

令和 6 年 4 月 23 日現在

機関番号：22605

研究種目：基盤研究(C)（一般）

研究期間：2020～2023

課題番号：20K11751

研究課題名（和文）FPGAを含む計算資源管理のためのオペレーティングシステム基盤

研究課題名（英文）Operating System Framework for Computing Resource Management including FPGA

研究代表者

追川 修一（Oikawa, Shuichi）

東京都立産業技術大学院大学・産業技術研究科・教授

研究者番号：00271271

交付決定額（研究期間全体）：（直接経費） 3,300,000円

研究成果の概要（和文）：本研究では、FPGAをCPUを含めた計算資源の1つとする概念を提案し、実現した。従来はFPGAは1つのプロセスによって専有されている場合には、他のFPGAを用いるプロセスは実行を開始することはできず、解放されるのを待つ必要があった。本研究では、FPGAが専有されている場合は、他の利用可能な計算資源を代替として用いて実行するという選択肢を与えることで、計算資源全体での多重化を可能とした。

研究成果の学術的意義や社会的意義

FPGAはプログラマブルな計算資源としてソフトウェアと同様に利用できるべきとの考え方にに基づき、OSによりFPGAを用いるアプリケーションにソフトウェアと同等の利便性を提供可能にした点に、本研究の学術的な意義がある。また本研究は、FPGAを用いるアプリケーションの開発を容易とする方向では、例えばRaspberry Piのように、Proof of Conceptの開発への波及効果が期待できる点に、本研究の社会的意義がある。

研究成果の概要（英文）：This study proposed the operating system framework, of which goal is to ease software programmers of the development FPGA IPs by making FPGA a programmable computing resource. The proposed framework aims at being able to run programs that utilize FPGA just as those that run on CPUs. In order to step toward the goal, the proposed framework makes FPGA the first-class citizen that provides programmable computing resources by providing the function to multiplex of FPGA by finding available computing resources among CPU cores and FPGA and dispatching computation.

研究分野：システムソフトウェア

キーワード：ソフトウェア オペレーティングシステム 計算機システム

1. 研究開始当初の背景

処理性能向上のために、コンピュータはヘテロジニアス化が進んでいる。2000年代前半にはプロセッサの高クロック化による性能向上は消費電力増大のため限界を迎え、その後、マルチコア、SIMD、そしてGPUの汎用化(GPGPU)により、高性能化を実現してきた。しかしながら、SIMD、GPGPUを含めた汎用的アーキテクチャは特定のアルゴリズムを前提とはしないため、その高性能化は多くのハードウェア資源を必要とするものとなる。そこで、実行時に回路をプログラム可能なFPGA(Field Programmable Gate Array)が注目されている。FPGAを用いることで、アプリケーションが必要とするアルゴリズムに特化したハードウェアを実装でき、さらなる省電力と高性能の両立が期待されている。

FPGAは、VerilogやVHDL等のハードウェア記述言語によりプログラムされてきた。これらのハードウェア記述言語は、文法はC言語に類似しているものの、ハードウェアの動作を記述するため、ハードウェアの動作を理解したうえでのプログラミング技術が必要となる。そのため、汎用プロセッサ用にプログラミングをするソフトウェア技術者には、FPGAのプログラミングは困難であった。

高位合成(HLS: High Level Synthesis)技術は、そのようなハードウェア記述の困難さを軽減するため、汎用プロセッサ用のプログラムをハードウェア記述言語のプログラムに変換するものである。HLSは長年にわたる研究開発の結果、実用化一般化がすすみ、FPGAをプログラムするツールの一部として配布され、広く利用可能になっている。HLSを用いることで、C、C++、Java、Scala等のプログラムのハードウェア化が可能となり、ソフトウェア技術者にもFPGAのプログラミングが可能となる環境が整備されてきている。

HLSと、近年、研究開発が盛んになっているアプリケーションのドメインに特化したプログラミング言語(DSL: Domain Specific Language)を組み合わせることで、FPGAの可能性はさらに大きく広がりを見せる。DSLの1つであるHalideは、画像処理の高速化に特化した言語として開発された。処理アルゴリズムとターゲットに合わせた高速化手法を独立に記述し、FPGA、CPUそれぞれに合わせた高速化手法を用いたプログラムを生成できる。従って、単一のDSLプログラムからFPGA、CPUそれぞれに最適化されたプログラムを生成可能である(図1)。

画像処理は基本的には行列演算であるため、同じく行列演算を用いる深層学習の高速化にも利用可能である。例えば、Keras、PyTorchといった深層学習フレームワークからHalideの中間表現に変換可能であり、即ち、そこからHLSを生成することで、FPGA化も可能となる。そもそもFPGAはアルゴリズムをハードウェアで実現できるところに強みがあり、特定用途のアルゴリズムの記述に適しているDSLを用いることで効率的な開発が可能となり、FPGAが強みを発揮できる環境が整ってきていると言える。

