Vol.2015-HPC-150 No.44
2015/8/6

OpenCLによるFPGAの予備評価

丸山 直也[1,2,a)]　Hamid Reza Zohouri[2,b)]　松田 元彦[1,c)]　松岡 聡[2,d)]

概要：本稿では Altera Stratix V を対象に OpenCL による FPGA の利用についてその予備的な評価を行い、その現状結果を報告する。FPGA は従来 HDL によるハードウェアレベルの記述が必要であったが、OpenCL のようなある程度の抽象度を持った記述系が利用可能になってきつつあり、今後のより広い応用分野での有効性が期待されつつある。本稿では今後の詳細な調査に向けて、まず既存の OpenCL ベンチマークを用い、性能について予備的な評価を行う。その結果、現状では単純に既存の OpenCL プログラムを用いただけでは高い性能を得ることは難しく、より FPGA に適したプログラミングが必要であることがわかった。

1. はじめに

科学技術シミュレーションなどの高性能計算において電力効率の改善を目的としたアクセラレータの利用が進んでいる。GPU や Xeon Phi などのすでに大規模スーパーコンピュータにおいて採用が進んでいるデバイスや、組み込み系プロセッサを高密度実装したメニーコアアクセラレータなど今後の展開が期待されるデバイスなど少なくない種類のアクセラレータが登場している。その一つとして、再構成可能プロセッサ、特に Field Programmable Gate Array（FPGA）が注目されつつある。FPGA はこれまでもその再構成可能なロジックによって、対象アプリケーションによっては汎用プロセッサに比べて顕著な性能優位性が存在することが多くの研究において示されてきている。特に今日においては電力効率の観点からも注目を浴びつつあり、代表的な例としては FPGA ベースのデータセンタアーキテクチャである Catapult 等があげられる [1]。Catapult ではウェブサーチ等のワークロードを FPGA によって電力効率を大幅に向上させられることを実証した研究であり、現在大規模な実運用が進められていると言われている。一方、これまで FPGA のプログラミング、すなわち「再構成」のためには Verilog 等のハードウェア記述言語によってハードウェアレベルでアルゴリズムを実装する必要があり、一般のアプリケーションプログラマにとって敷居が極端に高い問題がある。数十行程度の通常の CPU 向け C コード等を FPGA 向けに記述するだけでも数ヶ月程度のプログラミングが必要とも言われ、一般的な計算科学等における利用シナリオには利用は非常に限定的であると言える。

FPGA におけるプログラミングを容易にするための取り組みとして各種のハイレベル記述系の研究開発が実施されてきた。特に、アクセラレータ向け標準プログラミング仕様である OpenCL [2] への FPGA の対応が進んできており、主要な FPGA ベンダーからは OpenCL コンパイラおよびランタイムが提供されている。OpenCL は標準化団体 Kronos によってまとめられている標準仕様であり、2009 年に最初のバージョンがリリースされ、現在はバージョン 2.0 が最新版である。各社の GPU や、Xeon Phi などの主要なアクセラレータ、また通常の Inte l 系 CPU をターゲットプロセッサとした実装が各社ベンダーから提供されており、これにより既存の CPU 向けプログラムとアクセラレータを使ったハイブリッド実装の機種独立なプログラミングを可能になっている。新たに FPGA のサポートが進展してきており、これにより仕様上は単一のプログラムを GPU や Xeon Phi から FPGA まで幅広い環境で動作させることが可能となっている。GPU の科学技術計算における利用は数年前から進んでおり、これまでに CUDA や OpenCL で記述されたアプリケーションも多く存在する。それらの既存アプリケーション資産も CUDA から OpenCL に移植するか、もしくはそのまま FPGA 対応 OpenCL コンパイラを使うことで FPGA を使うことが可能である。OpenCL が提供する抽象化は通常のプロセッサ向けプログラミングモデルとしては十分なものとは言えず、そのプログラミングの生産性は必ずしも高いものではないが、FPGA における既存のハードウェア記述言語に比

[1] 理化学研究所計算科学研究機構
[2] 東京工業大学
[a)] nmaruyama@riken.jp
[b)] zohouri.h.aa@m.titech.ac.jp
[c)] m-matsuda@riken.jp
[d)] matsu@is.titech.ac.jp

