

## エージェントによる分散データベースの 機能分化された検索システム

南雲行夫 江尻博尚 川合英俊 高山文雄  
いわき明星大学 理工学部

LANのアプリケーションの中のパーソナルデータベースアプリケーションが個人で使用する範囲の内容のものならネットワークとは無縁に動作していても構わないが、多くの人が同時利用するのに向けた良好なデータを管理していることも珍しくなくなってきた。ネットワーク上に多数点在するデータベースを統合検索する為には、データを一つのコンピュータにまとめてしまうか、DBMSを用いて検索することになる。しかし、データベースがネットワーク上に多数点在するという事は、データと管理者を分散させたまま、管理者の労力やコンピュータの負荷が集中していないという点で、有利な環境であるとも考えられる。

本研究では、DBMSの検索部分に焦点を絞り、移行前の環境を維持した統合検索機能を実現する為、検索に必要な機能を分化し、それぞれの機能を持つエージェントをネットワーク上に分散して配置する。これらのエージェントにより、検索処理が分散される為、処理が集中してコンピュータを占有してしまうことはなく、データも元の分散したままなので、管理にかかる負担を増やさずに、元の環境をほぼ保ったまま統合検索機能が実現された。この論文では、本研究の提案するシステムによって、元の環境を維持したまま統合検索機能が実現される事など本システムの利点について実験的に述べる。

### Agent applied to search data on dispersed database.

Nagumo Yukio, Ejiri Hironao, Kawai Hidetosi, Takayama Fumio  
Iwakimeisei University

A personal database application system in a LAN is also available even without network, but in most case the system is used to have usefull data for multiuser concurrently. In order to be unified many personal databases distributed in a LAN, they should be collected on to a computer in a LAN or be retrieved by use of a newly developed database management system. Personal databases distribution might be effective for reducing of computer power centralization and a personal database administration.

In this paper, to implement a unified retrieving function without changing any personal database system enviroment, three kinds of agents are allocated on a personal database, on a user interface at his client system and on a data address manager. By these three agents, retrieving process are distributed in a LAN, in addition to that, a personal database remains as is and a personal database administration does not increase, are experimentally reported.

#### 1 はじめに

現在、コンピュータ環境の多くはLANによって結ばれたネットワークが構築されており、そのネットワーク上ではFIPクライアントやWWWブラウザなどのいくつものアプリケーションが動作している。それらのアプリケーションの中にはネットワークとは無縁に動作しているものも多い。

パーソナルデータベースアプリケーションであっても意外とネットワーク上に孤立して存在している場合がある。パーソナルデータベースアプリケーションは個人で使用する範囲の内容のものならそれでも構わないが、パーソナルデータベースアプリケーションを用いて多くの人が利用できたほうが良い

データを管理していることも珍しくない。これはパーソナルデータベースアプリケーションが安価であったり、操作が容易である為だろうと思われる。そのようなデータベースがネットワーク上に多数点在する場合、内容によっては統合検索したくなる。本学の事務処理LANでもそんな要求があった。一般的に、統合検索をする為には、データを一つのコンピュータにまとめてしまうか、DBMSを用いて管理することになる。しかし、どちらの方法も、その後の管理やシステムの構築が大変であるし、コンピュータにかかる負荷も大きくなり、移行前と同じような環境では無くなる為、好ましくない。なぜなら、データベースがネットワーク上に多数点在するということは、データと管理者を分散している為、管理者の労力やコンピュータの負荷を分散しているという点で、有利な環境であるとも考えられるからである。

本研究では、DBMSの検索部分に焦点を絞り、移行前の環境を維持した統合検索機能を実現する為に、検索に必要な機能を分化し、それぞれの機能を持つエージェントを作成した。統合検索に必要な機能を、データベースから実際に検索する機能、データベースのあるアドレスを管理する機能、検索を行なうユーザーのインターフェイス機能の三つとし、それぞれのエージェントを作成し、ネットワーク上に分散して配置しする。これらのエージェントが、検索処理を分散する為、処理が集中してコンピュータを占有してしまうことはなく、データも元の分散したままなので、管理にかかる負担もほとんど変わらなく、元の環境をほぼ保ったまま統合検索機能が実現される。

この論文では、本研究の提案するシステムによって、環境を維持したままで統合検索機能が実現される事と本システムの利点について述べる。また、今回作成したプロトタイププログラムに組み込まれていないセキュリティ案についても少し触れる。

