

マルチPSI上への KL1分散処理系の実装

ICOT
第4研究室

中島 克人

マルチPSIの開発目的

- (1) **KL1 並列処理系の研究開発および評価**
 - － 要素プロセッサ (PE) 内での高速処理方式
 - － 大規模並列マシンに通用するPE 間処理方式
 - － 実用的なシステムとしての機能の実装
- (2) 並列ソフトウェアの研究開発

分散処理系設計のキーポイント

非共有メモリ型マルチプロセッサ

=PE 内/PE間のデータアクセス所要時間の比が大きい
(マルチPSIでは二桁程度)

⇒ PE間の通信量を抑えることが重要

1. PE間データ管理

- 単独PEでのゴミの回収(GC)を実現
- PE間にまたがるゴミの効率的な回収
- コピーの重複を防止

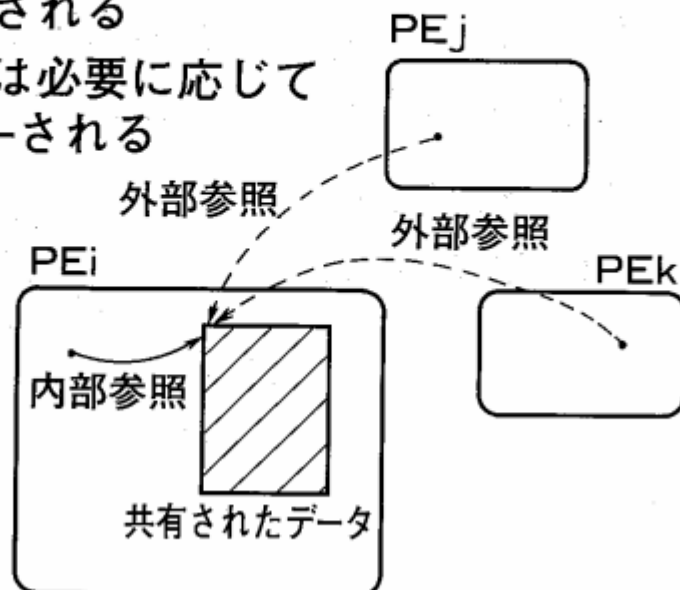
2. PE間ゴール実行管理

- 効率の良い実行制御
- 効率の良い終了判定

1. PE間データ管理

PE間データ参照

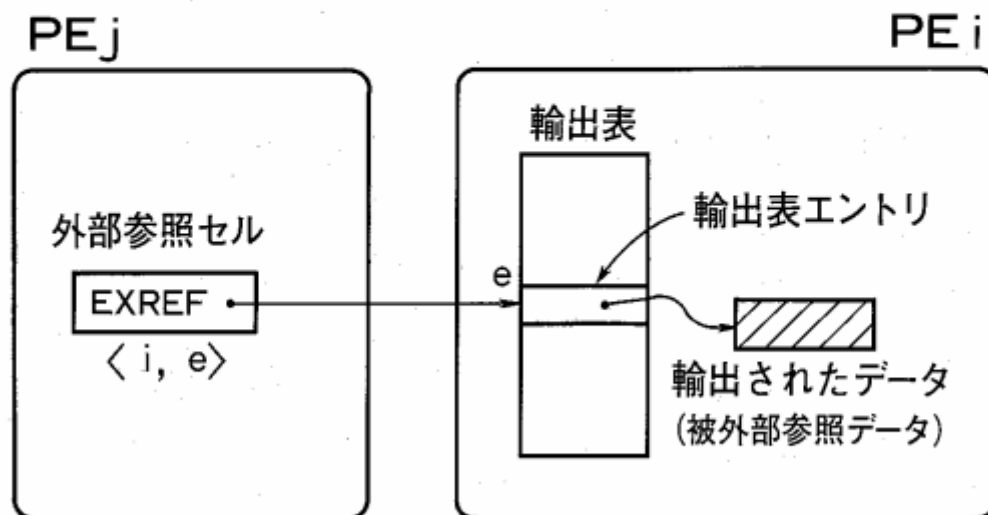
- 未定義変数と構造体データは「PE間外部参照ポインタ」を用いて共有される
- 構造体データは必要に応じてPE間でコピーされる



単独PEでのゴミの回収(GC)を実現

⇒ 輸出表により外部からの参照を管理

- 外部参照ID = $\langle \text{PE番号}, \text{PE内ID} \rangle$



PE間にまたがるゴミの効率的な回収

- 一括型……×局所性なし，×処理が止まる
- 参照カウントを用いた即時型
……○局所性あり
参照カウントの増減により通信量増大？
⇒○WEC方式によりカウント操作を
局所的に行なう

Weighted Export Counting(WEC)方式

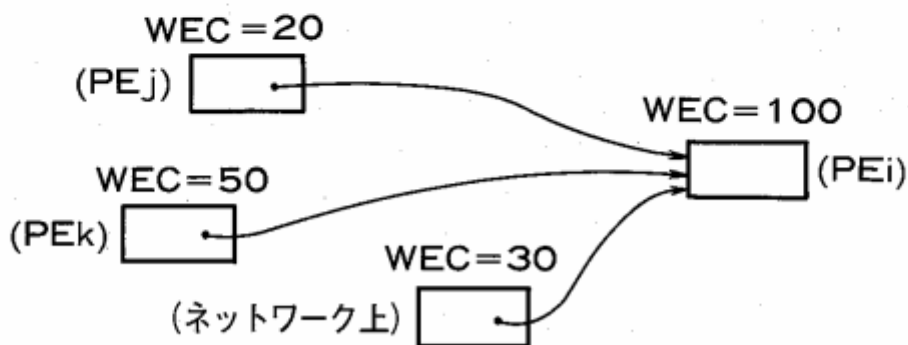
- 「重みつき参照カウント方式」の応用
= 参照ポインタ側にも参照カウントを持たせる
- 輸出側に無断で輸入側のポインタをコピーできる

