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

本多・近藤研究室
学籍番号:1153004
岩下　光弘

平成 23 年 5 月 9 日

発表する論文について

- タイトル
  Stencil Computation Oprimization and Auto-tuning on State-of-the-Art Architectures

- 著者
  Kaushik Datta,Mark Murphy,Vasily Volkov,Samuel Williams,Jonathan Carter,Leonid Oilker,David Patterson,John Shalf,Katherine Yelick

- 出典
  SC 08:Proceedings of the 2008 ACM/IEEE conference on Supercomputing.Piscataway,NJ,USA:IEEE Press,2008,pp. 1-12.

目次

- 背景と目的
- ステンシル計算
- マルチコアアーキテクチャ
- 最適化手法と Auto-tuning
- 最適化の効果とパフォーマンスの比較
- まとめ

背景と目的

マルチコアプロセッサ技術の台頭によって，プロセッサのアーキテクチャは多様化している．

そのため，1つの最適化手法が全てのアーキテクチャに有効であるとは限らない．

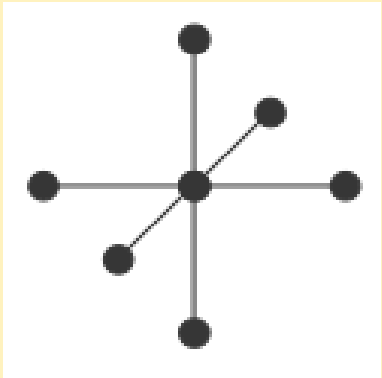
本研究では，アーキテクチャ上それぞれ異なる特徴を持つ5種類のマルチコアプロセッサに対し，並列ステンシル計算プログラムの最適化を行う．

この実行解析によって，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する．

それにより，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開発に与える影響について議論する．

ステンシル計算

- ステンシル計算は近傍計算の一種で，偏微分方程式の解法に用いられる．



- 3D の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．
- ステンシルは構造がシンプルなこと，点の値が一意に定まっていることから，複数のコアやスレッドを用いた並列計算に適している．
- そのため，ステンシル計算はプロセッサのベンチマークプログラムに用いられる．

マルチコアアーキテクチャ

本研究では，以下に示す 5 種類のマルチコアアーキテクチャを用いた.

Intel XEON(Clovertown)
AMD Opteron(Barcelona)

- 設計が複雑なコアを 8 個搭載している.
- ハードウェアの階層構造によるメモリ管理

マルチコアアーキテクチャ

Sun UltraSPARC(Victoria Falls)
STI Cell
NVIDIA GeForce GTX280

- 設計がシンプルなコアを 8 個〜240 個搭載している.
- ソフトウェアベースによるメモリ管理.

