

Thesis Title: Dependable and Scalable FPGA Computing Using HDL-based Checkpointing.

Full name: Hoang-Gia Vu
Supervisor: Professor Yasuhiko Nakashima
Co-supervisor: Professor Michiko Inoue
Co-supervisor: Associate Professor Takashi Nakada
Laboratory: Computing Architecture

Abstract:

Thanks to high computational capabilities, reconfigurability, power efficiency, and the great advantages of customizing hardware for domain-specific applications, Field Programmable Gate Arrays (FPGAs) are now widely deployed in modern datacenters and high-performance computing systems. However, this deployment compounds the dependability of the computing systems due to their growing size and complexity. On the other hand, it challenges designers to scale computing systems.

In this doctoral dissertation, we present how FPGA computing can be dependable and scalable using HDL-based checkpointing. First, we study a method to guarantee the consistency of snapshots between FPGA and other components. Such consistency is essential for the snapshots to be resumed correctly on FPGA. We then propose two checkpointing architectures along with a checkpointing mechanism on FPGA: CPRtree – a tree-based checkpointing architecture, and CPRflatten – a ring-based flattened checkpointing architecture. The two checkpointing architectures are transparent to applications and portable across different hardware platforms. Third, we investigate a static analysis of the original HDL source code for CPRflatten from fundamentals to algorithms in order to re-use hardware resources for the checkpointing purpose, thus reducing hardware consumption caused by checkpointing functionality. Fourth, we introduce two Python-based tools in structures and algorithms to generate checkpointing infrastructures according to CPRtree and CPRflatten so that designers' task in writing checkpointing source code can be removed completely. The two tools can be integrated seamlessly into hardware design flows. The position of the tools in design flows ensures that our checkpointing architectures are independent of other tools and technology. Fifth, we study a checkpoint/restart scheme for dependability of FPGA computing. In this scheme, we also introduce a software stack with application programming interface (API) functions for “coarse-grained” management from the host. The stack is also transparent to applications and portable across hardware platforms. Sixth, we present two schemes for scalability of FPGA computing employing our above checkpointing architectures. The first scheme – on-the-fly multitasking on FPGA allows multiple users to efficiently share a limited reconfigurable fabric. The second scheme – on-the-fly hardware task migration in heterogeneous FPGA computing allows a hardware task to be migrated between different FPGA fabrics with different technology.

We evaluate our proposals from hardware overhead, maximum clock frequency degradation, data footprints, and performance overhead to power consumption. Although the hardware overhead is still significant, the performance degradation and the additional power consumption is small. Our proposals show a potential for bringing FPGAs to hyper-scale computing, such as hyper-scale data centers and hyper-scale clouds while taking advantages of software-based computing.

(論文審査結果の要旨) (A4 1枚 1、200字程度)

FPGA(Field Programmable Gate Arrays)は高いエネルギー効率と高い柔軟性を兼ね備えており、カスタマイズ可能な特定用途向け高性能計算基盤として非常に有用であるため、組込み機器からデータセンターまで幅広く採用されている。一方、FPGAが大規模かつ広範囲に応用されるにつれ、信頼性の向上が課題となっている。本論文では、一貫性を保障したチェックポイントアーキテクチャを提案することによりこの問題を解決する。まず、第一の貢献として、稼働中のFPGAに対する一貫性のある状態保存と状態復元を実現するチェックポイントアーキテクチャを提案した。具体的には、元アプリケーションの構造を利用し、木構造を持つCPRtreeと実装効率を重視したリング構造を持つCPRflattenの2つのアーキテクチャを提案し、状態保存・復元速度と回路面積オーバーヘッドを比較評価した。第二の貢献として、提案チェックポイントアーキテクチャを用いて、状態保存時と異なるアーキテクチャを持つFPGAでの復元を実現することにより、異種FPGA間における実行中アプリケーションの動的な移行手法を提案した。これにより、多種類のFPGAが存在するデータセンターにおける柔軟かつ動的な負荷調整を可能とする。第三の貢献として、アプリケーションコードの静的解析により、アプリケーション中に存在するすべての記憶素子の内容を保存するのではなく、最低限保存すべき状態情報を特定することにより、必要なハードウェアコストおよび時間オーバーヘッドの削減を実現した。第四の貢献として、上記を実現するためのPythonフレームワークを作成することにより、既存のRTLコードに対して自動的にチェックポイント機能を挿入可能とし、多くのアプリケーションにおいて自由なチェックポイントを実現した。最終的に13%~78%の回路オーバーヘッド、2.6%以下の性能低下、0.08ms以下のレイテンシでチェックポイントが実現可能であることを確認した。

以上、本論文は学術上、實際上寄与するところが少なくない。よって、本論文は博士(工学)の学位論文として価値あるものと認める。