

機関番号：14603

研究種目：若手研究(B)

研究期間：2009～2010

課題番号：21700111

研究課題名(和文)

マルチコアプロセッサ時代における高速データベース処理方式の実現

研究課題名(英文)

A study on a fast database processing method toward the era of multicore processors

研究代表者

宮崎 純 (MIYAZAKI JUN)

奈良先端科学技術大学院大学・情報科学研究科・准教授

研究者番号：40293394

研究成果の概要(和文)：

本研究では、マルチコアプロセッサを活用するデータベース処理について、ディスクのバッファ管理方式ならびに時系列パターンマイニングに焦点を当て、それらのアルゴリズムの研究と実現方式の研究を行った。その結果、バッファ管理に関してはコア数に対してほぼ線形のスループットを実現し、また、時系列パターンマイニングに関しては従来の手法よりもキャッシュミス回数を減らすことにより2倍の高速化、さらに6スレッドで4倍以上となるスケーラビリティを実現した。

研究成果の概要(英文)：

In this study, we focused on the topics of disk buffer management and sequential pattern mining, and studied on their algorithms and implementation methods so that we can make use of the potential of multicore processors. As a result, as for the former case, we showed almost linear throughput of accessing the buffer to an increase in the number of cores. As for the latter case, the proposed cache-conscious sequential pattern mining algorithm obtained 2x faster than the existing one because of the reduction of the number of cache misses, and more than 4x speedups when six threads are used in its scalability.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	1,800,000	540,000	2,340,000
2010年度	1,500,000	450,000	1,950,000
年度			
年度			
年度			
総計	3,300,000	990,000	4,290,000

研究分野：総合領域

科研費の分科・細目：情報学・メディア情報学・データベース

キーワード：データベース、マルチコアプロセッサ、情報システム、データマイニング

## 1. 研究開始当初の背景

近年、ハードウェアの微細化が急速に進み、動作クロックを上げるよりも、一つのチップに複数のCPUコアを搭載するプロセッサ、いわゆるマルチコアプロセッサにより計算処理の高速化を目指す方向へ進んでいる。コア

数は増加の一途をたどり、例えばSun SPARC T2プロセッサのように8コア構成で、1コア当り8スレッドの同時実行、つまりプロセッサ当り同時に64スレッドを処理可能なプロセッサも存在する。また、IBM、ソニー、東芝が共同開発したCellプロセッサや、

NVIDIAが開発したグラフィックプロセッサのように高速計算に特化するため、普通のプロセッサが持つべき機能を簡略化した複数の計算用コアを持つものも現れており、汎用プロセッサよりも格段に高速な計算が可能となっている。

一方、計算機の最も重要な応用の一つであるデータベースは、ますます高速なデータベース処理が求められるようになってきている。例えば意思決定支援システムやデータマイニング等は高い処理性能が要求される。しかしながら、データベースのソフトウェアは極めて保守的であり、最新のマルチコアプロセッサの能力を享受できず、高速処理の要求に十分に答えられていない。

## 2. 研究の目的

本研究では、マルチコアプロセッサ上で高速なデータベース処理を実現するために、(1)マルチコアプロセッサ上でのデータベースの共有データに対するペナルティの小さな排他制御方式の開発、(2)マルチコアプロセッサ上での、主記憶メモリの性能を最大限に活かせるデータアクセス方式とそれを利用したデータベース処理アルゴリズム、の研究開発を行なう。より具体的には、(1)に関しては、データベース処理の根幹部分であるバッファ管理について、効率の良い排他制御アルゴリズムを提案し、(2)に関しては特に時系列データマイニング処理に着目し、マルチコアプロセッサの性能を活かすためのキャッシュ指向の並列処理アルゴリズムの提案を行なう。双方ともに、現在の、さらには今後のマルチコアプロセッサ時代に不可欠な技術であり、本研究の成果は非常に意義の高いものである。

## 3. 研究の方法

本研究の目的を達成するために、(1)プロセッサのCAS命令を利用したロックフリー同期によるバッファ管理データ構造の実現とその評価、(2)時系列パターンマイニングアルゴリズムPAIDのキャッシュ指向化と並列化によるマイニング処理の高速化とその評価、について取り組んだ。

(1)については、近年のマルチコアプロセッサアーキテクチャでのデータベースのボトルネックがバッファ管理部にあるとの報告に基づき、データ構造上複雑な排他制御を必要としないバッファ管理手法の選定、ならびにその手法のロックフリー化により、当該目的を達成する。(2)については、性能の最も良いとされる既存のPAIDアルゴリズムを利用し、そのデータ構造のキャッシュ指向化による主記憶バンド幅の有効利用、ならびにマ

