

マイグレーションを支援する 分散集合オブジェクト

30388 高橋 慧
近山・田浦研究室

powered by [Phoenix Library](#)

PCグリッド環境での計算

特徴

- ▶ インターネット上の「普通の」計算機で並列処理
- ▶ 専用並列計算機に匹敵する潜在性能

問題点

- ▶ 所有者の判断でプロセッサ数が増減
- ▶ メモリに比べて通信路は非常に遅い
- ▶ 記述が難しい(次ページ参照)

並列プログラムの記述

- ▶ 大きなデータ(配列など)を操作
- ▶ 時々他のプロセッサの要素にアクセス
自プロセッサ内の処理をカプセル化できる、
オブジェクト指向が適している

現状

- ▶ 既存の並列オブジェクト指向は、無駄が多く遅い
- ▶ このため・多くは手続き的に通信を記述
- ▶ 逐次プログラムに比べて、記述が難しい

グリッド環境での計算

求められる記述モデル

- ▶ 動的プロセッサ数の増減に対応
- ▶ 無駄な通信が発生しない
- ▶ 記述が容易(例えばオブジェクト指向)

これらを備えた記述モデル
「分散集合オブジェクト」を提案

背景・目的

既存のモデル

- ▶メッセージパッシング
- ▶分散オブジェクト

分散集合オブジェクトの提案

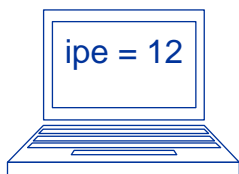
現状・今後の計画

本報告の構成

[5]

メッセージパッシング

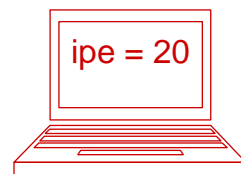
- ▶相手プロセッサを番号で指定して通信
- [+]無駄の無いプログラムが書ける
- [-]手続き的・記述が難しい
- [-]プロセッサ数の増減に対応が難しい



```
/* ipe = 12 */  
double d = 3.14;  
send(20, &d);
```



```
to: 20  
data: 3.14
```

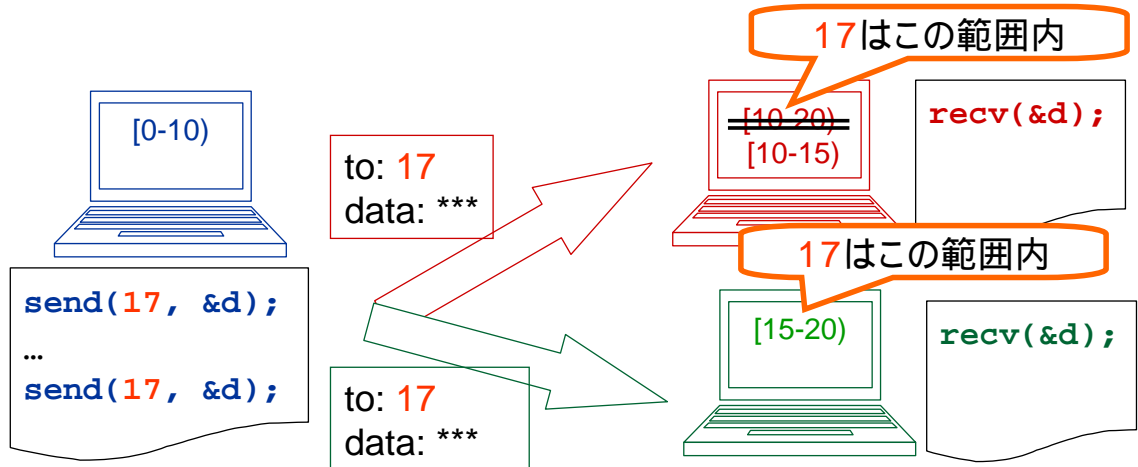


```
/* ipe = 20 */  
double d;  
recv(&d);
```

