グリッドコンピューティング授業

08M37284　浜野智明

取り上げる論文

- On the Design and Analysis of Irregular Algorithms on the Cell Processor: A Case Study of List Ranking
  - David A. Bader, Virat Agarwal, Kamesh Madduri
  - IPDPS2007

背景：Cell Broadband Engine(1/2)

- Sony、東芝、IBMにより開発
- 特定の処理目的に特化し設計されたプロセッサ
  - 大量のデータ処理に反復的な処理を行うアプリケーション
- 高いピーク性能：単精度204.8GFlops(3.2GHz)
- 現在、PS3の他にアクセラレータとしてRoadrunner（IBM）等にも搭載

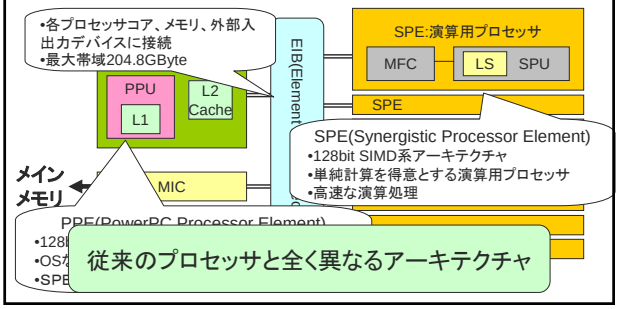
➢Dual core Opteron 1.8GHz：6562

➢PowerXCell 8i 3.2GHz：12240

⇨ 1,376TFlops

背景：Cell Broadband Engine(2/2)

- ヘテロジニアスマルチコア
  - 1個の汎用的なプロセッサコアと8個の演算用プロセッサコア



現状の問題点

- Cellのアーキテクチャは従来のプロセッサと全く異なる

→Cell用の新しいアルゴリズムを考える必要

- しかし、従来の計算モデルはCell用アルゴリズムの解析には使えない
  - RAMモデル→命令数≒実行時間

⇒Cellアーキテクチャ用の新しい計算モデルが必要

目的と成果

- 目的
  - Cellアーキテクチャ用計算量モデルの構築
- 成果
  - 計算量モデルの提案
  - モデルを用いたシステマティックなアルゴリズム解析手法の提案
  - List Ranking問題にモデルを適用し、アルゴリズムの解析、パフォーマンスの改善
    - DMA転送レイテンシの隠蔽

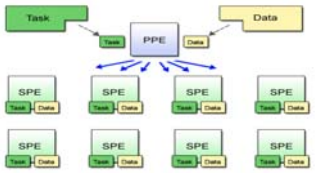
提案計算モデル

- 次の3つからパフォーマンスを解析

1. PPE、SPEそれぞれにおける計算量
2. DMA転送のレイテンシ
3. SPEにおける分岐命令

PPU、SPEそれぞれにおける計算量

- 一般的なCellプログラムではPPEでメインプログラムを、SPEでサブプログラムを実行
  - PPEで、対象となる一連の処理やデータを分割してSPE上で実行されるサブプログラム に振り分ける
  - SPEは振り分けられた処理を実行し、PPEへ結果を返す



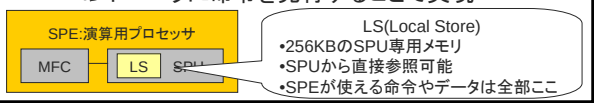
PPU、SPEそれぞれにおける計算量を考慮すべき

DMA(Direct Memory Access)転送のレイテンシ

- DMA転送とは
  - CPUを介すことなく、デバイス⇔メモリ間のデータ転送を行なう手法
  - CPUがDMAコントローラにDMA転送命令を発行することで実現
  - 命令はキューイングされ、CPUの実行中に動作が可能

CellにおけるDMA転送

- SPEはLS以外のメモリには直接アクセスできない
  - メインメモリにはアクセス不可
  - SPEがデータに対し処理を行う場合、ここにload
- DMA転送によりLS⇔メインメモリ間のアクセスを実現
  - SPEがMFC(memory flow contraller)内のDMAコントローラに命令を発行することで実現



DMA(Direct Memory Access)転送によるレイテンシ

- SPEはLSしか直接アクセスできない
- LSは256KBと小さい

⇒メインメモリ上の不連続なデータに対する処理が必要なプログラムを実行する場合、多くのDMA転送が必要

- また、DMA転送のレイテンシは90ns (270clock sycle)と比較的大きい
- データ転送を頻繁に行うプログラムの場合、DMA 転送レイテンシ > 演算処理時間となるかも