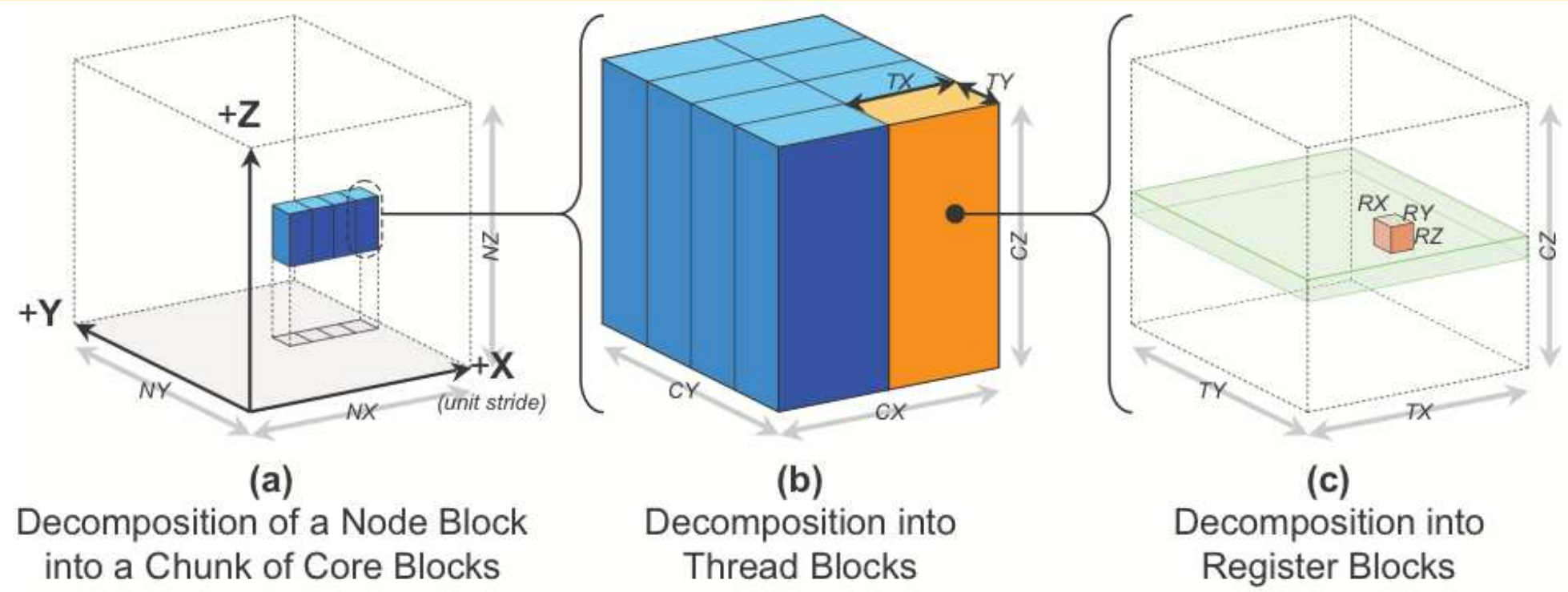
最適化と Auto-tuning

プロセッサの最適化のための 4 つの手法を示す.

1. 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

最適化と Auto-tuning



(a)
Decomposition of a Node Block into a Chunk of Core Blocks

(b)
Decomposition into Thread Blocks

(c)
Decomposition into Register Blocks

最適化と Auto-tuning

2. Data Allocation

データを適切に配置することによる最適化

e.g : padding,NUMA を意識したデータ配置

3. Bandwidth Optimization

メモリレイテンシの隠蔽と，メモリトラフィックの改善による最適化

e.g : プリフェッチング,DMA

4. In-core Optimization

キャッシュへのアクセス回数軽減や，SIMD に特化したコード生成によるコア内のパフォーマンス最適化

e.g : ループアンローリング

最適化と Auto-tuning(2)

Auto-tuning

これまで紹介した最適化技法に加えて，本研究ではプロセッサのパフォーマンス向上のための Auto-tuning 環境を構築した．

- 従来，最適化パラメータの調整は人間による手入力によって行われてきた．
- しかし，人間による手入力では調整に時間がかかる．
- そこで，本研究では，最適化パラメータの調整を自動的に行う Auto-tuning 環境を構築した．
- この Auto-tuning 環境を用いることで，最適化パラメータをより短時間で，かつ自動的に調整できる．

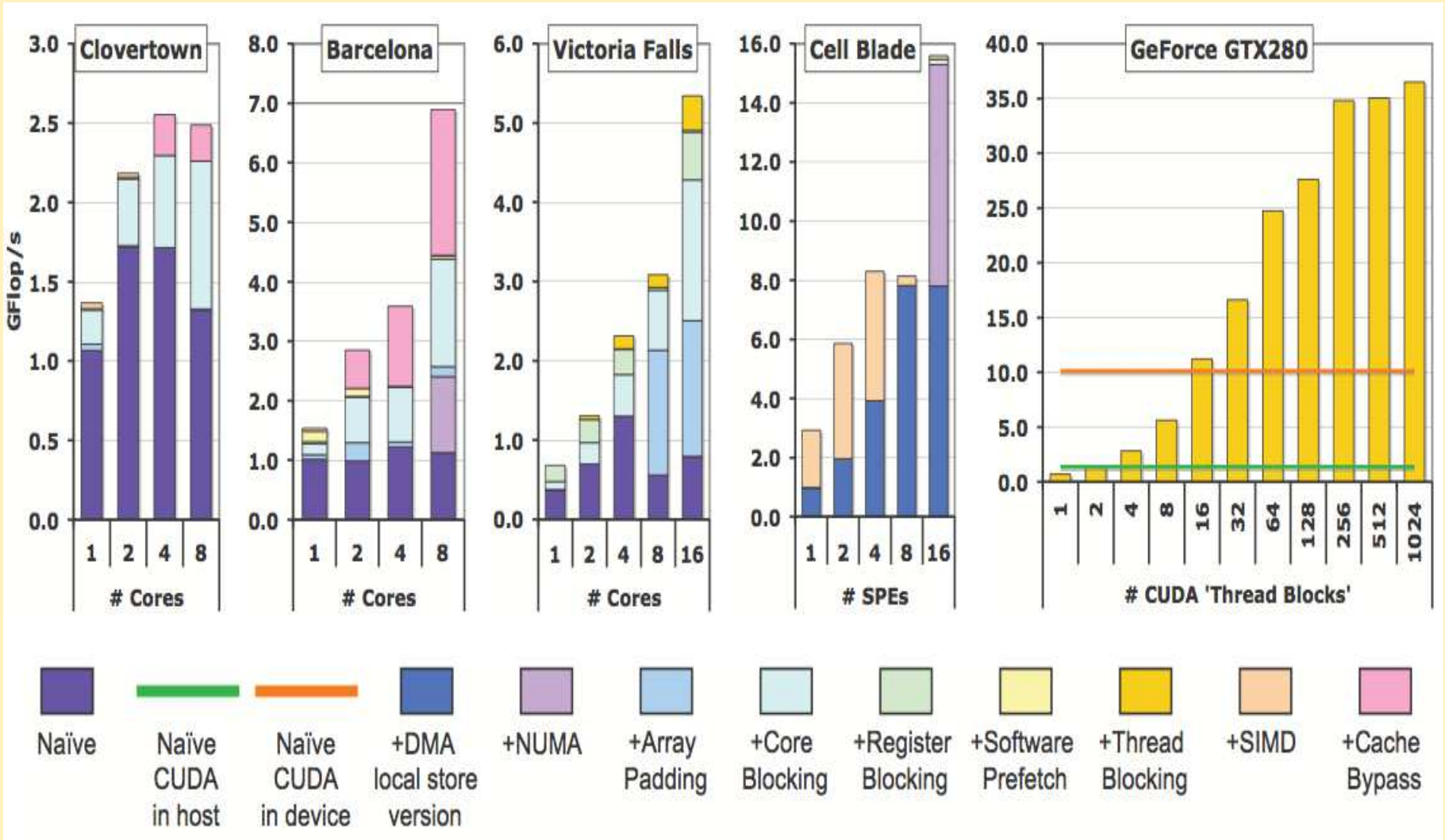
最適化と Auto-tuning(3)