## 2 エージェントによる分散化

データベースを扱う場合、SQLサーバなどのDBMSによって統合されていたほうが管理しやすく、機能的にも豊富である。しかし、処理を行なうDBMS自体が一つである為、DBMSを実行して

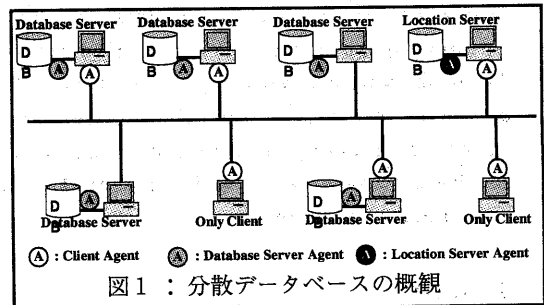
いるマシンには非常に多くの処理が集中することになる。このような処理の集中するマシンは、より処理の速い、高性能のものとなり、日頃仕事に使用されているようなマシンに、更にインストールするのは賢明ではない。新たにSQLサーバを設置するにはコスト面でも労力でも負担が増すことになる。

しかし、元々データベースは分散配置されているのだから、処理する部分を集中させる必要はない。この研究ではこの処理する部分をエージェントとして分散配置されたデータベースそれぞれに分散して配置することにより、処理の集中を回避しようという試みを行なっている。

DBMS全体を分散して作成することが理想ではあるが、今回はデータベースの検索部分のみに限って行なうこととする。これは、実際に使用する頻度の大きな処理は検索であると考えての事である。

この研究では、分散化する方法としてエージェントを用いている。エージェントには移動型エージェントと固定型エージェントがあるが、この研究では固定型のエージェントを使用している。移動型エージェントはインタプリタ型のスクリプトで書かれることが多く、スクリプト転送時のネットワークトラフィックの増加や、現在JAVAなどで指摘されているようなセキュリティー問題などがあることなどから、今回は固定型エージェントを使用した。

## 3 実現方法



今までデータベースの統合には、SQLサーバが多く用いられてきたが、この研究では検索面に於けるデータベースの統合を行なう為、SQLサー

バの代わりにエージェントを使用する。これは、SQLサーバの処理を分散し、サーバ構築の負担を軽減することををねらった方法である。この研究では図1に示すような次の三つのエージェントを使用した。

- 1) クライアントエージェント (CA): データの検索を行なうユーザーと直接接するエージェントである。
- 2) データベースサーバエージェント (DA): 直接データベースに検索を行なうエージェントである。
- 3) ロケーションサーバエージェント (LA): データベースサーバ(DA)のIPアドレスなどを保持している。

これらのエージェントを使用することにより、データベースをLAN上に点在させたままで、ユーザーが統合的にこれらを検索することを可能にする。

エージェントによる分散化の方法として用いた今回の方法では、ロケーションサーバが重要な役割を果たす。ロケーションサーバエージェントはデータの有るマシンのIPアドレスを保持している、問い合わせに対してその情報を与えるだけなのだが、ロケーションを保持することによって問い合わせをするユーザーがデータベースの在処やマシンの在処、ましてやデータの在処などまったく意識せずに問い合わせを行なうことが出来るのである。ほかのエージェントは、お互いのエージェントをまったく意識せずに動くので、ロケーションサーバエージェントがほかのエージェントとの間をつなぐ唯一のエージェントとなっている。しかし、このような機構だとロケーションサーバエージェントに、アクセスが集中してしまうことが考えられるが、DBMSの集中処理の場合と違って、処理が単純でごく短時間で済む為、ある程度アクセスが集中しても問題にならないのである。

#### 4 エージェント・プログラムの説明

##### 4.1 クライアントエージェント

データを検索するためにユーザーは、例1のような検索文字列をクライアントエージェントに入力

するとロケーションサーバへ送信される。  
例1)

black & moon | white & star

ここで&はAND、|はORの意味としている。

これはごく単純な規則に基づく問い合わせ文字列で、検索したいキーワードとandやorを組み合わせたものである。これは、ほかのサーバエージェントによってSQL文に変換される。

