

Arria 10 SoC を用いた Full-PIC 法における割付処理の高速化

野田 裕之† 酒井 諒太郎† 宮島 敬明‡ 天野 英晴†

†慶應義塾大学 理工学部 情報工学科 ‡宇宙航空研究開発機構 航空技術部門 数値解析技術研究ユニット

1 はじめに

Full Particle-In-Cell(Full-PIC) 法は、ホールスラストの研究開発で用いられる数値シミュレーション手法である。ホールスラストは、電気エネルギーを利用して推力を得る電気推進エンジンの一種で、燃費の指標である比推力が高いことから人工衛星の姿勢制御や軌道間輸送に利用される。Full-PIC 法は、イオン・中性子・電子の全てを粒子として扱うため、粒子を流体モデルに近似させる他の手法と比べ高精度であるが、計算コストが高いことが問題とされる [1]。宇宙航空研究開発機構 (JAXA) が開発を進めるシミュレーション・コード”NSRU-Full-PIC”は、Full-PIC 法を用いており、処理に膨大な時間を要する。

我々は NSRU-Full-PIC の高速化のため、エネルギー効率に優れた SoC FPGA クラスタによるパラメータサーベイを検討している [2]。Altera 社のミッドレンジ SoC FPGA である Arria 10 SoC は、ARM プロセッサを内蔵することで簡単に Linux が動作し、浮動小数演算用の DSP ブロックを持つことから演算性能にも優れている。本稿では、このボードを用いたホールスラスト・シミュレーション用 FPGA クラスタ構築の第一段階として、Altera 社が提供する FPGA 向け OpenCL を用いて、Full-PIC 法の中で特に高負荷である割付処理のうち、リダクション演算部のオフローディングを行った結果を述べる。

2 Full-PIC 法における割付処理

2.1 Full-PIC 法の流れ

Full-PIC 法の流れを図 1 に示す。Full-PIC 法では、電場を計算するためのグリッドが用意されており、各粒子はそのモデル上を自由に移動する。図 1 におけるステップ 4 は、「電荷の外挿処理」を行うステップである。「電荷の外挿処理」には大別して 2 つの処理が存在し、1 つ目は粒子をグリッドの四隅に割り付ける処理 (割付処理)、2 つ目は粒子の物理量をもとにポアソン方程式

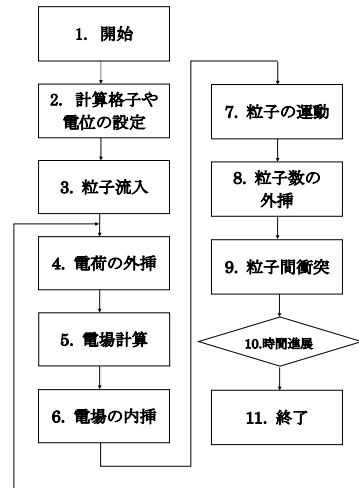


図 1: Full-PIC 法の流れ

を解く処理である。

2.2 予備評価

京都大学が所有するスーパーコンピュータ Comphor (CRAY XE6) 128 コアにおける、NSRU-Full-PIC の処理時間の分布を図 2 に示す。「電荷の外挿処理」の処理時間は、NSRU-Full-PIC の全処理時間のうち約 4 割を占めた。また、Arria 10 SoC 上の ARM Cortex-A9 における「電荷の外挿処理」の処理時間の分布を図 3 に示す。割付処理の処理時間は「電荷の外挿処理」全体の約 7 割を占めた。

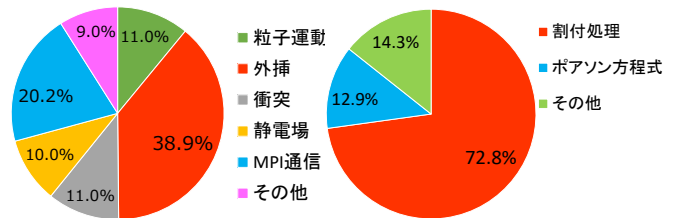


図 2: NSRU-Full-PIC 全体 図 3: "電荷の外挿処理" における処理時間の分布 (ARM (CRAY XE6 128 コア) Cortex-A9)

2.3 Full-PIC 法における割付処理

ひとつのグリッド内に複数の粒子が存在するため、割付処理の並列化には Read After Write(RAW) ハザード

A study of acceleration of interpolation part in Full-PIC simulation using Arria 10 SoC
 Hiroyuki Noda† Ryotaro Sakai† Takaaki Miyajima‡
 Hideharu Amano†
 †Keio University
 ‡Japan Aerospace Exploration Agency

