

Title	主記憶データベースを対象とした高機能メモリアーキテクチャに関する研究
Author(s)	府川, 智治
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1694
Rights	
Description	Supervisor: 田中 清史, 情報科学研究科, 修士

主記憶データベースを対象とした高機能 メモリアーキテクチャに関する研究

府川 智治 (110110)

北陸先端科学技術大学院大学 情報科学研究科

2003 年 2 月 14 日

キーワード: 主記憶データベース, DRAM, メモリコントローラ, 質問処理.

1 はじめに

近年, プロセッサと主記憶間の性能格差の拡大およびデータの大規模化により, データベースの応答時間が増大し, 質問処理の高速化が重要になっている. 主記憶データベース (以下, MMDB) [1] はディスクを使用したデータベースに比べて高速でかつランダムアクセスにも強い. しかしさらなる高速化を図るため, メモリからデータを連続してプロセッサに転送する方式と, ポインタを介したアクセスを見かけ上一回のアクセスでデータを取得する方式を提案する. これらの機能をメモリコントローラ (以下, MC) に組み込み, シミュレーションにより質問処理を評価する.

2 主記憶データベース (MMDB)

MMDB 方式におけるリレーション構成は, メモリ使用量の削減から各セルには実体へのアドレス (ポインタ) を格納する. ポインタサイズより小さいデータはセルに直接値を格納する. このようなリレーションの特性から条件探索を実行する場合, データの一致 / 不一致はポインタ比較で判定できるが, データの大小比較はポインタを介した実体同士の比較が必要になる.

3 データ転送方式

本研究で提案する高速大規模データ転送方式について述べる.

一つ目の転送方式として Stride Data Transfer (SDT) を説明する. 従来の MC に次の機能を付け加えることでメモリ内のデータが連続してプロセッサに転送される. SDT 開始前にプロセッサから MC 内のレジスタにデータ転送数とデータ間隔値を格納する. その

後プロセッサはメモリリクエストを発行する．MC はそのリクエストに対するアドレスをレジスタに記憶し，DRAM アクセスを行ってデータをプロセッサに転送する．以下，レジスタ内に記憶したアドレスにデータ間隔値を加算して次アドレスを生成し，順次読み出しを行う．データ転送の終了はデータ転送数を越えた場合，間隔値の加算による Bank，Row アドレスのいずれかが変化した場合，あるいはページ境界を跨いだ場合である．またプロセッサのキャッシュミスによるメモリアクセスが発生した場合は SDT 転送を一時中断する．

二つ目の転送方式として Two-Phase Data Transfer (TPDT) を説明する．ここでは実体データとして文字列を扱う．これはメモリ資源の節約のためにポインタを介した二段階の実体アクセスを短縮する高速転送方式である．TPDT 方式を実現する MC は一段階目のアクセスで取得したポインタをプロセッサに転送せず，そのポインタを使って二段階目のアクセスをして読み出された文字列をプロセッサに転送する．TPDT 転送の終了は文字列の終了を示す NULL 文字を検出した場合，あるいはプロセッサが文字列の比較途中で条件に一致しないとわかった場合である．

4 評価・考察

質問処理における提案した転送方式の効果をシミュレーションによって示す．このシミュレータは SPARC 命令セットのアセンブラコードを入力とするシミュレータである．1 命令実行を 1 プロセッサクロックサイクルとする．各質問処理を実行してその総実行サイクル数を求め，従来方式と比較する．評価対象のリレーションは Wisconsin Benchmark[2] の一部を MMDB 用に変更した．SDT 方式および TPDT 方式によって転送されるデータは FIFO バッファ[3] に格納する．これはプロセッサ内のキャッシュの一部を再構成してバッファとしたものである．

評価結果から SDT 方式を適用した場合，プロセッサの命令実行に並行してメモリから予め必要とするデータを読み出して FIFO に格納する．そのためプロセッサはデータ読み出し毎にメモリアクセスのペナルティを被ることなく条件探索が行える．

一方，TPDT 方式を適用した場合，リレーションに格納する文字列の種類や長さによって実行サイクル数が変わる．つまり文字列が短くその種類が少ない場合，キャッシュにポインタと実体を全て格納することが可能であるため TPDT 方式による効果は見られない．しかし，リレーションサイズが大きくキャッシュにそれらを全て格納できる保証がない場合，TPDT 方式は効果が発揮される．さらに，長い文字列を検索する場合，従来方式では比較途中でメモリアクセスが発生する可能性があるが，TPDT では発生しないため条件探索を高速化できる．

5 おわりに

本研究では大規模データに対するデータベースの高速な質問処理を実現するために高速大規模データ転送方式として MMDB のデータ構造を利用した方式と DRAM のハードウェア特性を利用した方式を提案し、これらの機能を実現する機構を MC に組み込み、実装した。

シミュレーションによる質問処理の結果から、キャッシュにリレーション全体が収まり、データが繰り返しアクセスされる場合ではキャッシュが有効利用されているため、SDT 方式および TPDТ 方式での性能向上は見られなかった。しかし SDT 方式は時間的および空間的局所性が共に存在しないデータに対して条件探索を行う場合にプロセッサとメモリ間の性能差を吸収したと言える。また TPDТ 方式ではリレーション全体をキャッシュに格納できる保証がない場合に効果があり、さらにキャッシュ内の一つのブロックより大きい文字列を条件探索する場合、FIFO バッファによって文字列の比較途中でメモリアクセスが発生しないためプロセッサはストールなしで命令実行を継続することができる。

以上のことから本方式を適用することで質問処理の高速化を実現したと考えられる。

参考文献

- [1] H. Garcia-Molina, K. Salem, “Main Memory Database Systems: An Overview.” IEEE Trans. on Knowledge and Data Engineering, Vol.4, No.6, pp.509–516, 1992.
- [2] DeWitt. D. J, “The Wisconsin Benchmark: Past, Present, and Future.” The Benchmark Handbook, pp.269–316, J.Gray ed., Morgan Kaufmann, 1993.
- [3] Khairuddin bin Khalid and Kiyofumi Tanaka, “Implementation of FIFO Buffer Using Cache Memory.” IPSJ SIG Notes, ARC, Vol.2002, No.112, pp.83–88, 2002.