OpenCL による FPGA の予備評価, 情報処理学会研究報告 (2015-HPC-150) (2015).
- [3] 大島 聡史, 塙 敏博, 片桐 孝洋, 中島 研吾: FPGA を用いた疎行列数値計算の性能評価, 情報処理学会研究報告 (2016-HPC-153) (2016).
- [4] ウィッデヤスーリヤハシタ ムトゥマラ他: OpenCL を用いたステンスル計算向け FPGA プラットフォーム, 情報処理学会研究報告 (2016-HPC-154) (2016).
- [5] Zohouri, H. R., Maruyama, N., Smith, A., Matsuda, M. and Matsuoka, S.: Evaluating and optimizing OpenCL kernels for high performance computing with FPGAs, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'16)* (2016).
- [6] 藤田 典久, 他: OpenCL を用いた FPGA による宇宙輻射輸送シミュレーションの演算加速, 情報処理学会研究報告 (2016-HPC-161), No. 12.
- [7] HLS Compiler: Fast Design, Coding, and Hardware: (online), available from ([https://www.altera.co.jp/content/dam/altera-www/global/en\\_US/pdfs/literature/wp/wp-01274-intel-hls-compiler-fast-design-coding-and-hardware.pdf](https://www.altera.co.jp/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01274-intel-hls-compiler-fast-design-coding-and-hardware.pdf)).
- [8] Vivado Design Suite: (online), available from ([https://japan.xilinx.com/support/documentation/sw\\_manuals\\_j/j\\_ug1197-vivado-high-level-productivity.pdf](https://japan.xilinx.com/support/documentation/sw_manuals_j/j_ug1197-vivado-high-level-productivity.pdf)).
- [9] 塙 敏博, 伊田 明弘, 大島 聡史, 河合 直聡: FPGA を用いた階層型行列ベクトル積, 情報処理学会研究報告 (2016-HPC-155) (2016).
- [10] 塙 敏博, 伊田 明弘, 星野 哲也: 階層型行列計算の FPGA への適用, 情報処理学会研究報告 (2017-HPC-161) (2017).
- [11] 塙 敏博, 伊田 明弘, 星野 哲也: OpenCL を用いた FPGA による階層型行列計算, 情報処理学会研究報告 (2018-HPC-163) (2018).
- [12] Intel Corporation: *Intel FPGA SDK for OpenCL Best Practice Guide 19.3* (2019).
- [13] Barnes, J. and Hut, P.: A hierarchical  $O(N \log N)$  force-calculation algorithm, *Nature*, Vol. 324, pp. 446–449 (online), DOI: 10.1038/324446a0 (1986).
- [14] Aarseth, S. J.: Dynamical evolution of clusters of galaxies, I, *Monthly Notices of the Royal Astronomical Society*, Vol. 126, p. 223 (online), DOI: 10.1093/mnras/126.3.223 (1963).
- [15] McMillan, S. L. W.: The Vectorization of Small-N Integrators, *The Use of Supercomputers in Stellar Dynamics* (Hut, P. and McMillan, S. L. W., eds.), Lecture Notes in Physics, Berlin Springer Verlag, Vol. 267, p. 156 (online), DOI: 10.1007/BFb0116406 (1986).
- [16] Miki, Y. and Umemura, M.: GOTHIC: Gravitational oct-tree code accelerated by hierarchical time step controlling, *New Astronomy*, Vol. 52, pp. 65–81 (online), DOI: 10.1016/j.newast.2016.10.007 (2017).
- [17] Miki, Y.: Gravitational Octree Code Performance Evaluation on Volta GPU, *Proceedings of the 48th International Conference on Parallel Processing, ICPP 2019*, New York, NY, USA, ACM, pp. 62:1–62:10 (online), DOI: 10.1145/3337821.3337845 (2019).
- [18] Makino, J. and Aarseth, S. J.: On a Hermite Integrator with Ahmad-Cohen Scheme for Gravitational Many-Body Problems, *Publications of the Astronomical Society of Japan*, Vol. 44, pp. 141–151 (1992).
- [19] Nitadori, K., Makino, J. and Hut, P.: Performance tuning of N-body codes on modern microprocessors: I. Direct integration with a hermite scheme on x86\_64 architecture, *New Astronomy*, Vol. 12, pp. 169–181 (online), DOI: 10.1016/j.newast.2006.07.007 (2006).
- [20] Intel Corporation: *Matrix Multiplication Design Example*.