考慮すべき

SPEにおける分岐命令

- SPEは分岐予測器を持たない
  - コアを単純化することにより高クロックを実現
- 代わりに、コンパイラが作成する分岐ヒント命令によりソフトウェア的に分岐予測が行われる
- しかし、分岐予測ミスによるペナルティはかなり大きい
  - 18サイクル
- したがって、分岐命令を考慮すべき

提案計算モデル

- 実行時間を<T_C,T_D,T_B>で表現
  - T_C:プロセッサコアにおける計算量
    - ($T_{C,SPE}$)と($T_{C,PPE}$)を合わせたもの
    - ($T_{C,SPE}$):max{($T_{C,SPE(i)}$),1≦i≦p}
  - T_D:DMA命令数の最大値(全SPEにおける)
  - T_B:各SPEにおける分岐命令数

⇒支配的な項を見つける

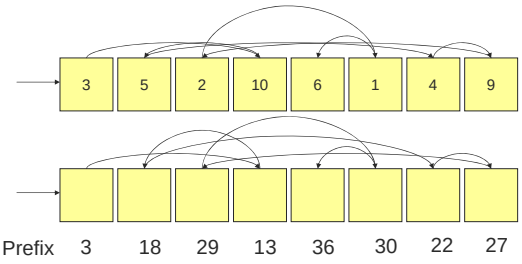
$T_C=O(n^3)$,$T_D=O(n^2)$,$T_B=O(n^2)$
⇒プロセッサコアにおける計算がボトルネック

提案モデルを用いたアルゴリズム解析手法

- 以下の手順で解析を行う
  1. $T_{C,SPE}$の解析
  2. T_Dの解析
     - 定数であれば、つまり、O(t)であればDMA転送は無視できることがこの時点でわかる
  3. DMA転送中にSPE上で有効な計算ができるか検討
     - できるのであれば、DMA転送によるレイテンシは隠蔽することができる
  4. 分岐命令の数を調べる
     - できるだけ少なくする

List Rankingとは(1/2)

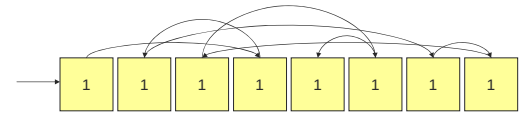
- Prefix Problem
  - X(i).prefix = X(i).value ※ X(predecessor).prefix
  - ※ = ＋ のときの例が下



Prefix 3 18 29 13 36 30 22 27

List Rankingとは(2/2)

- Prefix Problemの特殊系
  - ※ = ＋ かつ X(i).value = 1 for all i



Ranks 1 3 6 2 8 7 4 5

- メモリへのランダムアクセスが必要

⇒Cellで実装すると多くのDMA転送が必要

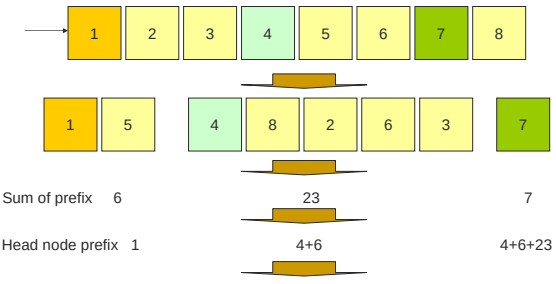
- 逐次に行うとO(n)

Prefix Problemの並列アルゴリズム

- SMP algorithm [Helman & JaJa,1999]
  1. ランダムにsコノードを選び、それらをヘッドノードとするサブリストをsコ作る
  2. それぞれのサブリストでprefixの合計を計算
  3. 2.を用い、それぞれのサブリストのヘッドノードのprefixを計算
  4. それぞれのサブリストでprefixを1ノードずつ計算
- これを単純にCellへ適用

