

マルチメディアデータベース検索言語 MMQL における機能 拡張機構

家富誠敏 河信司 富井尚志 有澤博

横浜国立大学 工学部 電子情報工学科
〒240 横浜市保土ヶ谷区常盤台 79-5

E-mail: {eto,kawa,tommy,arisawa}@arislab.dnj.ynu.ac.jp

あらまし

画像等のメディアに対して処理を行なうためには、特殊なハードウェアや画像処理等に優れた特性を持つコンピュータなどを必要とする場合がある。

さまざまなメディアを扱うマルチメディアデータベースでは、メディア毎のさまざまな処理を行なうために異なる特性を持つコンピュータ等を統合し、扱うメディアとコンピュータのもつ能力に応じて、処理を分散させるような仕組みが必要となる。

本論文ではこのようなデータベースのマネージメントシステムとして、新たなメディア(処理)が組み込まれた際など、処理の分散を巧妙に行なうことのできるシステム構成を紹介する。

キーワード：マルチメディアデータベース、機能分散、DBMS

Extention of Operators in MultiMedia-Query-Language MMQL

Masatoshi IETOMI, Shinji KAWA, Takashi TOMII, Hiroshi ARISAWA

Division of Electrical and Computer Engineering
Faculty of Engineering
Yokohama National University
79-5, Tokiwadai, Hodogaya-ku, Yokohama-shi, Japan

E-mail: {eto,kawa,tommy,arisawa}@arislab.dnj.ynu.ac.jp

Abstract

As multimedia database has to maintain various media, it should integrate computers with different characteristics, such as specialization in handling graphic data, audio data, etc.

This paper describes a database management system which joins together several different computers and decentralizes properly the query processing according to the maintained media and the abilities of the computers.

The system allows new operators to be included into MultiMedia-Query-Language MMQL in an easy and natural way.

Keywords: multimedia database, decentralize function, DBMS

1 はじめに

近年、データベース技術は飛躍的に進歩している。

従来のデータベースで扱われるデータといえば、数値、文字などのデータ型に限られていたが、現在では、それに加えて、テキスト、画像、音声、映像など様々なメディアにわたるデータの取り扱いについて提案がなされている。さらには三次元情報または3次元+時間軸情報をデータベースで扱おうという動きもあり、この先も新しいメディアを扱いたいという要求が出てくることは十分考えられる。

このような新たなメディアをデータベースで扱うためには、メディアをデータとして格納できるというだけでなく、そのメディア特有の操作(メディア依存処理)を行なえることが不可欠である。

メディア依存処理の例としては、数値メディアに対する数値演算(+、-、×、÷等)、画像メディアに対する画像処理(圧縮、合成等)など様々なものが考えられる。そして、近年の多種多様な分散計算機環境では例えば数値であれば数値演算が得意なマシン、画像であれば画像処理が得意なマシンというように、コンピュータごとにメディア(処理)の得手不得手がある。また、コンピュータによっては特殊なハードウェアを持つものもあり、これをデータベースシステムで用いたいという場合もあるだろう。

著者らによって提案されているリアルワールドデータベース[1]のように、森羅万象、ありとあらゆる情報を扱い、さまざまな処理を行なうためには、このような異なる特性をもつコンピュータを統合し、扱うメディアとコンピュータのもつ能力に応じて、処理を分散させるようなシステムが必要となる。従来の1つのマシンでデータベースに必要な全ての処理を行なうものに対して、このような種々のコンピュータ・特殊なハードウェアの統合によって複数マシンに機能を分散させたものをここでは機能分散型データベースシステムと呼ぶが、このような機能分散型データベースシステムを管理するDBMSの実現にはいくつかの問題点がある。一つにはデータベースシステムの各処理を行なうオペレータについての情報をどのようにして管理するかという問題である。そのオペレータがどのマシン上で動作し、どのようにして呼び出されるのか、どのような特性を持っているのか等、これら膨大な情

報をオペレータの数だけ全て管理するのは非常に難しい。また検索時において各処理をどのようにマネージメントするかという問題もある。これは処理の最適化などを考えると、各コンピュータの負荷を考慮し、巧妙に処理を分配することが必要となり、極めて困難である。このような問題を解決するためには、処理のマネージメントを行なうために非常に賢いアルゴリズムと高いスペックの演算能力を必要とし、現状でこれを実現するのはかなり難しいといえよう。

従って本論文ではこのような問題を全く違ったアプローチから解決する手法を提案する。

本手法ではシステムのマネージメントを各処理を実行するプロセス任せに行う。これにより分散されたさまざまなコンピュータにおける処理を非常に容易にマネージメントすることができ、新たな処理等を組み込む際にも、システム全体に対して複雑な更新動作を必要としないようにシステムを実現できる。