▶ Phoenixモデルによる改良

- 各プロセッサは番号の集合(例えば0-10)を持つ
- 実行中に番号の配分を変更可能

[+]プロセッサの増減に対応 (他の得失は同様)

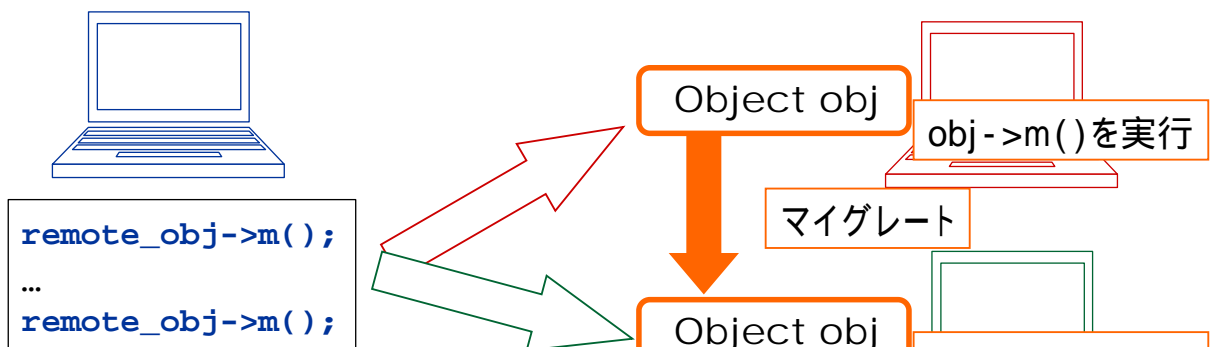


既存モデル

[7]

分散オブジェクトモデル

- ▶オブジェクトと、それを用いるプログラムを記述
 - ▶リモートのメソッドを呼び出せる(RMI)
 - ▶オブジェクトを他プロセッサに移送(マイグレート)
- [+]記述がオブジェクト指向で楽



[-] プロセッサ増減 性能又は記述性が低下

▶ 例えば、大きな配列を操作する例

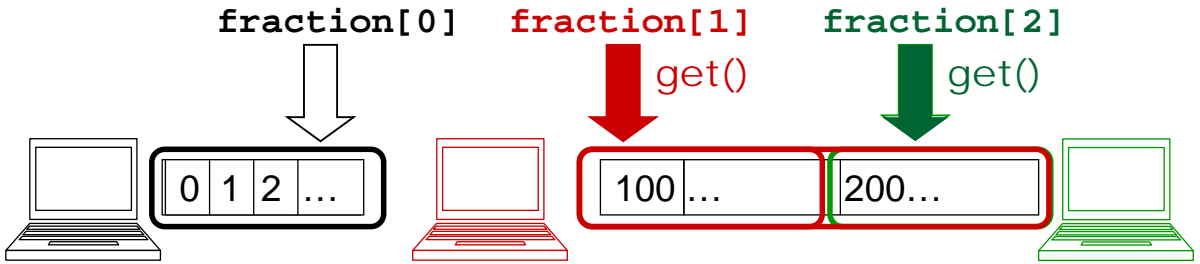
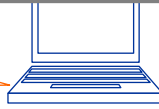
□ (1) 断片オブジェクトの配列

断片の配分が実行中に変化

プログラマが要素配分を把握する必要がある

215番の要素が欲しい

```
fraction[1]->get(215-offset);  
fraction[2]->get(215-offset);
```



既存モデル

[9]

