4AC－7

並列論理型言語KLICによる並列データベースのI/O処理の高速化

宮崎 純　横田 治夫

北陸先端科学技術大学院大学 情報科学研究科

1 はじめに

我々は，並列論理型言語KLIC[1]によるアクティブデータベースシステム[2]のプロトタイプParade[3]を提案している．KLICの論理変数によるプロセス間のメッセージ通信は，無共有並列計算機上でデータベース処理を実現する際に非常に適合性が良い．これまでの研究により，データベース処理をKLICのプロセス間のメッセージ通信を利用することにより実現し，汎用ワークステーションならびに超並列計算機nCUBE2上で動作させ，その記述能力と移植性の高さを実証してきた．しかしながら，データベース処理において非常に大きな鍵を握る二次記憶およびPE間のデータ転送などのI/O処理が，システムの性能のボトルネックとなっていることが分かってきた．これは，KLIC標準のI/Oが性能を重視して設計されたものではないことに起因する．

本稿では，KLICによる並列データベース処理を実用的な速度で動作させるために，KLICのジェネリックオブジェクトを用いて低レベルのI/O性能の向上を目指し，その初期評価を並列計算機nCUBE2を用いて行なった．

2 I/O処理の高速化

2.1 ページオブジェクト

KLIC標準のI/Oは，外部表現と内部表現を交換するためにparse/unparse機構を必要とし，かつそのデータアクセス単位は小さい．そのため，ソフトウェアオーバヘッドの大きいものとなっており，しかもディスクI/Oの面から見ても，スループットを高くすることは困難である．また，PE間のデータ転送は基本的に1レベルごとの小さなメッセージで行なわれるため，メッセージハンドリングオーバヘッドの大きな並列計算機では，そのコストは重い(図1参照)．

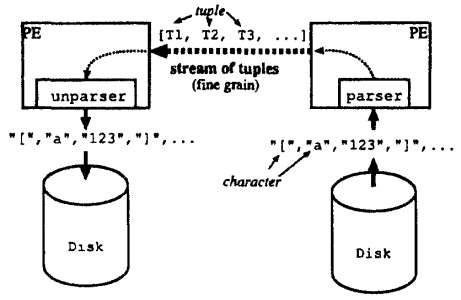

図 1: KLIC 標準I/O

一般的にデータベース処理では，I/Oが全体のボトルネックとなることが多く，このボトルネックを解決することがデータベース全体の処理速度の向上につながる．通常のデータベースでは，ディスクをページ単位といった大きなデータ単位でアクセスしてディスクI/Oのスループットを高くすることが行なわれている．これをKLICで行なうためには，KLICのデータ型を拡張するためのジェネリックオブジェクトを用いるのが，最も簡単である．そこで，新しいデータ型としてページ単位でディスクをアクセスできるよう，ジェネリックオブジェクトを用いてページオブジェクトを定義し，ディスクI/Oのスループットの向上を図った．ページ本体は，KLICのヒープとは別空間に取り，ゴミ集めによる無駄なデータの移動を減らすことも試みている．このページオブジェクトによるページ単位のディスクアクセスは，B-Treeなどのアクセスパス構造との親和性の向上にも貢献する．また，このページオブジェクト単位でPE間のデータ転送を行なうことにより，通信のスループットの向上も同時に達成することができる．一般に無共有並列計算機は，メッセージのセットアップコストがメッセージバッファの獲得やロックのため，通信のスループットに比較してかなり大きい．しかし，KLICはプロセッサ間通信の際に，通常1レベルずつの細かなメッセージが転送される．KLICはマルチレベルでのメッセージ通信もサポートしているが，これはメッセージのencode/decodeというソフトウェアの

Acceleration of I/O Performance for Parallel Databases Using Parallel Logic Language KLIC
Jun Miyazaki, Haruo Yokota
School of Information Science, Japan Advanced Institute of Science and Technology

オーバヘッドが伴う．しかしながら，前述のページオブジェクトを用いれば，連続空間上の大きなデータ単位でメッセージを送受信することが可能となり，メッセージのセットアップコストを相対的に小さくでき，さらにデータの encode/decode といったソフトウェアのオーバヘッドも削減できる．以上によりトータルな通信オーバヘッドを大きく改善することができる (図2参照)．

