卒業研究報告書

題目

並列処理によるデータベース

指 導 教 員

石水 隆 助教

報告者

04-1-47-175

三宅健太

近畿大学理工学部情報学科

平成 21 年 1 月 31 日提出

概要

膨大な量のデータから成るテーブルに対し検索し、1 つの応答時間が非常に大きなものの場合がある。その原因には SQL 文の文法が悪い、あるいはインデックスの張り方が悪いなどデータがきちんとそれぞれのテーブルに割り振られていない場合や、そもそもデータサイズが巨大すぎて、1 台のマシン自体でクエリを行うこと自体に限界がある場合などがある。後者の原因に対しては、データを複数のプロセッサに均等に割り振り、SQL 文の効率を上げるパラレルクエリと呼ばれる手法により、解決できる。これは、複数のマイクロプロセッサなどに処理を分散して割り当て、同時に計算・処理を行ない、システム全体のパフォーマンスを向上させるという技術である。このパラレルクエリを使用して、この問題を解決していく。

本研究ではデータベースのパラレルクエリに必要な環境(MySQL、XAMPP、TeraPad)をインストールし、これらの機能を使用し、膨大なレコードからなるテーブルへのクエリを、データベースによる「通常のクエリ」とパラレルコードによる「パラレルクエリ」の 2 通り行う。

本研究では、パラレルクエリの有用性を検証するためにパラレルクエリを使用可能なデータベース環境を構築し、膨大なレコードからなるテーブルの検索を通常の逐次クエリに対してパラレルクエリの時間短縮が得られるかどうかを測定する。

目次

1 序論

1.1 本研究の背景

膨大な量のデータから成るデータベースを検索する場合、1 つの応答時間が非常に大きなものとなる場合がある。その原因には、SQL 文の効率が悪い、あるいはインデックスの張り方が悪いなど、パフォーマンスチューニングの基本動作が出来ていない場合や、そもそもデータベースのサイズが 1 台の計算機で検索するには大きすぎる場合がある。後者の原因に対しては、複数のプロセッサに検索処理を分担するパラレルクエリと呼ばれる手法により解決できる場合がある。

1.2 データベース

データベース(DB)は、特定のテーマに沿ったデータを集めて管理し、容易に検索・抽出などの再利用をできるようにしたものである。データベースには、 OS が提供するファイルシステム上に直接構築されるものや、データベースをコンピュータ上で管理するためのデータベース管理システム(DBMS)を用いて構築されるものが含まれる。

単純なファイルシステムには、ファイルシステム自体にデータを統一的手法で操作する機能はない。このため、ファイルシステムでデータ管理をするためには、データの操作機能を「応用プログラム側」に持つしかない。応用プログラムがデータ操作を行なうためには、データ操作前に対象となるファイルの物理的格納状態を調査し、また物理的格納状態に変更があった場合、随時対応していかなければならない。データベースは、データ操作機能を自ら持つことにより、応用プログラム側でデータの物理的格納状態を知らずとも操作でき、かつ、データの物理的格納状態に変更があった場合にも応用プログラム側の処理に影響が及ばないことを保障することができる。データの物理的格納状態が応用プログラムの処理に影響しないことをデータとプログラムの独立だと呼ばれ、データベースの前提条件となっている。これをプログラムとデータの独立性という。

1.3 並列処理(パラレルクエリ)

パラレルクエリとは、複数のマイクロプロッセッサなどに処理を分散して割り当て、同時に計算・処理を行なうことで、システム全体のパフォーマンスを向上させる技術である。また、そのような環境を効率的に活用するためのソフトウェアやプログラミングの手法の総称である。

今日の計算機では、1 台の計算機にマイクロプロセッサを複数搭載するマルチプロセッサやプロセッサ内部に複数の処理装置を実装するマルチコアなどが採用されており、1 台の計算機でも一部の処理は並列に行なうことができる。また、複数の計算機をネットワーク接続し、計算機全体を 1 台の並列計算機とする。クラス処理なども現在注目されている。このため、パラレルクエリも様々な環境で使用できるようになった。

1.4 本研究の目的

本研究では、パラレルクエリの有用性を検証するために、高負荷なクエリを並列化することによって、クエリの処理速度の向上、クエリの処理のパフォーマンスの向上が、どの程度得られるかを測定する。検証方法としては、サイズの大きいテーブルを２つ以上のテーブルに分割して、各テーブルへの負荷を均等に分割することによって、得られる処理時間が１つのテーブルに対して逐次処理を行なった場合と比べて、どの程度短縮されるかを測定する。

1.5 本報告書の構成

以下に本報告書の構成を述べる。

2 章では、本研究を行なうための準備として、パラレルクエリを行なえるデータベース環境の構築の仕方について述べ、3 章では本研究の内容として、研究の検証手順について述べる。4 章ではその検証によって、得られた結果及び考察を行なう。5 章では、まとめ及び今後の課題について述べる。

2 準備

本章では、パラレルクエリを行なえるデータベース環境の構築の仕方について説明する。

Windows の標準環境にはデータベースは付属していないため、データベース環境を構築するためにはいくつかのソフトウェアをダウンロードし、インストールしなければならない。本研究では、XAMPP、MySQL、TeraPad、の３つをインストールした。以下に上記の３つのソフトウェアについて述べる。

