

## オープンな協調型データベースの構成法

八木 哲 高橋直久

NTT 未来ねっと研究所

東京都武蔵野市緑町 3-9-11

yagi@core.ntt.co.jp naohisa@core.ntt.co.jp

あらまし 本稿では、インターネット上に散在するデータベースをコモディティとみなし、自在に協調動作させるための、システムの構成法について述べる。提案するシステムでは、各々のデータベースで実行する、各々のデータベースの仕様に基づいたクエリ文の組みをノード、クエリ文の組の依存関係をアーチとする、データフローグラフ形式でクエリを表現する。このクエリ表現に、実行に必要な補助情報を付加し、モバイルコードとして各データベースと組をなすサーバに分配し、データ駆動方式に基づいて実行制御することにより、データベースを協調動作させる。本稿では、システムの構成と簡単な適用実験について示す。

キーワード マルチデータベース 分散処理 データ駆動 XML

## The construction method of cooperative open database system

Satoru Yagi and Naohisa Takahashi

NTT Network Innovation Laboratories

3-9-11 Midori-Cho Musashino-shi Tokyo Japan

yagi@core.ntt.co.jp naohisa@core.ntt.co.jp

**Abstract** This paper presents the system for making heterogeneous databases which are commodities connected to the Internet cooperate. The query is input to the system as a data-flow diagram. In the query, nodes express the set of queries executed by the individual databases. Arcs express dependences among nodes. The query and supplementary are translated into a mobile code. The mobile code is distributed to servers paired with the databases. Servers execute the mobile code based on the data-driven method. This paper describes the design of the system and an experiment in which the system was applied to a simple application.

key words Multi database Distributed processing Data driven XML

## 1 はじめに

WWWベースのシステムの普及やネットワークの高速化とあいまって、ネットワーク接続された様々な規模や種類のデータベースが運用されている。これらのデータベースを関連付けて共同利用することは、極めて有意である。例えば、a) 地理分散した観測点のデータや研究データ(例:ゲノムデータベース<sup>1)</sup>)の相互利用、b) 多数のベンダーが隨時入れ替わるB2B市場(例:リンク集<sup>2)</sup>)における取引条件の相互利用、c) 複数のWWWサーバで構成された電子モール(例:PC関連<sup>3)</sup>)や負荷分散しているサーバ群のログ解析、d) 地図、画像、音など多様なデータを連動させる電子図書館、e) アプリケーション・サービス・プロバイダが提供する複数のデータベースサービスの連携利用などである。即ち、ネットワーク上に散在するデータベース群をコモディティとみなし、自在に協調動作させて共同利用する、オープンな協調型データベースである。

これには、地理分散したデータベースの運用主体が、独自に運用しているデータベースをそのまま利用して、データベースを協調動作させるコミュニティに、柔軟に参加、脱退できることが要件となる。即ち、1) 広域性と規模への対応、2) データベースを協調動作させるコミュニティへの参加と脱退が容易、3) 柔軟なデータベースのアクセス権の設定、4) 各データベースの独自仕様が利用可能、5) 統一的なインターフェースの提供が課題となる。異種データベースを統括するアプローチに、マルチデータベース<sup>4)5)6)</sup>がある。また、情報収集に重点を置いたメディエータ<sup>7)8)9)</sup>がある。これらアプローチでは、何らかの水準のスキーマを利用して、システムを統括するマスタ機能がある。結果として、マスタへの制御機能の集中に起因する、拡張性や構成変更作業に伴う利用者への影響範囲の問題により、上記1)2)3)の条件を満たすことが難しい。スキーマを利用したシステム統括に起因する、構成変更作業に伴うシステムと利用者への影響範囲の問題により、上記2)3)4)の条件を満たすことが難しい。これらの問題は、データベース間の結合が強く、データベースの自律性が不十分なことに起因する。

このような問題に対して、データベース間の結合を弱め、データベースの自律性を高めるために、相互運用システム<sup>10)</sup>を基礎として、クエリをデータフローグラフ形式で表現し、データ駆動方式で実行制御するアーキテクチャ、MRC(Making Resource Cooperate)<sup>11)</sup>の研究を進めている。クエリのノードは、個々のデータベースで実行される、個々データベースの仕様に従ったクエリ文の組みである。アーカンは、あるノードを構成するクエリ文を実行するために必要なデータが、他のノードを構成するクエリ文の実行により生成されるという、ノード間の依存関係である。トーンンは、クエリ文の実行結果である。このクエリのノードを、各データベースで発火規則<sup>12)</sup>に基づいて実行制御することで、クエリを実行する。このようなクエリ表現を、DFGクエリと呼ぶことにする。実行と制御をデータベースを単位に分散することで、前記1)2)3)の問題に対処する。データベースの独自仕様をデータベースを単位に局所化することで、前記2)3)4)5)の問題に対処する。データ駆動方式という簡明なメッセージパッシング型実行制御方式により、前記5)の問題に対処する。