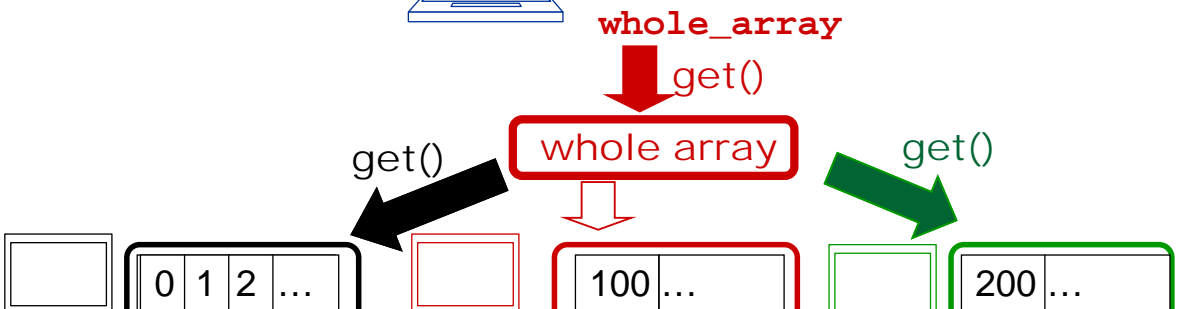
分散オブジェクトモデル

[-] プロセッサ増減 性能又は記述性が低下

(2) 断片オブジェクト + 全体オブジェクト

**メッセージが全体オブジェクトに集中
性能上ボトルネック**

```
whole_array->get(0);  
whole_array->get(201);
```



メッセージパッシング

- ▶ 性能が良い
- ▶ プロセッサ増減に対応 (Phoenixモデル)
- ▶ 記述は手続き的で難しい

分散オブジェクト

- ▶ プロセッサ増減に対応する場合、性能と記述性のトレードオフ
 - ▣ 性能を重視すると、記述が難しい
 - ▣ 記述性を重視すると、性能が悪い

これを改善するモデルを提案

既存モデル

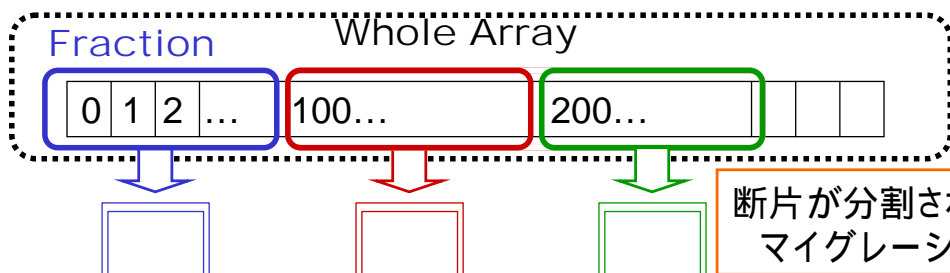
[11]

分散集合オブジェクトの提案

基本は分散オブジェクト

オブジェクトの分割保持

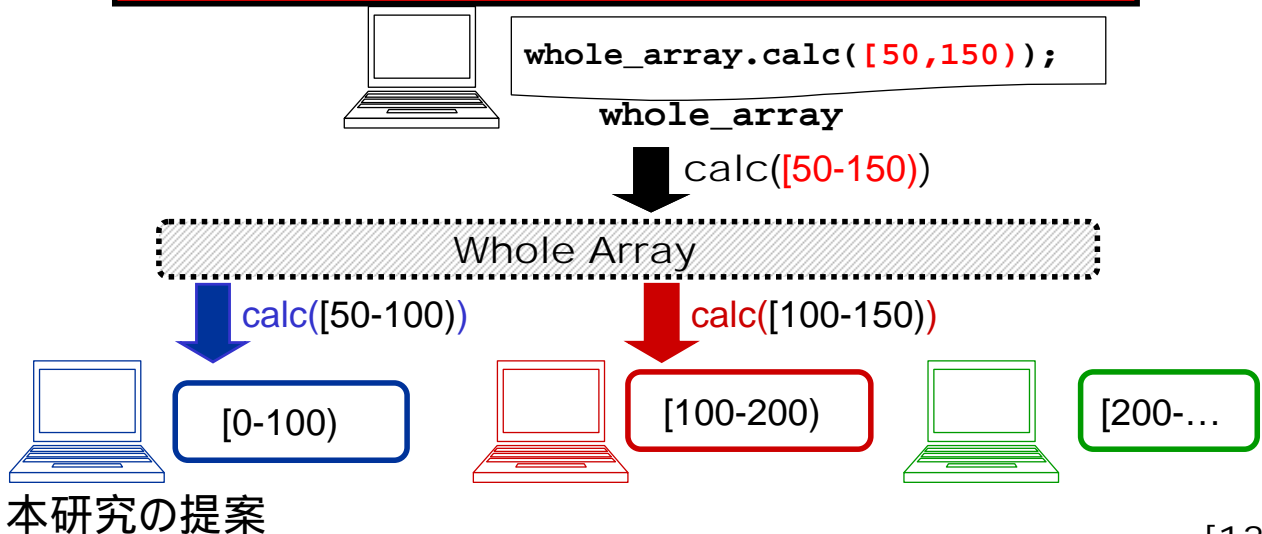
- ▶ 仮想的な「全体オブジェクト」がある
- ▶ 各プロセッサに断片オブジェクトが存在
- ▶ 断片は分割・併合・マイグレートできる