- [21] Miki, Y. and Umemura, M.: MAGI: many-component galaxy initializer, *Monthly Notices of the Royal Astronomical Society*, Vol. 475, pp. 2269–2281 (online), DOI: 10.1093/mnras/stx3327 (2018).
- [22] Einasto, J.: On the Construction of a Composite Model for the Galaxy and on the Determination of the System of Galactic Parameters, *Trudy Astrofizicheskogo Instituta Alma-Ata*, Vol. 5, pp. 87–100 (1965).
- [23] Navarro, J. F., Hayashi, E., Power, C., Jenkins, A. R., Frenk, C. S., White, S. D. M., Springel, V., Stadel, J. and Quinn, T. R.: The inner structure of  $\Lambda$ CDM haloes - III. Universality and asymptotic slopes, *Monthly Notices of the Royal Astronomical Society*, Vol. 349, pp. 1039–1051 (online), DOI: 10.1111/j.1365-2966.2004.07586.x (2004).
- [24] Navarro, J. F., Ludlow, A., Springel, V., Wang, J., Vogelsberger, M., White, S. D. M., Jenkins, A., Frenk, C. S. and Helmi, A.: The diversity and similarity of simulated cold dark matter haloes, *Monthly Notices of the Royal Astronomical Society*, Vol. 402, pp. 21–34 (online), DOI: 10.1111/j.1365-2966.2009.15878.x (2010).
- [25] de Vaucouleurs, G.: Recherches sur les Nebuleuses Extragalactiques, *Annales d'Astrophysique*, Vol. 11, p. 247 (1948).
- [26] Geehan, J. J., Fardal, M. A., Babul, A. and Guhathakurta, P.: Investigating the Andromeda stream - I. Simple analytic bulge-disc-halo model for M31, *Monthly Notices of the Royal Astronomical Society*, Vol. 366, pp. 996–1011 (online), DOI: 10.1111/j.1365-2966.2005.09863.x (2006).
- [27] Fardal, M. A., Guhathakurta, P., Babul, A. and McConnachie, A. W.: Investigating the Andromeda stream - III. A young shell system in M31, *Monthly Notices of the Royal Astronomical Society*, Vol. 380, pp. 15–32 (online), DOI: 10.1111/j.1365-2966.2007.11929.x (2007).
- [28] Gilbert, K. M., Guhathakurta, P., Beaton, R. L., Bullock, J., Geha, M. C., Kalirai, J. S., Kirby, E. N.,

- Majewski, S. R., Ostheimer, J. C., Patterson, R. J., Tollerud, E. J., Tanaka, M. and Chiba, M.: Global Properties of M31's Stellar Halo from the SPLASH Survey. I. Surface Brightness Profile, *The Astrophysical Journal*, Vol. 760, p. 76 (online), DOI: 10.1088/0004-637X/760/1/76 (2012).
- [29] Ibata, R. A., Lewis, G. F., McConnachie, A. W., Martin, N. F., Irwin, M. J., Ferguson, A. M. N., Babul, A., Bernard, E. J., Chapman, S. C., Collins, M., Fardal, M., Mackey, A. D., Navarro, J., Peñarrubia, J., Rich, R. M., Tanvir, N. and Widrow, L.: The Large-scale Structure of the Halo of the Andromeda Galaxy. I. Global Stellar Density, Morphology and Metallicity Properties, *The Astrophysical Journal*, Vol. 780, p. 128 (online), DOI: 10.1088/0004-637X/780/2/128 (2014).
- [30] Navarro, J. F., Frenk, C. S. and White, S. D. M.: Simulations of X-ray clusters, *Monthly Notices of the Royal Astronomical Society*, Vol. 275, pp. 720–740 (online), DOI: 10.1093/mnras/275.3.720 (1995).
- [31] Sérsic, J. L.: Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy, *Boletín de la Asociación Argentina de Astronomía La Plata Argentina*, Vol. 6, p. 41 (1963).
- [32] Hernquist, L.: An analytical model for spherical galaxies and bulges, *The Astrophysical Journal*, Vol. 356, pp. 359–364 (online), DOI: 10.1086/168845 (1990).
- [33] Tenjes, P., Tuvikene, T., Tamm, A., Kipper, R. and Tempel, E.: Spiral arms and disc stability in the Andromeda galaxy, *Astronomy and Astrophysics*, Vol. 600, p. A34 (online), DOI: 10.1051/0004-6361/201629991 (2017).