の回避が必要である。先行研究では、NSRU-Full-PICの高負荷部を Graphics Processing Unit(GPU) にオフロードし、割付処理部には排他処理の一種であるアトミック処理を適用した [2]。アトミック処理は、ある操作が完了するまで他の操作を停止させるため非効率である。一方、ひとつのグリッド内に存在する粒子の物質量を全て加算してからその結果を四隅に割り付けることで、RAW ハザードを回避しつつ効率のよい演算を行うことが可能であると考えた。

3 実装と評価

Full-PIC 法の割付処理のうち、リダクション処理部を Arria 10 SoC に実装し、CPU 実行と比較した。Arria 10 SoC は、ARM Cortex-A9 と Field-Programmable Gate Array(FPGA) が 1 つに結合されており、CPU-FPGA 間の効率の良いメモリアクセスが可能である。また、科学技術計算を扱う上で重要な素子となる、浮動小数点演算用の Digital Signal Processing(DSP) が搭載されている。先行研究として、Xilinx 社のローエンド SoC FPGA である Zynq を用いて Full-PIC 法の高速化を行った研究が存在する [3]。Zynq は搭載されている FPGA リソースが少ないため、オフロード箇所が限定される。このため、本研究ではミッドレンジ SoC FPGA である Arria 10 SoC を用いた。実装には、FPGA 向け OpenCL ベース設計環境である Altera SDK for OpenCL を用いた。図 4 に、Antel SDK for OpenCL における並列モデルの概要を示す。カーネルコードによってパイプラインが生成され、スレッド空間 (work-item) が順次流し込まれる。今回は、Single work-item と呼ばれる並列モデルを使用した。このモデルでは、コンパイラが並列性を抽出し、ループをパイプライン化する。

実装環境を以下に示す。

- SoC: Arria 10 SoC
 - CPU: ARM Cortex-A9 (1.5GHz)
 - FPGA: SX660
- OS: Linaro Ubuntu 14.04.5 LTS
- Kernel: 4.1.22-ltsi
- コンパイラ: aocl 16.0.2.222 , gcc 4.8.3
- ツール: Altera SDK for OpenCL 16.0.2 Build 222

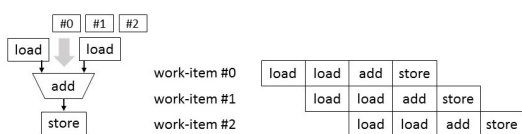


図 4: Antel SDK for OpenCL における並列モデルの概要

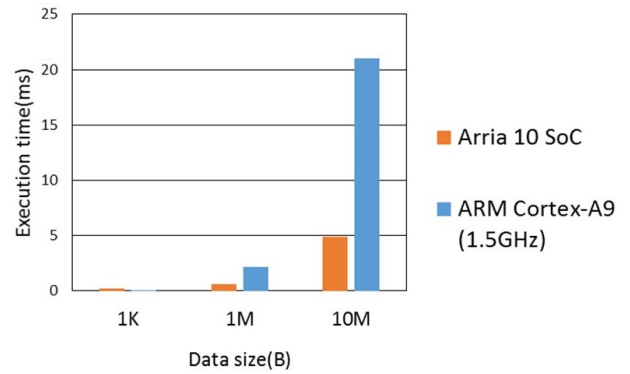


図 5: Arria 10 SoC と ARM Cortex-A9 の実行時間の比較

表 1: 本実装における FPGA リソース使用量

Logic utilization	ALUTs	Registers	Memory blocks	DSPs
8%	5%	4%	17%	8%

図 5 に、Arria 10 SoC と ARM Cortex-A9(1.5GHz)でのリダクション処理の実行時間の比較を示す。結果として、CPU 実行と比較して最大で 4.30 倍の高速化を達成した。また、表 1 に本実装における Arria 10 SoC の FPGA リソース使用量を示す。

4 結論

本稿では、Full-PIC 法の高負荷部である割付処理のうち、リダクション演算部を Arria 10 SoC に実装した。CPU 実行と比較して、最大で 4.30 倍の高速化を達成した。

参考文献

- [1] Sigeru Yokota, Kimiya Komurasaki, and Yoshihiro Arakawa. Plasma density fluctuation inside a hollow anode in an anode-layer hall thruster. Reston, Va, 2006. American Institute of Aeronautics and Astronautics.
- [2] Takaaki Miyajima, Shinatora Cho, and Naoyuki Fujita. A study of gpu acceleration of "source" part in hall-thruster simulation. In *IEICE Tech. Rep.*, Vol. 115 of *CPSY2015-62*, pp. 7–12, Dec. 2015.
- [3] R. Sakai, N. Sugimoto, T. Miyajima, N. Fujita, and H. Amano. Acceleration of full-pic simulation on a cpu-fpga tightly coupled environment. In *2016 IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSOC)*, pp. 8–14, Sept 2016.