ルチコアプロセッサを活用するための並列処理アルゴリズムの開発により、研究目標を実現する。

## 4. 研究成果

### 4.1 データベースバッファ管理の効率化

データベースのバッファ管理の問題点は、ディスクページをバッファフレームに固定する際、データ構造の操作に排他制御が必要な点である。このデータ構造が複雑なほど、排他制御の回数や複雑度が増し、特にマルチコアプロセッサでは、ロックの競合が確率的に増加してしまう。これまで長年バッファ管理として、LRUポリシーが多く使われてきたが、LRUは長いリスト構造と、リスト構造内のデータを高速にルックアップするためのハッシュ表から構成され、あるページが参照されるごとに、データ構造の複数箇所で排他制御を行った上で変更が必要であった。この複雑なLRUのデータ構造の排他制御を減らすことは困難であり、データ構造が簡潔な他のポリシーを選択しなければ、ロックフリー化は難しい。その中でCLOCKポリシーはデータ構造が簡潔であり、バッファのヒット率もLRUと遜色がない。そこで、本研究ではCLOCKポリシーをもとに、そのデータ構造のロックフリー化を実現し、Nb-GCLOCKと名付けた。

人工的なデータを用いて、予備実験を行ったところ、Nb-GCLOCKは同時実行64スレッドからバッファにアクセスしても、ほぼ線形なスループットが得られることが判明した。さらに現実のオンライントランザクション処理を想定するTPC-Cベンチマークで提案手法を評価した。その結果Nb-GCLOCKは従来のバッファ管理手法と比較して、同時実行スレッド数を増加させても排他制御の過度な競合がないため、性能が劣化しないことが判明した(図1)。また、バッファ管理とは関係の無い部分である索引構造がボトルネックとなっていることも分かった。索引構造については、OLFITと呼ばれるロックフリー化されたB木が存在し、これを利用すればさらなるOLTP処理性能が期待できる。

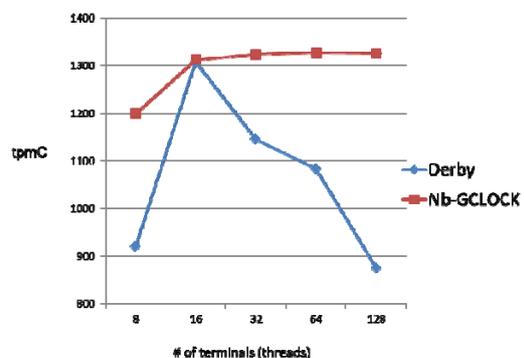


図1: TPC-Cベンチマークの結果

#### 4.2 高性能時系列パターンマイニング

時系列パターンマイニングとは、時系列データにより構成されたデータベースから時系列パターンと呼ばれる出現順序を維持した状態の特定の閾値を満たす部分列を発見する手法である。時系列パターンマイニングアルゴリズムは、データ解析、行動予測、情報推薦といった分野において重要な技術とされており、過去の時系列データから得られた今まで気づかなかった時系列パターンを利用することで、顧客の行動予測といったマーケティングや、生物の代謝経路の発見など、様々な応用ですでに利用されている。

しかしながら、一般に時系列パターンマイニングは、データ規模、処理の複雑さから、非常に時間のかかる処理であり、特に最小支持度とよばれる閾値の設定によっては、発見されるパターンの組み合わせが膨大となり、その計算量は非常に大きなものとなる。このため、時系列パターンマイニングアルゴリズムの高速化のための研究が数多くなされてきた。その多くはアルゴリズム自体の改良、あるいは並列化による高速化であった。しかしながら、マルチコアプロセッサ時代においては単なる並列化だけでは不十分である。なぜならば、マルチコアプロセッサは、主記憶を各コアが共有しており、主記憶へのアクセスが競合すれば、データ待ちのために著しく処理性能が落ちるためである。これを回避するためには、並列化とともに主記憶アクセスを減らすためのキャッシュ指向化が必須となる。キャッシュ指向化はアクセス競合だけでなく、メモリのアクセスレイテンシを著しく減少させ、プログラムの処理性能を大きく上げることができる。すなわち、マルチコアプロセッサ時代では、並列化とキャッシュ指向化の両立が要求される。

