

無線3次元積層チップを用いたDeep Learningアクセラレータのシステムレベルシミュレーション

大久保徹以† 小島拓也† 天野英晴† 高田遼‡ 石井潤‡ 坂本龍一‡
近藤正章‡ 中村宏‡

†慶應義塾大学理工学部情報工学科 ‡東京大学大学院情報理工学系研究科システム情報学専攻

1 はじめに

電力効率に優れたDeep Learningアクセラレータの実現において、汎用性・柔軟性の面から、無線3次元積層チップによるメニーコアアーキテクチャを用いたアプローチが有力であると考えられる。しかしながら、このようなアプローチにおいては、現実の大規模Deep Neural Networkを、いかに効率的にアクセラレータ上の命令セットで実装するかが大きな課題となる。本稿では、3次元積層チップによるシステム全体を再現する、システムレベルシミュレータを開発することにより、効率的にシステム全体の評価を行う手法を提案する。

2 背景

ThruChip Interface[1]による無線3次元チップ積層を用いた、ビルディングブロック型計算システムは、チップ間通信を電氣的に無接触で実現し、柔軟性などの面において有利である。Cube-1[2]は最初のプロトタイプであり、ホストプロセッサであるGeysier 1チップと粗粒度再構成可能アクセラレータであるCMAを最大で3チップまで積層することができた。

現在この他のアクセラレータが開発中であり、Deep Learning向けアクセラレータSNACC[3]はその1つである。SNACCはDeep Learningを効率的に実行するためのSIMD命令を含む独自の命令セットを持つ。SNACCを搭載したビルディングブロック型計算システムをSNACC-Cubeと呼ぶ。各チップ上にルータがあり、TCIによって隣接するチップと通信する。チップ間はTCIを介したルータ間の

System Level Simulation of Deep Learning Accelerator with Wireless 3D Stacked Chips

Tetsui Ohkubo† Takuya Kojima† Hideharu Amano† Ryo Takata‡ Jun Ishii‡ Ryuichi Sakamoto‡ Masaaki Kondo‡ Hiroshi Nakamura‡

†Keio University ‡University of Tokyo

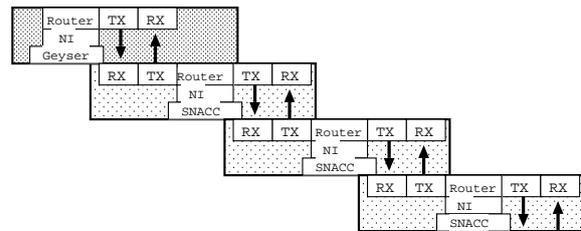


図1: SNACC-Cubeの構造

パケット通信によってコミュニケーションすることができる。

このようなシステム向けのソフトウェアの開発には、当該システムのハードウェア記述を用いたRTLシミュレーションを用いる方法が考えられ、これはサイクル単位で正確であるが、大規模Deep Neural Networkの実装にあたっては、実行速度やソフトウェア開発者の利用可能性の面で難がある。本稿ではシステムレベルシミュレータ上で、実ハードでの実行サイクルをなるべく正確に評価しつつ効率的にシステムレベルシミュレーションを実施する手法を提案する。

3 システムレベルシミュレータ

ホストプロセッサはMIPS R3000命令セット、アクセラレータは独自命令セットを採用しているため、システム全体のシミュレーションに2種類のインタプリタが使われている。ホストは、GPLライセンスで公開されているシミュレータVMIPSを採用した。

正確なターゲットソフトウェア評価のためには、実行サイクル数およびTCI転送のオーバーヘッドをなるべく正確に見積もる必要がある。以下に誤差の原因を考察する。

ホストプロセッサのサイクルカウント古典5段パイプラインを実装したインオーダー実行プロセッサである。単純に実行された命令数を数えるだけでは

パイプラインストールを考慮することができず、これが誤差の原因となりうる。また、Geysler のキャッシュフラッシュメカニズムをシミュレータは正確に再現していないのでこれも第二の原因となりうる。

