3次元流体コードIMPACT-3DによるOmni XcalableMPの評価

村井 均 （理研）
坂上 仁志 （核融合研）

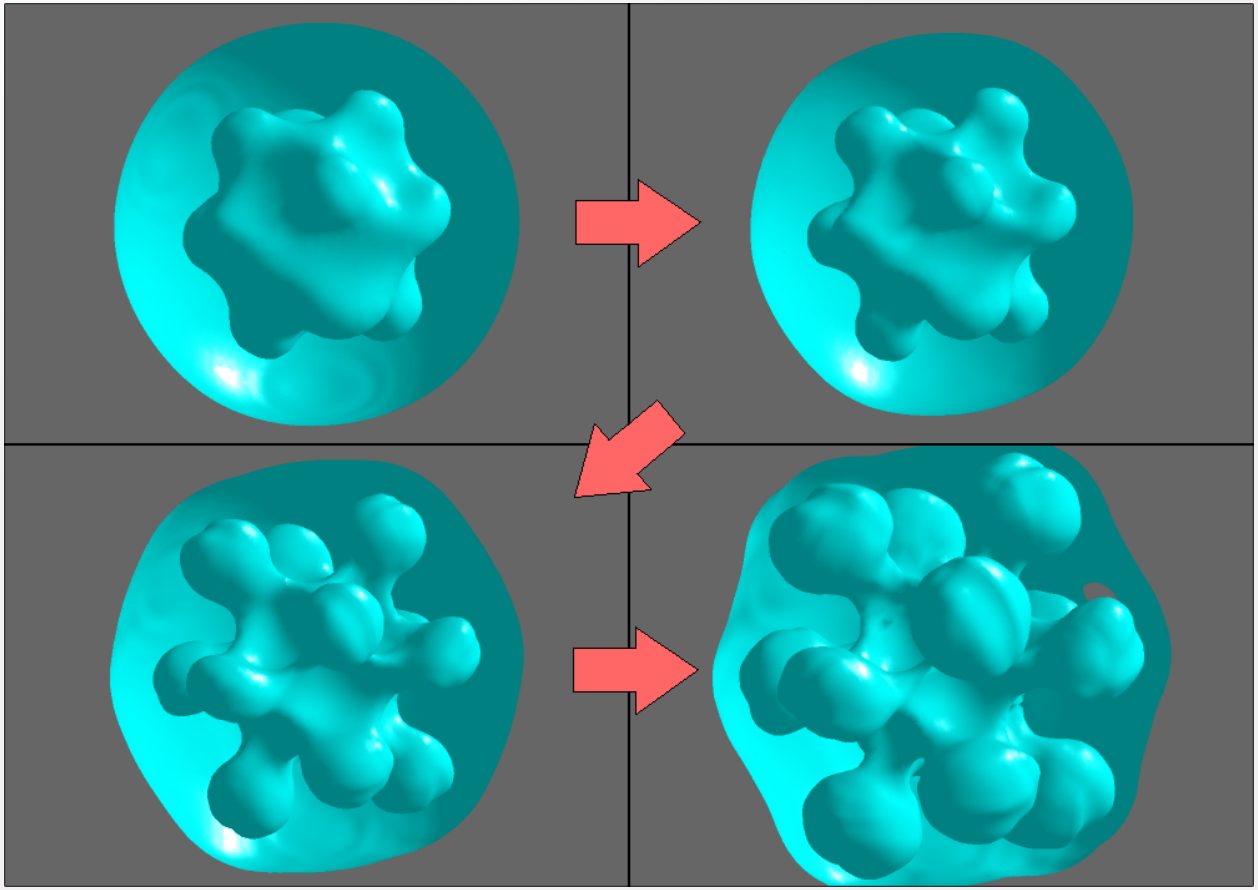
はじめに(目的)

- 3次元流体コードIMPACT-3DをXcalableMPで並列化することにより、XcalableMPの生産性を評価する。
- IMPACT-3Dの性能を京コンピュータで評価することにより、Omni XcalableMPの能力を評価する。

3次元流体コードIMPACT-3D (1)

- レーザー核融合の爆縮過程のシミュレーション
  - 球対称ターゲット
  - スタグネーション相(減速相)
  - レイリー・テイラー不安定性
- ３次元流体方程式(非粘性，圧縮性)
- カーテシアン座標系
- ５点差分による空間微分
- 陽的解法による時間積分
- 多次元の時間発展は分ステップ法

3次元流体コードIMPACT-3D (2)



3次元流体コードIMPACT-3D (3)