較すれば大幅にその生産性を改善するものであり、実際に幅広く利用可能となるためには最低でも OpenCL 程度のプログラミングの抽象化は必須である。また、特に科学技術計算では浮動小数点演算性能が重要であるが、これまでの FPGA では主に整数演算向けに構成されており浮動小数点演算のサポートは限定的であった。しかし、Altera 社の Stratix 10 などの FPGA では単精度ながら 10TFLOPS 級の性能がアナウンスされており、DSP の搭載やハードマクロでの浮動小数点演算サポートなどによって今後改善されていくと想定される。このように、ハードウェアおよびソフトウェアから様々な要素技術の整備が進んでおり、スーパーコンピュータ等の高性能計算システムにおいて構成するデバイスとして着目されるべき周辺技術が整いつつある。

我々は特に大規模な計算規模を必要とするアプリケーションを対象とした、FPGA の有効性の評価検討を進めている。初期検討として OpenCL を用いた場合の性能やプログラミングについて評価を進めており、本稿では既存の OpenCL ベンチマークセットを用いた評価について報告する。具体的にはベンチマークとして Rodinia [3] を用いる。Rodinia は motif [4] と呼ばれる種々の計算パターン毎にベンチマークプログラムを提供しており、それぞれについて OpenMP, CUDA, OpenCL など共有メモリ環境向け並列実装が複数提供されている。我々は Rodinia ベンチマークから 3 本のベンチマークを選択し、その Altera FPGA である Stratix V における評価実験を行った。評価にはオリジナルの Rodinina ベンチマークの実装を可能な限り変更せずに利用した場合の性能、および Altera 社から提供されているプログラミングガイド等で推奨されている最適化を一部施した場合の性能を評価した。現状では評価は進行中であり、これまでに得られている結果は限定的なものであるが、今後の網羅的な評価に向けた指針を得ることができた。

2. 評価手法

我々の目的は FPGA の OpenCL による有効性の評価であり、本稿では FPGA として Bittware 社の PCI ボード S5-PCIe-HQ (S5PHQ) [5] を用いる。S5PHQ は 4GB の DDR3 DRAM を 2 バンク搭載しており、また 18MB の QDRII メモリを 4 バンク搭載し、FPGA として Altera の Stratix V GS を搭載している。Stratix は Altera のハイエンド FPGA であり、Stratix V は 2010 年にリリースされた比較的古い FPGA であるが、現在一般に利用可能な Altera 製品としては最新世代のものである。Stratix V GS は Stratix V の一種であり、表 1 に示すように Adaptive Logic Module とよばれる論理ユニットを 262,400 備えており、また 3926 の DSP ブロック、2,567 のメモリモジュールなどを搭載している。

表 1 Stratix V GS Specification

#Logic units	262,400
#Memory modules	2,567
#DSP blocks	3,926

S5PHQ をターゲットとした OpenCL 環境として本稿では Altera 社から提供されている Altera OpenCL SDK [6] を用いる。これは Stratix V などの Altera FPGA を対象とした OpenCL コンパイラやランタイムから構成され、Altera 社から 2013 年より有償で提供されている。FPGA を対象とした OpenCL 実装としてはその他に Owaida らによる SOpenCL [7] などが提案されており、我々と同様に OpenCL ベンチマークを用いた評価も一部報告されている [8]。本稿では広く入手可能な処理系として Altera OpenCL SDK を用いるが、今後その他の処理系およびその他の FPGA についても評価を進める予定である。

我々の評価の目的の一つに標準化された OpenCL によって他のアクセラレータと同様に実際に FPGA が利用可能かどうかを調査することがあげられる。従って、評価に用いるプログラムは既存の FPGA 向けに記述された OpenCL プログラムだけでなく、既存、特に GPU 等のアクセラレータ向けに記述されたプログラムを用いることが望ましい。また、他のアクセラレータや通常の CPU との性能比較も重要であり、OpenCL に限らずそれぞれのプロセッサに適した実装を用いた評価が望ましい。さらに、特定の限られた数の計算カーネルによる評価ではなく、なるべく広い範囲の並列計算を網羅した評価を行うことを狙っている。

