2G－1

MAPPLEによる超並列オブジェクト指向メインメモリデータベースの試み

今崎 憲児　福見 幸一　牧之内 顕文
（九州大学 工学部 情報工学科）

1. はじめに

近年，プロセッサを数百台，数千台備えた分散メモリ型並列計算機が商用化されている．このような分散メモリ型並列計算機上で共有メモリが実現できれば，その計算機は巨大なメモリ空間を持つ１台の計算機とみなすことができる．我々はこの点に着目し，現在データベースの分野で盛んに研究されているメインメモリデータベースを並列計算機上で実現することを考えている．これにより，トランザクションの並列実行やクラッシュ・リカバリ処理の高速実行などが可能となる．本論文では，現在我々が開発している超オブジェクト並列プログラミング言語MAPPLEを用いたDBの設計について述べる．特にここではデータベースの標準的なベンチマークであるOO1(Object Operation Version 1)を取り上げる．

2. 超オブジェクト並列プログラミング言語MAPPLE

我々が現在開発しているプログラミング言語MAPPLE[AI94]には次のような特徴がある．

- 並列処理の対象として，数値や文字列などの基本的なデータ型以外に，データの集まり(collection)としての集合型も扱うことができる．さらに，各プロセッサに分散したオブジェクトを管理するオブジェクトとして和集合オブジェクトを用意している．
- オブジェクト並列言語である．オブジェクト並列ではオブジェクト(データ構造とそれを操作するための関数(メソッド)を合わせ持ったもの)を並列計算機の各プロセッサに分散させ，メソッドを１つずつ同期を取りながら並列に実行する．
- 実行時にオブジェクトを動的にプロセッサに割り当てることが可能である．

Design of Massively Parallel Object-Oriented Main Memory Database Using MAPPLE
Kenji Imasaki, Koichi Fukumi, and Akifumi Makinouchi
Department of Computer Science and Communication Engineering,Kyushu University

- オブジェクトの個数の変動に対処できる．

これらの特徴をいかして，メインメモリデータベースの設計を超並列計算機上で行なう．

3. MAPPLEを用いたメインメモリデータベースの設計

3.1 メインメモリデータベース(MMDB)

MMDB[GS92]は現在，DBの分野で盛んに研究されているトピックである．MMDBはディスクを用いたDBに比べて次の様な利点が存在する．

- ダイレクトアクセスによるアクセス時間の短縮
- トランザクションの高速終了
- 高速なロックの解放
- データがポインタで表されることによる領域の節約

しかし，DBのシステムがメモリに直接アクセスするため，信頼性に欠ける．そこで，クラッシュ時のためにディスク等にバックアップやログを残すような処理が必要となる．

3.2 MAPPLEを用いた超並列MMDB

MAPPLEでは和集合オブジェクトによって，各プロセッサに分散した同一型のデータを統一的に扱うことが可能となる．超並列MMDBにおいても同様のことが必要となる．したがって，この超並列MMDBにおいても，データベースのデータは全て和集合オブジェクトによって管理され，それに対する操作はメソッドの一斉起動によって実現される．

4. 超並列MMDBでのベンチマーク

4.1 OO1(Object Operation Version1)ベンチマーク

OO1ベンチマーク[CS92]では，CAD.CASE等のエンジニアリングアプリケーションで良く用いらるオペレーションに対してDBの性能を計る．ある適当な大きさ(4MByte:20000個)の部品のDBを作り，それぞれの部品を接続する．接続の９０％はIDが近い部品に接続する．残りの１０％は全体の部品の中から任意に接続

する．そのDBに対して，次の処理を行なう．

- Lookup:100個の部品を持ってくる
- Traveral: ある部品に接続している全ての部品を持ってくる(深さ7まで)
- Insert:100個の部品を挿入する