- 典型的な3次元ステンシルコード
- XMPの「グローバルビュー」並列化に非常に適する。
- かつては、High Performance Fortran (HPF)でも並列化され、地球シミュレータ(初代)の512ノードを使った評価の結果は、2002年のゴードン・ベル言語賞を受賞。
- つまり、IMPACT-3Dで高性能を達成できることは、XMPの普及に向けて必須。

IMPACT-3Dの並列化(1)

```
!$XMP NODES proc(32,32)
!$XMP TEMPLATE t(lx,ly,lz)
!$XMP DISTRIBUTE t(*,BLOCK,BLOCK) ONTO proc
      real*8 :: sr(lx,ly,lz), ...

!$XMP ALIGN (*,j,k) WITH t(*,j,k) :: sr, ...
!$XMP SHADOW (0,0:1,0:1) :: sr, ...

!$XMP REFLECT (sr, ...)

!$XMP LOOP (iy,iz) ON t(*,iy,iz)
      do iz = 1, lz
      do iy = 1, ly-1
      do ix = 1, lx
         ... = sr(ix,iy,iz)
      end do
      end do
      end do
```

ノードとテンプレートの宣言

データの分散

隣接通信

ループ並列化

IMPACT-3Dの並列化(2)

- 指示文のわずかな修正により、分散を変更することが可能。

1D分散

```
!$XMP NODES proc(512)

!$XMP DISTRIBUTE t(*,*,BLOCK) ONTO proc

!$XMP ALIGN (*,*,k) WITH t(*,*,k) :: sr, ...
!$XMP SHADOW (0,0,0:1) :: sr, ...

!$XMP LOOP (iz) ON t(*,*,iz)
```

2D分散

```
!$XMP NODES proc(32,32)

!$XMP DISTRIBUTE t(*,BLOCK,BLOCK) ONTO proc

!$XMP ALIGN (*,j,k) WITH t(*,i,j) :: sr, ...
!$XMP SHADOW (0,0:1,0:1) :: sr, ...

!$XMP LOOP (iy,iz) ON t(*,iy,iz)
```

3D分散

```
!$XMP NODES proc(8,8,16)

!$XMP DISTRIBUTE t(BLOCK,BLOCK,BLOCK) ONTO proc

!$XMP ALIGN (i,j,k) WITH t(i,j,k) :: sr, ...
!$XMP SHADOW (0:1,0:1,0:1) :: sr, ...

!$XMP LOOP (ix,iy,iz) ON t(ix,iy,iz)
```

IMPACT-3Dの並列化(3)

- モジュールの利用
  - Omni XMPの現在の実装では、モジュールで分散配列を宣言した方が効率がよい（∵ 分散された仮配列は一次元化されるため、バックエンドの最適化に影響する）。

```
module mod
real a(100,100)
!$xmp align a(i,j) with t(i,j)
```

```
subroutine sub
use mod
```

```
program main
real a(100,100)
!$xmp align a(i,j) with t(i,j)
call sub(a)
```

```
subroutine sub(a)
real a(100,100)
!$xmp align a(i,j) with t(i,j)
```

IMPACT-3Dの並列化(4)

- ステンシル通信
  - reflect指示文のwidth節を用いて、割り当てたステンシル（シャドウ）の一部のみの更新を明示。

```
!$xmp shadow a(1:1, 1:1, 1:1)

!$xmp reflect (a) width (0, 0, 1)
...
!$xmp reflect (a) width (0, 1:0, 0)
```

