

高速 ORDBMS LiteObject の設計と実装*

4 A a - 4

岡田 敏 鬼塚 真 小林 伸幸 小西 史和†

NTT 情報通信研究所‡

1 はじめに

近年、データベースは画像等のマルチメディアの分野でも使用されるようになってきている。これらの分野では、例えば画像検索においては画像同士の類似度で検索を行うなどデータの特性に合わせた検索が必要となる [1]。

これらの要求に対応する DBMS として、オブジェクトリレーショナル DBMS [2] (ORDBMS) が注目されている。ORDBMS では、データの特性に応じた処理をユーザ定義メソッド機能として提供している。そして、我々はデータベースへの高速アクセスを実現するためメモリデータベースアーキテクチャを使用した ORDBMS "LiteObject" を開発している。

LiteObject ではユーザ作成のプログラムをユーザ定義メソッド機能として、DBMS に組み込み、高速に実行する方法を提供している。

本稿では、LiteObject におけるユーザ定義メソッドの実現方法について述べる。

2 従来技術の問題点

従来の ORDBMS ではユーザ定義メソッドの実行に関して以下のような問題点があった。

DBMS とユーザ定義メソッドのリンク方法：

DBMS とユーザ定義メソッドは動的リンクによって関連付けられていた。リンクを張られたメソッドはメソッドの起動時に DBMS にロードされ、実行されるため、起動速度が遅いという問題があった。

アクセスバスの最適化： 従来の DBMS では、ユーザ定義メソッド実行時の I/O コスト、演算量等をユーザが入力し、この情報を基にアクセスメソッドの最適化を行っていた [3]。この方法ではアクセスメソッドの最適化のために DBMS が行う処理が増加するため、データへのアクセス速度が低下してしまうという問題があった。

次節ではこれらの問題に対する LiteObject での実装法について述べる。

3 ユーザ定義メソッド

LiteObject ではユーザ定義メソッドの機能を以下の 3 項目に分けて実現している。

ユーザ定義関数： データの持つ特徴に合わせてユーザが独自に作成した関数である。

ユーザ定義オペレータ： データの持つ特徴に合わせてユーザが独自に作成した二項演算子である。

ユーザ定義インデックス： ユーザが独自に作成したインデックスであり、データに適したアクセス方法を提供する。

以下にこれらのユーザ定義メソッドの管理法、および使用法について述べる。

3.1 DBMS への登録

ユーザ定義メソッドは DBMS の機能としてユーザに提供するため、起動前に DBMS に登録しなくてはならない。

ユーザ定義メソッドを DBMS に登録する処理の順序を以下に示す。

1. ユーザによる関数 (オブジェクトファイル) の作成。
2. ユーザによるインデックス情報ファイルの作成。本ファイルはユーザ定義インデックス使用時に参照する情報を保持する。主な内容はインデックス名、インデックスが使用できるデータ型等を含んだインデックスの適用条件等である。
3. DBMS とユーザ定義メソッドとの間に静的リンクを形成する。

LiteObject ではユーザ定義メソッドを静的リンクによって組み込んでいる。

静的リンクの採用により、DBMS 起動時にはユーザ定義メソッドも実行可能な状態となるため、従来の動的リンクと比較して、ユーザ定義メソッドの起動を高速に行うことが可能となる。

3.2 ユーザ定義メソッドの利用

ユーザ定義メソッドを利用するための SQL 文例と処理内容を以下に示す。本例では画像データを扱う "picture" クラスを使用する。このクラスに関数 "similarity" とオペレータ "similar" を追加し、クラス内の属性 "image" に対しインデックスを付与する処理を示す。なお、インデックス情報ファイルにはインデックスとして "treeindex" が登録済みとする。

ユーザ定義関数の追加：

```
ALTER CLASS picture ADD METHOD similarity
FILE similarity.o;
```

関数名 similarity と DBMS に登録済の実行ファイル (similarity.o) との対応関係を生成する。

ユーザ定義オペレータの追加：

```
ALTER CLASS picture ADD OPERATER similar
FILE similar.o;
```

オペレータ名 similar と DBMS に登録済の実行ファイル (similar.o) との対応関係を生成する。

*Design and implementation of high-speed ORDBMS "LiteObject"