2. 研究の目的

FPGAによる特定アルゴリズムの高性能化の研究は盛んに行われてきた一方で、プロセッサとは異なる汎用性を持つFPGAをどのようにシステムの一部として組み込み、実行環境を管理するオペレーティングシステム(OS: Operating System)からどのように扱うのか、という課題についての研究は少なかった。それは、FPGAは時間的空間的に細粒度での分割が困難であり、排他的に利用せざるを得ず、多重化ができないことが技術的な障害となっていたためである。しかしながら、HLSによりFPGAはソフトウェア技術者がプログラミングする対象となることで、プログラマブルな計算資源であるFPGAをソフトウェアと同様に利用できる。FPGAをソフトウェア同様に利用できる実行環境とは、FPGAを用いる複数アプリケーションの並列実行、および単一アプリケーション内でのFPGA、デバイス、ソフトウェアの透過的な協調処理を実現するものである。そのような実行環境を実現する一歩として、本研究ではFPGAを含む計算資源の多重化の問題を解決することで、FPGAを用いるアプリケーションにソフトウェアと同等の利便性を提供することを本研究の目的とする。

3. 研究の方法

本研究では、FPGAは1つのプロセスが占有して使用するものであり、他のFPGAを使用するプロセスは実行を開始することができないという問題点を解決するため、計算資源としてCPUはFPGAを代替する、またその逆も成り立つという、CPUとFPGAを並列に考えるという概念を取り入れる。そのためには、単一のプログラムから複数の利用可能な計算資源で実行可能なバイナリ等を生成できる環境が必要となる。対象となるプログラムから、CPUについてはコンパイルして実行可能バイナリを生成し、FPGAについては高位合成(HLS)によりFPGA回路に変換してIPを生成する開発環境を開発し、研究を行う。開発環境は、複数のオペレーティングシステム間でのポータビリティを考慮し、Dockerを用いる。実行環境としては、汎用性を重視し、Windowsを用いる。WindowsではDocker for WindowsとWSL2上のDockerの2通りの方法を用いることができるため、それら両方に開発環境を構築する。

4 . 研究成果

FPGA を含む計算資源の多重化を実現する機構:計算資源としてのFPGAの利用は、FPGAに実行させるプログラムを高位合成(HLS)等によりFPGA回路に変換し、CPUで実行するメインプログラムからFPGA回路を呼び出すことによって実現する。FPGAに回路をロードできるプロセスを1つとすると、FPGAは1つのプロセスによって専有され、他のFPGAを用いるプロセスは実行を開始することができない。本研究では、FPGAを計算資源の1つとする概念を導入する。この概念により、FPGAが専有されている場合は、他の利用可能な計算資源を用いて実行するという選択肢が生まれることになり、計算資源の多重化が可能になる。そのためには、FPGAに実行させるプログラムは基本的にCPUでも実行可能なプログラムとなっている必要があり、単一のプログラムから、高位合成(HLS)によりFPGA回路に変換して生成するIPとコンパイルして生成するバイナリの両方が必要となる。そこで、単一のプログラムからFPGAならびにCPUの両方で実行可能な形式を生成し、計算資源の利用状況に応じて用いる計算資源を選択、実行する評価をおこなった。実験環境は、Linuxが動作する評価ボード上に構築した。

構築した機構を用いて行った実験からは、1)FPGA回路の再構成を含むFPGAを用いるための初期化処理は、プロセスの起動と比較して長時間かかることから、必要とする回路がFPGA上に無く再構成が必要な場合、FPGAの初期化処理と並行して、CPU等の他の計算資源で実行することで、FPGAを用いるアプリケーションの起動時間を大幅に削減することが可能であること、2)比較的単純な処理であってもFPGAを用いることで処理の高速化が可能であること、またFPGAの初期化処理はコストが大きいだけでなく処理時間のばらつきが非常に大きいことを明らかにした。

FPGAを用いるユーザプログラムの軽量な実行環境:本研究の提案方法で問題となるのは、CPUで実行するプログラムの呼び出しコストとFPGAで実行するIPの呼び出しコストの差異である。CPUで実行するプログラムの場合、関数呼び出しとなるため、その呼び出しコストは非常に小さい。一方、FPGAで実行するIPの場合、ハードウェアであるFPGAはデバイスドライバにより管理されているため、オペレーティングシステムカーネルを経由した呼び出しとなり、そのコストは大きくなる。そのため、FPGAを用いることが有用となる場合は、FPGAの呼び出しコストを吸収することのできる、ある程度の実行時間が必要となる処理に限定されることになる。

ここで、FPGAを呼び出すユーザプログラムをカーネル内で実行することができれば、FPGAの呼び出しコストを低減することができることになる。そこで、ユーザプログラムをカーネル内で実行することで軽量な実行環境を提供するUnikernelの1つであるUKL(Unikernel Linux)を用いた実行環境について実証実験を行い、本研究の提案方法についての有用性について明らかにした。

5. 主な発表論文等

〔雑誌論文〕 計0件

〔学会発表〕 計3件（うち招待講演 0件 / うち国際学会 3件）

1. 発表者名 Shuichi Oikawa
2. 発表標題 Transparent Transition of a User-Level Service to Kernel-Level Functionality
3. 学会等名 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE) (国際学会)
4. 発表年 2023年

1. 発表者名 Shuichi Oikawa
2. 発表標題 Operating System Framework for Transparent Execution on a CPU and FPGA
3. 学会等名 2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA) (国際学会)
4. 発表年 2021年

1. 発表者名 Shuichi Oikawa
2. 発表標題 Operating System Resource Management for FPGA
3. 学会等名 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE) (国際学会)
4. 発表年 2020年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8 . 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------