並列推論マシンＰＩＭ

－クラスタ内実験処理系－

佐藤正俊　清水肇

（財）新世代コンピュータ技術開発機構

１．はじめに

ＩＣＯＴでは，ＧＨＣ［上田］をベースとした並列論理型言語（ＫＬ１）を実行する並列推論マシン：ＰＩＭを開発中である［後藤］．ＰＩＭは，要素プロセッサ（ＰＥ）を１００台規模で接続するマシンであり，その構成はクラスタにより階層化する．クラスタは１０台程度のＰＥを密結合した構成をとり，クラスタ間は疎結合構成をとる．

ＰＩＭのＫＬ１処理方式は，ＰＩＭの構成に従い密結合向き処理方式と疎結合向き処理方式とに分け検討を進めている．以下では密結合向きＫＬ１処理方式の検討において開発されたクラスタ内実験処理系についてその概要と処理方式について報告する．

２．クラスタ内実験処理系の概要

２．１目的

クラスタ内実験処理系の開発の目的は，①クラスタ内処理方式の処理アルゴリズムの検証，②処理方式案の比較検討のためのベースの作成，③処理系の拡張の調査，④ＰＩＭの設計のためのデータを収集する等にある．特に③においてはクラスタ内実験処理系の開発と並行して並列キャッシュの設計を行っているが，この設計のための基礎データ（ＫＬ１実行におけるメモリアクセス情報）を収集する．

我々は，これらの目的を達成するためにクラスタ内実験処理系を開発した．この処理系は逐次マシン（ＶＡＸ／１１）上でマルチプロセスサポートツールを用いて作成し，１プロセスでクラスタ内の１ＰＥをエミュレートしている．また処理系よりのメモリアクセス情報を基に現在設計中の並列キャッシュを評価できるように，並列キャッシュ・シミュレータ［松本］を開発した．

２．２特徴

処理系の特徴を以下に列挙する．

①共有メモリ構成向きのＫＬ１処理モデル（以下局所実行モデルと呼ぶ）［佐藤］の検討に従って処理系を実現した．

クラスタ内のＫＬ１の実行は各ＰＥの局所性を活かした実行が必要であり，局所実行モデルの検討は，上記並列キャッシュ・シミュレータからの結果をフィードバックしながら行った．

②負荷分散方法の比較機能を盛り込んでいる．

分配方法は局所実行モデルの分配方法（以下局所分配と呼ぶ）を評価するために，比較対象としてゴールの分配をランダムに行なう分配方法（以下ランダム分配と呼ぶ）を盛り込んだ．

③各種統計情報を収集できる．

統計情報は，処理時間，領域等の資源の使用状況，各処理の回数等である．ここで領域とは，ゴール，メタコール，サスペンド，ヒープ，コードである．

④メモリアクセス情報は並列キャッシュシミュレータの入力データとなり，並列キャッシュの評価が行なえる．

メモリアクセス情報は，各領域へのアクセス情報を収集している．ここでアクセス情報とは，各メモリ領域に対するRead／Write，ロック操作等の情報である．

２．３システム規模と処理速度

現在の処理系は，ソースプログラム（言語Ｃを使用）で約３０００行のプログラム規模である．またクラスタ内ＰＥのシミュレート台数は１～１６台である．

ＫＬ１実行のみの処理速度は，ＶＡＸ／１１－７８５において８ｑｕｅｅｎｓプログラムを実行する場合，ＰＥ台数１台のシミュレーションで約１．５ＫＲＰＳであり，ＰＥ台数のオーバヘッドはそれほど多く無く，ＰＥ台数４台のシミュレーションで約１．４ＫＲＰＳである．またデータ収集を行なうオーバヘッド（主にファイルへの出力）は２０倍ある．

２．４実験環境

処理系は，ＰＩＭの機械語であるＫＬ１－Ｂコードをオブジェクトコードとして ＫＬ１プログラムを実行する．このＫＬ１－Ｂを生成するために，我々はＶＡＸ上に言語系を用意している．言語系は，他のシステムと共有できる抽象ＫＬ１－Ｂを生成するコンパイラ［木村］と，この抽象ＫＬ１－Ｂからターゲット用のＫＬ１－Ｂコードを生成するアセンブラ／リンカより構成される．

３．処理方式

処理方式はＫＬ１の逐次処理方式をベースに局所実行モデルに拡張したものである．その主な拡張点は共有領域の管理方法，ユニフィケーションにおけるロック操作［清水］，ＰＥ間の通信方式及び負荷分散方式である．

３．１逐次処理方式

逐次処理方式でのゴールリダクションは逐次版エミュレータ［久門］と同様に行われる．ただし本処理系ではＰＩＭのシミュレーションを行なうために，処理系で扱うＫＬ１－Ｂコード及びデータをＰＩＭのワード構成と同じデータ部３２ビット，タグ部８ビットとしている．

３．２局所実行モデルへの拡張点

（１）局所的なメモリ配置

局所実行モデルは，実行におけるメモリ参照の局所性を高める工夫として領域の共有方法に局所的なメモリ配置を導入している．そのメモリ配置とは，レディキューやゴールレコード等を共有せずに使用する領域（以下ローカル領域）に置き，ヒープやサスペンドレコード，メタコールレコードを共有して使用する領域（以下グローバル領域）に置くことである．

