計算機アーキテクチャ 141－3
（2001．1．26）

分散メモリ並列計算機向け OpenMP 拡張仕様

山中栄次[†‡] 金子正教[†‡] 堀田耕一郎[†‡]

OpenMP Application Program Interface(API)は，ユーザがプログラム中に並列性を記述するのに適しているが，共有メモリ並列計算機向けの言語仕様であるため，分散メモリ並列計算機(DMP)向けに適用しようとした場合に必要となる複数プロセッサ間のデータ分散と通信を記述するための機能が不足している．本稿では，DMP やワークステーションクラスタにおいて効果的に実装可能な OpenMP Fortran API の拡張を提案し，プロトタイプによる NPB ベンチマーク 2 題での評価により，本拡張が DMP に対して有効であることを確認した．

OpenMP Extensions for Distributed Memory Parallel Machines

Eiji Yamanaka,[†‡] Masanori Kaneko,[†‡] Koichiro Hotta[†‡]

OpenMP Application Program Interface (API) is suitable for describing user-directed parallelism in program. However, the current OpenMP lacks important features to be applied to distributed memory parallel computers (DMP). These features are data distribution and communication among multiple processors. We propose a set of extensions for OpenMP Fortran API that can be implemented effectively on DMP machines and workstation clusters. We have roughly designed the set of extensions and made a compiler prototype for the extended OpenMP Fortran API. The performance results show that our extensions are effective for DMP.

1. はじめに

OpenMP Application Program Interface(API)[1]は，High Performance Fortran (HPF)[2]と比較するとユーザが直接プログラム中に並列性を記述するのに適している．しかし，現在の OpenMP は共有メモリ並列計算機向けの並列言語仕様であるため，これをそのまま分散メモリ並列計算機(DMP)に適用しようとしても，複数プロセッサ間におけるデータ分散およびデータ通信を明示的に記述することができないために，プログラムの効果的な並列化ができないことがある．

我々は DMP とワークステーションクラスタに効果的に実装可能な OpenMP Fortran API の拡張仕様を設計し，富士通のスーパコンピュータ VPP シリーズ向けのプロトタイプコンパイラを試作した．本拡張仕様を NAS パラレルベンチマークの 2 題に適用して DMP に対する有効性を確認した．

本稿では OpenMP API の DMP 向け拡張仕様を提案し，プロトタイプコンパイラを用いた性能評価により拡張仕様の有効性を示す．次節では我々の提案する OpenMP Fortran API の DMP 向け拡張仕様の詳細について述べる．3 節では本拡張を NAS パラレルベンチマーク 2 題に適用した場合の性能評価について述べ，4 節で本稿をまとめる．

2. OpenMP API の DMP 向け拡張仕様

OpenMP API を DMP 環境に適用するために，「プロセッサグループ」と「インデックス」という 2 つの概念を導入する．

「プロセッサグループ」は仮想プロセッサの集合である．ユーザはこのプロセッサグループ上でデータを明示的に分散する．

「インデックス」は配列の添字とループの回転数の

† アドバンスト並列化コンパイラ研究体
Advanced Parallelizing Compiler Project
‡ 富士通株式会社
Fujitsu Limited

両方を記述するための抽象概念である.

これら2つの概念をベースに拡張仕様を設計した.

2.1 プロセッサグループ

プロセッサグループは, プロセッサグループの名前 p と仮想プロセッサの数 n とを用いて PROCESSOR ディレクティブによって指定される.

```
!$OMP PROCESSOR p(n)
```

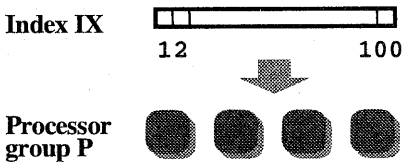
プロセッサグループ上にデータが分散されるとともにプロセッサグループ上でプログラムが実行される. プロセッサグループの形状は1次元配列である.

PARALLEL ディレクティブによって, パラレルリージョンが定義される. パラレルリージョン内では複数のスレッドが生成され, 各スレッドはプロセッサグループ内の仮想プロセッサに割り付けられる. 各スレッドが割り付けられた仮想プロセッサに分散されたデータは PRIVATE として扱うことができる.

