



XcalableMP規格部会 活動報告

PCクラスタコンソーシアムXMP規格部会
岩下 英俊(富士通)

内容

- ▶ プログラミング言語XcalableMP (XMP)を3分で紹介
- ▶ XMP規格部会のこれまでの活動
- ▶ この1年の議論と、V1.1仕様の紹介
- ▶ まとめと今後

XMPプログラミング はやわかり

① 動作確認した逐次プログラムを用意。ここから“並列化”が始まる。

```
real A(50,100)
```

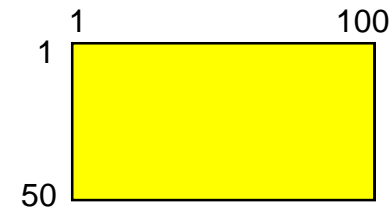
```
...
```

```
do j=1,100  
  do i=1,50
```

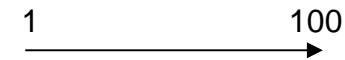
```
    ... = A(i,j) + A(i,j-1) + A(i,j+1)
```

```
  enddo  
enddo
```

配列A



ループ j



XMPプログラミング はやわかり

- ① 動作確認した逐次プログラムを用意。ここから“並列化”が始まる。
- ② 使用するノードをN次元配列で宣言。ノード数は実行時に決めてもよい。
- ③ テンプレートの形状と分散を決める。これがプログラム全体の並列化戦略になる。

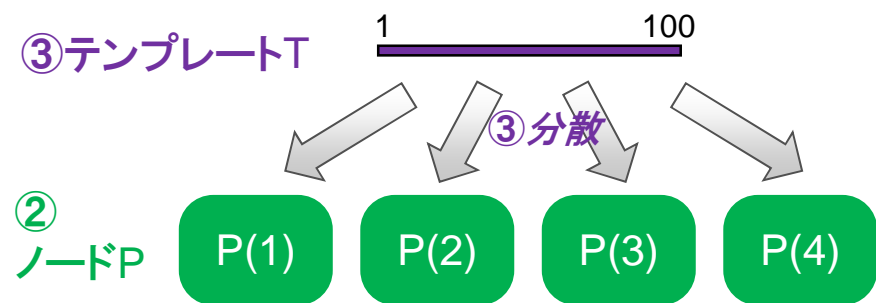
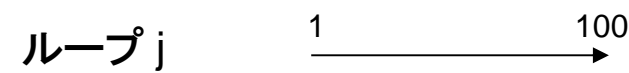
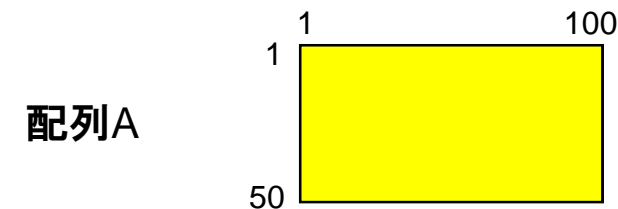
```

!$xmp nodes P(4)=*           ! ②
!$xmp template T(100)        ! ③
!$xmp distribute T(block) onto P ! ③
  real A(50,100)

  ...

  do j=1,100
    do i=1,50
      ... = A(i,j) + A(i,j-1) + A(i,j+1)

    enddo
  enddo
  
