XcalableMPのCoarray機能の実現に向けて

理化学研究所／富士通

岩下 英俊

内容

- XMPにCoarray機能を加える意味
- 言語仕様のマージ、結果何ができたのか
- XMP/CのCoarray仕様、最新状況
- OMNI XMP上の実装の状況

Coarray機能とは

- Fortran2008仕様の一部（Fortranは今や並列言語）
  - Coarray Fortranの実行は、複数「イメージ」の同時実行
  - 宣言文の例

    real a(10,10)[*], s[*]
    - これで、変数aとsは全イメージからアクセスできる。
  - 参照と定義の例

    b = a(i,j)[k2] + s[k3]　　a(i,j)[k1] = f(x)
    - イメージk2のa(i,j)とイメージk3のsを取ってきて加算する。
    - イメージk1にあるa(i,j)に右辺式f(x)の結果を送る。

- XMPはこの機能を含む。
  - XMP/FはCAF1.0仕様を包含。XMP/Cにも採用。
  - XMP仕様にマージする上で必要な機能も定義

XMPにCoarrayを加える意味

- 言語仕様の相補性
  - グローバルビュー（XMP）だけでは表現できないアプリがある。
  - ローカルビュー（Coarray）だけでは煩雑、不便。
  - → プログラム内で適材適所の使い分けができるようになる。
- 大規模並列に向けた狙い
  - 連成シミュレーションでは、異種アプリを同時に実行させる。
  - → Coarrayアプリも、部分ノードで実行される１タスクとして結合できるようになる。
- コンパイラ開発の相補性
  - two-sided通信、collective通信はXMPの実装に活用され、既に安定した性能を出している。
  - one-sided通信の活用には選択肢・課題がまだ多い。
    - 通信層の選択肢には、GASNet、MPI3、京ならTofu上のRDMA、低レベル通信ライブラリ、バックエンドコンパイラのCoarray機能‥
  - → Coarrayの実装でone-sided通信の利用技術を確立し、グローバルビュー（タスク間gmove等）にもフィードバックできる。

CoarrayをXMP仕様の中に入れる（XMP規格部会の活動）

- 要件
  - Coarray Fortran（CAF）1.0仕様（Fortran2008に包含される範囲）に対し上位互換
  - XMP指示文と同時に（同じ手続内で）使えること
  - Coarrayで書かれた手続をXMPから呼べること
- 無矛盾な仕様体系とするための検討
  - 突き詰めれば、Coarrayの「イメージ」は、XMPにとって何なのか？という問いだった

並列化単位の比較

XMPの「ノード」

- XMPの並列実行の単位
  全ノードが同時にプログラムを実行
- [data mapping]
  サブノードを再帰的に定義でき、データは任意のサブノードに対して配置できる。
  - 例えば、配列aはnodes p(1:4)に分割配置。配列bはnodes p(5:16)に分割配置。
- [work mapping]
  - loop指示文またはarray指示文で、workとノードの対応を指示。
  - task指示構文の中でこれらを使えば、サブノードでの実行が自然に記述できる。

➔ 相対的なノード群へのマッピングが考えられている。

Coarrayの「イメージ」

- Coarrayの並列実行の単位
  全イメージが同時にプログラムを実行
- [data mapping]
  データは常に全イメージに同じ大きさで確保。明示的・自動的なallocationは、全イメージが同時に実施しなければならない。
- [work mapping]
  - イメージ番号を使って明示

```
if (this_image()==1) then
  (イメージ1が実行するコード)
end if
```

  - イメージ番号は常に全イメージの中の番号。

➔ SPMDだけが想定されていてタスク並列には機能不足

XMPにおけるCoarrayの「イメージ」

- 手続を実行しているノードに１対１に対応付ける。
  - サブセットノードで実行されるXMP手続では、（全ノードでなく）サブセットノードが全イメージ

```
!$xmp nodes p(8)
      real a(10,10)[*]

!$xmp task on p(1:4)
        call tantan    !(a)
!$xmp end task
!$xmp task on p(5:8)
        call tantan    !(b)
!$xmp end task
```

```
      subroutine tantan
!$xmp nodes q(*)=*
      real a(10,10)[*]

      if(this_image()==1)  &
        a(i,j)[2]=1.0
```

Nodes p	1	2	3	4	5	6	7	8
Nodes q	1	2	3	4	1	2	3	4
Image-id	1	2	3	4	1	2	3	4

- この仕様により、Coarrayで書かれた手続は、ほぼそのままXMPのタスク手続として利用できる。

XMP/C向けCoarray仕様の検討（XMP規格部会の活動）

- Cにおけるcodimension記法（V1.0）
  - Fortranのcodimension記法に似せた。ただし、':' で区切る必要がある。

宣言	int a[10]:[*]	int s:[*]
参照	a[i][j]:[k1] + s:[k2]	
定義	a[i][j]:[k] = 式	

  - 宣言は、具体的にどこに出現してよいか曖昧だった
- 宣言の仕様の明確化（11月予定V1.2.1）

仕様変更（見通し）

- Cの文法に、以下のルール（赤色の範囲）を追加する。
  - declarator（宣言子）を拡張したcodeclaratorを定義

    codeclarator : *declarator* ':' *[* '[' init-expr ']' ... *]* '[' '*' ']'
  - 1. 初期化付きdeclaratorが許される宣言文なら、codeclaratorを許す。

    init-declarator

    : *declarator [* '=' *initializer]*

    | *codeclarator [* '=' initializer *]*
  - 2. 関数のパラメタ（仮引数）の型宣言に、codeclaratorを許す。

    parameter-declaration

    : *declaration-specifiers declarator*

    | *declaration-specifiers codeclarator*

    | *declaration-specifiers abstract-declaratory*

    | *declaration-specifiers*
  - 3. 構造型の要素の型宣言に、codeclaratorを許す。

    struct-declarator

    : ':' *constant-expression*

    | *declarator* ':' *constant-expression*

    | *declarator*

    | *codeclarator*

まだ議論あり

宣言が書ける例/書けない例

- 書ける例
  - ファイル直下
    ```
    int a:[*], b:[*];
    struct { float re, im; } c1:[*];
    ```
  - ブロックの中
    ```
    if (...) {
        int a:[*] = 0;
    }
    ```
  - 関数パラメタ
    ```
    int foo(float a:[*], float b:[*], int n)
    { … }
    ```
  - 同、旧仕様
    ```
    int foo(a, b)
    int a:[*];
    int b:[*];
    { … }
    ```
- 議論が残っている
  - 構造型要素、type構文の中
    ```
    static struct {
        int n;
        double a:[*];
    } ss;
    ```
- 書けない例
  - forループのパラメタ
    ```
    for (int i = 0, s:[*] = 0; ...) {  ... }
    ```

実装の状況

- XMP/C Coarray機能
  - 現在のサポート範囲: V1.0のサブセット
  - 通信層: 汎用にGASNet、京のTofu向けにRDMA
    - One-sided通信の研究
- XMP/F Coarray機能
  - XMP/Cをベースに今年度から開発中
    - RuntimeはXMP/Cと共通
    - 12月プロトタイプ版完目標

まとめ

- XMPにCoarray機能を加える意味
  - 同時に使えること、相互に呼べること
  - 大規模並列に向けて
- 言語仕様のマージの結果
  - XMPのタスクに、Coarrayの世界をマッピングできる。
- XMP/CのCoarray仕様
  - 実装を進めつつ、仕様の詳細化を進めている。
- OMNI XMPの実装
  - 詳細はまた別の機会に