特集 7　UDC 681.3：661.68

データパラレルマシンを用いた超並列関係データベース処理

Massively Parallel Relational Database Processing on Data Parallel Machines

岡田英明*・喜連川優**
Hideaki OKADA and Masaru KITSUREGAWA

1．はじめに

従来，関係データベースシステムの性能向上を目的とし数多くの研究がなされてきた．現在，GAMMA[1]，SDC[2]などに代表される無共有（shared nothing）型アーキテクチャに基づく並列データベースシステムは最も有効なアプローチの一つと考えられており開発，評価等が進められている．一方でコネクションマシンをはじめとするデータパラレルマシンとよばれる，制御レベルではなくデータレベルの並列性を利用する並列計算機が注目されつつある．現在，データパラレルマシンはデータ並列性を有する大規模数値計算や画像処理に主として利用されているが，データ並列性を有し大容量データを扱う関係データベース処理も有効な応用分野となりうるのではないかと予想される．しかしながら，筆者らの知る限りではデータパラレルマシンによる関係データベース処理の研究はほとんど報告されていない．データパラレルマシンの関係データベース処理への有効性を明らかにすることを目的とし，二次記憶を含めた大規模なジョイン処理をコネクションマシンCM-2上に実装し，性能を測定した．本稿ではその処理方式ならびに評価結果について報告する．

2．データパラレルマシンCM-2

コネクションマシンCM-2[4]はSIMD型の超並列計算機であり，1bitのPEを16PE単位にハイパーキューブ網で結合し，最大構成時では65,536PEで構成される．命令はフロントエンドからシーケンサと呼ばれるハードウェアを通してブロードキャストされ，各PEはその命令を内部状態により選択して実行し，SIMD動作を行う．CM-2のプログラミングではPARISとよばれる並列命令セットを用いるが，そこでは仮想プロセッサ（VP）なる考え方が大きな特長となっている．仮想プロセッサ数と物理プロセッサ数の比を仮想プロセッサ比（VP比）と呼び，各物理プロセッサはVP比数分の仮想プロセッサの命令を実行する．したがって，プログラム実行時にはシステムの物理プロセッサ数とプログラムの仮想プロセッサ数に応じて動的に仮想プロセッサ比が決定される．PARISの仮想プロセッサの機能は，1つのデータ要素を1つの仮想プロセッサに割り当てることを基本とする．あるデータ集合が割り当てられている仮想プロセッサの集合のことを，仮想プロセッサセット（VPセット）と呼ぶ．VPセットはPARISへの関数呼び出しによって生成される．その大きさ（VPの数）は生成された時点で決まり，以後変化しない．複数のVPセットを使用することもできるが，制御対象は常に1つのVPセットであり，VPの数は物理プロセッサ数の2の冪乗倍の数でなければならない．メモリ割り当てなどもこのVPセットに対する命令によって行われる．このように任意の大きさのデータベースを物理プロセッサ数とは独立に処理することが可能となる．CM-2にはデータボルトと呼ばれるRAID-2型のディスクアレイが二次記憶システムとして付加されている．32台のディスクを並列駆動することにより転送速度は25MB/secを維持できるとされている．データボルト上のファイルにも仮想プロセッサの概念があり，コネクションマシン上の主記憶とのデータのやり取りは同じVP比のVPセットに対してのみ可能となっている．

