

共有資源の競合に着目したCMP向け
実行フェーズスケジューリング手法
についての研究

高性能コンピューティング学講座 近藤研究室
0853009 久米正人

概要

チップマルチプロセッサ(CMP)では、キャッシュやメモリバスなどの資源が共有になっているため競合が生じやすい→**チップ全体の性能が著しく低下する**

↓

このため、共有資源の競合をできる限り回避することが重要

競合による性能低下

競合が生じると.....

- 共有キャッシュに関して
 - データの追い出し合い, etc...
- メモリバスに関して
 - メモリにアクセスできずコアがアイドル状態となる, etc...

以上の現象により、**CMPが持つ性能を十分に得られない可能性がある**

提案

競合に着目したプロセススケジューリングなどが従来研究としてあるが、一般に1つのプロセス内でも共有資源の要求率は変化

↓

共有資源の競合を減らし、より細粒度な実行フェーズ単位でのスケジューリング手法を提案する

研究の手法

実行フェーズの中から、要求帯域幅が共有資源のバンド幅を超えないような組み合わせを選択してスケジューリングを行う

- 実行フェーズ: プログラム(=プロセス)を単位時間ごとに分割した単位

Manipulator

- (1つのコア上では1つのプロセスしか同時に実行されないものとする)
- 全てのプロセスの動きの制御を司るManipulatorを考える

提案手法のモデル

動かすプロセス

提案手法のモデル

Manipulatorはプロセスに対し、パフォーマンスカウンタ値を要求するシグナルを送る

提案手法のモデル

プロセスは、Manipulatorに対し、パフォーマンスカウンタ値を共有メモリ等を用いて送信

2. 研究の手法

提案手法のモデル

Polling Interval

Manipulatorは、得られたデータから予測を行い、次に動かすプロセスを選択する

HPC演習センター 100
7 September '09 10

2. 研究の手法

提案手法のモデル

Polling Interval

Manipulatorはプロセスに対し、動きを制御するシグナルを送る

HPC演習センター 100
7 September '09 11

3. 最近の実験

最近の実験

- 提案手法が従来手法に比べて優位な結果を示すケースを見つけるための実験を行っている
- この実験に使用したプログラムを次に示す

HPC演習センター 100
7 September '09 12

3. 最近の実験

使用プログラム (prg0.c)

```

1 int i, j, k;
2 char a = 0, *j;
3 U = ((char *)malloc(268435456 * sizeof(char)));
4 for(i = 0; i < 268435456; i++) U[i] = 1;
5 for(k = 0; k < 6; k++){
6   for(j = 0; j < 808; j++){
7     for(i = 0; i < 268435456; j += 256)
8       a += U[i]; // cache miss
9     for(j = 0; j < 58; j++){
10      for(i = 0; i < 268435456; i++)
11        a += U[i]; // cache hit
12    }
13 printf("solution = %d\n", a);
14 free(U);

```

HPC演習センター 100
7 September '09 13

3. 最近の実験

使用プログラム (prg0.c)

- このプログラムは、キャッシュヒット部分とミス部分が両方とも10秒になるように作られている
- 単独で実行すると、このセットを6回繰り返し、120秒程度で終了する

HPC演習センター 100
7 September '09 14

3. 最近の実験

実験手順

- 実験手順
 - Core i7 940 (k008) 上にて、8個のprg0を同時に動作させて計測を10回行い、その平均を結果とした
- 予測方法 (カウンタ値はキャッシュミス回数を使用)
 - (a) 過去15個のカウンタのデータから最小二乗法を用いて近似式を求め、次のフェーズのカウンタ値を予測
 - (b) 1個前のデータをそのまま使用

HPC演習センター 100
7 September '09 15

3. 最近の実験

(a) 最小二乗法

HPC演習センター 100
7 September '09 16

3. 最近の実験

(b) 1つ前をそのまま

HPC演習センター 100
7 September '09 17

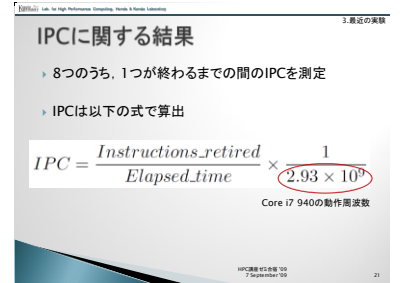
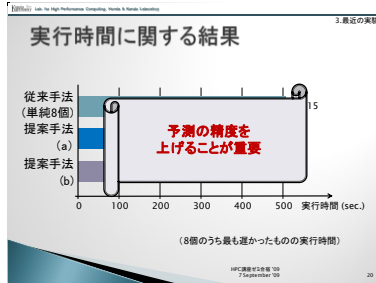
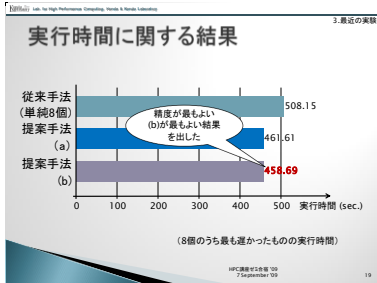
3. 最近の実験

実行時間に関する結果

手法	実行時間 (sec.)
従来手法 (単純8個)	508.15
提案手法 (a)	461.61
提案手法 (b)	458.69

(8個のうち最も遅かったものの実行時間)

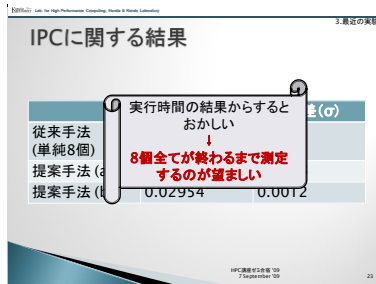
HPC演習センター 100
7 September '09 18



IPCに関する結果

	IPC	標準偏差 (σ)
従来手法 (単純8個)	0.03010	0.0011
提案手法 (a)	0.02950	0.0008
提案手法 (b)	0.02954	0.0012

IPC測定システム 100
7 September '09



- 今後の方針
- SPECなどのベンチマークで測定
 - 実行の公平性の測定
 - 予測の精度の評価
予測値と実測値の差の測定
 - スケジューリング機構をもっと作りこむ
- IPC測定システム 100
7 September '09

まとめ

- 提案手法が従来手法に比べて優位な結果を示すケースを見つけるための実験を行った
- 「予測の精度が高くなると実行時間が短縮される」ということより、本研究においては予測の精度を上げることが重要である

IPC測定システム 100
7 September '09