  - 処理系の内部では、通信スケジュールの再利用、MPIの永続的通信機能の利用といった最適化が適用されている。

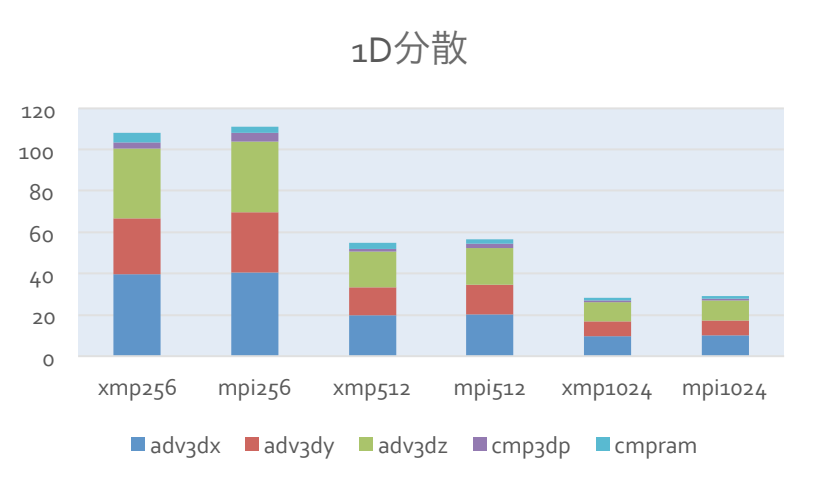
評価環境

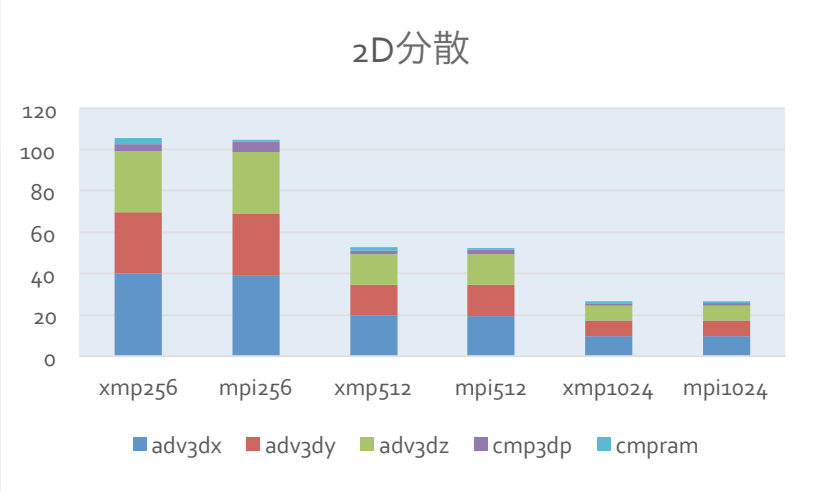
- 京コンピュータの256, 512, 1024ノード
- フラット並列
  - 各ノードに8プロセスを配置
- 言語環境 K-1.2.0-14
- Omni XcalableMP R1410
- ノード1(ランク0)の計測結果を採取