アクセラレータの評価を広い範囲の並列計算について実施することを目的としたベンチマークセットして Rodinia ベンチマークがあげられる [3]。同ベンチマークでは motif と呼ばれる構造格子、密行列計算などの計算パターン毎にベンチマークプログラムを提供しており、それぞれについてマルチコア CPU 用 OpenMP 実装および NVIDIA GPU 用 CUBA 実装に加えて GPU 向けに記述された OpenCL 実装も提供されている。本評価では Rodinia ベンチマークを用いることで、GPU 等既存アクセラレータと比較した FPGA の性能およびプログラミングの評価を行う。具体的には現状では同ベンチマークセットより B+ Tree、Needleman-Wunsch、SRAD の 3 つのベンチマークについて Altera OpenCL による評価を行っている。B+ Tree は motif としてはグラフ探索に相当し、Needleman-Wunsch は動的プログラミング、SRAD は構造格子に相当する。前者 2 本は主に整数演算主体であり、SRAD は浮動小数点演算であり、本稿では単精度バージョンを評価する。

上記 3 本のベンチマークプログラムについて Altera 社によるプログラミングガイド [9] や最適化ガイド [10] を参考に、異なる並列化モデルと最適化による 4 つのバージョン（表??）を作成し、それぞれについて比較考察を行う。

表??にあるとおり、v0 では GPU 向けに記述された既存 OpenCL プログラムを動作させた場合の性能を見積もるためのバージョンである。v1 はそれを FPGA のパイプライン処理向けに変更したバージョンであり、通常の GPU 向け OpenCL 実装である細粒度スレッド並列からパイプライン並列へと並列化を変更したバージョンである。これは最適化ガイドにおいて推奨されている並列化モデルであり、具体的には、OpenCL では細粒度スレッド並列は NDRange と呼ばれるスレッドの階層的グループによってカーネルを実行するモデルであり、これは CUDA におけるグリッド、スレッドブロック等によるカーネル実行に相当する。元の Rodinia ベンチマークではすべて NDRange カーネル実行として並列化されており、それを v1 では OpenCL におけるタスクカーネル実行に変更した。同実行モデルは実質的には NDRange カーネルにおいて 1 スレッドのみを起動する場合に相当し、OpenCL コンパイラによってスレッド内のループを FPGA の論理ユニットにパイプライン並列処理としてマッピングすることを狙ったものである。Altera OpenCL コンパイラではタスクカーネルについては詳細な最適化レポートが作成され、プログラミングガイド等ではそれを元に段階的に最適化を進めることが推奨されている。本稿では OpenCL ではより一般的と想定される細粒度スレッド並列に加えて v1 においてパイプライン並列についても評価を行う。

上述の 2 つのバージョンは異なる並列化モデルを表現したものであるが、v2 および v3 はそれぞれそれらに対して比較的軽微な最適化を施した場合の効果を評価するためのバージョンである。具体的には、カーネルのポインタ引数に `restrict` キーワードを指定することによるメモリアクセス効率化、また NDRange の場合はローカルワークグループサイズの静的な指定、SIMD 並列化の指定を適用した。Task カーネルにはループアンローリングを適用した。これらの最適化はすべて元のプログラム構造を変更することなく attribute や pragma によるものであり、比較的簡易に適用可能である。そのため特に v2 の NDRange の場合には高い効率が達成できることがわかれば、GPU 等との性能可搬性を崩さずに移植できることを示すものであり、重要な知見と言えるだろう。一方で、効率化のために FPGA のパイプライン処理を考慮し v1 や v3 に構成を変更する場合は、単純にスレッド数を 1 にするだけでなくローカルメモリの使用をやめ、カーネル内に並列性を持たせる構造に変更する必要がある。これはむしろ比較的 CPU 向けループ構造に近い形態ともいえ、既存 CPU プログラムからの移植への親和性を示すものとも言える。