近年の研究報告から、PAIDと呼ばれる優れた時系列パターンマイニングアルゴリズムに着目し、本研究ではPAIDをベースに、そのキャッシュ指向化と並列化を同時に実現するCC-PAIDアルゴリズムを提案した。

時系列パターンマイニングでは発見された時系列パターンを辞書式順序木で表わすことができる。PAID アルゴリズムでは、長さ $k+1$ の時系列パターンの抽出処理の対象を決定する際、辞書式順序木上で深さ優先探索により、一つの長さ $k$ の時系列パターンを選択する。

ここでPAIDのILP-listと呼ぶデータ構造のキャッシュミスを考える。図2は長さ2の時系列パターンA→Bの処理開始時のCPUキャッシュの状態を表し、破線は処理対象となっていることを表している。この例では時系列パターンA→Aの処理終了時にCPUキャッシュに時系列データのID (SID) 4, 5のILP-listしか存在していない。このため、時系列パ

ターンA→Bの処理開始時にSID 1, 2, 3 をフェッチする必要があり、これがキャッシュミスの原因の一つとなる。時系列パターンマイニングでは、基本的にSIDに対して昇順にアクセスされるため、SIDの番号が小さいと早くにキャッシュにフェッチされてしまう。このため、規模の大きいデータベースの処理ではSIDの番号が小さいILP-listはキャッシュから外れやすくキャッシュミスが増加する。

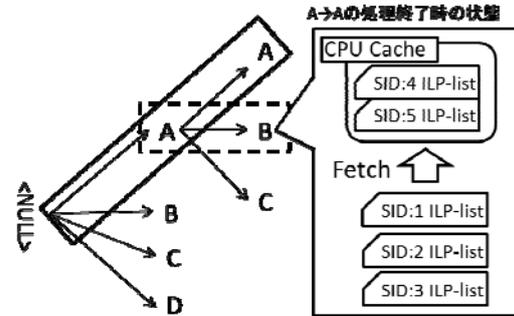


図2: PAIDにおけるキャッシュの利用

ILP-list へのアクセスを考えると、長さ $k-1$ の時系列パターンの共通接頭辞を含む長さ $k$ の時系列パターンが複数存在する場合、それらの間では同じILP-listへのアクセスが行われる。すなわち空間的参照局所性が存在する。PAIDでは長さ $k$ のパターンの一つを発見した後、そのパターンを接頭辞に長さ $k+1$ のパターンを発見するが、もし長さ $k-1$ の共通接頭辞となる複数の長さ $k$ の時系列パターンを幅優先で探索すれば、ILP-listへのアクセスに時間的参照局所性が生じる(図3)。つまり、共通接頭辞を含む複数の長さ $k$ の時系列パターンを一括して処理することにより、空間的参照局所性だけでなく時間的参照局所性を向上させることが可能なる。これに基づく手法をCache-Conscious PAID (CC-PAID)と呼ぶことにする。

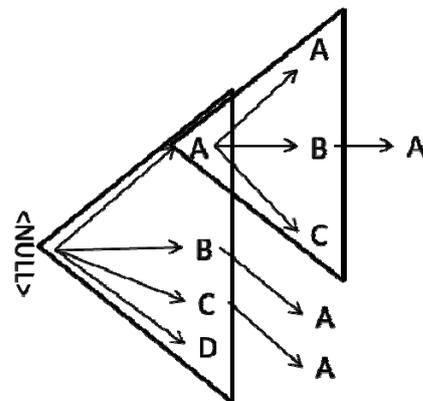


図3: CC-PAIDにおける幅優先探索

提案手法であるCC-PAIDを実装し、実験を

通してもとのPAIDとの比較を行った。実験に利用したプロセッサは、6 コアのIntel Core i7 980Xである。データは、IBMのデータジェネレータを利用し、時系列データ数 50000 のデータセットを作成して利用した。

図 4 は、コア数 1 のときのCC-PAIDとPAIDの実行時間(折れ線グラフ)と、L3 キャッシュのミス回数(棒グラフ)を示している。グラフから分かるように、CC-PAIDはPAIDに対して、キャッシュミスが最大 61%の削減、実行速度は約 2 倍となった。