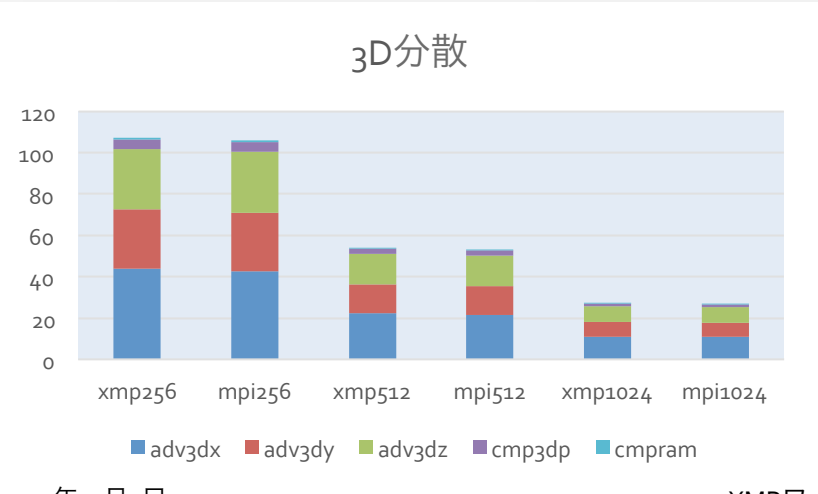
3D分散版のライン数

	XMP	MPI
ライン数	1373	1675

評価結果(1) 全体

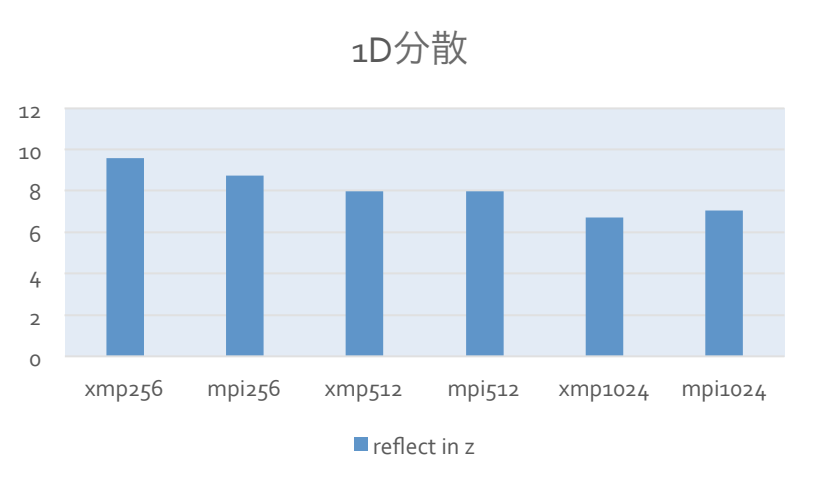


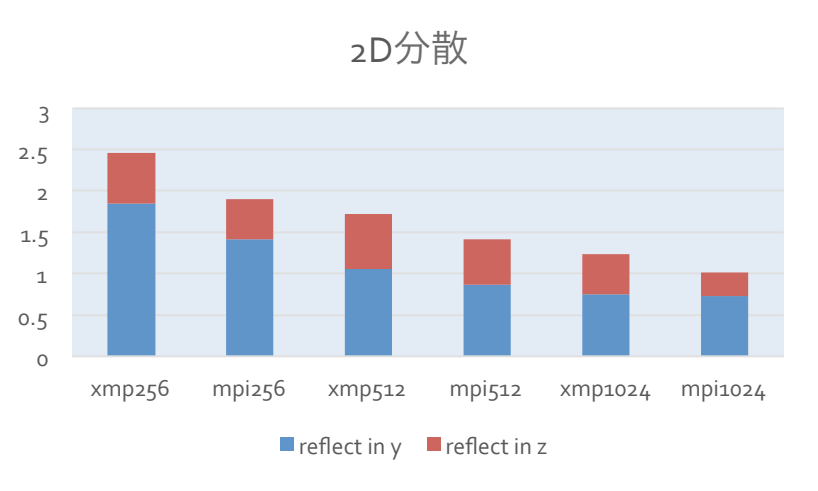


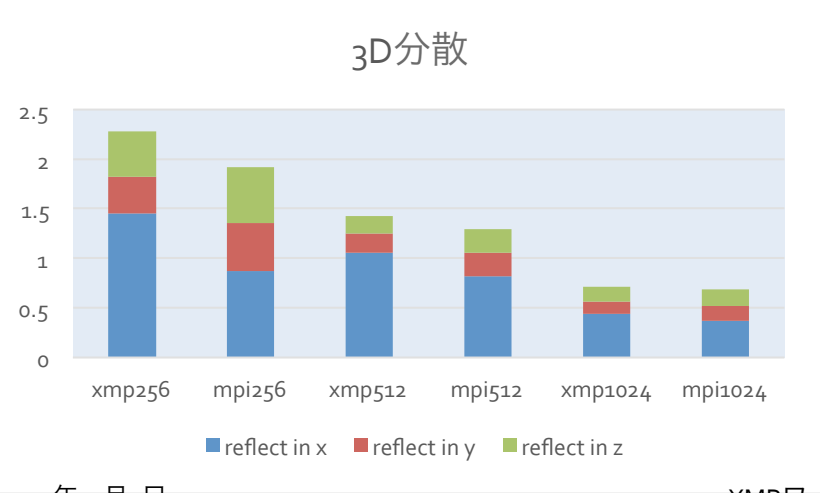


- XMP版はMPI版とほぼ同等の性能
- わずかながら、2D分散が最も速い。

評価結果(1) ステンシル通信







- Omni XMPランタイムによるステンシル通信の性能は、MPI版のそれに迫る。
- 配列の一次元目(不連続)の通信に、やや課題が残る。

疑問？

- 以下の最適化を適用しないと、MPI版が異常に遅い？

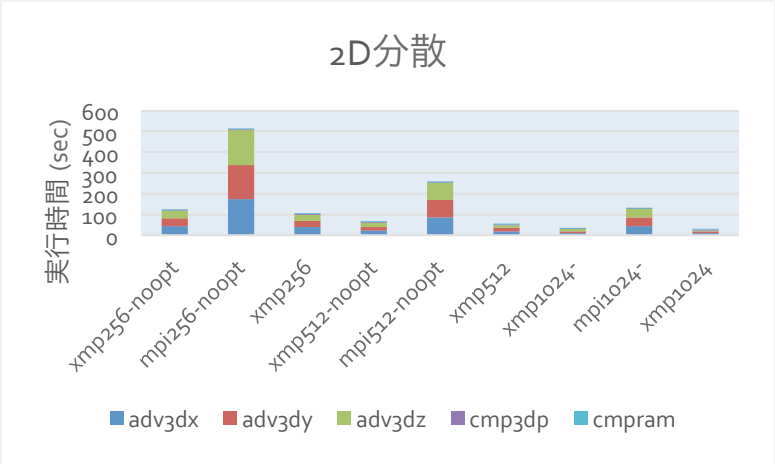
```
real a(n,n,n), b(n,n,n), c(n,n,n)
```