3. 評価

以降の実験には Bittware 社の S5PHQ ボードを PCI に接続したシステム用いて評価する。同システムは Intel Core i7 920 CPU、12GB メモリを搭載したワークステーションであり、OS として CentOS 6.6 を用いた。FPGA を使うためのソフトウェアとしては、Altera 社による Altera OpenCL SDK v15.0.1 および Quartus v15.0.1 を用いた。また、性能比較として Intel Xeon E5-2670 (8 コア、2.6GHz 動作) および NVIDIA Tesla K20X による評価も行った。それぞれコンパイラとしては gcc v4.4 (Red Hat 4.4.7-4) および CUDA v7.0 を用いた。また本評価では OpenCL および CUDA での実行時間にはアクセラレータデバイスとホスト間のデータ転送時間の評価は省略し、今後の課題とする。

Rodinia ベンチマークは評価時点で最新版であるバージョン 3.0 を用いた。B+ tree, SRAD, Needleman-Wunsch (NW) のそれぞれの入力データ等は各ベンチマークに付属する run ファイルに従った。具体的には B+ Tree は Rodinia に付属する入力ファイルである mil.txt および command.txt を用いた。SRAD は 2048x2048 の問題サイズ、繰り返し回数を 100 回とした。NW は 2048x2048 の問題サイズとした。

3.1 NW

図 1 に NW の各バージョンの実行時間をミリ秒で示す。各バーの高さは経過時間を示すものであり低い方が性能が高いことを意味する。この結果からオリジナルに最も近い v0 が 277 ミリ秒程度であるのに比べて、それに対して restrict キーワードなどの追加やワークグループサイズの指定など簡易的な最適化を施した v2 では 9 倍弱高速化していることがわかる。一方で Task カーネルへ並列化を変更した v1 および v3 では 2000 秒弱まで大幅に実行時間が増加しており、効率の大幅な低下が確認された。これらのバージョンについては Altera OpenCL コンパイラによってパイプライニング適用の成否や遅延時間の見積もりなど比較的詳細なレポートがコード生成時に作成されるが、現状ではそれを参考にした最適化は実施できていない。今後プロファイラなども活用しつつ、細粒度スレッド並列およびパイプライン並列モデル両方についてさらに効率化の検討を進める予定である。

図 2 に Intel Xeon E5-2670 および NVIDIA Tesla K20X との比較を示す。FPGA の実行時間は先に示した複数バージョンの評価結果から最速であった v2 を用いた。この結果からは FPGA は CPU に対してはわずかながら高速である一方、GPU と比較すると 10 倍以上低速であることがわかる。消費電力を考慮すると今回用いた S5PHQ は PCI からの供給電力で動作するボードであり、GPU や CPU と比べると電力消費は抑えられており、電力性能では GPU と比較した差も縮まると言えるが、絶対性能で CPU とそれほど差が無いこと、GPU と比べて大きな差をつけられており、既存 OpenCL プログラムをそのままもしくは簡易に

表 2 ベンチマークバージョン

バージョン	変更内容	評価目的	カーネル呼び出し方法
v0	元のベンチマークを Altera OpenCL で動作させるために必要な最小限の変更のみ	GPU 向けに書かれた既存 OpenCL プログラムをそのまま FPGA で実行した場合の性能評価	NDRange
v1	v0 のベンチマークから Task カーネルへと変更	FPGA のパイプライン処理を意識した並列化を施した場合の性能評価	Task
v2	v0 のベンチマークから Task カーネルへと変更	スレッド並列における怪異最適化の評価	NDRange
v3	v1 のベンチマークを簡易最適化	FPGA のパイプライン並列における簡易最適化の評価	Task