本稿では、先ず、MRCに基づいたMRCシステムの概要を示す(2章)。次に、クエリの表現(3章)、MRCシステムの動作(4章)、データベース間のデータ(トーンン)転送用のデータフォーマットを示す(5章)。また、実験システムにより簡単な適用実験を行う(6章)。最後に、MRCシステムの適性を考察し(7章)、本稿の内容をまとめ、今後の課題を示す(8章)。

## 2 MRC システムの概要

MRCシステムの構成を図1に示す。機能要素

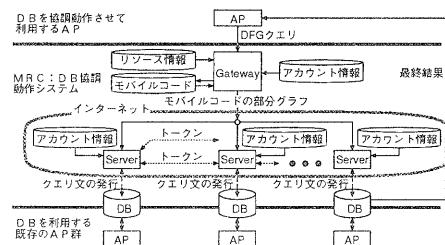


図1: MRC システム

として、Server と Gateway がある。Server は、DFG クエリのノードを構成するクエリ文を、データベースに発行する。Server は複数の Engine から構成され、一つの Engine が一つのノードを受け持ち、同一データベースに対する複数のノードの実行を可能にする。Gateway は、利用者やアプリケーションが作成した DFG クエリを受け付け、リソース情報を利用し、DFG クエリのノードに対して、実行を担当するデータベースと Server の組を割り当てる。割り当ての終った DFG クエリを、モバイルコードと呼ぶ事にする。リソース情報は、データベースのカタログ情報と Server のリストであり、データベースと Server の運用主体が登録する。Gateway は、MRC システムの利用者ごとに複数あってもよい。更に Gateway は、モバイルコードを部分グラフに分解し、割り当てた Server に分配する。Server は、Gateway から分配されたモバイルコードの部分グラフを、発火規則に基づいて実行制御し、組をなすデータベースにクエリ文を発行する。クエリ文の実行結果は、トークンとして他の Server に転送され、組をなすデータベースに一時保存される。最終結果は、アプリケーションがネットワークを介して、結果が置かれたデータベースを直接アクセスする。Server 間のデータ転送について、協調動作可能なデータベースの種類は、データ転送に用いるデータフォーマットの表現能力に依存するため、データフォーマットに、表現能力の高い XML を利用する。また、広域で安定した通信を行うために、通信プロトコルとして TCP を利用する。

MRC システムの利用モデルとしてのアカウントの扱いついで、データベースを協調動作させるコミュニティに参加する、利用者やデータベースの構成変更に伴う作業の影響を局所化するために、利用者のアカウントである“メンバ”と、データベースのアカウントである“ユーザ”に加え、コミュニティの同意のもとに作られた、ある協調動作のためのアカウントである“コミュニティ”からなる、3 階層のモデルを採用する(図 2)。利用者は、“メンバ”的アカウントで Gateway の“コミュニティ”的アカウントにログインする。Gateway は、Server にモバイルコードを配布するときに、“コミュニ

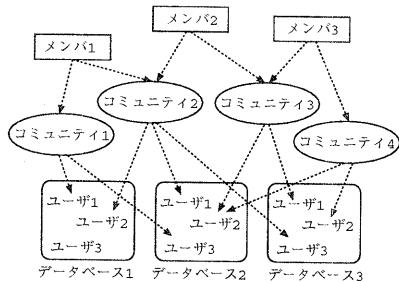


図 2: アカウント・モデル

ティ”的アカウントで Server にログインする。Server は、“コミュニティ”に属する“ユーザ”的アカウントでデータベースにログインし、クエリ文を発行する。“メンバ”が利用可能な“コミュニティ”を Gateway で管理し、“コミュニティ”が利用可能な“ユーザ”を Server で管理すれば、“コミュニティ”に属する利用者の構成が変わった場合には、利用者側で運用する Gateway の設定ファイルを操作するだけでよい。また、“コミュニティ”で利用するデータベースの構成が変わった場合には、データベースの運用主体側で運用する Server の設定ファイルを操作するだけでよい。これら二つの設定ファイルを、アカウント情報と呼ぶことにする。