3．主記憶内ジョイン処理アルゴリズム

データパラレルマシンの関係データベース処理応用の第一歩として主記憶内にリレーションが収まる時のジョイン処理アルゴリズムについて，ソートマージ法を用いたアル

*三菱電機㈱　情報電子研究所
**東京大学生産技術研究所　付属機能エレクトロニクス研究センター

研　究　速　報

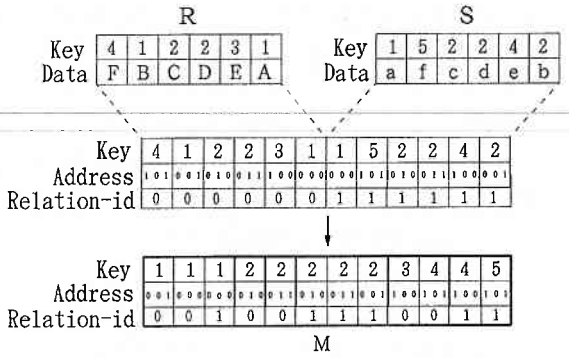

図1　ソートフェイズ

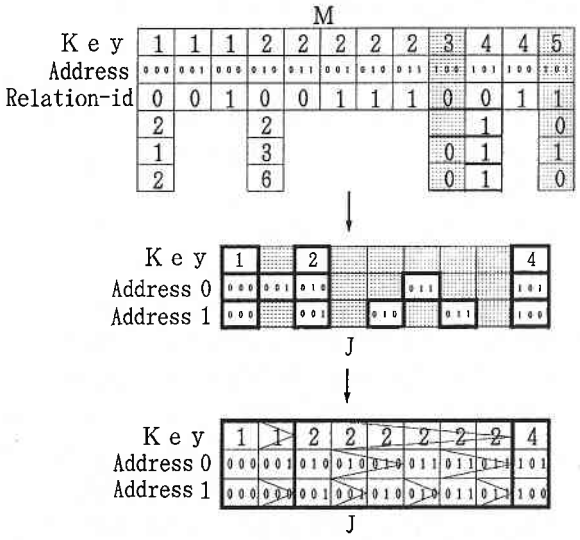

図2　マージジョインフェイズ

ゴリズムとハッシュによる分割を用いたアルゴリズムを提案し，コネクションマシンCM-2上に実装，評価した[3]．その結果データパラレルマシンでは，データ分布によらず一定の性能を維持できるデータパラレルソートマージジョイン（DPSMJ）が優れていることから，本論文ではこれを利用することとした．リレーションの仮想プロセッサへのマッピングは，1タプルが1仮想プロセッサに対応するものとする．DPSMJアルゴリズムは大きく三つのステップに分けられ，それぞれソートフェイズ，マージジョインフェイズ，置換フェイズと呼ぶ．各フェイズにおける動作を図1，2，3に示す．

ソートフェイズ：最初に，リレーションRとSが併合されて1つのリレーションMを作成する．ここでMのタプルはキー属性とアドレス，リレーションのID（Rが0，Sが1とする）のみからなる．タプルのキー属性に従ってMを昇順にソートする．同じキー属性を持つタプルについては，リレーションIDを利用してRのタプルの後にSからのタプルが並ぶ．

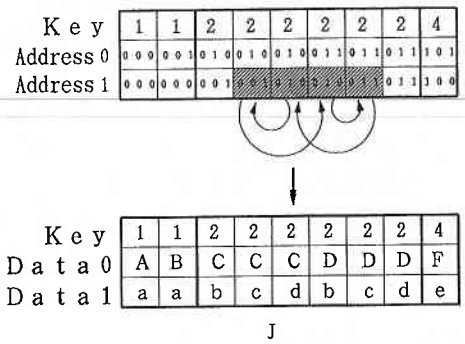

図3　置換フェイズ

マージジョインフェイズ：リレーションMの上で必要な処理を行い，ジョイン後のリレーションJを生成する．Mのそれぞれのタプルは，自分と同じキー属性を持つタプルが，Jの中にいくつ存在することになるかを計算し，スキャン加算してJのタプル数を得る．その後，リレーションJのための空間をアロケートして必要なデータをそこに転送する．このフェイズはデータ送信とデータ複製からなる．

置換フェイズ：マージジョインフェイズが終了した段階では，まだデータの組み合わせが適切ではない．求めるリレーションを得るために，Sのタプルから送られてきたデータを再配置する．このようにして，最終的に望むリレーションを，Jの中に作ることができる．

4．データボルトを用いた大規模ジョインアルゴリズム

4.1　アルゴリズムの概要

本節では，二次記憶を含めた大規模リレーションに対するジョイン処理方式について述べる．本アルゴリズムはスプリットフェイズ，ジョインフェイズの2つのフェイズからなる．

ジョイン処理されるリレーションをR，Sとする．スプリットフェイズにおいてN個のバケットに分割するとき，リレーションR，Sから生成されたi番目のバケットをR_i，S_iとする．リレーションR，Sはそれぞれ，長さl_R，l_Sのタプルからなり，C_R，C_S個のタプルを持つものとする．また，キー属性の長さをl_{key}と表す．

はじめに，リレーションR，Sはデータボルトにファイルとして書き込まれている．このデータボルト上のファイルにもコネクションマシンの主記憶と同じくVP比の概念があり，あるVP比であるVPセットにより書き込まれたファイルは同じVP比のVPセットによってしか読み込むことができない．したがって，もしデータボルト上のファ

イルと異なるVPセットを読み書きする場合には，データボルト上のVPセットに等しいVPセットを確保しておき，それをI/Oバッファとして用いる．たとえば，読み込み時にはI/Oバッファに読み込んだリレーションをsend命令を用いて，そのリレーションを必要としているVPセットに転送する必要がある．このデータボルト上のVPセットのVP比をリレーションR，SについてそれぞれV_R，V_Sとする．また，物理プロセッサ数をPとする．以下に，ジョインアルゴリズムの詳細について述べる．