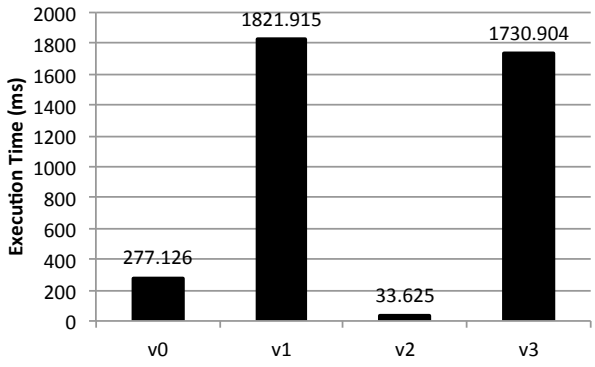

図 1 NW 性能結果

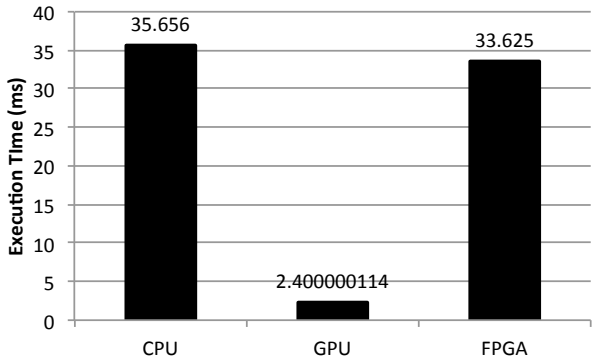

図 2 NW 性能の CPU および GPU との比較

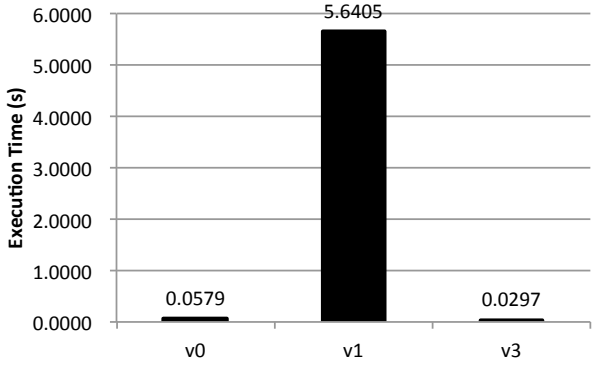

図 3 B+ Tree 性能結果

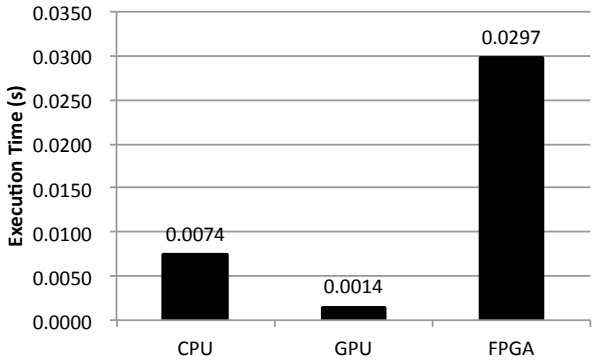

図 4 B+ Tree 性能の CPU および GPU との比較

最適化した程度では FPGA を用いる優位性は限定的と言える。

3.2 B+ Tree

図 3 に B+ Tree の各バージョンの実行時間（秒）を示す。B+ Tree では v2 は実行が完了しなかったため計測から除外した。原因は調査中である。v2 の実行が計測から除外された影響はあるものの、本ベンチマークでは NW とは逆に細粒度並列バージョンではなくパイプライン並列を指向した v3 が最も高速であった。この結果からプログラムによって適した並列化のモデルが変わることを示しており、興味深い結果である。今後より詳細な調査を進める予定である。

CPU および GPU と比較した場合では共に性能は見劣りする結果となった（図 4)。FPGA バージョンは GPU 向けに書かれたプログラムを流用していることなど最適化の余地は大きく残されており、今後さらに高速化が可能かどうか評価を進める必要がある。

3.3 SRAD

図 5 に各バージョンの SRAD 実行時間（秒）および図 6 に CPU、GPU との比較を示す。詳細な解析は今後の課題だが B+ Tree と同様に性能的には見劣りする結果となった。またパイプライン並列はこれらのバージョンでは有効に適用されていないことが推定される。