アクセラレータのサイクルカウント 現在はパイプラインなし 4 段マルチサイクルプロセッサとみなせる。いくつかの SIMD 命令は 4 サイクルで終了せずより複雑な制御フローを持つが、これを考慮することは容易である。最もシミュレーションが困難なのは、アクセラレータの持つシェアードメモリのアービトレーションを考慮した場合で、実ハードでは、各サイクルごとに 4 つのコアのうち何れかがアクセス権を与えられる比較的単純なアプローチを取っているが、4 サイクルで終了しない SIMD 命令が存在することを考えると、シミュレータ上でのタイミングは完全に予測不可能である。したがってこれがサイクルカウントの主要な誤差の原因となりうる。

TCI コミュニケーションのレイテンシ チップ間コミュニケーションにパケットルーティング方式を用いているため、実ハードウェア上で動作する場合に全体のレイテンシについて何の保証も持たない。現在のシミュレータは読み書きされたバイト数に定数をかけストールさせる単純なアプローチを取っている。定数は TCI の設計から求まる現実的な値でありホストメインメモリへの読み書きよりはるかに大きい。1 バイト転送に TCI は 25 サイクル要し、さらに通信時に余裕をみて 50% のマージンを取ると仮定する。

4 評価

ハードウェア記述は Verilog で、RTL シミュレーションは Cadence NC-Sim 10.20-s131 上で行った。システムレベルシミュレータは C++ で記述し、GCC Version 6.2.1 上でコンパイルした (-O2 あり)。両シミュレータは CentOS 6.5 の動作する CPU Intel Xeon E5-2667 マシン上で実行した。シミュレータ評価に用いたコードはハンドアセンブルされた小規模 4 レイヤー CNN のものである。

シミュレーション結果を表 1 に示す。システムレベルと RTL システム全体は SNACC-Cube システム全体のシミュレーションである。シングルコアと RTL シングルコアの実行時間はどちらも十分に短

いためさらなる検討の余地があるが、システム全体ではシステムレベルシミュレータが有意なパフォーマンスの改善を示し、ソフトウェア開発のラウンドタイム短縮に貢献することが示された。システムレベルシミュレーションと RTL シミュレーションの不自然なまでの所要時間差は、RTL シミュレータが与える情報の粒度を考えると妥当なものである。

表 1: シミュレータ評価

	所要時間	サイクル数
システムレベル	0.04s (4390 倍速)	375253 (誤差 1.27%)
シングルコア	0.01s (188 倍速)	31793 (誤差 7.85%)
RTL システム全体	175.60s	380097
RTL シングルコア	1.88s	34501

5 結論と今後の課題

SNACC-Cube のシステムレベルシミュレーション環境によって、効率的にシステム全体の評価を行う手法を提案した。実際にシステムレベルシミュレータを実装し、4390x 倍高速でありながら誤差 1.27% 以内でサイクル数をカウントすることができることを示した。今後の課題としては、TCI コミュニケーションレイテンシーのより正確なモデリングなどが挙げられる。

参考文献

- [1] Tadahiro Kuroda. ThruChip interface (TCI) for 3D networks on chip. In *2011 IEEE/IFIP 19th International Conference on VLSI and System-on-Chip*, pages 238–241. IEEE, oct 2011.
- [2] Noriyuki Miura, Yusuke Koizumi, Yasuhiro Take, Hiroki Matsutani, Tadahiro Kuroda, Hideharu Amano, Ryuichi Sakamoto, Mitaro Namiki, Kimiyoshi Usami, Masaaki Kondo, and Hiroshi Nakamura. A scalable 3D heterogeneous multi-core with an inductive ThruChip interface. *IEEE Micro*, 33(6):6–15, 2013.
- [3] A study on building-block computing systems using inductive coupling interconnect. http://www.am.ics.keio.ac.jp/kaken/_s/.