Auto-tuning 環境によるステンシルコードの最適化パラメータ

Category	Optimization Parameter	Name	parameter tuning range by architecture: Clovertown	Barcelona	Victoria Falls	Cell Blade	GTX280
Data Allocation	NUMA Aware		N/A	✓	✓	✓	N/A
	Pad to a multiple of:		1	1	1	16	16
Domain Decomp	Core Block Size	*CX*	NX	NX	{8...NX}	{64...NX}	{16...32}
		CY	{8...NY}	{8...NY}	{8...NY}	{8...NY}	CX
		CZ	{128...NZ}	{128...NZ}	{128...NZ}	{128...NZ}	64
	Thread Block Size	*TX*	CX	CX	{8...CX}	CX	1
		TY	CY	CY	{8...CY}	CY	CY/4
	Chunk Size		$\{1...\frac{NX \times NY \times NZ}{CX \times CY \times CZ \times NThreads}\}$				N/A

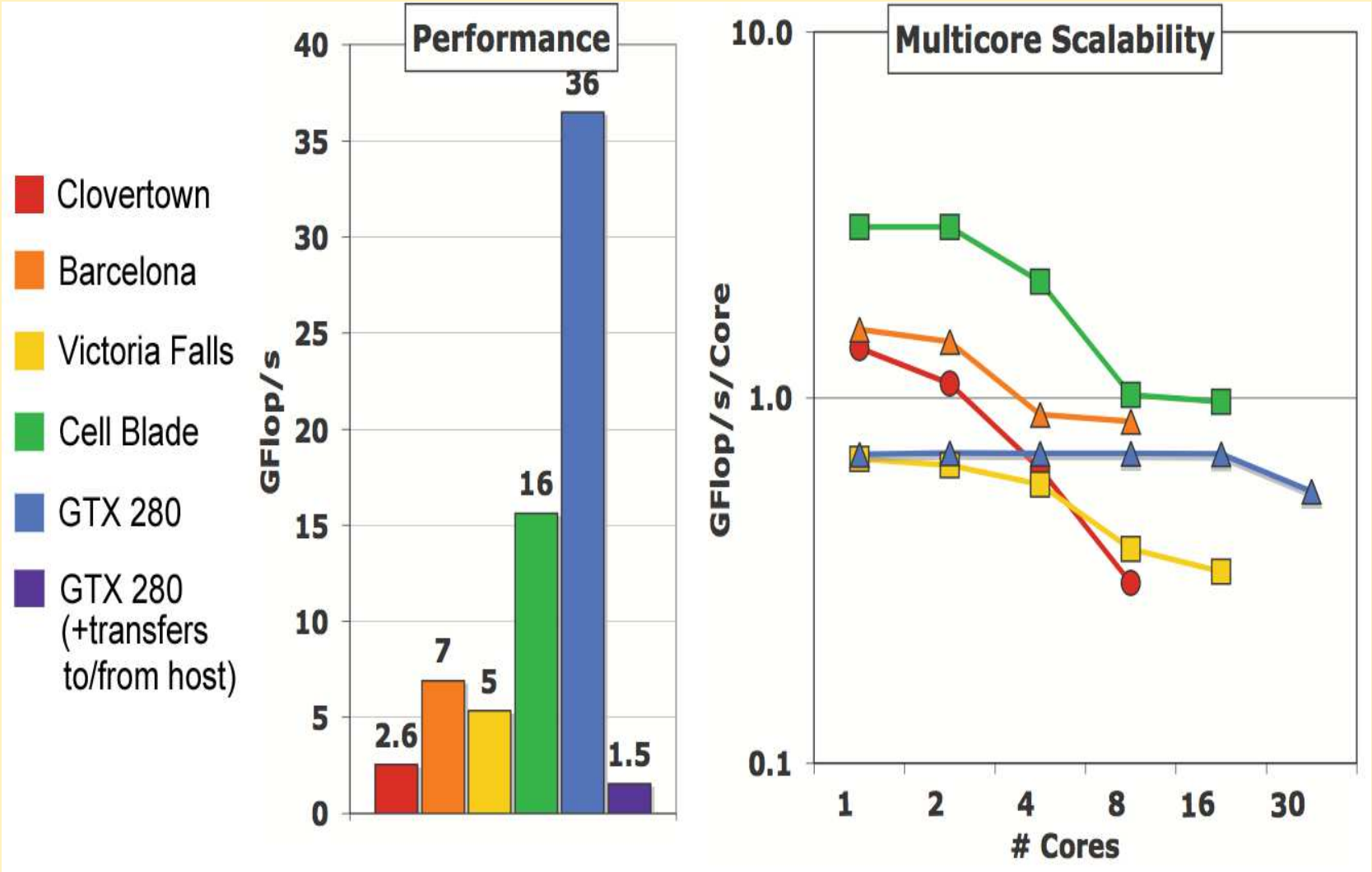
最適化・フォーマンスの比較

グラフ中のそれぞれの色のバーが最適化の効果を表す.
ただし，紫色のバーは最適化を施す前のプロセッサの性能を表す.



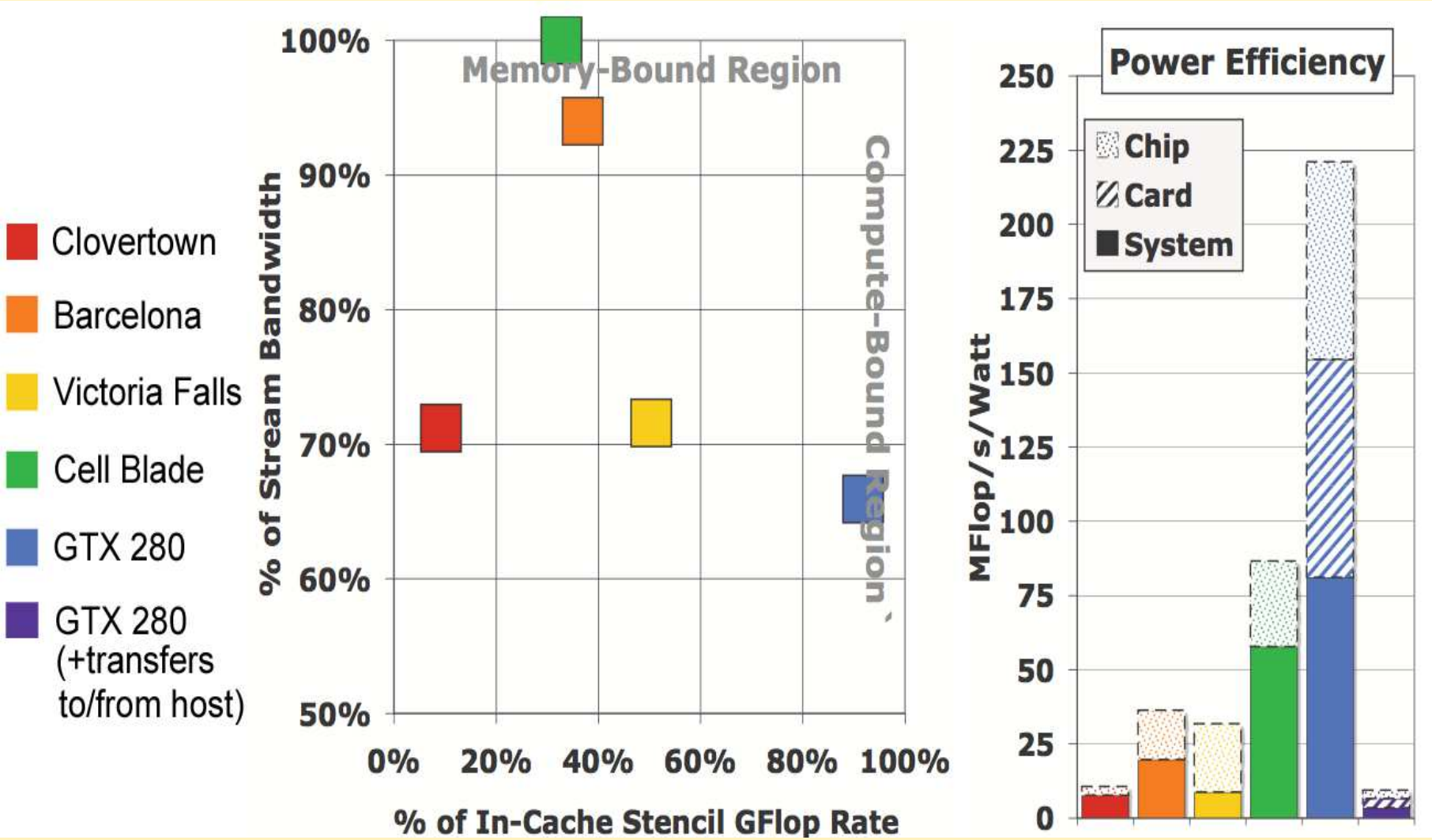
最適化・パフォーマンスの比較 (2)