2章ではまずこのような機能分散型データベースマネージメントシステムの思想と概要を紹介し、3章では、データベース検索システムにおける実装例を示し、その特徴を述べる。

2 システムの構成

さまざまなメディアを扱うマルチメディアデータベースでは、異なる特性を持つコンピュータ(3次元画像処理等に優れたもの、映像専用ストレージを持ち映像の扱いに優れたものなど)を統合した、いわゆる機能分散型システムの扱いが要求される。

このような機能分散型データベースのマネージメントシステムを、ここではクライアント・サーバモデルに基づく構成で考える。

2.1 クライアント・サーバ

クライアント・サーバモデルは、操作の対象となる情報をサーバが管理し、これに対してクライアントが処理を行なうという機構である。

クライアントは複数存在することができ、それぞれのクライアントはそれぞれの処理を担当する(全く同じ処理を行なうクライアントが複数存在しても構わない)。

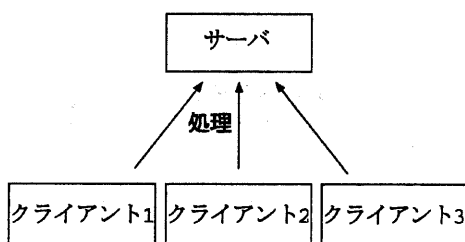


図 1: クライアント・サーバシステム

例えば、数値演算を行なうシステムを考えると、演算結果およびその途中結果をサーバが管理し、+、-、×、÷等の実際の演算をそれぞれのクライアント(+演算を行なうクライアント、-演算を行なうクライアント、×演算を行なうクライアント、÷演算を行なうクライアント等)が行なう。

サーバとクライアントの情報のやりとりはプロセス間通信で行なわれ、異なるコンピュータ上に存在するクライアントからでもネットワーク等を介してサーバで処理を行なうことができる。

2.2 クライアント・サーバの問題点

さて、何かの演算を実行するためには、処理の流れに従って必要なクライアントを呼び出す必要がある。先述した数値演算の例では、 $3 + 4 \times 5$ のような計算を行なうとき、まず×演算を行ない、次に+演算をするというように処理と処理の間には守られなければならない順序の制約というものがある。システムにはこのような処理の順序制約を管理するものが存在しなければならない。

また2つの同じ処理を行なうクライアントが異なるコンピュータ上に存在するならば、負荷の低い方に処理を発行するのが望ましい。従って同一な処理を行なうクライアントが複数存在したときなど、実際の処理をどのクライアントに割り振るかといったようなことも管理することが必要となる。

ここでは、このような機能を持つものを想定してマネージャと呼ぶことにする。

マネージャは、処理の流れに従い、必要に応じて最も適当なクライアントを呼び出すという操作を行なうが、これを行なうためには非常に優秀なアルゴリズムと高い計算能力が必要とされる。

またマネージャは、どのようなクライアントが存在するのか、そのクライアントをどのような書式で呼び出すのか等、クライアントに関する情報を保持していなければならないが、そのためにはクライアントの数だけこれらのデータを管理せねばならない。このようなマネージャの実現は完全に不可能ではないが、非常に困難が予想される。

2.3 リゾルバ(解決者)の概念

問題となるのは、マネージャのようにクライアントの情報を管理して、必要に応じてクライアントを呼び出すという存在があることである。

そこで、一つの解決策としてマネージャのようなものを設けず、処理を行なうタイミングをクライアント自身で判断するというシステムを考える。

このようなシステムでは、どのようなクライアントが存在しているか等、クライアントに関する情報を統括しているものが存在せず、またクライアントへの処理の分配を行なうものも存在しない。

このように、自分で問題を解決する(仕事を見つけて、処理を行なう)ようなクライアントをここではリゾルバ(Resolver:解決者)と呼ぶことにする。

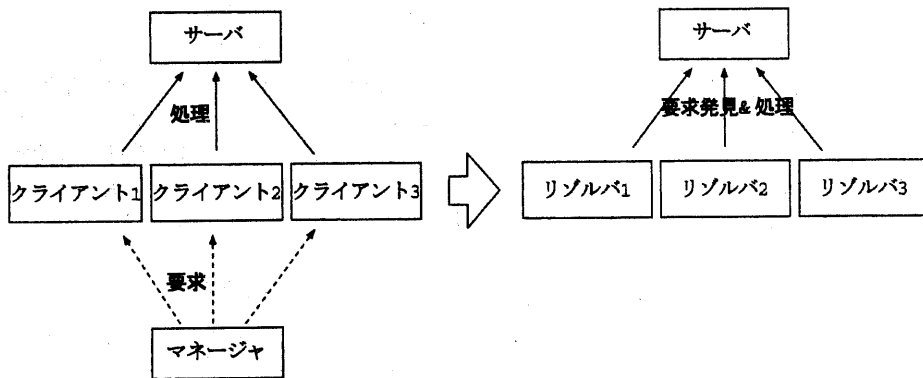
しかし、各リゾルバが自分だけで処理の順序制約を判断することは不可能であるため、どのような順序でどのような処理を行なうのかという情報はどこかに管理しておかなければならない。

