2011 年 5 月 9 日 情報システム基盤学合同輪講発表資料

Stencil Computation Optimization and Auto-tuning on State-of-the-Art Multicore Architecture

著者： Kaushik Datta, Mark Murphy, Vasily Volkov, Samuel Williams ..etc
出典： *Supercomputing 2008 PP.1—12*
発表者： 本多・近藤研究室 1153004 岩下　光弘

1　はじめに

本研究では，アーキテクチャ上の異なる特徴を持つ 5 種類のマルチコアプロセッサ [1] に対して，並列ステンシル計算プログラムの最適化を行った．その実行解析の結果から，それぞれのアーキテクチャに適した最適化手法とそのパラメータを明らかにし，プロセッサ設計におけるトレードオフと，それがアルゴリズムの開発に与える影響について議論する．

2　ステンシル計算

ステンシル計算（近傍計算の一種）[2] は，偏微分方程式の解法などに用いられる計算で．プロセッサアーキテクチャのベンチマークプログラムにも用いられる．3 次元空間におけるステンシル計算では，中心の 1 点の値をその周辺の 6 点の値から求める．

図 1: 3 次元空間におけるステンシルの概念

3　マルチコアアーキテクチャ

本研究で対象とする，5 種類のプロセッサは，そのアーキテクチャに基づいて 2 つに分類できる．1 つ目は，複雑なコアを少数搭載し，メモリ管理をハードウェアベースで行うプロセッサ，2 つ目は，シンプルなコアを多数搭載し，メモリ管理をソフトウェアベースで行うプロセッサである．

前者のプロセッサとして，Intel XEON(Clovertown) E5355, AMD Opteron 2356(Barcelona) を用いる，これらは，アウトオブオーダー実行のキャッシュベースプロセッサである．また，後者のプロセッサとして，Sun UltraSPARC T5140 T2+(Victoria Falls)，STI Cell(QS22 PowerXCell 8i)，NVIDIA GeForce GTX280 を用いる．Victoria Falls は，ハードウェアベースのマルチスレッドを搭載したプロセッサである．Cell は，POWER プロセッサベースのコア 1 個 (PPE) と，ローカルストアを持ったシンプルなコア (SPE) を 8 個搭載したプロセッサである．GeForce GTX280 は 240 個のシンプルなコアを搭載した GPGPU である．

4　最適化手法と Auto-tuning

4.1　最適化手法

本研究で用いる最適化手法は次の 4 種類である．

Problem Decomposition

処理対象のデータを計算資源に合わせて適切な大きさに分割する最適化手法である (図 2)．

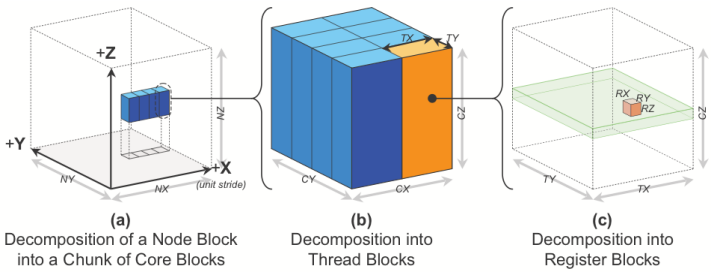

図 2: Problem Decomposition

問題全体 (Node Block) を Core Block，Thread Block，Register Block に階層的に分割する．これらの分割を行うことによって，キャッシュ中のブロックサイズ調整による容量ミスの防止．共有キャッシュやローカルメモリの局所性の利用，レジスタによるデータレベルでの並列性を得られる．

Data Allocation

データを適切なメモリやメモリ中の位置に配置することによる最適化手法である．NUMA-aware Allocation や Padding などがこの最適化手法に該当する．

Bandwidth Optimizations

メモリレイテンシの隠蔽とメモリトラフィックの向上のための最適化手法である．DMA や巡回待ち行列の実装，キャッシュバイパス命令などがこの最適化手法に該当する．

In-Core Optimizations

コア内の実効性能を改善する最適化手法である．ループアンローリングや，SIMD に特化したコードの生成する方法などがこの最適化手法に該当する．

4.2 Auto-tuning environment

Auto-tuning 環境は，本来人間の手で調整していた最適化パラメータを，アーキテクチャごとに自動かつ迅速に適切な値へ調整する機構である．

5 評価結果

図 3 に示すグラフは各種最適化手法が各プロセッサの性能にどの程度効果があったかを示している．グラフ中の紫色のバーは最適化を施す前の性能を，その上のいくつかの異なる色のバーは最適化による性能を表している．ただし，Cell に関しては，DMA の実装とその他の最適化による性能を，GTX に関しては，Thread Blocking による最適化後の性能を表している．

図 4(a) から (d) は，各プロセッサのピーク実効性能，コアのスケーラビリティ，達成できた計算性能とメモリバンド幅の利用率の相関，エネルギー効率をそれぞれ表している．