```



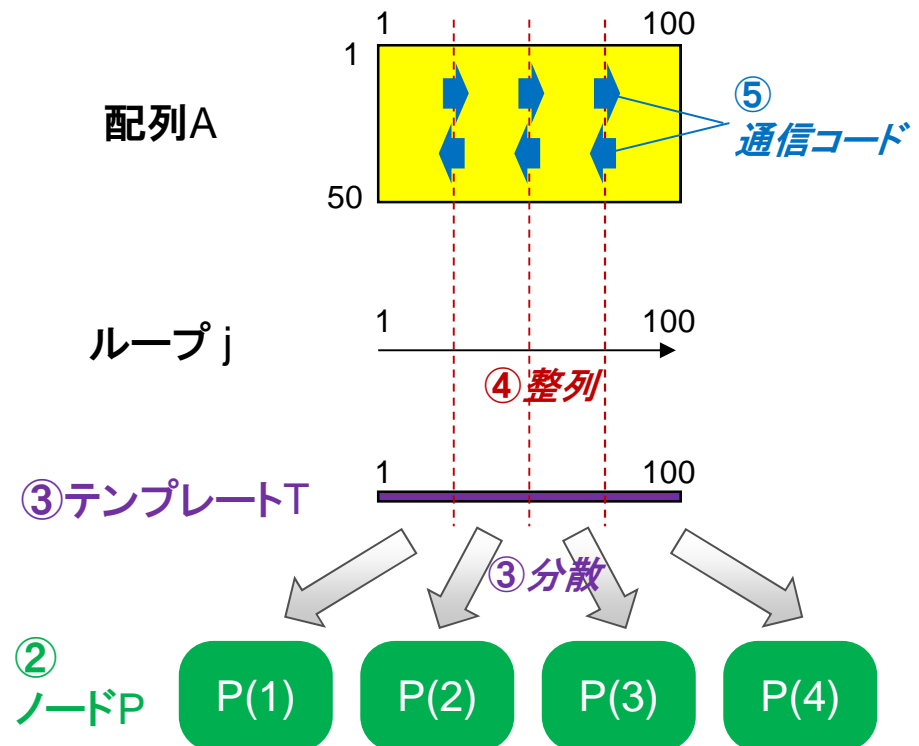
XMPプログラミング はやわかり

- ① 動作確認した逐次プログラムを用意。ここから“並列化”が始まる。
- ② 使用するノードをN次元配列で宣言。ノード数は実行時に決めてもよい。
- ③ テンプレートの形状と分散を決める。これがプログラム全体の並列化戦略になる。
- ④ 配列と計算ループをテンプレートに整列させる。通信が最小になるように気を付ける。
- ⑤ 通信を記述する。reflect(袖通信)、reduction、bcast、gmove(その他一般)など

```

!$xmp nodes P(4)=*           ! ②
!$xmp template T(100)        ! ③
!$xmp distribute T(block) onto P ! ③
    real A(50,100)
!$xmp align A(*,j) with T(j) ! ④
...
!$xmp reflect A              ! ⑤
!$xmp loop on T(j)           ! ④
    do j=1,100
        do i=1,50
            ... = A(i,j) + A(i,j-1) + A(i,j+1)
        enddo
    enddo

```



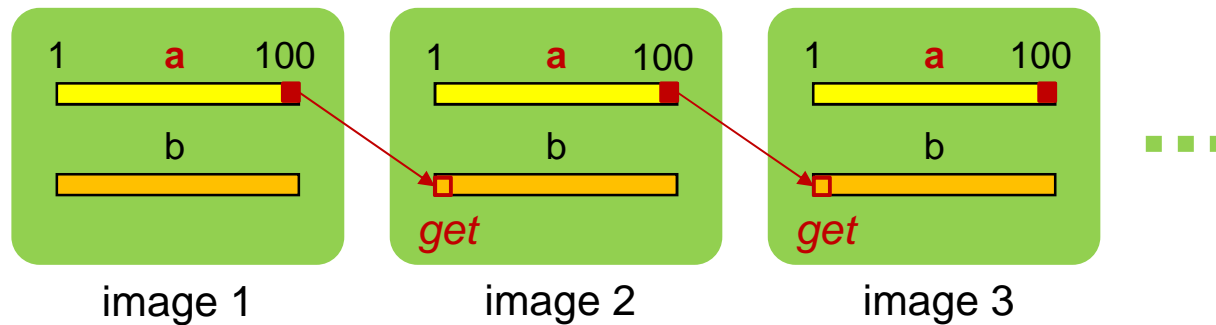
これまでの活動

- ▶ 2011年11月、V1.0仕様を公開
 - ▶ 文科省e-Scienceプロジェクトの一環として
- ▶ 2011年6月、XMP規格部会発足
WG開催は、2011年12月第1回 ~ 2012年12月第9回
 - ▶ PCCC会員以外の参加も自由。ただし投票権は正会員のみ
 - ▶ 開催通知案内をご希望の方は、PCCC事務局(アドレスは後述)まで
 - ▶ V1.0仕様の修正と拡張を議論
 - ▶ 影舞による進捗管理
- ▶ 2012年11月、V1.1仕様を公開
<http://www.xcalablemp.org/spec/xmp-spec-1.1.pdf>
 - ▶ V1.0からの主な拡張
 - ▶ Coarray機能対応 XMPによる並列化との混在の意味付け
 - ▶ プライマリノード配列 物理ノードのトポロジへの対応
 - ▶ マッピング問合せ関数 数値計算libやMPIとのインタフェース
 - ▶ 並列I/O仕様の詰め アトミック性などについての議論

Coarray機能対応 (1) Coarrayとは

- ▶ Fortran2008規格の一部。並列の概念
 - ▶ これまでも並列性の概念はあったが(配列式・配列代入、Forall文)、並列実行を陽に意識させる仕様はなかった。
- ▶ 書式は、“イメージ番号”(ノードのid)を変数に沿える。