ここではサーバにこの処理の順序に関する情報を持たせることとした。

これによって、リゾルバはまずサーバにアクセスして、処理の順序についての情報を取得し、処理の進行状況から、現在自分が行なうべき仕事があればその処理を行なうということになる。

一見すると、今までのマネージャの機能をサーバに持たせただけのようにも思えるが、リゾルバが自分で仕事を見つけて、自分で解決するため、誰かがリゾルバ(クライアント)を呼び出すというフェーズがないのが前節までに述べたクライアント・サーバモデルと異なる点である。

このようなサーバとリゾルバの働きによって非常に簡単にシステムのマネージメントを行なうことが可能となる。



マネージャがクライアントに要求を出し、クライアントはサーバに処理を行なう。

リゾルバは自分で要求 (サーバに格納されている) を探し出して、処理を行なう。

図 2: リゾルバの概念

3 データベース検索システムの実装

前章で述べた機能分散型 DBMS の考えに基づいて、実際にデータベースを検索するシステムを試作した。

ここではデータベースとして関数型データベース [2] を用い、これを関数型検索言語 MMQL [3] で検索するシステムを例にとる。

本章では、まず関数型検索言語 MMQL について簡単に解説し、次に検索システムの実装例を示す。

3.0.1 関数型検索言語 MMQL

関数型検索言語 MMQL は関数型データベースの検索言語である。

マルチメディアデータベースにおける検索とは、ある視点 (view) に従って必要な情報のみをデータベースから抜きだし、その結果を構造化した複合オブジェクトを生成することであると考えている。

MMQL はこのような検索結果として、オブジェクト式 (Object Expression = OE) と呼ばれる階層構造を持つ複合オブジェクトを効率良く生成することを目的として提案された検索言語である。

MMQL は処理の並列性の抽出やユーザ定義の埋め込み関数の取り扱い等の容易さなどから、大容

量データ処理、メディア拡張に優れ、マルチメディアデータベースの検索言語に適していると考えられている。

ここでは詳しい説明は省略するが、MMQL の埋め込み関数により特定メディアに対する処理 (メディア依存処理) を定義することができ、マルチメディアデータに対するさまざまな処理を行なうことができる。

これを実現するシステムでは、多様なメディア依存処理をサポートすることが必要となり、種々の特性を持つ異なるコンピュータ同士を統合した機能分散型データベースシステムが要求される。

3.1 関数型データベース検索システムの実装例

検索言語 MMQL でデータベース検索を行なうシステムを前章で述べたような考えに基づき構築した。このシステムは以下のような構成要素から成り立っている。

1. MMQL コンパイラ

検索は、まずユーザが検索要求である MMQL 検索文を発行することから始まる。

MMQL コンパイラは与えられた MMQL 検索文を構文解釈して、処理およびその評価順

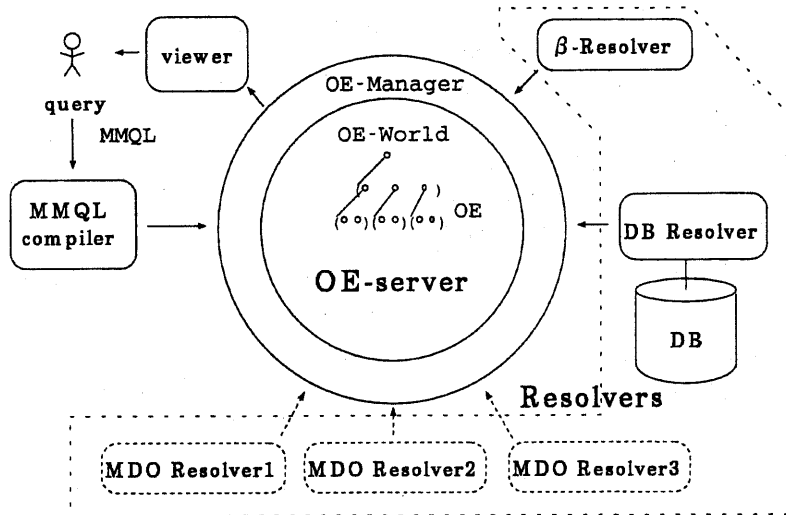


図 3: データベースシステムの構成

序に関する情報に翻訳するものである。

翻訳後、コンパイル結果は後述の OE サーバに渡され、管理される。

2. OE サーバ(サーバ)

OE サーバが管理する情報は大きく以下の 3 つである。

- 検索の途中結果 (オブジェクト式)
- 検索要求 (MMQL が翻訳されたもの)
- 検索の進行状況

サーバはこれらの情報を管理し、クライアントの要求に応じて途中検索結果や検索要求および検索の進行状況 (あわせて、現在どのような処理要求があるか等の情報) を提供する。