クライアントエージェントは、ロケーションサーバから該当するデータの有るデータベースサーバのアドレスを受け取り、そこへ検索文字列を再送し、検索結果を受け取る。

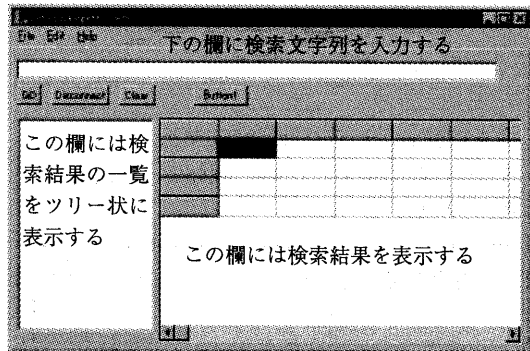
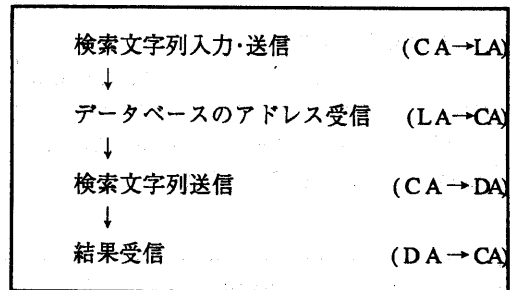


図2 : クライアントエージェントのGUI

エージェントの処理の流れは以下ようになる。



また、データの送受信にはTCP/IPを用いている。これは、現在一番普及しているプロトコルであり、エージェントをほかのプラットフォームに移植する場合に有効であると考えられる。

##### 4.2 データベースサーバエージェント

データベースサーバエージェントは、ロケーションサーバに次の登録をする必要が有る。そのため、データベースサーバ管理者は、ロケーションサーバのアドレス、データベースの公開するテーブル、

検索するフィールドを設定し、それをロケーションサーバに登録する。今回のプログラムでは検索するフィールドは図3に示すように三つまで設定できる仕様になっている。

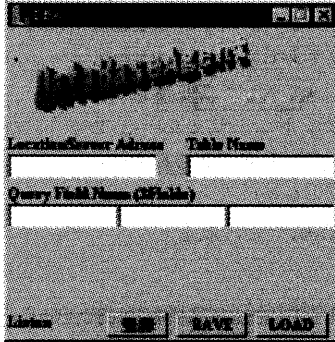
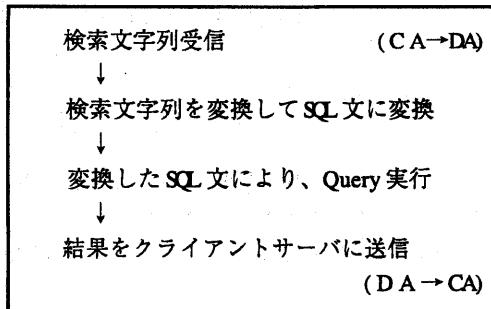


図3：設定画面

ロケーションサーバへの登録以降のデータベースサーバの検索サービスは、次のような処理の流れになる。



エージェントからのデータベースへのアクセスにはODBCドライバを通して行なう(図4)。

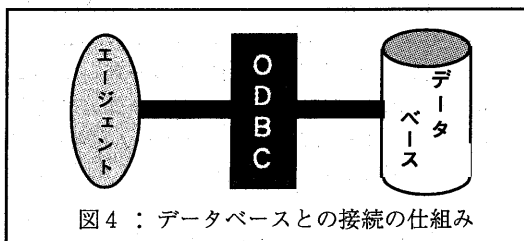


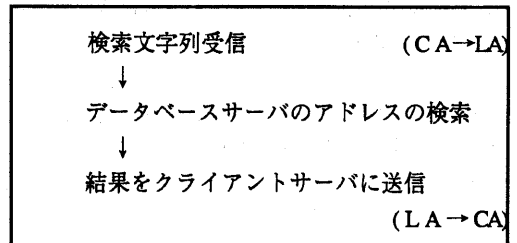
図4：データベースとの接続の仕組み

#### 4.3 ロケーションサーバエージェント

ロケーションサーバは、データベースサーバの

IPアドレスと、各データベースサーバの検索フィールド内の内容を保持する。ロケーションサーバエージェントは、クライアントエージェントからの検索文字列に該当するデータを保持するデータベースサーバのIPアドレスをクライアントエージェントに返す。

処理の流れは以下ようになる。



#### 5 実用例案

1) 大学における研究室単位の学生住所録の統合利用

現在、大学において学生の住所録が研究室単位で作成されているような場合に、それらのデータベースは異なったアプリケーションによって作成されているかもしれないが、このようなデータベース群を統合して利用できたなら、学生間の相互連絡に非常に役立つものとなるだろう。このようなデータベース群を統合するにはSQLサーバを、いちいち設置していたのでは、統合に必要な労力や金銭面での負担が問題となることは目に見えているが、本研究の方法であるエージェントによる分散データベースの検索を用いれば、最小限の労力だけで分散されたデータベースの統合利用が出来るシステムが構築できるだろう。