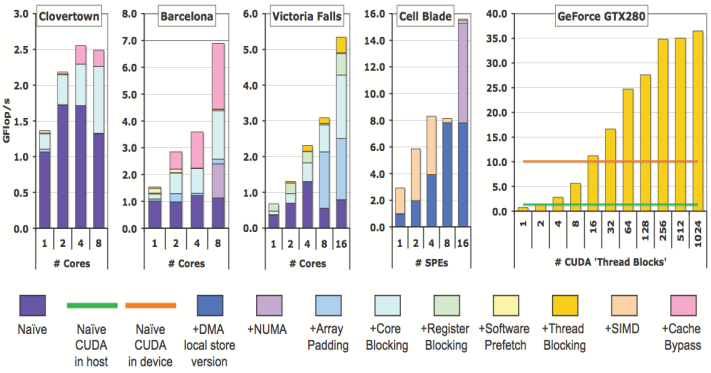

図 3: 最適化の効果

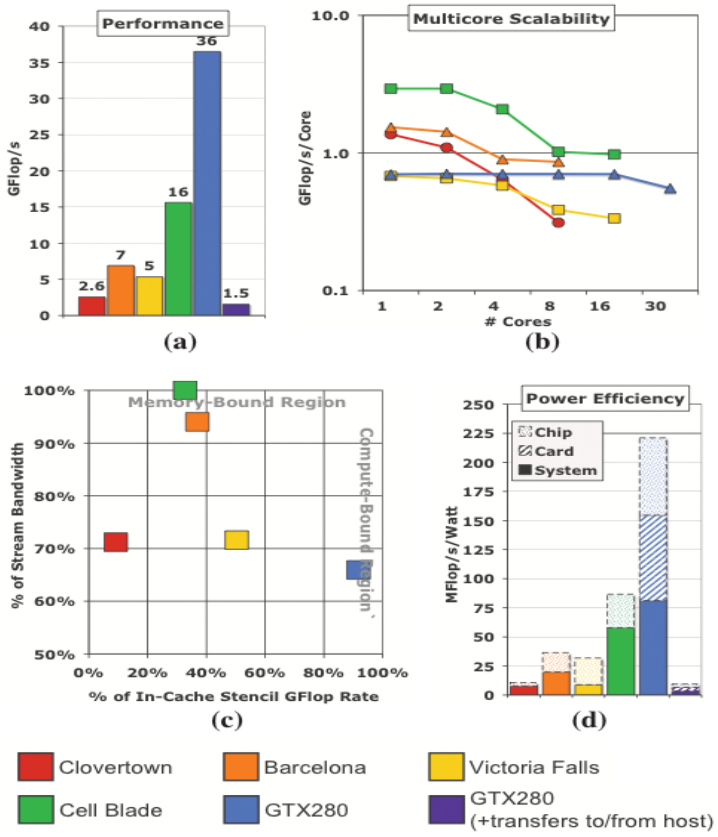

図 4: プロセッサの特性

キャッシュベースのプロセッサ Clovertown，Barcelona，Victoria Falls では，いずれも，Blocking やキャッシュバイパスによる効果が高い (図 3)．このうち，Clovertown は FSB の転送速度がボトルネックとなっているため，コアスケーラビリティが低く，それにより，最適化によるピーク性能は，最適化前と比べて約 2 倍程度に留まっており，Barcelona や Victoria Falls に見られるような 6 倍以上の最適化の効果は得られなかった．どのプロセッサもコア数が増えるに従いバンド幅の影響を受けるが，Barcelona ではコアを増やすことで NUMA を意識したデータ配置による最適化の影響が強くなるため，スケーラビリティはそれほど低下しない (図 4(b))．Victoria Falls では，特にコア数を 16 個とした時のパディングやコアブロッキングなどの最適化の影響が強く出ている (図 3)．

ローカルストアベースの Cell では，実効性能，エネルギー効率の両方において高い値を示している．これは，SPE1 個あたりのクロック周波数や，DMA などの最適化の効果が高いことが理由である．ただし，SPE の数が増えるに従い，バンド幅が使いきられてしまうため，コアスケーラビリティの低下は免れない (図 4(b))．

測定に用いたプロセッサの中で唯一の付加プロセッサである GTX280 では，ホストとデバイスを用いる場合とデバイスのみを用いる場合の 2 つの場合で実効性能を測定した．その結果，デバイス単体で計算することにより，GPU の特徴である並列性と高速なメモリバンド幅を活かせるため，処理速度，エネルギー効率の両面において良好であった．一方，ホストも用いる場合，ホスト－デバイス間における PCI Express バスの転送速度がボトルネックとなってしまい，著しく性能が低下することが分かった (図 4(a))．これは，エネルギー効率についても同様である (図 4(d))．

6 おわりに

本研究では，複数種類のプロセッサアーキテクチャに対して，ステンシル計算の最適化を行った．

この最適化によって，複雑なコアを少数搭載したプロセッサに対し，シンプルなコアを多数搭載したプロセッサのほうが高い実効性能とエネルギー効率を得られることが実証された．また，メモリレイテンシの隠蔽については，ハードウェアベースの階層構造による実装に対し，ソフトウェアベースによる管理方式を用いるほうが効果的であった．

プロセッサ毎に適切な最適化パラメータ を設定する Auto-tuning 環境を用いたことで，比較的コア数の少ないプロセッサにおいても，実効性能を高めることに成功した．

参考文献

[1] ”マルチコア”，IT 用語辞典 e-words，http://e-words.jp/w/E3839EE383ABE38381E382B3E382A2.html，(2011-4-18)

[2] ”Stencil(numericala_nalysis)”，Wikipedia，http://en.wikipedia.org/wiki/Stencil_(numerical_analysis)，(2011-4-20)．