Weighted Export Counting(WEC)方式(続き)

- INVARIANT :

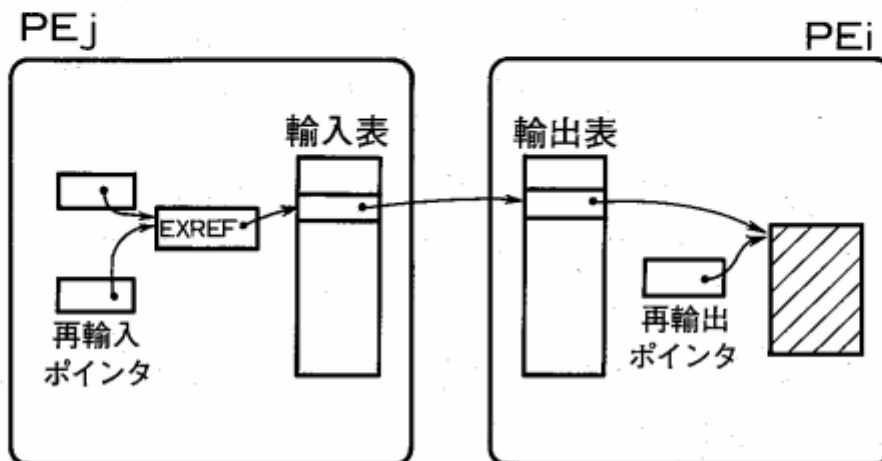
輸出側のWEC = 輸入側のWECの合計
+ ネットワーク中の外部参照ポインタ
が運ぶWECの合計

- 輸入側では輸入表によりWECを管理する



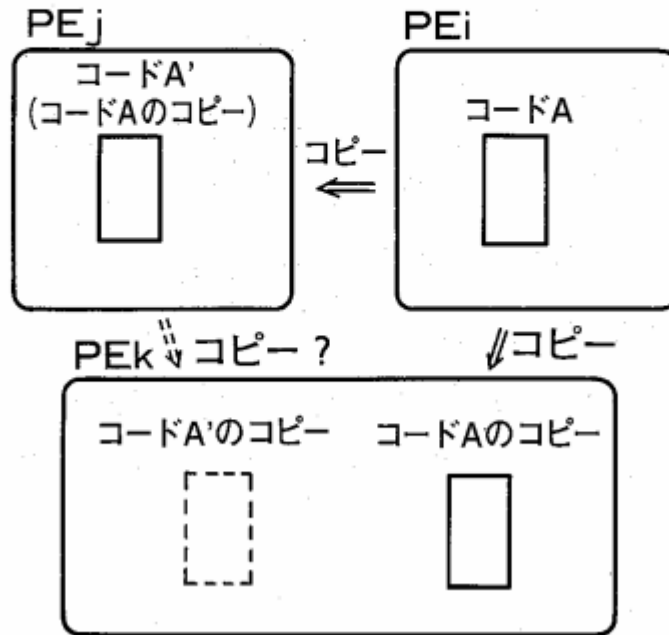
コピーの重複の防止(その1)

- 特定のPEが、他のあるPE内の同じ構造データを何回もアクセスすることの防止
⇒ ハッシュ機構を用いて、輸出入済みのデータを検出



コピーの重複の防止(その2)

- プログラム・コードのように色々な経路で同じ構造データが輸入されることの防止
⇒ システムに一意的な**構造体ID**を付与して再コピーを防止



MRBを用いた簡略輸出入

PE内で唯一の参照ポインタ(MRB=○)を輸出する場合は、WECの管理や再輸出入のためのハッシュ表などを省略した輸出入表を用いる……**白輸出入表**(もともとのものを**黒輸出入表**と称する)

2. PE間ゴール実行管理

効率の良い実行制御

⇒ 荘園・里親を用いたゴール管理の最適化

- 荘園 (= タスク: KL1 言語定義の実行単位) の情報を各PEに分散管理 (里親)
- ある荘園に属するゴールを保持するPEにはその荘園に対する里親がただ1つ生成される
- 里親では **KL1 言語定義の資源, ゴール数**, および, **荘園の状態** (run/stop/abort) をキャッシュ

効率の良い終了判定

⇒ Weighted Throw Counting (WTC) 方式を用いた終了判定

- 荘園内のゴールの終了判定がレーシングの心配なしに行なえる
- WECと同様, 「重みつき参照カウント方式」の応用
- WTCは送出されたゴールによって運ばれ, 到着先PEの里親に積算される
- INVARIANT :
 荘園のWTC = 里親のWTCの合計
 + ネットワーク中のゴールが運ぶ
 WTCの合計

まとめ

● PE間データ管理

- **輸出表**を用いて単独PEでのゴミの回収を実現
- **WEC方式**でPE間にまたがるゴミを効率的に回収
- **輸出入ハッシュ表**や**構造体ID**でコピーの重複を防止

● PE間ゴール実行管理

- **荘園・里親方式**で効率の良い実行制御を実現
- **WTC方式**で効率の良い終了判定を実現

KL1分散処理系はすべてファームウェアで実装
(約12Kステップ)

今後の課題

- 各種方式の評価と改良
- 大域GC
- 実装面からのKL1言語仕様へのフィードバック
- 分散処理系のPIMへの実装