2.1 XAMPP

XAMPP[5]とは、Web アプリケーションの実行に必要なフリーソフトウェアをパッケージとしてまとめたもので、apachefriends.org から提供されている。主として開発用あるいは学習用ではあるが、イントラネットなどにおいて実運用環境として使われることもある。

Apache(Web サーバ),MySQL(SQL データベースサーバ)と Web プログラミング言語である PHP や同目的で使われる Perl の 4 つの主要ソフトウェアと phpMyAdmin などの管理ツール、さらに SQLite など、いくつかの補助的なソフトウェアが含まれている。名前の 2 文字目以降の AMPP は主要 4 ソフトウェアの頭文字である元々は対応 OS は Linux のみであり、その頭文字 L を付け LAMPP と称したが、後に複数の OS に対応したため L を X に変え XAMPP となった。現在、Windows、Linux、Mac OS X、Solaris で利用可能である。

本来、前述の複数のソフトウェアを個別にインストールする必要があり、非常に手間がかかるが、XAMPP は一括してインストールするだけで、すぐに開発や運用が開始できる。PHP でのプログラミングを始めるユーザの多くが、XAMPP を使用している。パッケージとしての特性上、個々のソフトウェアのバージョンが必ずしも最新版で揃えられてはいないが、開発用・学習用としては十分である。

2.2 MySQL

MySQL[1]は、RDBMS(リレーショナルデータベース管理システム)の実装の 1 つである。オープンソースで開発されており、世界で最も有名なオープンソース・データベースとして知られている。他のフリーRDBMS と比較して高速性に定評があり、特に更新よりも参照の頻度の高いアプリケーションに向くとされている。具体的には Web アプリケーションの多くが該当する。データストレージエンジンは SQL エンジンとは分離独立しており、用途に応じた機能を持つストレージエンジンを選択できるマルチストレージエンジン方式となっている。

2.3 TeraPad

TeraPad[7]は、SDI タイプのテキストエディタである。ファイルの種類に応じた色分け表示機能や、外部ツールの呼び出し機能を持つ。XAMPP 上で動く PHP は初期設定で EUC-JP 文字コードで動くようになっている。

3 研究内容

本章では、本研究で行なったパラレルクエリの検証方法について、説明する。

本研究では、サイズの大きいテーブルへのクエリを MySQL による通常の逐次クエリと PHP で記述したパラレルコードによるパラレルクエリの 2 つを実行し、それぞれ処理時間を測定する。

3.1 検証の手順

本研究で使用するデータは、300 万レコードから成るテーブルであるパラレルクエリに使用するために、本研究ではパラレルクエリに先立ち、このテーブルを 2～10 個に分割する。

以下に本研究の手順を述べる。

[1]データの分割

入力データを各クエリごとに分割する。本研究では、レコード数が均等になるように 2～10 個のデータに分割した。

[2]データの挿入

上記 1 で分割したデータを各テーブル(T_1-----Ti(2<=i<=10))に挿入する。また、通常の逐次クエリと比較するために、テーブルへのデータ挿入後、テーブル全体をマージして比較用のテーブル T_0 を作成する。

[3]MySQL による各テーブルへのクエリ処理

各テーブルそれぞれに対して MySQL でクエリを行なう。

[4]パラレルコードによる各テーブルへのクエリ

各テーブルにそれぞれに対して PHP によって記述したパラレルコードを使用して、クエリを行なう。並列処理の主要な部分であって、各テーブルに同時に処理を行い、結果を作成した一時ファイルに追記し、プロセスの状態の管理を行なう。