### 3 クエリの表現

DFG クエリの仕様を、拡張 BNF<sup>13)</sup> を用いて表 1 に示し、各要素を説明する。

- DFG クエリ：データフローグラフ形式のクエリ。ノードとなるクエリ文の組に、アークのディストネーションである入力と、アークのソースである出力の情報を附加した形式で記述する。ノードの属性として、ノードの ID であるノード名と、実行回数など、ノードの実行形態を表す実行形式がある。
- 入力：アークのディストネーションを示す。属性として、入力データをトークンとして消費するか、初期値として再利用するなどを示す入力形式、入力データを受信する時のフォーマットを示すデータ形式、入力データの格納先を指定する変数名、送信元ノードを示すノード名、入力データの名前、入力データのうちの格納する部分を示す要素名がある。

表 1: DFG クエリ

DFG クエリ	=	* ("QUERY" 実行形式 ノード名 {" } * 入力 * クエリ * 出力 {" })
入力	=	"INPUT" 入力形式 データ形式 変数名 {" } ノード名 {" } データ名 {" } 要素名 {" ; }
出力	=	"OUTPUT" データ形式 ノード名 {" } データ名 {" } 変数名 {" ; }
クエリ	=	[ "BEGIN{" クエリ文の組 {" }" }" ] [ "ERROR_BEGIN{" クエリ文の組 {" }" }" ] [ "BODY{" クエリ文の組 {" }" }" ] [ "ERROR_BODY{" クエリ文の組 {" }" }" ] [ "END{" クエリ文の組 {" }" }" ] [ "ERROR_END{" クエリ文の組 {" }" }" ] [ "ERROR{" クエリ文の組 {" }" }" ]

表 2: モバイルコード

モバイルコード = \*データベース指定 \*サーバ指定 DFG クエリ  
 データベース指定 = "DATABASE" データベース名 "=" DFG クエリの部分グラフの指定 ";"  
 サーバ指定 = "SERVER" Server のアドレス "=" DFG クエリの部分グラフの指定 ";"

- 出力: アークのソースを示す。属性として、出力データを送信する時のフォーマットを示すデータ形式、送信先を示すノード名、出力データの名前、出力データの格納元を示す変数名がある。
  - クエリ: 各データベースが提供するトランザクションに関するクエリ文を利用して、分散トランザクション処理を実現するこのために、各データベースのクエリ言語や、Serverで展開されるマクロにより記述するクエリ文の組を、7つのブロックに分割して記述する。
    - BEGIN ブロック: 初期化処理やトランザクション開始の宣言、ロック操作を記述する。全てのノードの BEGIN ブロックが正常に実行された場合に、BODY ブロックを実行する。
    - BODY ブロック: データベース処理の本体を記述する。DFG クエリの「入力」「出力」の項は、BODY ブロックにのみ作用する。全てのノードの BODY ブロックが正常に実行された場合に、END ブロックを実行する。
    - END ブロック: 一時データの削除やコミット操作を記述する。全ての END ブロックが正常に実行された場合に、DFG クエリを正常に実行したと判断する。
    - ERROR-BEGIN ブロック: BEGIN ブロックでエラーが発生した時に実行する。一時データの削除やロールバック操作を記述する。
    - ERROR-BODY ブロック: BODY ブロックでエラーが発生した時に実行する。一時データの削除やロールバック操作を記述する。
  - Gateway は、Serverによるノードの実行制御に関して、BEGIN ブロック、BODY ブロック、END ブロックを単位に同期を取る。あるブロックの実行でエラーが生じた場合、エラーが発生した Server では、そのブロックに対応するエラーブロックを実行し、その他の Server では、ERROR ブロックを実行し、モバイルコードの実行を終了する。また、ロック期間を局所化したい場合には、DFG クエリを分割し、順次実行する方法がある。
  - 次に、モバイルコードの仕様を、拡張 BNF を用いて表 2 に示し、各要素を説明する。

表 3: データ引渡しの指定方法の例

入力の項の変数名	= 表の名前 / "ITEM(" 行指定 "," パラメタ指定 ")"
出力の項の変数名	= "ITEM(" 行指定 ")"

DFG クエリの、「入力」、「クエリ」、「出力」間のデータ引渡しの指定方法について、例えばクエリ言語として SQL を用いる場合、「入力」の属性の変数名として、表の名前か、 CLI 形式<sup>14)</sup>で記述された SQL 文のパラメタを指定する関数名を用い、入力データをクエリ文に渡す。また、「出力」の属性の変数名として、SQL 文を指定する関数名を用い、クエリ文の実行結果を取得し、出力データとして送信する。具体的な指定法を、拡張 BNF を用いて表 3 に例示する。「行指定」は、BODY ブロック内の行番号により記述する。ラベルの使用も考えられる。「パラメタ指定」は、CLI 形式で記述されたクエリ文内での、パラメタの出現順番により記述する。