左側の棒グラフは各プロセッサのピークパフォーマンスを表す．また，右側の折れグラフは，コアのスケーラビリティを表す．



最適化・パフォーマンスの比較 (3)

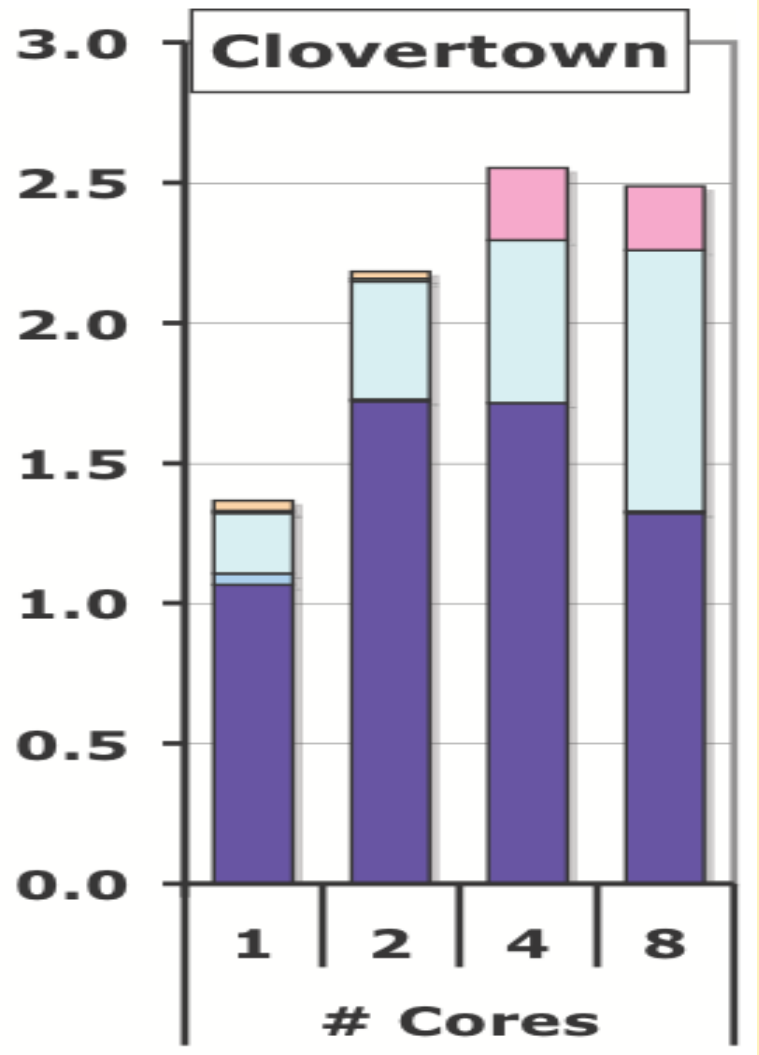
左側のプロット図は，計算性能とメモリバンド幅の利用率を表す．右側の図棒グラフは，エネルギー効率を表す．



プロセッサのパフォーマンス

Clovertown のパフォーマンス

- NUMA を意識したデータ配置の効果は薄い.
- コアを増やしても，最適化の効果は低い.
- その理由として，FSB の実装によるバンド幅のボトルネックがあげられる.