†Satoshi OKADA, Makoto ONIZUKA, Nobuyuki KOBAYASHI, Fumikazu KONISHI

‡NTT Information and Communication Systems Laboratories

ユーザ定義インデックスの生成：

```
CREATE INDEX idx1 ON
picture(image) AS treeindex;
```

インデックスの生成は以下の処理を実行する。

1. インデックス名称と共に DBMS に登録されているインデックス構築、およびデータアクセスのための実行ファイルとの対応関係を生成する。
2. インデックス構築のための手続き (図 1 "tree_build") を実行し、インデックスを構築する。

生成処理後の各機能の位置付けを図 1 に示す。

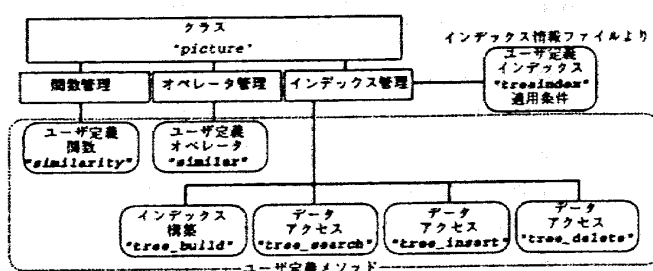


図 1: 各機能の相関関係

3.3 ユーザ定義メソッドの実行

図 2 にユーザ定義メソッドを使用した検索の例を示す。本例では画像データからなるクラス "picture" を使用する。picture には属性として画像の名称データを格納する "name"、画像の特徴データを格納する "image" 等を持っている。

本例は picture クラスからオペレータ "similar" を使用し、"apple" という画像に似た画像データの名称と類似度を示す関数 "similarity" の演算結果を返却するものである。

以下では本例にしたがって処理の流れを説明する。

```
select name, similarity(image, "apple")
from picture
where image similar "apple"
```

図 2: 検索例

3.3.1 アクセスパスの決定

SQL 文を解析し、データへのアクセスパスを決定する。

1. from 句からアクセスするクラス picture を特定する。
2. クラスの特定により、使用できるユーザ定義メソッド (図 1 のクラス picture と関連付けられたメソッド) を絞り込む。
3. オペレータ similar とその両辺 image, apple を取り出す。
4. オペレータとその両辺および、インデックス情報ファイルよりデータへのアクセスメソッドを決定する。

• インデックス情報ファイル中の適用条件に合致すればユーザ定義インデックス経由のデータアクセスを実行する。

• インデックス情報ファイル中の適用条件に合致しなければ全件検索を実行する。

3.3.2 インデックスを使用する場合

1. 検索用のデータアクセス手続きを起動し、インデックス経由のアクセスを行い、オペレータの検索条件に合致する複数のインスタンス群を得る。
2. データアクセスで得られた個々のインスタンスに対して画像の名称 (name) と類似度関数 similarity の結果を返却する。

• インデックスを使用しない場合

1. 属性 image のインスタンスに対して一件ずつオペレータ similar に対する評価を行う。

評価結果が "真" となる場合：

真となった image を含む画像の名称と similarity の結果を返却し、次のインスタンスの評価に進む。

評価結果が "偽" となる場合：

次のインスタンスの評価に進む。

以上の処理を全インスタンスに対して行う。

LiteObject では事前にインデックスの適用条件をインデックス情報ファイルに記述しており、クラスとインデックス適用条件からアクセスメソッドを決定することが可能である。本方式の導入により、LiteObject では従来多くの処理量を必要としていたアクセスメソッドの最適化処理の軽減が実現されている。そして、アクセスメソッドの最適化処理の軽減によりデータアクセスの高速化が実現できる。

4 まとめ

本稿ではユーザ定義メソッドの導入により、マルチメディアデータのような従来の DBMS では扱うことが困難であったデータを管理する ORDBMS "LiteObject" の実現方法について述べた。

LiteObject ではユーザ定義メソッドの静的リンク、アクセスメソッドの最適化処理の軽減を実現しており、処理速度の高速化を実現している。

参考文献

- [1] 赤間 浩樹, 紺谷 精一, 三井 一能, 申間 和彦, "画像内オブジェクトの自動抽出を使った画像検索システム - ExSight -," 信学会, 第 8 回データ工学ワークショップ, 1997.
- [2] 太田 佳伸 訳, Michael Stonebraker, "オブジェクトリレーショナル DBMSs," インターナショナル・トムソン・パブリッシング・ジャパン, 1996.
- [3] Surajit Chaudhuri, Luis Gravano "Optimizing Queries over Multimedia Repositories," ACM SIGMOD, 1996.