▶ メソッド呼び出し

- ▣ 外側からは、「全体オブジェクト」だけが見える
- ▣ インデックス集合を指定してメソッドを呼び出し

プログラマが要素配分を把握する必要がない



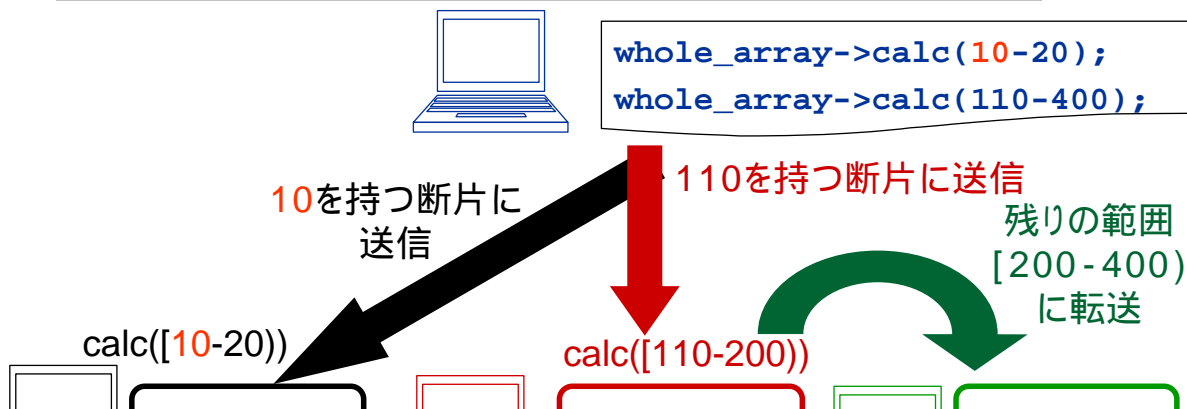
[13]

分散集合オブジェクトの提案

▶ メソッド呼び出しの実際の動作

- ▣ 呼び出し範囲の先頭要素にメッセージを送信
- ▣ 各断片がメッセージを中継 (将来的にはツリーを利用)

メッセージが特定のプロセッサに集中しない



■ プロセッサ数の増減に対応

- ▶ 断片のマイグレーション

■ 記述しやすい

- ▶ オブジェクト指向
- ▶ 「全体オブジェクト」というビューの提供

■ メッセージパッシングに劣らない速度

- ▶ メソッド呼び出しのボトルネックがない

進捗と今後の計画

■ 分散配列オブジェクトの作成中

- ▶ Phoenixライブラリを使用
- ▶ マイグレートに対応
- ▶ FFTやSORに使用

■ ライブラリの実装

- ▶ メッセージの自動生成

■ これを用いたアプリケーションの記述