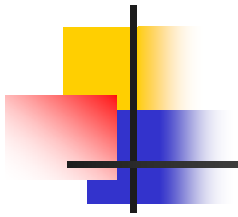


並列アルゴリズム

2005年度後期　火曜　2時限

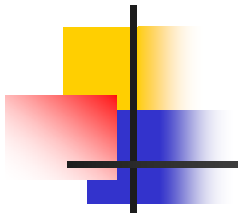
青柳　睦

Aoyagi@cc.kyushu-u.ac.jp

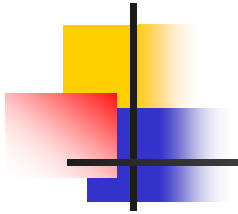
http://server-500.cc.kyushu-u.ac.jp/

1月10日（火）

10. OpenMP の概要



もくじ

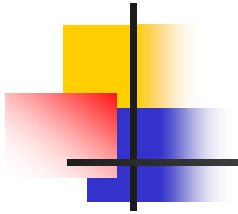


10. OpenMP の概要

- 共有メモリマルチプロセッサ向け並列プログラミングのためのプログラミングモデル

- ベース言語(Fortran/C/C++)をdirective（指示文）で並列プログラミングできるように拡張
  - 新しい言語ではなく，コンパイラ指示文（directives/pragma）、ライブラリ、環境変数によりベース言語を拡張
  - Fortran： !$OMPから始まる指示行
  - C: #pragma omp のpragma指示行

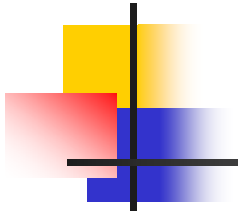
- 仕様
  - 米国コンパイラ関係のISVを中心に仕様を決定
  - OpenMP Architecture Review Board (ARB)
  - URL　http://www.openmp.org/



OpenMP の特徴(1)

- 共有メモリモデルの方が並列化が簡単
  - データ並列が簡単にできる．Owner Computes Ruleでは，まずデータ所在（分割）を考える．　SPMD並列モデルとして自然．
  - Single Address Space (利点・欠点あるが)
- Incremental な並列化が可能
  - まずはHotSpotのみをOpenMP並列化してみるなど，部分的な並列化が気軽にできる
  - （指示行は逐次コンパイラにとってはコメント行なので）逐次コードと並列コードが一元管理できる
- 大規模システムはSMPクラスタが主流?
  - 分散メモリマシン(PCクラスタなど）Onlyの大規模化は無理
  - SMP内ではまずはOpenMP(*)

(*) SMPクラスタでは，OpenMP + MPI ハイブリッドな並列化が必要



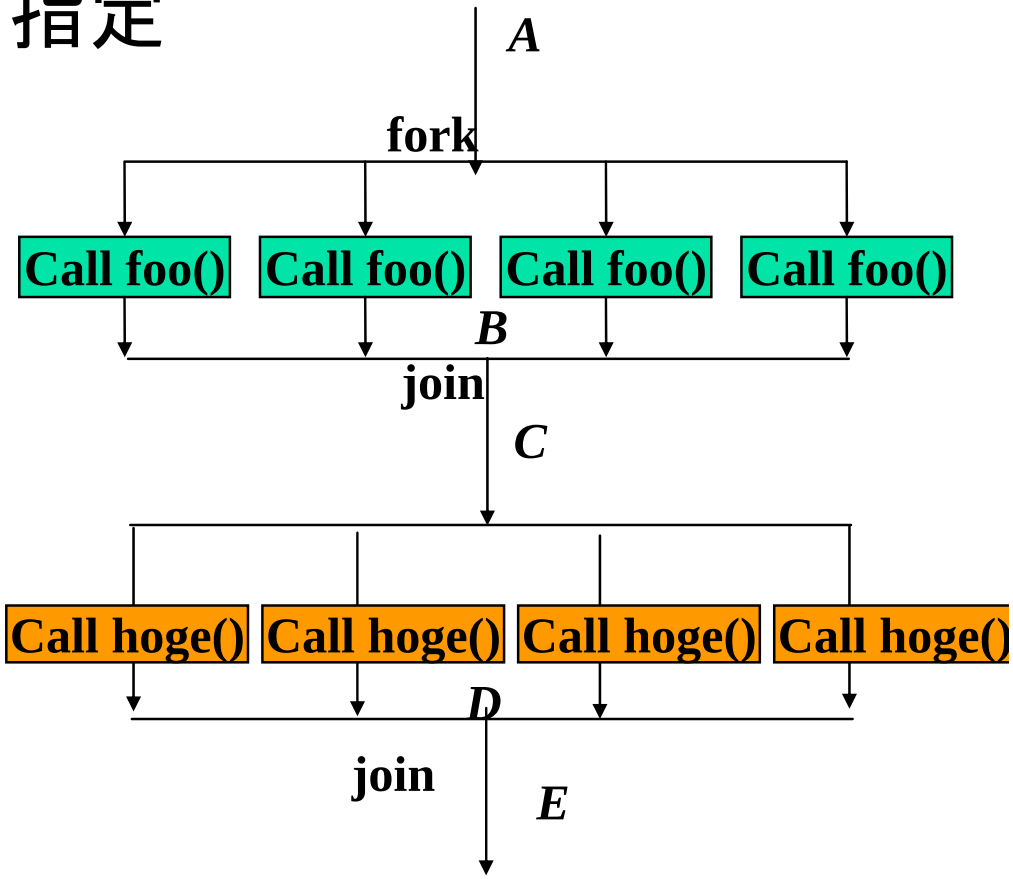
OpenMP の特徴(2)