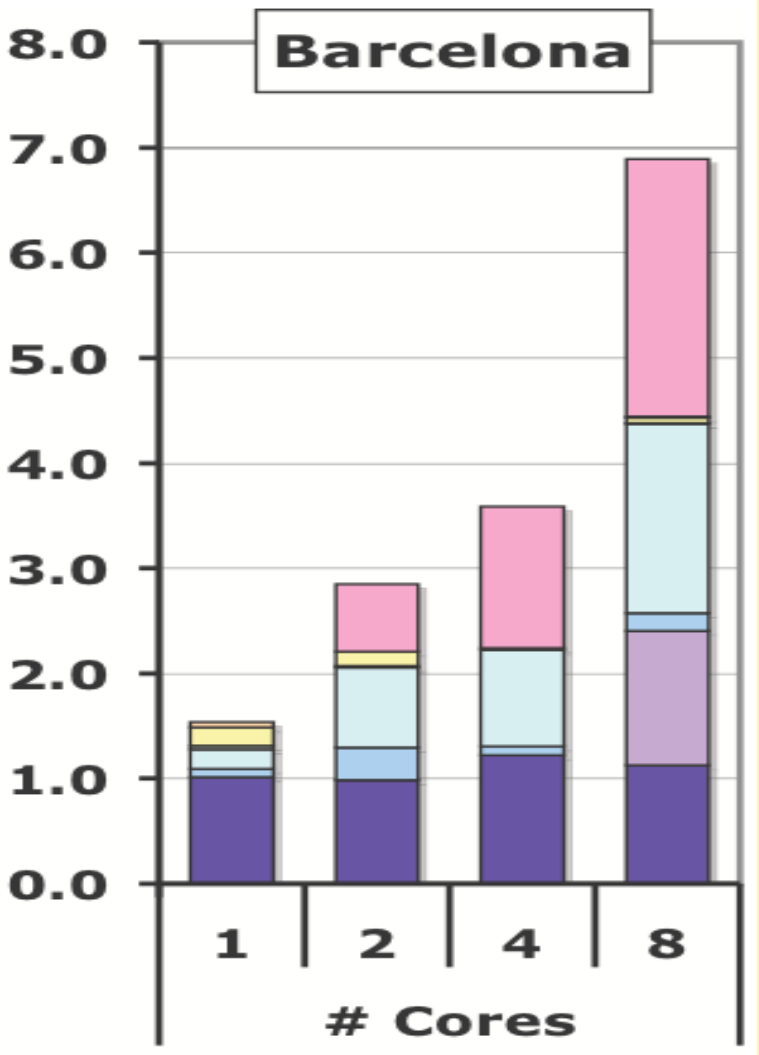


単位:GFlop/s

プロセッサのパフォーマンス (2)

Barcelona のパフォーマンス

- メモリコントローラの実装
- それによるバンド幅の改善
- ブロッキングとキャッシュバイパスによる効果が高い.
- 8 コアでは NUMA による効果が顕著である.