2.2 インデックス分散

プロセッサグループとデータの分散との対応関係およびプロセッサグループと処理の分散との対応関係を抽象化するためにインデックスという概念を導入する.

INDEX DISTRIBUTION ディレクティブによってインデックス分散の名前と属性が指定される. 図1にインデックス分散の指定例を示す.




```
!$OMP PROCESSOR P(4)
!$OMP INDEX DISTRIBUTION &
!$OMP& IX=(PROC = P,      &
!$OMP&     INDEX= 1:100,  &
!$OMP&     PART = BLOCK)
```

図1 インデックス分散の指定例

INDEX DISTRIBUTION ディレクティブによるインデックス分散の指定は次の4つの属性で構成される.

PROC

インデックスがマッピングされるプロセッサグループまたはその部分配列の名前の指定.

INDEX

インデックスの上下限の指定.

PART

プロセッサへのインデックスのマッピング方法の指定. この指定形式には OpenMP における SCHEDULE 節と SCHEDULE 節の省略形である BLOCK と CYCLIC のキーワードを含む.

OVERLAP

隣接するプロセッサに重複してマッピングされるオーバラップ領域の幅を指定. HPF では SHADOW と呼ばれている領域である.

2.3 データ分散

OpenMP において変数に PRIVATE または SHARED 属性を指定するための PRIVATE または SHARED ディレクティブを拡張してインデックス分散をパラメータとして付加することにより, プロセッサ上へのデータ分散を指定する. 図2にデータ分散の指定例を示す.

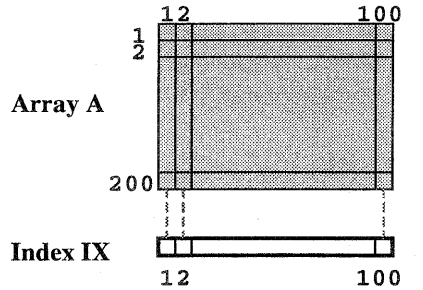


図2 データ分散の指定例

データ分散についてさらに以下のディレクティブを提案する.

RESIDENT ディレクティブ

SHARED データがマッピングされたプロセッサからのアクセスが PRIVATE 扱いで可能なことを指示する.

REDISTRIBUTE ディレクティブ

データの再分散を指示する.

2.4 ループ分散

PARALLEL DO と DO ディレクティブに対してインデックス分散を付加する拡張を行い，ループの各繰り返しのプロセッサへのマッピング方法を指定することでループ分散を記述する．図3にループ分散の指定例を示す．

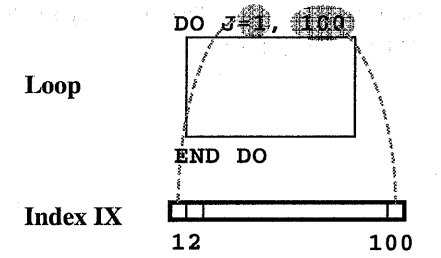


図3　ループ分散の指定

2.5 その他の拡張

効果的なデータ転送のために以下のディレクティブを提案する．

MOVE ディレクティブ

2 つの分散データオブジェクト間の一括データ転送を行う．

OVERLAPFIX ディレクティブ

オーバラップ域の各々の要素を，隣接するプロセッサの対応するデータの値で更新する．

BROADCAST ディレクティブ

PRIVATE データを指定された 1 つのプロセッサから他の全プロセッサに向けてデータを複製する．

UNIFY ディレクティブ

PRIVATE データをインデックス分散で指定された全プロセッサの間で相互に複製する．

2.6 プログラミング例

以下ではインデックス分散と OVERLAPFIX ディレクティブを用いたオーバラップ域を扱うプログラム例を示す．

3 行目と 4 行目で，インデックス分散 IX と IW を各々指定している．これらは IW にオーバラップ域があることを除いて同じである．

6 行目で，配列 A と B をインデックス分散 IW と IX で各々分散されたプライベート変数と宣言している．

配列 A を初期化するために 7 行目と 10 行目で，インデックス分散 IX で分散される並列ループを指定し，11 行目で，A のオーバラップ域を隣接するプロセッサの対応するデータの値で更新することを指定している．

