並列推論マシンPIM

−中期PIMの処理方式について−

佐藤正俊，清水肇，近山隆
(財)新世代コンピュータ技術開発機構

1．はじめに

中期ＰＩＭ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である。またこの構成はクラスタ内の密結合マルチ・プロセッサとクラスタ間の疎結合マルチ・プロセッサより成る［後藤］，このため，中期ＰＩＭの処理方式は，疎結合マルチ・プロセッサ向き処理方式と密結合マルチ・プロセッサ向き処理方式を融合したものとなる，前者は，Multi-ＰＳＩシステムとの共通課題として検討を進めている［宮崎］，ここでは後者について述べる，

ＰＩＭのクラスタにあたる共有メモリ（ＧＭ）を持つ密結合マルチ・プロセッサ向き処理方式を設計する上での課題は次の２つにまとめられる，

① 共有資源の操作における正当性の保証（ロック）
② 物理的に共有される資源をアクセスするときのアクセスパス競合の軽減

これらの課題を解決するために中期ＰＩＭ用のＫＬ１処理方式として，各プロセッサが局所的に操作できるデータを抽出し，これを共有メモリ（ＧＭ）と分けて，ローカル・メモリ（ＬＭ）に置く方式を検討した，

2．ＫＬ１処理イメージ

ＫＬ１の実行はゴールリダクションであり以下の操作に分けられる，

① ゴールのスケジューリング
　ゴールプールから実行可能ゴールを選ぶ，ここでゴールプールはゴールの実行のための制御用構造（後述）の集りである，
② 受動部のテスト
　ゴールの候補節群の受動部を実行し，ゴールをトラスト，フェイル，サスペンドのいずれかの状態にする，ここで実行は，ゴールの制御用構造をもとに環境（後述）を生成して行くことである，
③ 能動部のゴール生成
　受動部によってトラストされたゴールの能動部を実行する，ここでは新たなゴールの制御用構造を生成し，環境を更新する，
④ ゴール木の管理
　ゴール間の論理的関係をゴール木によって管理する，ここでゴール木とはメタゴールとゴールの関係を示すものである，

中期ＰＩＭでは，定義節（同一述語を持つ節の集合）を一つのコンパイル単位として機械語（ＫＬ１−Ｂ）におとし，これを逐次実行することで上記の４つの操作を実現する，

ＫＬ１−Ｂでは，受動部／能動部の実行をユニフィケーションと基本命令込み命令として表現し，スケジューリング／ゴール木の管理はゴール呼出し制御命令で表現する，

3．密結合マルチ・プロセッサ向き並列処理方式

3．1　課題の整理と解決策

密結合マルチ・プロセッサにおけるＫＬ１の実行とは，いくつかのＫＬ１−Ｂを解釈実行するプロセッサが，ＧＭ上で共有されたデータをもとに実行を行なうことである，この時，プロセッサはデータを共有メモリを介してアクセスするためレスポンス，スループット，ロックの課題を持つ，

この解決策は，ソフト／ハードの２つのアプローチがあり，ここでは２つを併用する，

また，この解決策は以下のとおりである，

①ソフトでのアプローチ
　プログラムに内在するデータ構造・アクセスの局所性に従い，そのデータ構造を処理方式上で切り分け，プロセッサごとにＬＭに配置する方法である，これにより，ＬＭ上のデータ構造については課題が解決される，

②ハードでのアプローチ
　個々のプロセッサの持つメモリ・アクセスの局所性を利用したキャッシュを並列実行環境でハード・ウェアとしてサポートする方法（並列キャッシュ）［松本］である，これにより，ＧＭ上のデータ構造についても課題が解決される，

3．2　データの分類

ＫＬ１−Ｂの実行でアクセスするデータは処理モデルで示したようにゴールの実行のための①制御用構造と②環境に分けられる，

①制御用構造とは，ゴール呼出し命令時にアクセスされるゴールレコードやメタゴールレコードである，

②環境とは，ストリーム変数や構造データである，

これらを並列実行環境にマッピングした場合，ゴールの実行のための制御用構造等は，ゴールを実行するプロセッサに固有に使用されるものでＬＭ上に切り出すことが可能である，ここで，試算によるとこの制御用構造に対するメモリ・アクセスは全メモリ・アクセスの約１／３程度を占める，

しかし，メタゴールの制御構造はすべての子ゴールに共有されるのでＧＭに置く，

またゴール間の同期通信を実現するストリーム変数は共通にアクセスされることが多く，構造データは共有するメリットが大きいため，ＧＭ上に置く，

Parallel Inference Machine : PIM
-On Execution Mechanism of the Intermediate Stage PIM-
Mnsatoshi SATO, Hajime SHIMIZU, Takashi CHIKAYAMA
Institute for New Generation Computer Technology (ICOT)

3．3　実現イメージ

下図に，この実行の概念イメージを示す。

（1）ユニフィケーション

ユニフィケーションは，受動部と能動部とで異なる。受動部ではゴール呼出し側引数の具体化は禁止され，他の能動部での具体化まで待ち合せを行なう機能が必要となる。このためユニフィケーションを目的別に以下の3種類に分ける。

a　受動部のユニフィケーション
b　能動部のユニフィケーション
c　引数表の生成

これらユニフィケーションは，主にストリーム変数や構造データに対して行われ，GM上の環境へのアクセスとなる（図の①），このためアクセス・スピードとロックの問題が生じ，これに対しては逐列キャッシュで解決する。