3. リゾルバ(クライアント)

個々のリゾルバは検索の単位的操作を行なう。リゾルバは操作毎に複数存在する。例えば、データベースにアクセスして情報を取り出すリゾルバ、数式演算を処理するリゾルバ、画像処理を行なうリゾルバなどがある。

これらはネットワーク上に存在する様々なマシン上に分散している。

リゾルバは

- (a) まず OE サーバにアクセスして現在処理すべき問題を取得する。
- (b) 処理すべき問題中にそのリゾルバが解決できる問題が存在するならば、その問題の解決を行なう。また、そのような問題がない場合には何も行なわない。

また問題解決のためには

- (a) 処理を行なうために必要な情報 (途中検索結果の一部) を OE サーバから取得する。
- (b) 処理を行なう。
- (c) 結果を OE サーバに返す (OE サーバは変更された情報を受けとり、検索結果に加える)。

という動作を繰り返す。

このような単位処理を行なうリゾルバがそれぞれ自分の担当の処理を行なうことにより、最終的な検索結果が生成される。

リゾルバには沢山の種類があり、代表的なものとしてはデータベースにアクセスし情報を引

き出すDBリゾルバや選択演算を行なうβリゾルバ、その他、特定メディアに対する演算を行なうMDO(Media Dependent Operation)リゾルバなどがある。

新たなメディア依存処理を埋め込みたいときには、その処理に対応するMDOリゾルバをシステムに加えることによって容易に機能拡張することができる。

4. ユーアー

各リゾルバがその処理を終了すると、OEサーバ内に最終検索結果が出来上がる。

ユーアーは検索結果のオブジェクト式を、ユーザに提示するためのもので、この動作によって検索は終了する。

3.2 システムの特徴

機能分散データベースシステムを扱う場合に、それぞれの処理をどのようにして、どのマシンで行なうかといったような処理のマネージメントは全体処理の最適化等を考えると非常に難解な問題である。

本来なら処理の特性やコンピュータごとの負荷を考慮し、各々のマシンに処理を最適に割り振る等が必要なのだが、これは必ずしも容易ではない。

ところが、このシステムではリゾルバが独自に仕事を見つけて処理するので、非常に煩わしいと考えられるこのようなマネージメントを行なう必要がない。

本システムでは、ある意味その処理を行なえる環境が整った瞬間に、行なうことが出来るリゾルバが(仕事を見つけて)処理を行なうので、全く自然に最適な処理の分配が行なわれる。

これは、新たなオペレータが定義された際(今まで存在していなかった新たな処理がシステムに組み込まれた場合)など、その処理の特性等が不明でどのようにマネージメントすればよいのかわからない場合にもかなり有効な手法であり、新たなリゾルバを生成するだけでシステムに余計な影響を与えることなく機能拡張できることと合わせ、非常に拡張性能が高いシステムであるということが言えるだろう。

4 まとめ

本稿では、さまざまな特性を持つコンピュータを統合した機能分散型データベースシステム実現のために、クライアント・サーバモデルと新たに導入したリゾルバという概念に基づくデータベース管理システムを提案し、関数型データベース検索システムの設計を行なった。

本来なら各マシンごとの負荷等を考慮し、処理を最適に分配することが必要なのだが、このシステムでは、個々のリゾルバが独自に仕事を見つけて処理を行なうので、新たな機能が加わる際など非常に煩わしいと考えられるマネージメントを自律的に行なうことが期待できる。

これにより、多様な処理を必要とするマルチメディアデータベースにおいて不可欠である機能の分散を効果的に行なうことを可能とし、非常にシンプルな機構で機能分散型データベース管理システムを実現することができる。

現在、本システムの試作タイプが試験的に運用されている。実装のプラットフォームとしてはUNIXワークステーションを用い、現在ではOEサーバとリゾルバが通信を行ない、簡単な検索処理が行なわれるところまでは実現されている。今後、さらなるパフォーマンスの向上を行ない、商用DBとの比較検討を行なう予定である。

参考文献

- [1] 有澤 博, “リアルワールド・データモデリングについての考察,” 電子情報通信学会技術研究報告, DE96-5, 1996.
- [2] H. Arisawa, T. Tomii, H. Yui, and H. Ishikawa, “Data Model and Architecture of Multimedia Database for Engineering Applications”, IEICE TRANS. INF. & SYST., vol E78-D, No.11, 1995.
- [3] 富井 尚志, 有澤 博, “マルチメディアデータベースにおける映像モデリングと操作言語,” 電子情報通信学会論文誌 Vol.J79-D-II No.4, 1996.