Parallel Inference Machine: PIM
－KL1 Experimental Implementation for PIM Cluster－
Masatoshi SATO, Hajime SHIMIZU
ICOT　　ICOT

またグローバル領域については，ローカル領域と同様に各ＰＥに分割して管理している（領域の生成／消滅はＰＥ毎，実際の使用は領域へのポインタを持つＰＥ間で共有される）．これにより領域の生成／消滅時のロック操作を省略でき，同時に負荷分散においてゴールの分配を抑えればグローバル領域に対しても局所的に領域を使用できる．

（２）ＰＥ間通信方式

ＰＥ間通信方式はローカル領域間の通信（メッセージ通信）とグローバル領域間の通信（共有メモリ通信）がある．メッセージ通信はＰＥ間の同期を取るための割込み機能，通信用バッファ及びメッセージ・ハンドラにより実現している．ここで割込み機能は共有メモリ上のフラグ（割込みフラグ）と１ゴールリダクション毎のフラグチェックでシミュレーションしている．また通信用バッファをグローバル領域にとり，各ＰＥは割込みによりメッセージ・ハンドラを起動する．

（３）負荷分散方式

［佐藤］で提案した要求駆動に基づく負荷分散方式（局所分配）と，その比較のためのランダム分配の２方式をサポートしている．

また負荷分散方式では分配の対象または候補を細かく指定し，方式の検討を行なうためにＫＬ１にプラグマを導入している．またＫＬ１－Ｂコードではプラグマ付ゴールとプラグマ無しゴールは別のenqueue 命令で表わす．

例えば以下のプログラムを実行する場合を考える．

```
p :- . . . | p, q, @r, . . .
```

ここで@はプラグマを表わし，能動部の最初に現れるゴールは再帰的に実行されるものとする．このＫＬ１－Ｂコードは以下のようになる．

```
p:    . . .
      enqueue( q ).
      . . .
      p-enqueue( r ).
      . . .
      execute( p ).
```

①局所分配方式

局所分配用のp-enqueue命令は，以下のアルゴリズムで実現される．

```
p-enqueue(goal):
      if (request_flag)
        then send goal to requested PE.
        else enqueue goal to self-PE queue.
```

この局所分配を実現するために，request_flagを共有メモリ上に置いた．このrequest_flagはidleＰＥからの要求の通信コストを抑えるために導入した．つまりこのrequest_flagをオンにすることで相手ＰＥのリダクションを中断させずに要求を伝えられる．また要求有りの時のプラグマ付ゴールの送出方法は，ゴールレコードのポインタを持つメッセージを組み立てメッセージ通信により行なう．メッセージは受信側のＰＥのメッセージ・ハンドラにより受け取られ，ゴールレコードは受信側のＰＥのレディキューに入れられる．

②ランダム分配方式

ランダム分配ではp-enqueue 命令は以下のアルゴリズムで実現される．

```
p-enqueue(goal):
      send goal to another PE at random.
```

ここでゴールの送出は，メッセージ通信により行われる．

③分配方式の比較

これらの分配について処理系と並列キャッシュ・シミュレータにより評価した結果を以下に示す．この結果はＰＥ台数４の時のもので，シミュレーションは１ゴールリダクションを１単位時間として行っている．

表１　局所分配とランダム分配の比較

	8queens		BUP	
	ランダム	局所分配	ランダム	局所分配
実行時間比	1.00	0.95	1.00	0.85
idle率	7.3 %	2.5 %	11.8 %	6.6 %
分配率	14.2 %	0.9 %	5.2 %	2.4 %
バスサイクル数比	1.00	0.42	1.00	0.49

表１より局所分配はidle率（全実行時間におけるゴール分配を待っている時間）を抑えながら，分配率（全リダクション数におけるゴール分配数）を抑えたことがわかる．また分配のオーバヘッド及び局所性の比較のためにバスサイクル数の比を示している．これによると局所分配はバスサイクル数を半分に抑え，ＰＥ間にまたがる共有メモリアクセスによるＰＥ間通信のオーバヘッドを削減していることがわかる．

４．　おわりに

密結合向きＫＬ１処理方式の検討において開発されたクラスタ内実験処理系についてその概要と処理方式を報告した．現在，この処理系を密結合マルチプロセッサシステム（Ｂａｌａｎｃｅ 21000）上に移植している．これによりロック操作を含めた処理アルゴリズムの検証及び実際の密結合マルチプロセッサシステムでの処理系の振舞を調査する予定である．

〈参考文献〉

［上田］K.Ueda, "GUARDED HORN CLAUSES",ICOT TR-103.

［後藤］後藤 他，" 並列推論マシンＰＩＭ"，情報処理 第３３回全国大会 ３Ｂ－５～７

［佐藤］佐藤 他，"共有メモリ構成クラスタ向きＫＬ１処理方式"，日本ソフトウェア科学会第３回大会，Ｄ－１－１

［木村］木村 他，" －ＫＬ１の抽象命令仕様とコンパイラー"，本大会予稿２Ｐ－１

［久門］久門 他，" －ＫＬ１の逐次版エミュレーター"，本大会予稿２Ｐ－２

［清水］清水 他，" －共有メモリ構造クラスタにおけるユニフィケーションー"，本大会予稿２Ｐ－３

［松本］松本 他，" －並列キャッシュとロック機構－"，本大会予稿２Ｐ－６