

FPGA-GPU 混載プラットフォームにおける HPC アプリケーションとシステム・ソフトウェアの開発

HPC application and system software development on FPGA-GPU combined platform

朴 泰祐

筑波大学計算科学研究センター

1. 研究目的

本研究では、FPGA の持つ潜在能力を HPC システムに最大限適用するため、これを GPU 等の演算加速装置と連携させ双方を相補的に利用し、さらに最新 FPGA の持つ高性能通信機能も含めた統合的な複合演算加速システムを実現する。GPU では十分な演算加速ができないアプリケーションについて、部分的な FPGA オフローディングと、高速並列通信を組み合わせた新しい FPGA の利用方法を実装・評価し、具体的な実アプリケーションを適用することにより、次世代の統合的演算加速並列システムの枠組みを構築しその有効性を実証する。さらに、最新型 GPU を用いた演算加速プログラミングに関する新しいフレームワーク適用の試みと単一コード記述による GPU と FPGA のプログラミングに関しても開拓する。

2. 研究成果の内容

- ・複数演算加速デバイス間のスケジューリングとデータ共有を簡単化するプログラミング環境として CHARM-SYCL を開発し、米国 Oak Ridge National Laboratory との共同研究により米国 ECP で開発されたベンチマークコードの実施を行った。ORNL との国際共同研究により、同研究書で開発された IRIS ランタイムシステムを我々が開発中の SYCL シングルプラットフォームによる異種アクセラレータ実行環境に適用し、これらのアクセラレータを効率的に利用可能であることを示した。
- ・高性能計算向け並列 FPGA 環境における FPGA 間通信システムを Ethernet 経由方式と直接リンク方式の両方式で実装した。特に後者に関しては、エラー検出機能を新たに導入し、安定した通信を実現するための予備実装を行った。メッセージサイズに一定の上限を設けることでエラーフリーで安定した FPGA 間相互通信が可能となり、Ethernet スイッチに依存しない、FPGA 上の演算と通信をシームレスにパイプライン実行する環境を構築した。
- ・GPU と FPGA を統一的にプログラミングすることを可能とする Intel oneAPI フレームワークを実アプリケーションである宇宙物理学コード ARGOT に適用し、GPU 実行における OpenACC 及び CUDA での実行ケースとの性能比較を行った。Cygnus における oneAPI 環境では、CUDA 実装に比べ oneAPI でも同等の性能が実現でき、

FPGA との統一的プログラミング環境が実現可能であることを原理的に実証した。

・演算加速装置間の通信フレームワーク開発の経験を活かし、富岳スーパーコンピュータ上の UCX 通信プラットフォームにおけるメニーコアプロセッサ間通信の最適化と性能評価を行った。

3. 学際共同利用プログラムが果たした役割と意義

本研究課題で行った GPU・FPGA の協業による多重複合演算加速手法は本プログラムで提供された Cygnus システムがなければ実施できなかった。また、FPGA 向け OpenACC コンパイラも Cygnus の Intel FPGA を用いて実装した。FPGA 間通信システムも同様に Cygnus 上で開発された。以上のように、本課題の研究は Cygnus という他に類を見ないシステムを用いて実施することができた。

4. 今後の展望

Cygnus スーパーコンピュータの運用終了に伴い、大規模な GPU・FPGA 統合プログラミング及びシステム構築の試みは一旦終了となるが、計算科学研究センター内の小規模複合演算加速システムで引き続き開発を継続する。また、本課題において発展的に開発された CHARM-SYCL システムを UniSYCL と改名し、引き続き ORNL との国際共同研究、またこれに加え Virginia Tech. 大学も加えたより大きな国際共同研究体制で異種複合演算加速プログラミングシステムの研究を R07 年度課題として継続的に実施する。

5. 成果発表

- (1) 学術論文
- (2) 学会発表
- (3) その他

使用計算機	使用計算機に ○	配分リソース※		
		当初配分	移行*	追加配分
Cygnus	○	10,000	-2,000	
Pegasus	○	0	+2,000	
Wisteria/BDEC-01				
※配分リソースについてはノード時間積をご記入ください。				
*バジェット移行を行った場合、「+2000」「-1000」のように記入				