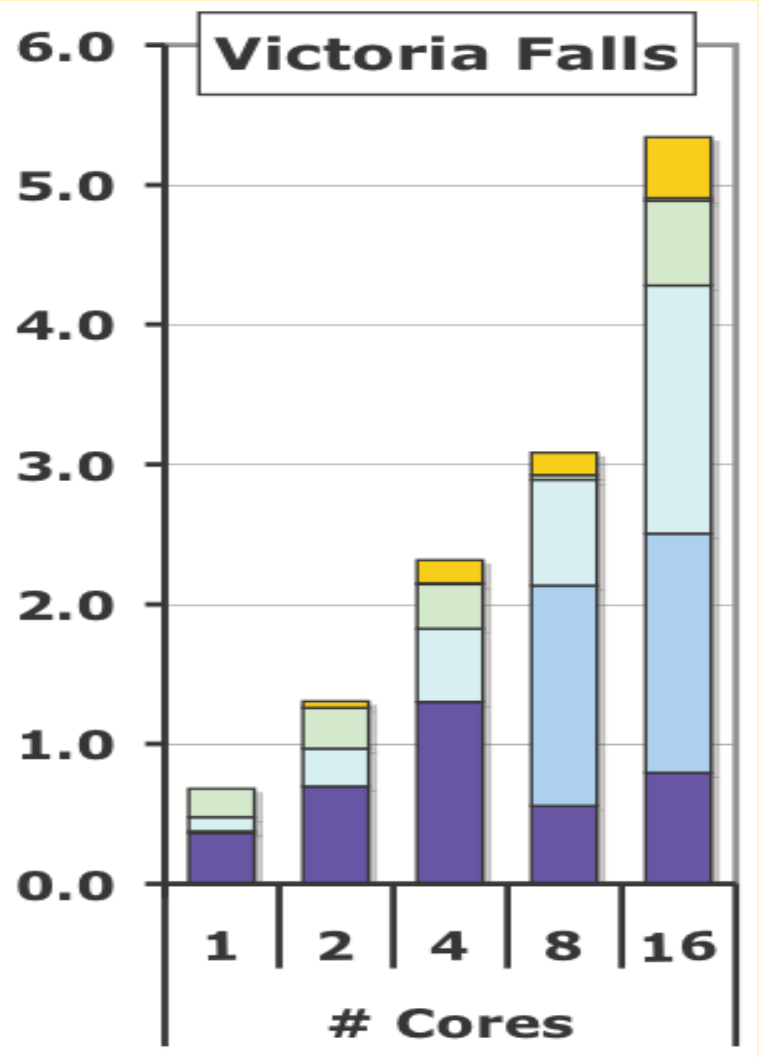


単位:GFlop/s

プロセッサのパフォーマンス (3)

Victoria Falls のパフォーマンス

- パディングとコアブロッキングによる効果が高い.
- コアを増やすほど最適化の影響が大きい.
- スケーラビリティの低下は少ない.

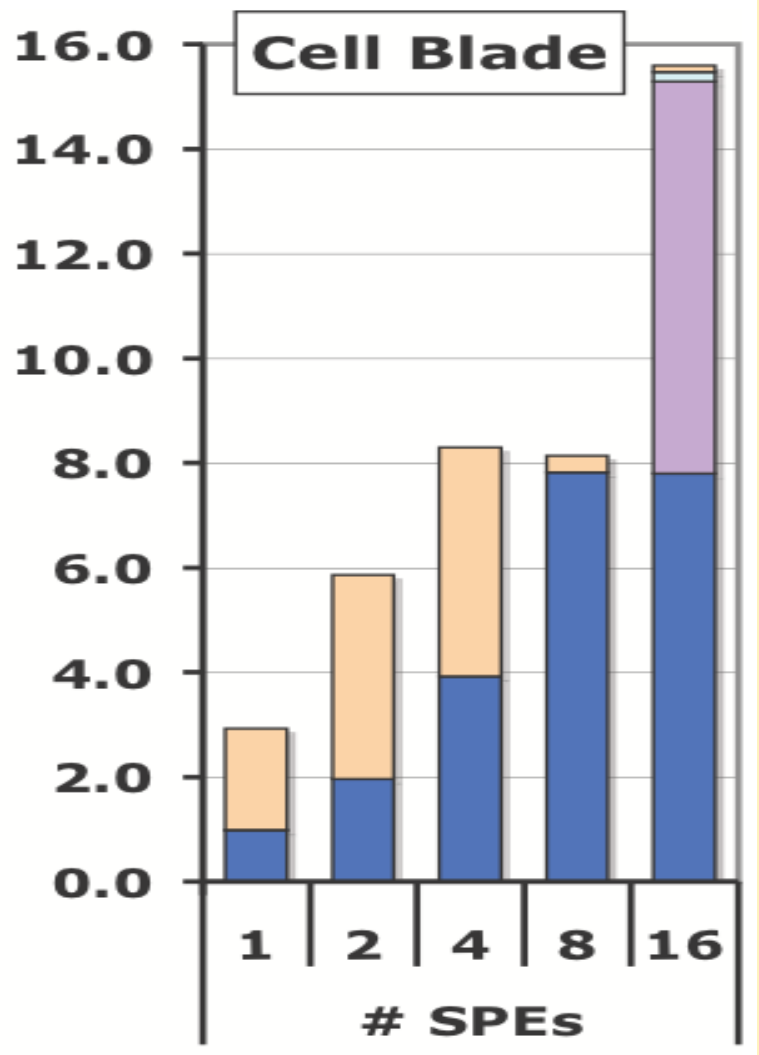


単位:GFlop/s

プロセッサのパフォーマンス (4)