4.2 スプリットフェイズ

スプリットフェイズでは，リレーションはキー属性値のハッシュ関数値に応じたバケットに分割され，データボルトに書き込まれる．リレーションR，Sについてそれぞれ独立に同じ処理を行う．

スプリットフェイズでは，スプリットバッファ，バケットバッファという2種類のバッファが少なくとも必要となる．スプリットバッファは読み込んだリレーションのキー属性に対してハッシュ関数値を計算して，各タプルがどのバケットに属するかを決定する．バケットバッファには，同じハッシュ関数値を持つタプルがスプリットバッファより送られる．それらをデータボルトに書き込む．このスプリットバッファとバケットバッファの割り当てを図4に示す．巨大なスプリットバッファとバケットバッファを1つだけ用意して共用する．スプリットフェイズではリレーションRを読み込むためのI/Oバッファを VP比V_Rで確保し，V_Rの倍数でありかつ，できるだけ大きいVP比V^R_{split}でスプリットバッファを確保する．バケットバッファに送られるタプル数は，平均的にはスプリットバッファに入るタプル数の分割数N分の1となるので$\frac{V^R_{split}}{N}$，もしくはキー属性値の分布を考慮して$\frac{V^R_{split}}{N}$より大きいVP比V^R_{bucket}のバッファを確保する．スプリットフェイズでは以下の処理を$\frac{C_R}{PV^R_{split}}$回行うことになる．

1．次の処理を最大で$\frac{V^R_{split}}{V^R}$回行う．

(a) データボルトから入力バッファにVP比V_Rでタプルを読み込む．

(b) 入力バッファからスプリットバッファへタプルを転送する．

2．スプリットバッファにおいてハッシュ値を計算する．

3．次の処理をiが0からN-1についてN回行う．

(a) ハッシュ関数値がiであるタプルを数え，バケットバッファに送る．

(b) バケットバッファに送られてきたタプルをデータボルトにRiバケットのファイルとして書き込む．

データ分布の不均一性などにより3(a)段階において，バケットバッファの溢れが生じる場合がある．そこで，溢れが生じる場合にはバケットバッファに入り切るタプルは出力し，入り切らないものはスプリットバッファに保存しておく．リレーション読み込み時にはI/Oバッファからスプリットバッファの空いている領域へタプルを転送する．

4.3 ジョインフェイズ

ジョインフェイズでは，図5に示すようにスプリットフェイズで生成されたバケットR_i，S_iをジョインバッファと呼ばれる2つのバッファにそれぞれを読み込んで主記憶内でのジョイン処理を行い，結果をデータボルトに出力することを分割数の数だけ繰り返す．主記憶内でのジョイン処理アルゴリズムには，DPSMJアルゴリズムを使用する．スプリットフェイズにおいてV^R_{bucket}のVP比でデータボルト上に書き込まれているバケットを読み込むには，V^R_{bucket}のVP比を持つバッファを用意しなければならない．そのバケットに対する書き込みが1度であれば，そのバッファがジョインバッファとなることができるが，複数回の書き込みが行われれば，そのバッファは単なる入力バッファとして使用され，バケットのサイズに応じたVP比V^R_{join}でジョインバッファを新たに確保する必要がある．スプリットバッファよりもリレーションのサイズが大きい場合には，バケットバッファへの書き込みが複数回行われる．したがって多くの場合，このような入力バッ

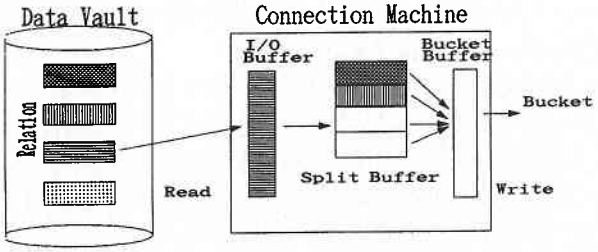

図4　スプリットフェイズにおけるバッファ割り当て法

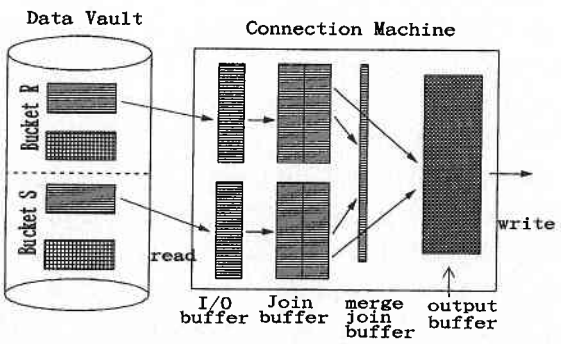

図5　ジョインフェイズにおけるバッファ割り当て法

ファが必要となる.

ジョインフェイズでは以下のような処理をN回行う.

