

計算機アーキテクチャ特論 (Advanced Computer Architectures)

レポート, TSUBAMEの見学

吉瀬 謙二 計算工学専攻
kise_at_cs.titech.ac.jp www.arch.cs.titech.ac.jp
W832 講義室 金曜日 13:20 - 14:50

1

1: 分岐予測の実装と評価

- **Bimode分岐予測**を実装し, その予測ミス率を測定せよ. また, Gshare分岐予測との予測精度の比較を示せ.
 - ハードウェア量を 2KB, 4KB, 8KB, 16KB, 32KB, 64KBとしてグラフを描け.
- **Bimode分岐予測**に工夫を施し (あるいは, ことなる方式の予測を実装し), 予測ミス率を測定せよ.
 - ハードウェア量を 2KB, 4KB, 8KB, 16KB, 32KB, 64KBとしてグラフを描け.

注意: 提出済みの場合, かならずしもこの問題を解く必要はありません.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

2: マルチコアプロセッサ (並列) プログラミング

(2-1)

プロセッサシミュレータSimMcを利用して, 与えられる行列積のプログラム (test64) を4個のコア用に並列化せよ.
4個のコアを用いて, 2倍以上の高速化を達成すること.
コンパイラの最適化オプションを利用しない (-O0を利用する) こと.
ソースコード及び性能向上率を示せ. また, この課題に要した時間を示すこと.

(2-2)

・(課題1)で用いたプログラムを (必要であれば) 修正して, コアの数(1,2,4,8,16)と性能向上率との関係をグラフに示せ. また, この課題に要した時間を示すこと. ここでも, コンパイラの最適化オプションを利用しない (-O0を利用する).
並列化しない逐次プログラムの性能を1として, グラフを描くこと.

(2-3)

コンパイラの最適化オプションをO3として, コアの数と性能向上率との関係をグラフに示せ.
並列化しない逐次プログラム (O3) の性能を1として, グラフを描くこと.
また, 最適化オプションの影響を議論せよ.
この課題に要した時間を示すこと.

Adapted from Superscalar Microprocessor Design, Mike Johnson

2: マルチコアプロセッサ (並列) プログラミング

(2-4)

プロセッサシミュレータSimMcを利用して, 与えられるDEM (Distinct Element Method) のプログラムを4個 (2 x 2) のコア用に並列化せよ.
コンパイラの最適化オプションをO3 とする.
ソースコード及び性能向上率を示せ. また, この課題に要した時間を示すこと.
ソースコードは /home/advance/kise/kadai/dem からコピーすること.

(2-5)

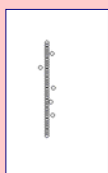
・(課題4)で用いたプログラムを (必要であれば) 修正して, 並列化しない逐次プログラムの性能を1として, グラフを描くこと.
4コアの場合の速度向上率が高いほど高得点とする.

(2-6, 重要)

・プロセッサシミュレータSimMcについての感想をまとめよ.
どこで苦労したか?
どの程度の時間が必要となったか?
期待する改良点.

Adapted from Superscalar Microprocessor Design, Mike Johnson

DEM (Distinct Element Method)



```
int main(int argc, char *argv){
    int i, ti;
    constructor();
    for (ti=0; ti<STEPS; ti++){
        contact_check(p, c);
        force_increment(p, c);
        coordinate_update(p, c);
    }
    draw_particle(t); // draw X
}

for (i=1; i<=ti; i++){
    printf("p[%d] %5.1f %5.1f\n", i, p[i].x, p[i].y);
}
return 0;
}
```

Adapted from Superscalar Microprocessor Design, Mike Johnson

3: 計算機システムの展望

- 10年後の計算機システム (パーソナルコンピュータ) はどのような構成になっているだろうか?
計算機アーキテクチャの視点から議論せよ.
- また, そのような計算機システムを活用するために解決すべき課題 (研究テーマ) について考察せよ.

Adapted from Superscalar Microprocessor Design, Mike Johnson

レポート 提出方法

- 2010年2月12日(金)
午後5時までに電子メール(PDFファイルで添付)にて提出
- kise at cs.titech.ac.jp

```
advance@sc440 dmj$ cd /home/advance/kise/advsl/dm [advance@sc440 dmj]$ make make -C /home/share/M-Core-S2bit/lib make[1]: ディレクトリ /home/share/M-Core-S2bit/lib' に入ります make[1]: 'all' に対して行うべき事はあります。 make[1]: ディレクトリ /home/share/M-Core-S2bit/lib' から出ます /home/share/Cwd/kysel/corbin/kysel-linux-gcc -Wall -O3 main.c -I/home/share/M-Core-S2bit/lib /home/share/Cwd/kysel/corbin/kysel-linux-gcc -static main.o /home/share/Cwd/kysel/corbin/kysel-linux-gcc -o test.out -la MLib.o [advance@sc440 dmj]$ make run make -C /home/share/M-Core-S2bit/lib make[1]: ディレクトリ /home/share/M-Core-S2bit/lib' に入ります make[1]: 'all' に対して行うべき事はあります。 make[1]: ディレクトリ /home/share/M-Core-S2bit/lib' から出ます /home/share/M-Core-S2bit/lib$ make -o2 -o2 -g -m32 module test. ## SimMC: Many-Core and MC Simulator V.1.0.5 2009-06-27 ## Simulator: Simple Coreside Simulator of MIPS Version 0.5.0 2008- ## DEBUG MODE 0, LOG MODE 0 ## core mode (1,0) = (2,3) ## Multi-Core library MLib v0.1.1 2009-07-29 Simple particle simulation version 2.1 by Kenji KISE DEW simulation, 40 particles, 40 steps p[0] 24439.7 22443.4 p[1] 31799.0 24498.2 p[2] 18049.0 27066.0 p[3] 26470.0 20732.0 p[4] 31981.4 22218.1 ## Simulation time 4.130 [sec] ## Simulation cycle 2244390 ## Simulation speed 545411.0 cycle / sec ## Simulation RSS: 3.62 MB [advance@sc440 dmj]$
```

Adapted from Superscalar Microprocessor Design, Mike Johnson

講義用の計算機の使い方

- ユーザ名 advance で 131.112.16.56 にログイン
 - linux など
 - ssh advance@131.112.16.56
 - 講義時に伝えたパスワードでログイン
- 学籍番号でディレクトリを作成して、そこで作業する。
 - mkdir myname
 - cd myname
- 参考ファイルをコピーして実行
 - tar ...

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005