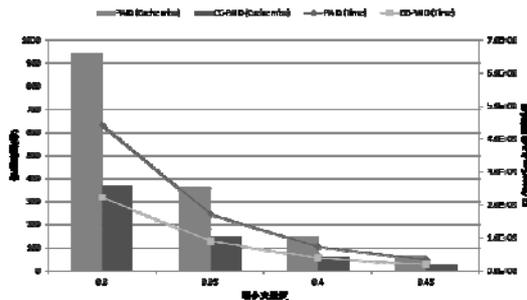


図 4: 実行時間とキャッシュミスの関係

次に、並列処理の効果を確かめるために、スレッド数(コア数)を1から6まで変化させ、スケーラビリティの検証を行った(図 5)。なお、CC-PAIDは並列化を前提に設計されているが、PAIDに関しても同様に並列化を行っている。

図 5 から明らかなように、スレッド数を増加させるに従い、実行時間が減少することが分かる。スレッド数 6 の時、スレッド数 1 の場合と比べて、CC-PAID、PAIDともに約 4.3 倍の高速化を実現している。一方、CC-PAID はPAIDに比べて、同時実行スレッド数に関係なく約 2 倍高速であることが分かる。

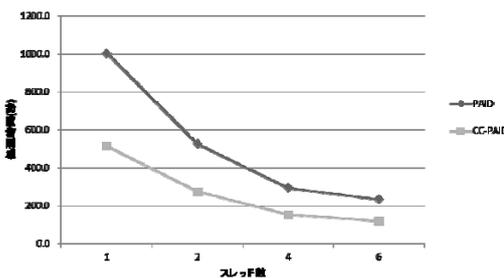


図 5: コア数に対するスケーラビリティ

以上から、CC-PAIDは処理のキャッシュ指向化による高速化だけでなく、並列化によってもさらに高速化されることを明らかにした。このことから、マルチコアプロセッサ時代向けのデータベース処理アルゴリズムの一つであると結論づけることができる。

今後は、CC-PAIDアルゴリズムのさらなる最適化、例えばデータ構造の動的な再構成により、よりキャッシュの効果が得られるような手法を研究する必要がある。さらには、意志決定支援などの大規模データベース処理について、マルチコアプロセッサ向けのキャッシュ指向アルゴリズムの研究を行う予定である。

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 2 件)

- ① 松原裕貴, 宮崎純, 藤澤誠, 天野敏之, 加藤博一: CC-PAID: CPUキャッシュを有効利用した並列時系列パターンマイニングアルゴリズム, 情報処理学会論文誌データベース, Vol. 4, No. 2, 2011, 査読有 (In Press)
- ② 油井誠, 宮崎純, 植村俊亮, 加藤博一, 山名早人: ロックフリーGCLOCKページ置換アルゴリズム, 情報処理学会論文誌データベース, Vol. 2, No. 4, pp. 32-48, 2009, 査読有

[学会発表] (計 5 件)

- ① 松原裕貴, 宮崎純, 藤澤誠, 天野敏之, 加藤博一: CC-PAIDにおけるデータベースサイズとCPUキャッシュ利用率の関係について, 第3回データ工学と情報マネジメントに関するフォーラム (DEIM2011), 修善寺, 2011. 2. 28, 査読無
- ② 福澤優, 宮崎純, 藤澤誠, 天野敏之, 加藤博一: カラムストアデータベースの処理性能と電力の関係について, 第3回データ工学と情報マネジメントに関するフォーラム (DEIM2011), 修善寺, 2011. 2. 28, 査読無
- ③ 松原裕貴, 宮崎純, 加藤博一: PUキャッシュを有効利用した並列時系列パターンマイニングアルゴリズム Cache-conscious parallel PAIDの提案, 情報処理学会データベースシステム研究会, 東京, 2010. 11. 12, 査読無
- ④ Jun Miyazaki: Database Buffer Management on Multicore Processors, The First Japan-China Workshop on Database and Digital Media Technologies in the Cloud, Dalian, China, 2010. 8. 23, 査読無
- ⑤ Makoto Yui, Jun Miyazaki, Shunsuke Uemura, Hayato Yamana, A Non-blocking Buffer Management based on the Generalized CLOCK, 査読有, Proc. of

IEEE 26th International Conference on  
Data Engineering (ICDE 2010),  
pp. 745-756, Long Beach, USA, 2010. 3. 4,  
査読有

[その他]  
ホームページ等

## 6. 研究組織

### (1) 研究代表者

宮崎 純 (MIYAZAKI JUN)

奈良先端科学技術大学院大学・情報科学研究科・准教授

研究者番号：40293394

### (2) 研究分担者

( )

研究者番号：

### (3) 連携研究者

( )

研究者番号：