6U-04

WWWによるDW処理の動的分散化

天野 龍明　　福岡 聡　　新居 佳守臣　　三浦 孝夫

法政大学工学部電気電子工学科

1 まえがき

本研究ではWWW上でのData Warehouse (DW) 処理をサーバとクライアントで動的に分散処理する方法について述べる。従来のDWのシステム構成には次のような特徴がある：

(1) 利用者がほしいデータに対してサーバが働き、そのデータを返してくれる。
(2) データ量が膨大になるにつれてサーバやネットワークへの負荷が増大する。
(3) クライアント側での処理速度の低下がおきる。

このような問題に対応するため、本稿ではDWにシステムの分散方式を提案する。すなわち、サーバとクライアントにわけ、負荷を軽減させる方法により、サーバ側ではデータの取得を、クライアント側ではデータの表示と操作を行う。他方、分散化したDWにおいて必要なことはサーバ側はいかに効率よくデータを転送し、負荷を軽減させるかにある。例えば、従来のDWにおいて［“時間”730日“商品名”3000品“店名”300店］の3次元で表現されるデータをスライシングする場合、その検索結果のデータ量は膨大なものとなり、サーバの処理が著しく低下する。また、スライシングするたびにサーバにデータを要求しているため、同じデータを何重にも取得していることになる。クライアント側の問題は、どの様にサーバ側からデータ情報をもらい受け、そして取得したデータを操作し、ブラウザ上で表現するかである。

2 DWと多次元データベース

DWは、基幹系のデータベースからデータを抽出、変換、統合し、物理的に多次元データベース (MDDB) として構築する。データは基本的に詳細データとそれから引き出される要約データの2種類とする。MDDBはサマリ化されたデータでその処理にオンライン分析プログラム (OLAP) を用いる。OLAPは利用者がMDDBをデータ分析操作する処理である。本実験ではOLAPを次のように定義する。

(1) 論理的な多次元ビューを持つこと。
(2) どのような検索に対してもパフォーマンス（レスポンス）が均一であること。
(3) 直感的なデータ操作ができること。

図1でキューブ（立方体）の形に表現した「データ・キューブ」は多次元ビューを意味する。RDBのテーブルが3次元の立体になったものである。利用者は，このキューブを「スライシング」、「ダイシング」、「ロールアップ・ドリルダウン」しながら多角的に分析する。それぞれの操作について次のように行う。

- スライシング
  キューブをさまざまな次元軸の組み合わせでデータを切り取り、その断面を見ることである。例えば、1月2月3月…と次々にスライスする事により、時系列的な推移を把握する事ができる。また、スライスすることによりN次元データは一つ次元が下げられN-1次元の表となる。
- ダイシング
  分析の視点（スライスする角度）を切り替えることである。クライアントの画面上では，X軸、Y軸、Z軸にとるデータ系列を入れ替えることに相当する。
- ロールアップ・ドリルダウン
  94/01/01～99/12/31→94/01～99/12→'94～'99といった集合レベルを増やしていく、つまりサマリ化する操作である。合計 (sum) や平均 (avg)、最大 (max) 最小 (min) 値などの値が取れる。本研究では合計 (sum) をとることにする。ドリルダウンはこの逆操作である。

本研究ではRDBMS、MDDBおよびOLAPをDWと定義する。

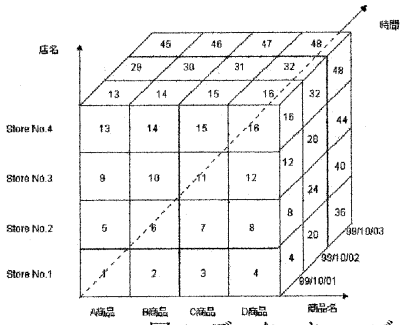

図 1: データ・キューブ

3 動的スライシングとOLAP操作

3.1 サーバ設計

サーバの負荷やネットワークへの負荷を軽減させるために動的スライシングを考える。
動的スライシングは以下の特徴を持つ。

(1) データを取得する状況に応じて動的にデータを分割する。
(2) すでに取得済みのデータを利用する。

また、動的スライシングは以下の場合に有効である。

(1) スライスするデータ量が多いほど効果がある。
(2) 通信速度が遅いほど効果がある。
(3) クライアント性能が低いほど効果がある。

さらに分割する上で表示範囲を動的に変化させることでより柔軟な検索が可能となる。

Distributing DataWarehouse Processing by WWW
Amano Tatsuaki, Fukuoka Satoshi , Arai Kazuo and Miura Takao
Hosei University, Dept.of Elec. and Elec. Eng., Kajino-cho 3-7-2, Koganei, Tokyo, JAPAN

例 1 店名 [NO.1 店, NO.2 店, NO.3 店, NO.4 店]、商品名 [A 商品, B 商品, C 商品, D 商品]、時間 [99/10/01, 99/10/02, 99/10/03] で構成された 3 次元のデータを時間 [99/10/01] でスライス (検索) した場合、結果が商品名 [A 商品, B 商品, C 商品, D 商品] ×店名 [NO.1 店, NO.2 店, NO.3 店,NO.4 店] の 16 件となる。これを件数で分割すると表の意味を失ってしまうため、商品名 [A 商品, B 商品, C 商品, D 商品] ×店名 [NO.1 店, NO.2 店] と商品名 [A 商品, B 商品, C 商品, D 商品] ×店名 [NO.3 店, NO.4 店] の二つに分割する。

