2012 年 12 月 25 日 情報システム基盤学合同輪講発表資料

CAPRI: Prediction of Compaction-Adequacy for Handling Control-Divergence in GPGPU Architectures

著者： Minsoo Rhu, Mattan Erez

出典： *Proceedings of the 39th International Symposium on Computer Architecture (ISCA '12) pp.61-71,2012.*

発表者： 本多・近藤研究室 1253016 松本 洋平

1 研究背景

GPU は大量の SIMD 型演算器を搭載する。SIMD 型演算器は 1 命令で複数のデータに対して並列演算ができるため、高い演算能力を持つのに加え制御回路を小さくできる利点がある。しかし、制御系の命令を効率良く実行できないという問題がある。従来の制御命令の実行効率を改善する方法は性能低下を起こす可能性があった。本論文では、それを解決する手法を提案する。

2 GPU のスレッド実行とその問題点

2.1 warp 実行

GPU では warp 単位でメモリアクセスと演算が実行される。warp は単一命令で同じ処理を実行する 32 スレッドの集合である。warp を実行する時は 8 個または 16 個の演算器をグループにして演算をする。例えば 16 個の演算器が搭載されている GPU で warp を実行する場合は、1warp を 2 回に分けて演算を実行する (図 1)。なお、GPU 上で 96 スレッドが生成された場合は 32 スレッドごとに 3 つの warp に分割される (図 2 以降では紙面の都合上 1warp を 4 スレッドで記載する)。

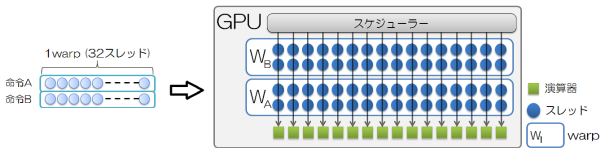

図 1: GPU での warp 実行

2.2 分岐実行における問題点

CPU において分岐命令を実行するとき、各スレッドは IF 文の THEN 節と ELSE 節の片方のみを選択して実行する。一方で、GPU では分岐命令の実行時に warp 中の各スレッドは THEN 節と ELSE 節の双方にそれぞれ分岐してしまう場合があるが warp 単位で同じ命令を実行するためその場合は双方のパスを実行する必要がある。その時、THEN を実行したスレッドは ELSE 節では休んでいることになる (図 2)。これは非効率であり SIMD 型アーキテクチャの構造上の欠点とされている。

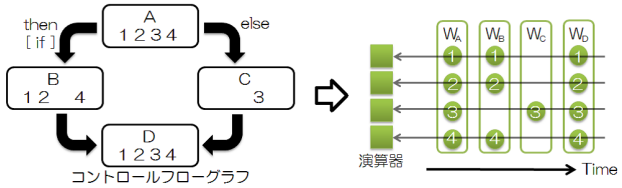

図 2: GPU での分岐命令実行順序

3 既存手法

提案手法のベースとなる 2 つの分岐実行の効率化手法について以下に述べる。

3.1 Warp-wide PDOM (WPDOM) [1]

分岐によっては warp 中の全スレッドが THEN 節と ELSE 節の片方のパスのみで実行されることがある。この場合は事前に片方のパスだけが実行されることがわかればもう片方のパスの命令実行を回避することができる。WPDOM は warp ごとにスタックを用意して分岐実行を管理する。このスタックでは分岐の際にそれぞれのパスで実行すべき warp 中のスレッドの番号を保持している。図 3 の例では THEN 節である命令 B で全てのスレッド '0123' が実行される。その後 ELSE 節である命令 C の各スレッドを実行する順番になるが実行するスレッドがないのでスタックは 'xxxx' となっている。この 'xxxx' の場合は命令を発行しなければ無駄な演算の実行を回避できる。

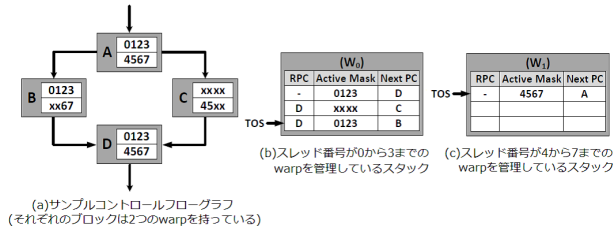

図 3: WPDOM の実行例

3.2 Thread Block Compaction (TBC) [2]

TBC は同じパスを実行する別々の warp 内のスレッド同士をまとめて実行 (コンパクション) する手法である。図 4 の命令 B では '0xxx' と 'xx67' の 2 つの warp がある。これらをまとめて '0x67' の 1 つの warp にコンパクションして実行すれば無駄な演算が削減される。なお、TBC ではコンパクションを行うために、複数ワープの実行管理を図 4 の Cooperating Thread Array (CTA) という 1 つのスタックで管理する。このため、複数ワープの命令実行の進行が同期化されてしまい命令実行の並列性が失われて性能低下の原因となる場合がある。