両指向機能はaで未定義の変数を読もうとした場合のサスペンド処理（図の②）とbでサスペンドされている変数に対して具体化をしようとした場合，サスペンド・ゴールを再開させるリジューム処理（図の③）で実現される。ここでサスペンド／リジューム処理のプロセッサが異なった場合，ゴールレコードをLMに置いているため，プロセッサ間通信が必要となる（図の④）。これはゴールレコードをGM上に置いた場合でもキャッシュのヒット率を上げるためにアクセスの局所性を保証しようとすれば生じる問題である。

またサスペンド／リジューム処理のコスト（同期のためのオーバヘッドは約2倍程度ある）を考慮すると回数を減らす工夫はメリットが大きい。

逐次処理系での評価によると，サスペンドの回数はスケジューリングに大きく依存し，深さ優先実行でかなり抑えることができる。またこの回数は不必要な非決定性を決定的にプログラムするプログラミングスタイルで減少させることができる。

またc引数表の生成では処理上ロックの不要であることを保証している。

（2）ゴール管理

ゴールの管理はゴールレコードとメタコールレコードによって行われる。

ゴールの実行はLM上のゴールレコードをもとに行われ，ゴールの生成／終了時のゴール木の管理はGM上のメタコールレコードの子ゴール数を変えることで行なう。

またメタコールの操作（GMのアクセスであり，処理コストは高い）は，全ゴールの終了，ゴールの失敗，他メタコールからの失敗／終了等があるが，その頻度は少ないので処理効率上，GMに置いても問題はないと予想される。

（3）ゴールのスケジューリング

ゴールのスケジューリングは，プロセッサでの局所性を引き出すために①プロセッサ主体にゴールを割り当て，深さ優先に実行できるものは実行する，ここで深さ優先実行とは，親／子ゴール間の局所性（子ゴールは親で生成された情報を用いて実行される）に従い，次に実行するゴールとしてその子ゴールを同一プロセッサに連続にマップし実行することで，これによりゴール間のプロセス間通信を軽減できる。

またゴールレコードをLM上に置くために，他のプロセッサにゴールを分配するときはプロセッサ間通信（図の④）で実現する。この通信はGM上の共有バッファを用いるプロセッサ間のメッセージ通信である。このようにプロセッサにまたがるゴールの分配はコストが高い，そのため分配方法はその頻度を抑える方法（要求駆動による一括通信）を考えている。

4．　おわりに

中間ＰＩＭのクラスタにあたる密結合マルチ・プロセッサ構成でのＫＬ１実行方式について述べた。今後は，現在作成中のソフトウェア・シミュレーションによりこの方式についての定量的評価を進めて行く予定である。

最後に，日頃御指導をいただくＩＣＯＴ４研内田室長をはじめ，御討論いただくＰＩＭグループの方々に深謝する。

＜参考文献＞

［上田］上田，"Guarded Horn Clauses"，ICOT TR-103.
［後藤］後藤，"-中期構想-"，本大会 3B-5.
［松本］松本，"-中期PIMのハードウェア構成について-"，本大会 3B-6.
［宮崎］宮崎，"Multi-PSIにおけるGHCの実行方式"，LPC'86.

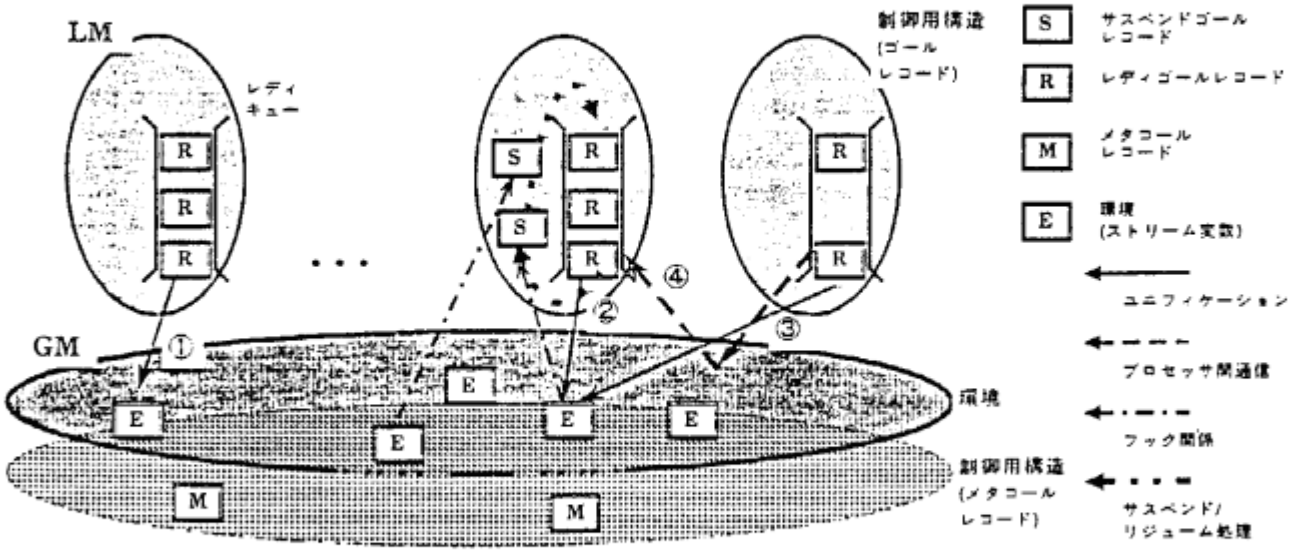

図 実行の概念イメージ