	A 商品	B 商品	C 商品	D 商品
NO.1 店	1	2	3	4
NO.2 店	5	6	7	8

	A 商品	B 商品	C 商品	D 商品
NO.3 店	9	10	11	12
NO.4 店	13	14	15	16

つづいて商品名 [D 商品] でスライスする。すると店名 [NO.1 店, NO.2 店, NO.3 店, NO.4 店] ×時間 [99/10/01, 99/10/02, 99/10/03] が結果として得られる。しかし、D 商品-99/10/01(4,8,12,16) は検索済みなので、この部分以外を検索対象とし、取得済みデータと結合する。

3.2 クライアント設計

通常サーバではデータは関係モデルを用いて 1 次元的に表現される。しかし、今回の実験で用いたサーバ (postgreSQL) にはクラス階層を直接支援するものではないので、OLAP 操作 (ロールアップ、ドリルダウン) が不可能である。よって、利用者にとってデータベースでなく DW を操作しているように見せる必要がある。アイデアとしてデータをサーバから読むときにブラウザ上に HTML 化させ,JavaScrip を用いてクライアント側でダイシング、ロールアップ、ドリルダウンを行わせる。こうする事により、利用者は DW を操作していることになる。サーバからのデータ量は膨大であるがそれは動的スライシングをする事で回避できる。以下、表示方法について述べる

例 2 3 次元のデータ情報を考えた場合ブラウザ上に 2 次元の表 1 のようにインスタンスに 3 次元データ属性の 1 つを書くことで表現できる。
3 次元のデータの属性が 2 つ重なった場合、インスタンスに属性を 2 つを書くことで表現できる。3 次元以上も同じく行う。

- *ダイシング*
  表 1 の時間次元と店名次元を変えることにより表現する。
- ロールアップ・ドリルダウン
  店名次元においてNO.1、NO.2 店は東京都にあり、NO.3 は千葉にあるという情報を持つとする。表 2 に表 1 のロールアップを示す。ドリルダウンはこの逆操作である。

表 1: 3 次元データの 2 次元表現

	99/10/1	99/10/2	99/10/3	99/10/4
A 商品	NO.1 店 34.1			
B 商品		NO.2 店 75.1 NO.3 店 54.7	NO.3 店 98.3	
C 商品				NO.3 店 13.2

表 2: 表 1 における店名によるロールアップ

	99/10/1	99/10/2	99/10/13	99/10/4
A 商品	東京 109.2			
B 商品		東京 109.2 千葉 166.2	千葉 166.2	
C 商品				千葉 166.2

4 システムの実現

4.1 サーバ

本実験システムでは、Pentium-II 550 MHz FreeBSD 2.8 をサーバとし、ここで PostgreSQL および PHP を稼動する環境を用いる。これを用いて動的にデータ取得量を変化させる。MDDB は各次元テーブルとファクトテーブルで構成される。各次元テーブルは正規化されたファクトテーブルに接続するただ 1 つの結合を持っている。サーバはクライアントの要求に対して、MDDB のスキーマを解析しロールアップ情報をクライアントに返す。同時に動的スライシングのためにセッションを生成する。セッションはデータの分割のためロールアップテーブルのスキーマを解析し、動的に変化する。検索結果は配列で自動的に保存される。サーバは保存されたデータを要求に結合できるか解析する。

4.2 クライアント

クライアント側の環境として MS-Windows 上に WWW ブラウザを稼動させ、OLAP 操作 (ダイシング、ロールアップ、ドリルダウン) を行うために JavaScript を使用する。クライアントはサーバで保存された配列 data[X,Y,Z]= α を受け取り操作する。例えば、3 次元のデータ情報、data[“店名”“商品名”“時間”] =売上という配列がある。ダイシングは配列のデータ情報の順番を変えることで操作できる。ロールアップ・ドリルダウンは配列 data[“NO.1 店”“A 商品”“99/10/1”“東京都”“テニス用品”“第 1 週”“スポーツ用品”“関東地方”“10 月”] =売上の情報がら操作する。動的スライスをされたデータ情報以外で利用者が要求するならばサーバへデータを取りに行くことになる。
まとめると次のようなことになる。

(1) プログラムの生成はサーバレベルで行う。
(2) サーバより受け取るデータは変数である。
(3) 不足データは明示的にサーバへと取りに行く。

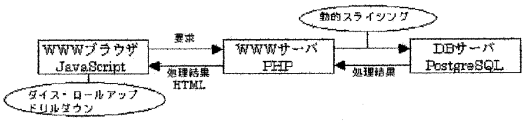

図 2: システム構成

5 むすび

WWW 上で Data Warehouse (DW) 処理をサーバとクライアントで動的に分散処理することにより、サーバの負荷やネットワークへの負荷を軽減し、柔軟な検索が可能となった。

参考文献

[1] Chaudhuri,S.: An Overview of DataWarehousing and OLAP Technology, SIGMOD Record, Vol.26, No.1, March 1997

[2] 石井義興: データウェアハウス (1996) 日本経営科学研究所

[3] Inmon,I.H.: データウェアハウス構築・活用・運用編 (1998) オーム社

[4] Jarke,M. et al: Fundamentals of DataWarehouses (1998), Springer-Verlag