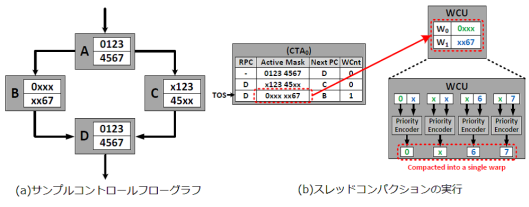

図 4: TBC の実行例

4　提案手法：CAPRI

TBC は実行する全ての warp に対してスレッドコンパクションを行う。この際、個々のスレッドに空きがなくコンパクションしても warp 数が減らないあるいはや分岐の片方のパスだけが実行される命令間など、スレッドコンパクションの効果がない warp 間でコンパクションを実行してしまうとメモリアクセスの不連続性や命令実行の進行が同期され warp 間の並列性が活用できずに性能低下を引き起こす可能性がある。例えば図 5(b) の例では TBC は使用している CTA により複数命令間の実行を同期化してしまっている。これにより命令 A 中のある warp で起きたキャッシュミスが後続の命令 B の実行を遅らせている。

上記のようなコンパクションは同じようなパターンで繰り返されることがしばしばある。そこで CAPRI ではスレッドコンパクションの効果がない分岐命令を記録することで次回以降でその分岐命令を実行する際にその部分でのコンパクションをしないようにする。具体的にはコンパクション前の命令中の warp 数とコンパクション後の warp 数を比較して数が一致してしまう場合、すなわちコンパクションの効果がない場合は次回以降のコンパクションを実行しないようテーブルに登録する。このようにすることで例えば図 5(c) のようにコンパクションの効果が得られない命令 A と命令 B のコンパクションをキャンセルすることでキャッシュミスのレイテンシを隠蔽することも可能になる。

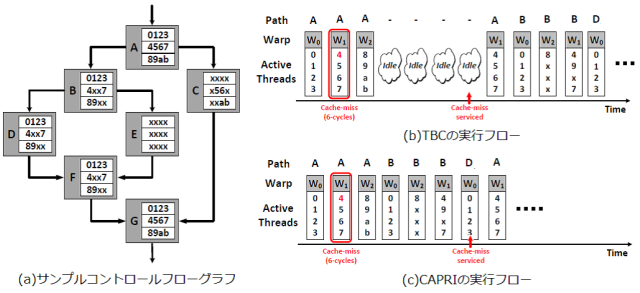

図 5: CAPRI の実行例

5　実験手法：GPGPU-sim[3]

GPGPU-sim は Tor Aamodt らによって開発された GPU アーキテクチャを研究するためのツールである。

本実験では NVIDIA's Quadro FX5800 をモデル化したものに WPDOM, TBC, TCB+, CAPRI(1bS), CAPRI (1bL) を実装して各種法を評価した。なお、TBC+は片方のパスだけ実行されるヒントをコンパイル時に挿入する手法、CAPRI(1bS) は一度コンパクションが不要と判断されるとそれ以降のコンパクションをしない手法、CAPRI (1bL) は 2bit カウンターを使用してコンパクション実行の可否を決定する手法である。

また、実行するベンチーマークは Parboil, Rodinia, CUDA SDK, GPGPU-sim に付随のベンチマークの中から性能検証の精度が高いものを選択する。

6　実験結果

6.1　IPC

分岐のあるベンチマークでは CAPRI(1bL) は WPDOM より 12.6%の性能向上、TCB+より 7.2%の性能向上が見られた。また、CAPRI(1bS) は CAPRI(1bL) ほどの性能向上は見られなかったが WPDOM、TCB+よりも高い性能が見られた。分岐のないベンチマークのでは TBC は WPDOM に対して平均で 10.1%、TBC+ は平均で 8.9%の性能低下が見られた。一方、CAPRI は WPDOM に対して± 1%の性能が変化した。

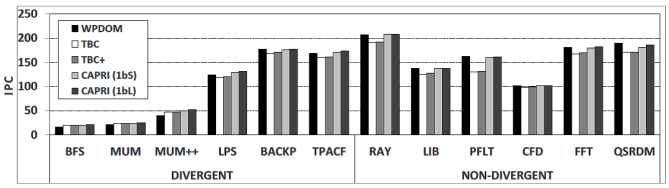

図 6: 各手法の IPC 比較結果

6.2　実装オーバーヘッド

プロセッサ設計では性能と消費電力と回路面積の 3 つはトレードオフの関係にあり、性能向上は達成したがその分だけ消費電力と回路面積が増えるということが多くある。CAPRI の場合は最大で 10.8%の性能向上を達成した。また、消費電力は演算実行数が減るため削減される。一方で 8160bit のレジスタ量だけ回路面積増加 ($1mm^2$ 程度) してしまうが、これはチップ全体の 1/300 程度である。よって、ほんの僅かな回路面積のオーバーヘッドで性能向上と消費電力の削減を達成した。

7　結論

本論文では GPU 上で分岐命令を効率良く実行する提案した。分岐命令がある場合は, 平均で 7.6% 性能向上が見られた。分岐命令がない場合は, ± 1% 性能が変化した。また、ほんの僅かな回路面積で提案手法が実装できることを示した。

参考文献

[1] W.W. Fung et al. Warp Formation and Scheduling for Efficient GPU Control Flow. MICRO-40 pp.407-420 , 2007.

[2] W. W. Fung and T. M. Aamodt. Thread Block Compaction for Efficient SIMT Control Flow. HPCA-17 pp25-36 , 2011.

[3] Ali Bakhoda et al. Analyzing CUDA workloads using a detailed GPU simulator. ISPASS 2009 pp163-174 , 2009