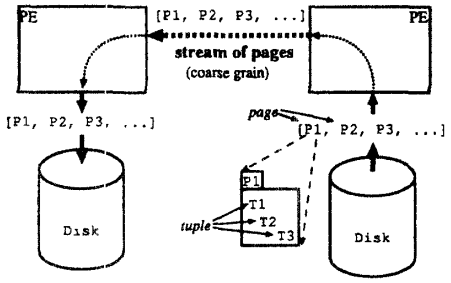

図 2: ページオブジェクト I/O

2.2 ページオブジェクトの拡張

このページオブジェクトに対して様々なメソッドを定義し，リストなどの KLIC の基本データ構造に変換することなく直接データを扱えるようにすることができる．例えば，選択，射影，結合演算などを直接メソッドとして定義し，ページ単位で効率よく処理することが可能である．

3 ページオブジェクトの I/O 性能

4KB 単位でディスクアクセスとメッセージ通信ができるよう，ジェネリックオブジェクトを用いてページオブジェクトを定義し，ページオブジェクトと KLIC 標準の I/O の性能の比較を nCUBE2 上で行なった．実験に用いた nCUBE2 は，表 1で示す I/O 性能を持つ．

表 1: nCUBE2 の I/O 性能

Disk Read	1.1 MB/s
Disk Write	0.46 MB/s
PE 間通信	2.2 MB/s

表 2は 1 タップル当たり 180B である 3000 タップルのリレーションにおける，ディスクからの読出し，書込み，PE 間通信の I/O 性能の比較を行なったものである．表中の termio は KLIC 標準の I/O，codedio は parser/unparser を専用の符合化を行ない最適化を行なったもの，pobjio は今回定義したページオブジェクトを用いた I/O の場合である．

この結果からも分かるように，ページオブジェクト

表 2: I/O 性能の比較 (単位: msec)

	termio	codedio	pobjio
Disk Read	2.56×10^4	934	907
Disk Write	2.35×10^5	2130	2030
PE 間通信	4170	4180	165

を用いた I/O は，KLIC 標準の I/O に比較して著しい I/O 性能の向上が得られることが明らかである．これは，ページオブジェクトが，KLIC 内部表現と外部表現の変換を行なう parser/unparser のオーバヘッドがなく，かつ大きなサイズのデータ単位で I/O を行なうため，I/O スループットが向上したからである．専用の符合化を行なう codedio の parser/unparser は，このオーバヘッドが小さく，ページオブジェクトに比較してディスク I/O がわずかながら劣る程度であるが，PE 間通信は 1 レベルずつであるため，通信性能の改善は得られない．

4 おわりに

並列論理型言語 KLIC を用いてデータベース処理を行なう場合，ディスクや通信の I/O がボトルネックとなる．これは，KLIC の I/O の粒度が小さく，また，I/O の際のソフトウェアのオーバヘッドが大きいためである．本稿では，KLIC のジェネリックオブジェクトを用いて新しくページオブジェクト型を定義し，これを用いて I/O をページ単位で効率よく行なう方法について述べた．さらに nCUBE2 上での実験を通して，このページオブジェクトを用いた I/O が，従来の I/O の性能に比較して大きな性能の改善が得られることを示した．

今後，このページオブジェクトに様々なメソッドを定義することにより，ページ単位で効率の良いデータベース処理を行なうことができるように拡張を行なう予定である．

参考文献

[1] T. Fujise, T. Chikayama, K. Rokusawa and A. Nakase: "KLIC: A Portable Implementation of KL1", Proc. of FGCS '94, pp. 66–79 (1994).

[2] J. Widom and S. Ceri: "Active Database Systems: Triggers and Rules For Advanced Database Processing", Morgan Kaufmann Publishers, INC., San Francisco, CA (1996).

[3] 横田, 宮崎, 土屋：“並列アクティブデータベースエンジン Parade の並列処理”, 文部省科学研究費重点領域研究『高度データベース』松江ワークショップ講演論文集, 第 2 巻, pp. 426–433 (1996).