4.2 並列版OO1ベンチマーク

部品オブジェクトは部品IDによって複数ブロックに分割される，そして各ブロックを1つずつプロセッサに割り当てる(例えば、プロセッサ1に部品1.2.3、プロセッサ2に4.5.6、...)．前述のルールに従い，それぞれの部品を接続する．接続は部品IDの値を保持することによって行なわれる．その様子を図1に示す．

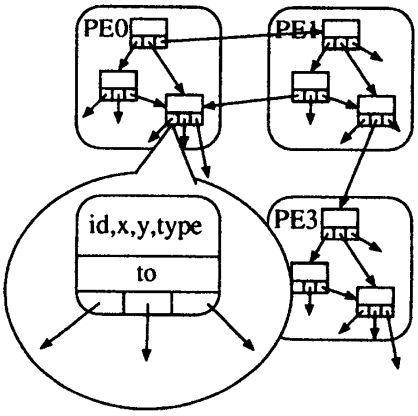

図 1: 並列版OO1ベンチマークの部品オブジェクト

前述の3つの演算についてはMAPPLEを用いて次のように行なう．

- Lookup: 各プロセッサの条件に合う部品を検索し，ホストプロセッサに送る．ホストプロセッサではその属性を表示する．
- Traverse: 次の手順で幅優先探索を行なう．
  - i) 指定の部品が存在しているプロセッサは探索を開始する．その他のプロセッサは停止したままである．
  - ii) 途中で外部への探索があった場合(図2の3)は`External Traversal Queue(ETQ)`に必要な情報(部品ID，深さ等)を挿入し，次の探索を継続する(図2の4.5....)．
  - iii) 探索するノードがなくなったら，探索を終了する．
  - iv) ETQのそれぞれの要素を対応するプロセッサに対して，メッセージとして送信する．
  - v) メッセージを受け取ったプロセッサは同様の探索を開始する．
  - vi) 全てのプロセッサのETQが空になったら終了する．
- Insert: 部品IDにしたがって挿入する．

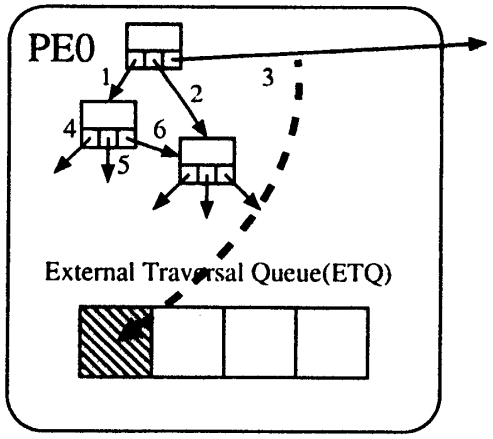

図 2: 各プロセッサでの探索

5. おわりに

本論文では，まず，プログラミング言語MAPPLEについて概説し，それを用いてどのようにして超並列メインメモリデータベースを設計するかについての構想を述べた．そして，OO1ベンチマークを例にとって，実装手順について簡単に述べた．

発表では本論文で述べたベンチマークを実際の並列計算機（富士通社製分散メモリ型並列計算機AP1000）上で実装し，評価をした結果について述べる．また，今後は，別の標準的なベンチマークであるOO7ベンチマークについても検討する．そして，リカバリーやロギングを実装し，最終的な超並列データベースシステムを設計する予定である．

参考文献

[CS92] R. G. G. CATTELL and J. SKEEN:"Object Operations Benchmark". ACM Transactions on Database Systems. Vol.17. No.1. Pages 1-31. March 1992.

[GS92] Hector Garcia-Molina. and Kenneth Salem:"Main Memory Database Systems:An Overview". IEEE Tansactions on Knowledge and Data Engineering. Vol.4. No.6. December 1992.

[AI94] 福見，今崎，天野，牧之内："オブジェクト並列プログラミング言語INADA/MPPの実現へ向けて"，文部省重点領域研究第5回シンポジウム予稿集,2-61 ～ 2-67. 平6.10.