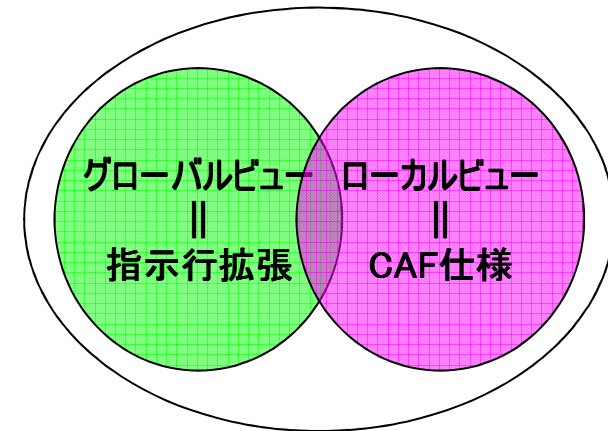
<code>real a(100)[*], b(100)</code>	! aはcoarrayと宣言したので、他イメージから
	! 参照・定義できる。
<code>me = this_image()</code>	! 私のイメージ番号の問合せ
<code>if (me>1) b(1) = a(100)[me-1]</code>	! 左隣のイメージのa(100)をgetしb(1)に代入
<code>sync all</code>	! 完了待ちと全イメージの同期



Coarray機能対応 (2) XMP V1.1仕様では

- ▶ Coarray機能は、XMPに必要
 - ▶ Fortranでは今後標準。無視できない。
 - ▶ ローカルビューであり、XMPの指示行拡張とは相補的に使えると期待
- ▶ 議論の末、解決した課題
 - ▶ Fortran2008仕様との互換性保障
 - ▶ Coarrayで書かれた関数・サブルーチンは、そのままXMPから呼出し可能
 - ▶ Coarrayの「イメージ」とXMPの「ノード」との対応付けを定義
 - ▶ XMPプログラム中に、coarray記法が混在可能
 - ▶ XMP/Cでも、同じ機能をサポート

XMP V1.1言語仕様



XMP/Cでのcoarray記述例

```
float a[100]:[*], b[100];  
  
me = xmp_node_num();  
  
if (me>0) b[0] = a[99]:[me-1];  
xmp_sync_all(&stat);
```


まとめ(1/2)

- ▶ XMPの特長 ……実用性、ユーザ資産を重視
 - ▶ Fortran, Cをベースとする他のプログラミング言語・モデルとの比較

XMPの特長	OpenMP (shared)	HPF	XPF	Coarray	MPI (library)
言語仕様拡張でなく 指示行(directive)拡張	YES	YES	YES	NO	ライブラリ
ローカルビューでなく グローバルビュー(PGAS, Data-parallel)	共有メモリ	YES	YES	NO	NO
性能透過性(performance-awareness) どんな通信が起こるかプログラムから分かる	YES	NO	YES	YES/NO	YES
言語間結合・混在(interoperability) 実装依存とせず規約されているか	NO	複雑・重い	不十分	NO	NO

- ▶ **指示行拡張**と**グローバルビュー**は、逐次プログラムからの段階的のため。
- ▶ **性能透過性**を重視。例えば、指示がない限り通信なしで処理することを保障。
- ▶ **言語結合**は、引数渡し、大域変数の扱い、コンテキストの継承などが課題。
言語混在は、用語・概念の対応付け、実行モデルの構築から。
単純に「同時に使えます」と言うだけでは、振舞いが実装依存になってしまう。

まとめ(2/2)

- ▶ 言語間結合・混在は、今も議論の中心
 - ▶ **Coarray**との混在実現は、V1.1(11月公開済)の大きな成果。
 - ▶ **数値計算ライブラリ**呼出しインタフェースも、V1.1で整備。
 - ▶ **スレッド並列**との混在は、V1.0仕様を一旦撤回して再議論中。
 - ▶ OpenMP指示文との混在時の課題を議論するため。

- ▶ 今後の議論
 - ▶ **GPU**向けプログラミングとの混在
 - ▶ 候補として、OpenACCとの混在を議論する。
 - ▶ **大規模タスク並列**: 複数の並列手続の同時呼出し
 - ▶ 個々の手続きは他言語/MPIも可。連成の一つのスタイルとして。

※ 協力者を求めます。特に、アプリ側から興味をお持ちの方
お問合せは、PCCC事務局 (pccc-sec <at> pccluster <dot> org) まで