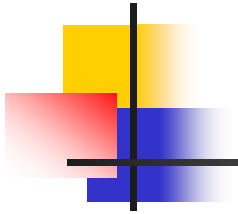
- 細かなtask並列指示，複雑な同期チューニングは難しい
  - データ並列向き？，MPIのような細かな同期や，nesting した並列構造に対応していない(c.f. MPIはCOMM_WORLDの階層を許す)
  - 通信はimplicitに実行される(send&recieve不要だが)，明示的な同期や通信制御を含むチューニングができない
- 一般にScalability 低い(大規模並列には不向き)
  - 16-64プロセッサ程度のSMPマシン向き
- 一般に分散メモリシステムでは利用できない
  - 分散メモリシステムでもソフトウェアで共有アドレスを実現するソフトウェア分散共有メモリ（Ditributed Shared Memory: DSM）システムが研究・開発されており(例： SCASH, TreadMarks)，これを用いてOpenMP並列処理は可能だが，通信ボトルネックに注意

OpenMP の並列実行モデル

- Fork-joinモデル
- parallel regionの指定

```
… A …
#pragma omp parallel
{
    foo(); /* ..B... */
}
… C ….
#pragma omp parallel
{
  hoge(); /* ..D..*/
}
… E …
```





Parallel Region

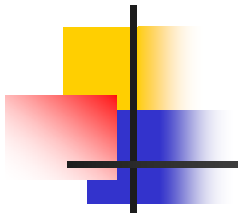
- 複数のスレッドによって、並列実行される部分をParallel Region と呼ぶ
  - Parallel構文で指定
  - 同じParallel regionを実行するスレッドをteamと呼ぶ
  - region内をteam内のスレッドで重複実行する

Fortran:

```
!$OMP PARALLEL
    …
    … parallel region
    …
!$OMP END PARALLEL
```

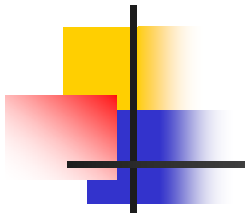
C:

```
#pragma omp parallel
{
    …
    … Parallel region…
    …
}
```



Parallel region とスレッド

- スレッド ID
  - 実行時ライブラリ関数 omp_get_thread_num()で得る(IDは、Team内の0から始まる正整数)
  - マスタスレッド ID=0, IDを使って違うデータにアクセス
- スレッド数
  - 実行時ライブラリ関数 omp_set_num_threads(*nthreads*)で設定(環境変数 OMP_NUM_THREADS)
- 同期
  - parallel regionの最後でjoin
  - 指示文による同期 critical, atomic, barrier
  - 実行時ライブラリを用いる ロック関数



Work sharing構文

- Team内のスレッドで分担して実行する部分を指定する構文
  - parallel region内で用いる
- for 構文
  - イタレーションを分担して実行
  - データ並列を記述
- sections構文
  - 各セクションを分担して実行
  - タスク並列を記述
- single構文
  - 一つのスレッドのみが実行
- parallel 構文と組み合わせた記法がよく使われる
  - parallel for 構文
  - parallel sections構文

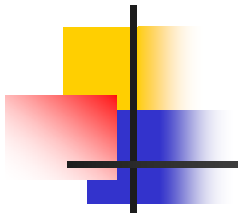
簡単なOpenMP 並列プログラミングの例

例1 ベクトルの和

```
#pragma omp parallel for reduction(+:s)
   for(i=0; i<1000;i++) s+= a[i];
```

例2 疎行列ベクトル積

```
Matvec(double a[],int row_start,int col_idx[],
 double x[],double y[],int n)
{
   int i,j,start,end; double t;
#pragma omp parallel for private(j,t,start,end)
   for(i=0; i<n;i++){
      start=row_start[i];
      end=row_start[i+1];
      t = 0.0;
      for(j=start;j<end;j++)
         t += a[j]*x[col_idx[j]];
      y[i]=t;
  }
}
```



レポート課題（2006年1月10日出題）

【課題】

FFT(高速フーリェ変換)の数値計算に関して，以下の各項目について調査しレポートにまとめて提出してください．

(1)　FFTの数値計算アルゴリズム

(2)　FFTの並列計算の方法

(3) 理工学分野でFFTを利用している具体的な応用分野の説明

提出先：aoyagi@cc.kyushu-u.ac.jp
Subject: 並列アルゴリズム課題
締め切り：平成18年1月24日（火）

形式：
text,pdf,ps,Word,Excel
ファイルの添付可