オーバラップ域は，12 行目から 15 行目に示すように，各々のプロセッサ上でプライベートに参照できる．

```
1         PROGRAM OVERLAP_SAMPLE
2  !$OMP  PROCESSOR P(4)
3  !$OMP  INDEX DISTRIBUTION IX=        &
   !$OMP&  (PROC=P,INDEX=1:12,PART=BLOCK)
4  !$OMP  INDEX DISTRIBUTION             &
   !$OMP&  IW=IX(OVERLAP=(1,1))
5          REAL A(12),B(12)
6   !$OMP PRIVATE A(/IW),B(/IX)
           ...
7   !$OMP PARALLEL DO /IX
8          DO I=1,12
9           A(I) = I
10         END DO
           ...
11  !$OMP OVERLAPFIX A
           ...
12  !$OMP PARALLEL DO /IX
13         DO I=2,11
14           B(I)=((A(I1)+2*A(I)+A(I+1))/4
15         ENDDO
           ...
16         END PROGRAM
```

3. 性能評価

我々が提案する OpenMP Fortran API の分散メモリ並列計算機向け拡張仕様を，富士通 VPP シリーズ用に試作したプロタイプコンパイラを用いて評価した．性能評価プログラムとしては NAS パラレルベンチマーク 1.0(Class A)[3]の MG と LU の 2 題を使用した．VPP700 における最高性能との比較として，拡張仕様の性能を評価した．図4，図5に MG，LU の評価結果を示す．VPP700 の 32CPU において，最高性能の 65～70%の性能が OpenMP の DMP 向け拡張仕様によって達成できることを確認した．

MG: Performs a simple multigrid calcula-tions. Highly structured, short and long distance communication.

LU: Regular-sparse, block(5x5) lower and upper triangular system solution. Typified at NASA Ames by the code INS3D-LU.

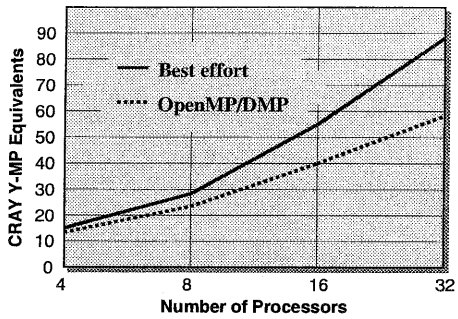


図4　CRAY Equivalents of VPP700 on MG

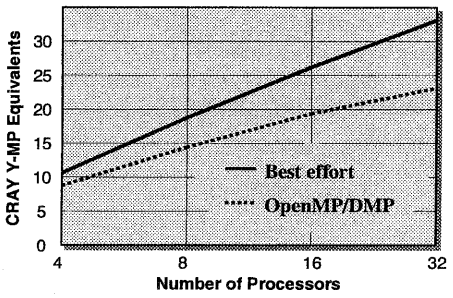


図5　CRAY Equivalents of VPP700 on LU

4. まとめ

本稿では，共有メモリ並列計算機向け言語仕様である OpenMP Fortran API の分散メモリ並列計算機向け拡張仕様を提案した．拡張仕様の VPP700 向けプロトタイプコンパイラを試作して NAS パラレルベンチマークプログラムの MG，LU で性能評価を行った．32CPU で最高性能の 65～70%の性能が得られることを確認し，本拡張仕様が DMP 向けに有効であることを示した．

謝辞

本研究の一部は，技術研究組合新情報処理開発機構における RWC プロジェクトの並列分散コンピューティング技術分野によるものです．

参考文献

[1] OpenMP Architecture Review Board, "OpenMP Fortran Application Program Interface Version 1.0," October, 1997.

[2] High Performance Fortran Forum, "High Performance Fortran Language Specification Version 2.0," January 31, 1997.

[3] D. Bailey, E. Barszcz, J. Barton, D. Browning, R. Carter, L. Dagum, R. Fatoohi, S. Fineberg, P. Frederickson, T. Lasinski, R. Schereiber, H. Simon, V. Venkatakrishnan and S. Weeratunga, "The NAS Parallel Benchmarks," RNR Technical Report, RNR-94-007, March, 1994