↓ 配列を融合

```
real x(3, n,n,n)
```

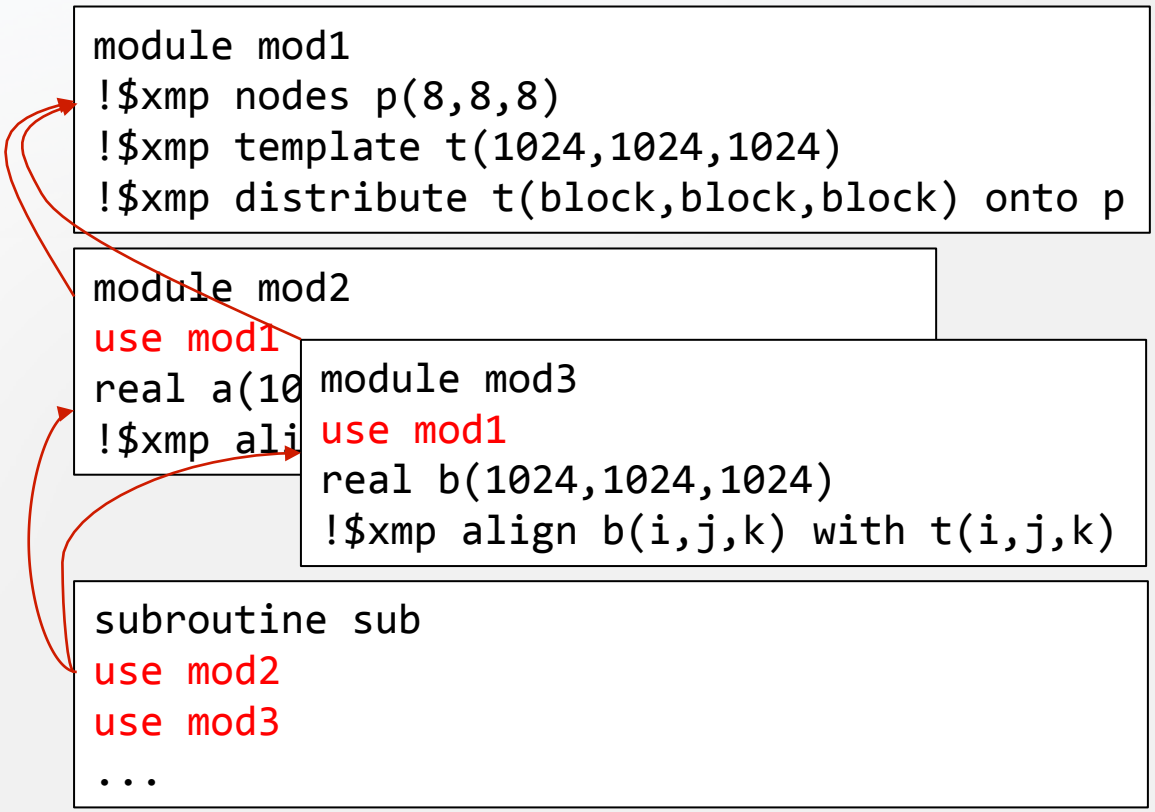
- 一回の通信で、複数の配列を対象にできる。

※ これまでの結果は、XMP、MPIとも、上記の最適化を適用したもの。



明らかになった制限事項

- XMP指示文を含むモジュールの階層的な参照



XMP並列実行環境（ノード配列およびテンプレート）を宣言

分散配列を宣言

各サブルーチン

おわりに

- 3次元流体コードIMPACT-3DをOmni XMPで並列化し、京コンピュータ上で評価。

  ➡ 手で作成したMPI版とほぼ同等の性能

- IMPACT-3Dのような規則的なコードの並列化に関し、XMPおよびOmni XMPが有用であることを示せた。

今後の課題

- ハイブリッド並列(スレッド並列)
  - 並列パック/アンパックに基づくステンシル通信により高速化の可能性あり。
  - フラット並列では、MPI派生データ型に基づくステンシル通信が用いられる。
- 非同期通信（通信と計算のオーバラップ）
  - `reflect`指示文の`async`節と`wait_async`指示文
- RDMAに基づくステンシル通信
  - 京において実験的に実装済み。