Cell のパフォーマンス

- 1〜4 コアでは，SIMD による効果が高い．
- 16 コアでは，NUMA による効果が顕著である．
- バンド幅を最大限まで活用している．
- SPE を 8 個以上に増やすと，スケーラビリティが大きく低下する．

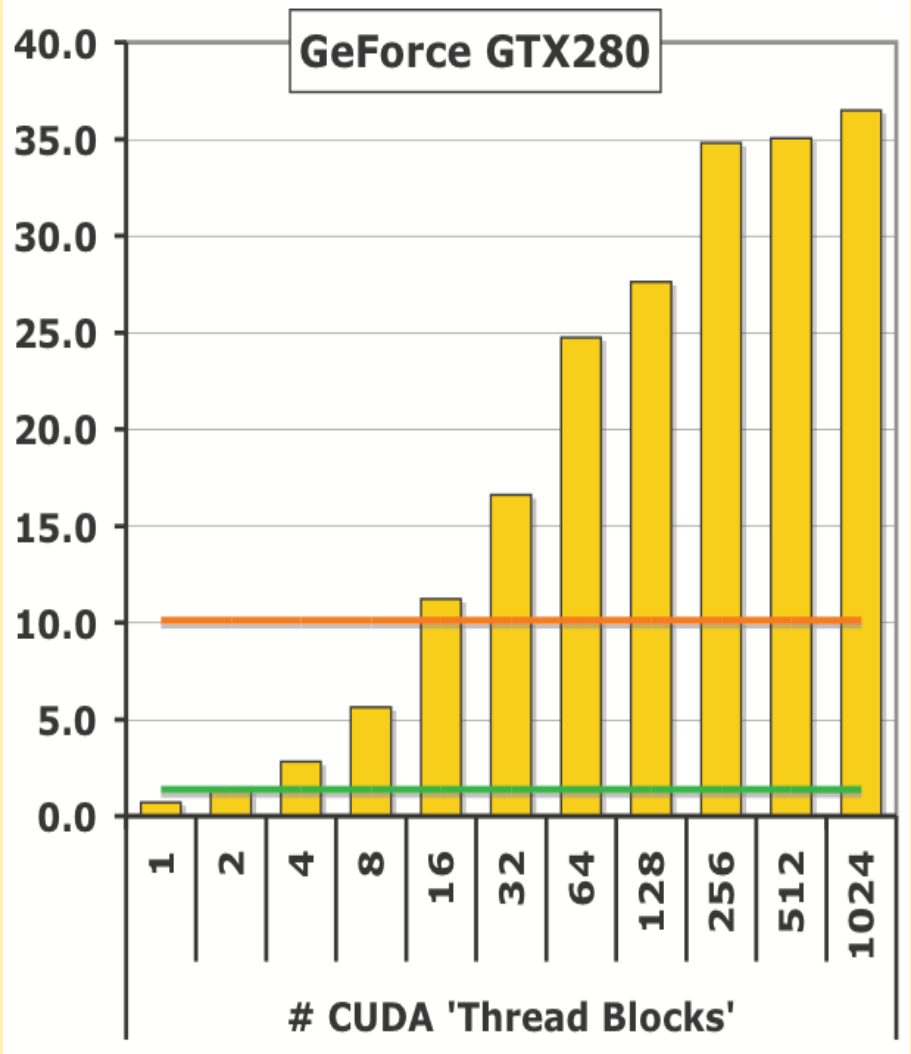


単位:GFlop/s

プロセッサのパフォーマンス (5)

GTX280 のパフォーマンス

- バンド幅を活かしきれないが，計算パフォーマンスは非常に高い．
- デバイスのみで計算することで GPU としての高い並列性とバンド幅を活かせる．
- ホストも利用する場合，PCI Express バスのバンド幅がボトルネックになる．
- このボトルネックは，計算パフォーマンスだけでなくエネルギー効率にも影響する．



単位:GFlop/s

まとめ

- ステンシル計算においては，複雑なコアを少数搭載したプロセッサよりも，単純なコアを多数搭載したプロセッサのほうが高いパフォーマンスが得られた．
- また，メモリのレイテンシを隠蔽する方法として，ハードウェアでメモリを管理する方式よりも，ソフトウェアでメモリを管理する方式のほうが効果的であった．
- Auto-tuning 環境は，プロセッサにおける最適化パラメータの設定を迅速かつ自動的に行うために大変に重要である．