1. バケット R_i をすべて読み込むまで，以下を繰り返す.
   (a) バケットを入力バッファに読み込む.
   (b) 入力バッファからジョインバッファへ転送する.
2. バケット S_i についても同様の処理を行う.
3. リレーションR，Sから生成されたバケットファイルのVP比を V^R_{join}，V^S_{join} とすると，キー属性とアドレス，リレーションIDからなる，
   $$V^R_{join} + V^S_{join} \leq V_{merge}$$
   をみたすVP比 V_{merge} のリレーションMを用意し，R_i，S_i のキー属性，アドレス，リレーションIDをMに送る.
4. リレーションMのDPSMJによるソート.
5. DPSMJにより生成される結果リレーションのタプル数を求め，その数に応じたVP比 V_{result} で長さが $l_S + l_R - l_{KEY}$ のバッファJを確保する．確保できない時は，利用可能な主記憶量だけ確保して，以下の処理を分割して行う.
6. リレーションMの内容に応じて，R_i，S_i からデータをJに送り，J内で結果リレーションの生成を行う.
7. 結果リレーションをデータボルトに出力する.

以上の処理をデータパラレル手法によりプログラム化した.

5. 性 能 評 価

データボルトを用いたジョイン処理アルゴリズムをコネクションマシンCM-2に実装し性能を測定した．使用したCM-2は8Kプロセッサ，ローカルメモリ8KB，全体では64MBの主記憶を有する．この測定では，タプルのキー値として1からタプル数までの値を重複なく設定している．測定時間は，コネクションマシンの処理時間のみであり，フロントエンドの処理時間は含まない．キー長は4バイト固定とし，タプル長は64，128，256，512バイトの4種類の値をとるものとする．両方のリレーションは同じタプル長，同じタプル数であるとし，リレーションのサイズを8，16，32，64MBと変化させた．タプル数は，タプル長とリレーションのサイズに応じて決定され，16Kから1024Kの値をとる．図6に全体の処理時間を示す.

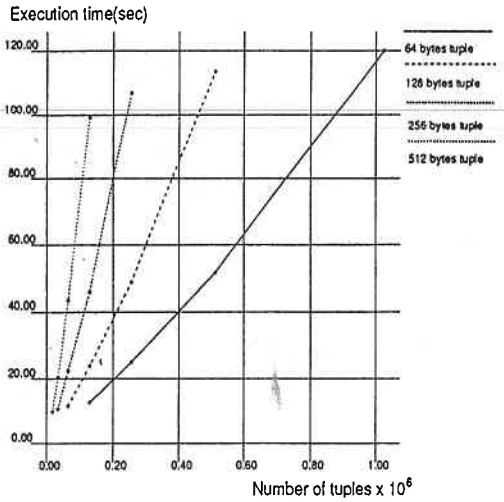

図6　データボルトを用いたジョインプログラムの実行時間

6. お わ り に

これまでにデータパラレルマシンの関係データベース処理への有効性に関する研究はほとんどなされていなかった．ここでは外部記憶装置とのI/Oをも含めたデータパラレルジョインアルゴリズムをデータボルトと呼ばれるディスクアレイを備えたコネクションマシンCM-2に実装し，その性能について評価した.

データパラレル記法により外部記憶装置を含めた関係データベース処理が可能であることを明らかにした.

現在のシステムでは充分な性能は必ずしも得られず，仮想プロセッサ機構におけるメモリ，通信，ファイルに対する制限が大きいことがわかった．データパラレルマシンの関係データベース処理への適用には，より柔軟なメモリ管理が望まれる．今後，データベース処理に不可欠なデータパラレル機構について明確化を進めてゆく予定である.

(1993年12月10日受理)

参 考 文 献

1) D.A. Schneider, et al: A Performance Evaluation of Four Parallel join Algorithms in a Shared- Nothing Multiprocessor Environment, Proc. of SIGMOD, pp. 110-121, 1989.
2) M. Kitsuregawa, et al: The Super Database Computer (SDC) System Architecture, Algorithm, and Preliminary Evaluation, Proc. of the 25th HICSS, pp. 308-319, 1992
3) M. Kitsuregawa, K. Matsumoto: Massively Parallel Relational Database Processing on the Con-nection Machine CM-2, The 2nd International Symposium on Database Systems for Advanced Applications, pp. 226-235, 1991.
4) Thinking Machines Corp.: Connection Machine Model CM-2 Technical Summary, Version 5. 1, 1989.
5) 岡田英明，松本和彦，喜連川優：コネクションマシンCM-2による大規模関係データベース処理とその評価，情報処理学会，第94回データベースシステム研究会，94-29, pp. 253-262, 1993.