[5]結果をまとめる

各テーブルの処理がすべて完了しているかをチェックし、作成された一時ファイルの内容を合計し、各テーブルの処理速度の結果を表示する。

3 結果・考察

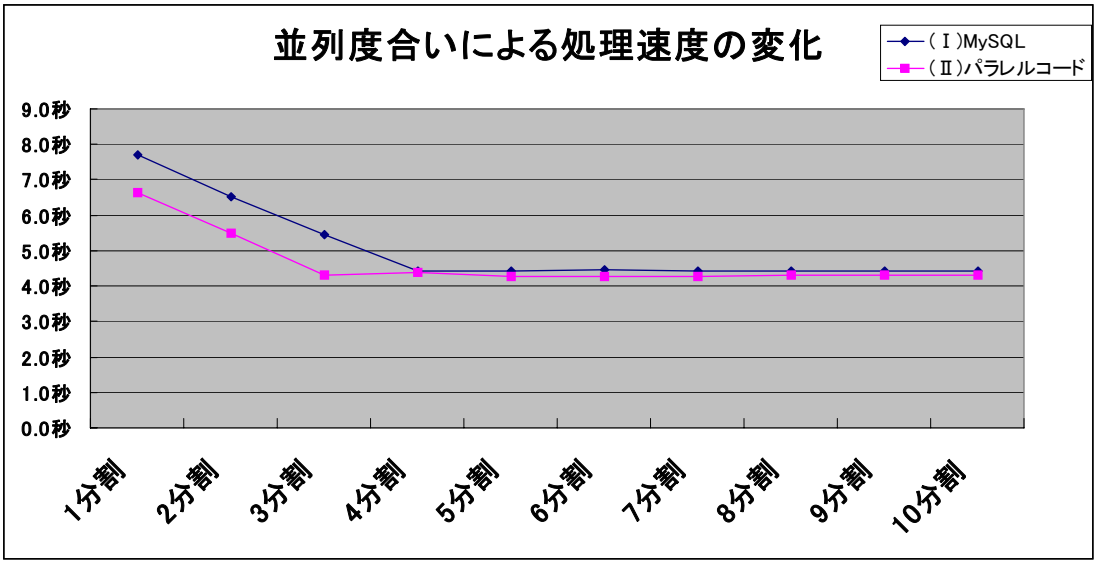

図.1　各分割数でのクエリの処理時間

本章では、本研究で得られた結果について述べ、その考察を行なう。
図 1 に逐次クエリおよび分割数を 2～10 分割に変化させた場合の各パラレルクエリの処理時間を示す。また、その詳細を付録 3 および付録 4 に示す。

4.1 MySQL による逐次のクエリの結果

図 1 および付録 3 より、逐次クエリにおいてテーブルを 2 分割してクエリを行うと、分割しなかった場合に比べて 1.5 倍ほどパフォーマンスが改善されることが示される。3,4 分割でも同様に 1.6～2 倍程度のパフォーマンスの向上が示された。しかし、分割数を 5 以上にしても 4 分割した場合に比べて処理時間の短縮は得られなかった。

4.2 パラレルコードによるパラレルクエリの結果

図 1 および付録 4 より、パラレルクエリにおいても、テーブル 2,3 分割してクエリを行うと、分別しなかった場合に比べて、1.3～1.5 倍程度のパフォーマンスの向上が得られることが示される。しかし、分割数を 4 以上にしても 3 分割した場合と比べて、処理時間の向上は得られなかった。

4.3 双方のクエリの比較の結果

図 1 より逐次クエリとパラレルクエリを比較した場合、テーブルの分割数が 1 ～3 分割のときは、逐次クエリに対してパラレルクエリが 1.5 倍程度高いパフォーマンスが行われることが示される。しかし、4 分割以上のときは、逐次クエリとパラレルクエリとの差は見られない。

4.4 考察

4.1～4.3 結果より、逐次クエリ、パラレルクエリ共にテーブルを分割することで、パフォーマンスの向上が得られ、また逐次クエリよりもパラレルクエリのほうがより高いパフォーマンスが得られることが示された。これは、テーブルを分割することより、個々のステップを並列に処理することが出来、そのため全体の処理速度が向上したと考えられる。しかし、テーブルの分割数がある一定の範囲を超えると、それ以上テーブルの分割数を増やしてもパフォーマンスの向上は得られない。これは、本研究で作った逐次クエリ、パラレルクエリは 1 台の計算機を用いての処理であるため、並列処理可能な処理数に限界がある

ためと考えられる。例えば、本研究の計算機の環境では並列度４が限界であった。

4 結論・今後の課題

本研究では、テーブルを分割して逐次クエリ、パラレルクエリを行ったときに、どの程度パフォーマンスの向上が得られるかの検証だった。

本研究より、データベースでクエリ処理を行う際には、テーブルを分割して、各テーブルを並列処理することにより､逐次クエリ､パラレルクエリ共にパフォーマンスの向上が得られることが示された｡また、逐次クエリよりもパラレルクエリのほうがより高いパフォーマンスが得られる。しかし、本研究で作った逐次クエリ、パラレルクエリは 1 台の計算機を用いての処理であるため、並列処理可能な処理数に限界が見られた。

1 台のマシンで、パラレルクエリのパフォーマンスを向上させていくには、複数の行を一気に挿入するマルチプル INSERT による方法や集約関数を使用する方法などがある。この手法を用いた場合にどの程度パフォーマンスが向上するのかを検証していくのが、今後の課題である。

また、本編ではクエリの重さから、300 万レコードのデータを一気にデータベースのテーブルに挿入することが出来ず、数回に分けて挿入しなければならなかった。よってデータ挿入部分を自動化し、ユーザがデータベースを使用する手間を省くことも今後の課題である。

5 謝辞

石水先生には、お子様も生まれたばかりで、仕事にもご家庭にも非常に忙しい時期にもかかわらず、遅い時間までご指導していただいたことに、深く感謝しています。本当にありがとうございました。

6 参考文献

[1] http://tomo.ac/goodstream/database/mysql/basic.htm //MySQL の基本操作.

[2] http://www.phppro.jp/school/phpschool/ //PHP,基礎講座

[3] http://www.thinkit.co.jp // Software Developer's Think IT

[4] http://us3.php.net/ //PHP：Manual

[5] http://phpspot.net/php/pgXAMPP.html //XAMPP：開発環境,php spot

[6] http://allabout.co.jp/internet/database/ //All About

[7] http://www5f.biglobe.ne.jp/~t-susumu/library/tpad.html //TeraPad