本研究のエージェントによるシステムを構成するには、まず、データベースがODBC対応であることが必要である。そうでないとデータベースサーバエージェントが対応していない為、そのデータベース独自のエージェントの作成を行なわなければならないだろう。最近ではほとんどのデータベースアプリケーションがODBC対応となりつつあるので、そのようなことは少ないと思われる。ODBCに対応してさえいれば、ほとんどのデータベース関連のアプリケーションに対応するシステムとなる。

ここでは、データベース群のすべてがODBC対応であると仮定した状況における設置方法について述べる。

データベースをエージェントによって統合する為には、ロケーションサーバをまず設置する。それには、ロケーションサーバエージェント本体とデータファイルの二つのファイルをマシンに設置し、OBCマネージャを使ってデータファイルをエージェントの認識できるようにし、本体を起動する。たったこれだけで、ロケーションサーバの設置は完了する。

次に、データベースサーバを設置する。まず、データベースサーバエージェント本体のファイルをマシンに設置し、OBCマネージャを使って公開する目的のデータベースをエージェントに認識できるようにする。その後、エージェントを起動し、ロケーションサーバのIPアドレスと公開するテーブル名と三つの検索に使用するフィールド名を設定して、これでデータベースサーバの設置は完了する。

後は、クライアントエージェントを起動してロケーションサーバのIPアドレスを設定するだけで統合検索が可能となる。

このように、本研究のシステムを設置するのは非常に容易である。

また、管理面においてもメリットがあると考えられる。データベース自体、もともとが研究室単位のデータベースなので、それぞれのデータベースに管理者がおり、統合して飛躍的に増えたデータの管理を集中して行なわずに、今まで通りの管理を行なっていればいいわけである。大量のデータを集中して管理する必要が無いのは大きな利点である。

エージェントによる方法は、統合検索をする場合にもメリットがある。例えば、学生の名前は知っているが、どこの研究室に所属しているのかを知らない場合、本研究の方法なら、何の意識もせずに検索すれば発見できるだろう。しかも、先に述べたような簡単な、なんの複雑な設定もしないシステムによってです。これがもしSQLサーバやほかのデータベースアプリケーションだったらある程度の複雑な設定が必要になる。このような検索面におけるメリットは、独立したロケーションサーバによってもたらされたものと考えられる。ロケーションサーバに登録されたデータ所在地は単純で小さなデータですが非常に重要な意味を持っています。

このように例にし示した場合には、エージェン

トによるデータベース群の統合利用は簡単な作業によって有効に働き得るシステムとなっています。

## 6 セキュリティ(案)

今回作成したプロトタイププログラムでは、セキュリティはまったく考慮されていない。今後の計画ではプログラムのソースコードのJAVAへの移植と共にRSAを用いたテンポラリーなキーによる送受信データの暗号化を考えている。

まず、ロケーションサーバに使用ユーザーの登録機能とキーの分配機能及びデータベースの格付けデータを付加し、セッション毎にキーを生成してクライアントおよびデータベースサーバに分配する。このように、セキュリティ機能を付け加える事によって、より様々なデータを取り扱う事が出来るようになる。

## 7 結論

本研究は、データベースをより簡単にグループウェアとして活用する方法を提案したものである。より多くの人に簡単に使ってもらう為には、より有用な情報を安易に検索できる必要があり、情報をより有用なものにするにはなるべく多くの情報の中から検索したほうがよい。そのためには、今ある情報のより簡単な統合法が必要になる。

データベース統合にSQLサーバを用いたとき、まずSQLサーバ自体を設置しなければならず、パーソナルコンピュータによって構成されたLAN内で使用中のコンピュータに設置するには処理が重く不都合である。SQLサーバを導入するのと異なって、今回示したエージェントを用いた方法は、現状のLAN環境を維持しつつグループ内のデータベースを疑似統合が出来る。また、SQLサーバでは、検索時に、どこに必要なデータが有るかを、或る程度知っている必要が有るが、エージェントを用いた方法では、ロケーションサーバに登録してあるデータベースなら、どこにデータが有るかを知らずに、検索することが可能である。

今回の研究で行なったエージェントを用いた方法のように、分散したデータそれぞれにエージェントを置いた形態は、処理の集中を避け、管理者の負担と経済的な負担を軽減でき得る。

本研究から、このような分散型のDBMSが、現在の状況に見合ったシステムであると考えられる。