  ⇒s = 8

Prefix Problemの並列アルゴリズム



提案解析手法を用いての解析

1. T_Cの解析
   - T_C = O(n/p) ； Step2,4における$T_{C,SPE}$の平均
2. T_Dの解析
   - T_D = O(n/p)；Step2

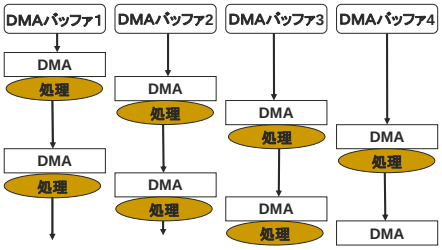
   ⇒DMA転送は無視できない
3. DMA転送レイテンシを隠蔽できるか検討
   - Step2においてDMA転送中にSPEは何もしていない

   ⇒DMA転送中にSPEに処理をさせればレイテンシを隠せればパフォーマンスの改善が見込める

DMA転送レイテンシの隠蔽(1/2)

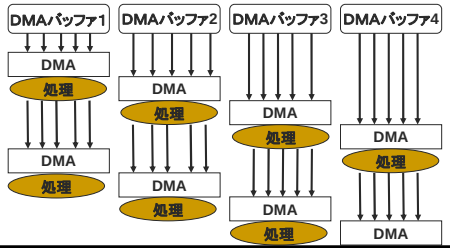
- 基本的なアイデア

  スレッドを複数つくり、あるスレッドがDMA転送の完了待っている間に他のスレッドを動かす



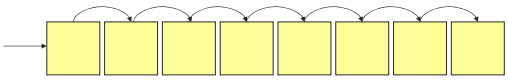
DMA転送レイテンシの隠蔽(2/2)

- 実装
  - 1スレッドに1個サブリストを割り当て
  - しかし、これではDMA転送が多くなりすぎる
  - b個のスレッドのメモリへのアクセスを1つのDMAリスト転送にまとめる

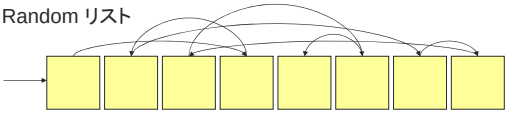


評価環境

- IBM BladeCenter QS20
  - 3.2GHz Cell BE *2
  - 1GB memory
- 対象リスト
  - Ordered リスト

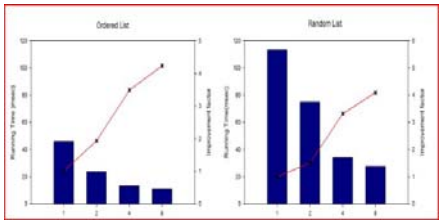


  - Random リスト



評価結果：DMAバッファ数変更による性能変化

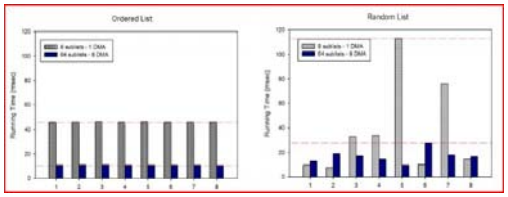
- list size 2^{20}



Ordered List、Random Listともにバッファ数が増えるにつれ性能改善

評価結果：SPE間のroad balance

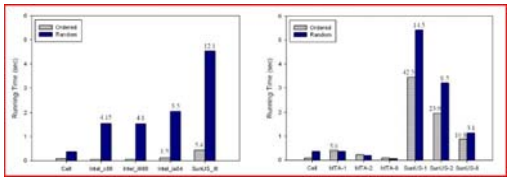
- list size 2^{20}



•Ordered Listは全てのサブリストの長さが等しくなるので、バランス
•Random Listはサブリストによって長さが異なるのでバランスしていない→サブリスト数を増やすことでバランス

評価結果：他プロセッサとの比較

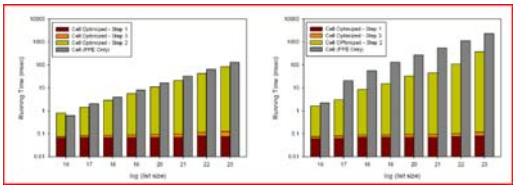
- List size 8 million



Cellにより他プロセッサと比較して高いパフォーマンスが実現したことがわかる

評価結果：逐次実装との比較

- PPEだけを用いた逐次実装との比較



•Ordered List,Random List共にパフォーマンスの改善
•特にRandom Listで大きな改善

まとめ

- Cell用計算モデルの構築、提案
- モデルを用いたアルゴリズム解析手法の提案
- リストランキング問題にモデルを適用し、パフォーマンスの改善