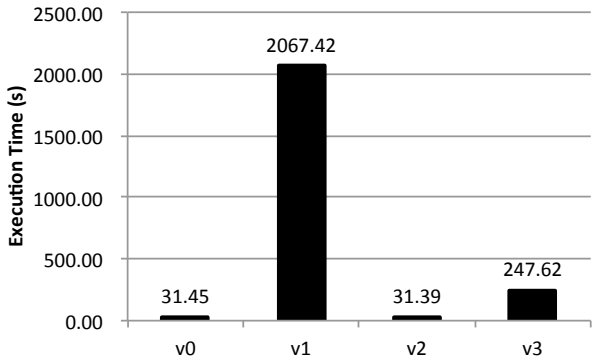

図 5 SRAD 性能結果

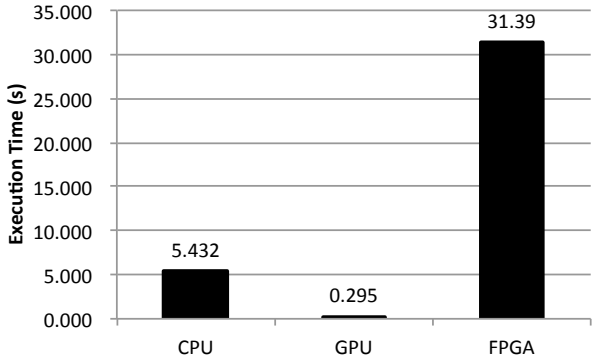

図 6 SRAD 性能の CPU および GPU との比較

4. おわりに

本稿では Altera Stratix V を対象に OpenCL による FPGA の利用についてその予備的な評価を行い、その現状を報告した。FPGA は従来 HDL によるハードウェアレベルの記述が必要であったが、OpenCL のようなある程度の抽象度を持った記述系が利用可能になってきつつあり、今後のより広い応用分野での有効が示される可能性がある。一方で本稿で示したように性能的には単純に既存 OpenCL プログラムを用いただけでは満足いくものは得られとは現状では言えず、より FPGA に適したプログラミングが必要と言える。我々は引き続き調査、評価をすすめ、特に HPC における FPGA の有効性について検討を進める予定である。

参考文献

[1] Putnam, A., Caulfield, A., Chung, E., Chiou, D., Constantinides, K., Demme, J., Esmaeilzadeh, H., Fowers, J., Gopal, G. P., Gray, J., Haselman, M., Hauck, S., Heil, S., Hormati, A., Kim, J.-Y., Lanka, S., Larus, J., Peterson, E., Pope, S., Smith, A., Thong, J., Xiao, P. Y. and Burger, D.: A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services, *41st Annual International Symposium on Computer Architecture (ISCA)* (2014).

[2] OpenCL Working Group, K.: The OpenCL Specification: Version 1.0 (2009).

[3] Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J., Lee, S.-H. and Skadron, K.: Rodinia: A benchmark suite for heterogeneous computing, *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pp. 44–54 (online), DOI: 10.1109/IISWC.2009.5306797 (2009).

[4] Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Keutzer, K., Kubiatowicz, J. D., Lee, E. A., Morgan, N., Necula, G., Patterson, D. A., Sen, K., Wawrzynek, J., Wessel, D. and Yelick, K. A.: The Parallel Computing Laboratory at U.C. Berkeley: A Research Agenda Based on the Berkeley View, Technical Report UCB/EECS-2008-23, EECS Department, University of California, Berkeley (2008).

[5] Bittware: S5-PCIe-HQ Datasheet (2014).

[6] Altera Corporation: Implementing FPGA Design with the OpenCL Standard (2013).

[7] Owaida, M., Bellas, N., Daloukas, K. and Antonopoulos, C.: Synthesis of Platform Architectures from OpenCL Programs, *Field-Programmable Custom Computing Machines (FCCM), 2011 IEEE 19th Annual International Symposium on*, pp. 186–193 (online), DOI: 10.1109/FCCM.2011.19 (2011).

[8] Krommydas, K., chun Feng, W., Owaida, M., Antonopoulos, C. and Bellas, N.: On the characterization of OpenCL dwarfs on fixed and reconfigurable platforms, *Application-specific Systems, Architectures and Processors (ASAP), 2014 IEEE 25th International Conference on*, pp. 153–160 (online), DOI: 10.1109/ASAP.2014.6868650 (2014).

[9] Altera Corporation: Altera SDK for OpenCL Programming Guide (2015).

[10] Altera Corporation: Altera SDK for OpenCL Best Practices Guide (2015).

謝辞 Microsoft Research の Aaron Smith 氏の助言に感謝する。FPGA の評価には株式会社エルセナから協力を受けたとに感謝する。