#### 4 MRC システムの動作

MRC システムの正常時の動作を、図 3 のシーケンス図に示す。実行中にエラーが発生した場合の概要について、1)Server や Engine は、エラーの発生を Gateway に通知し、2)Gateway は、他の Server や Engine に実行中止を指示する。3)Server は、エラー処理を行い結果を Gateway に返す。

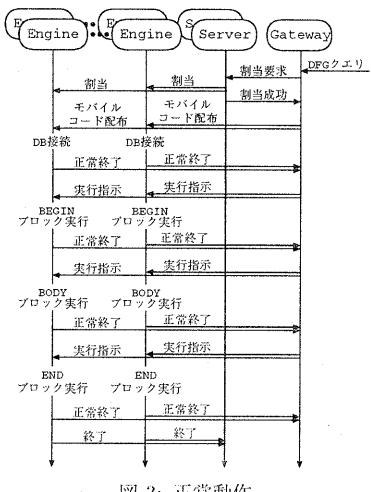


図 3: 正常動作

#### 5 データフォーマット

MRC システムの、Server 間のデータ転送用のデータフォーマットは、図 4 に示すツリー構造を取る。DFG クエリの記述と対応付ければ、「送信元ノード名」は、データの送信側ノードの名前であり、「データ名」は、送信側ノードの出力の項のデータ名である。この二つが、受信側ノードの入力の項の、ノード名とデータ名に一致した場合、変数名で指定された変数に格納される。「データ値」はデータ本体である。「要素」はアトミックなデータの値であり、名前や型等を属性を持つ。XML により、このような形式で表現されたデータと、データベースに格納するデータとの対応付け方法は、DFG クエリの入力と出力の項のデータ形式に記述する、キーワードにより指定する。各キーワードによる対応付けの内容は、デフォルトで用意する方法と、ユーザが直接指定する方法がある。関係データベースを対象としたデフォルトのキーワードは、例えば次の 3 つがある。

- CTL：制御用のシグナル。あるクエリ文が実行されたことを示し、「要素」は持たない。
- TUPPLE：タブルと対応付ける。「要素」の階層は一段であり、タブルの各項目を対応付ける。図 5 上参照。
- TABLE：表と対応付ける。「要素」の階層は多段であり、リーフの「要素」にタブルの項目を対応付ける、リーフの一つ上の「要素」にタブルを対応付ける。リーフ以外の「要素」が値を持つ場合は、タブルの項目として扱い、その「要素」の下位に属するタブルには、同じ値を入れる。図 5 下参照。

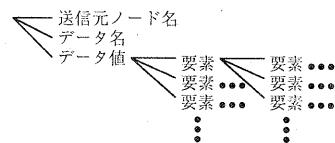


図 4: データフォーマット

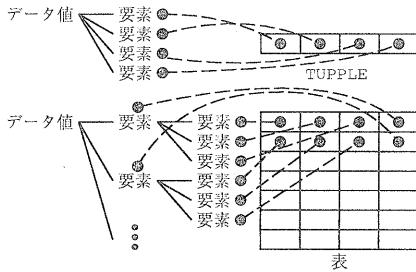


図 5: 対応付けの例

データモデルの制約により、データベース間で完全なデータの受渡しが不可能な場合には、インデックスとなる項を作成し、これをポインタとして利用する方法がある。

## 6 適用実験

実験用 MRC システムの目的は、MRC システムの動作と適用領域の検証にある。可搬性のために、Gateway と Server は JAVA で記述し、Engine はマルチスレッドにより実装した。DFG クエリのクエリ文は、一般的な SQL を記述対象とした。リソース情報の扱いは簡略化しており、データベースと Server の明示的な指定が必要である。実験用 MRC システムの動作環境を表 4 に示す。

MRC システムの適用領域に、多様なデータベースを連動させる電子図書館や、多数のベンダが隨時入れ替わる B2B 市場における、取引条件の相互利用などがある。この場合、キーとなるデータを配布してデータベース処理を行い、その結果を更にキーとして利用する形態が考えられる。例えば、最初のデータベース処理で制約条件の検索を行い、次のデータベース処理では、その制約条件に合致するものを選択する形態である。この種の単純な問題として、PC の構成案を作成する問題を取り上げる(図 6)。ベンダが提供するマザーボードとメモリのデータベース(表 5)が利用可能である時、

表 4: 動作環境

OS	FreeBSD3.2-RELEASE
JAVA	JDK1.1.8
XML Parser	XMLAJ3.0.0EA3 / SAX
RDB	PostgreSQL 6.5.3

M/B情報サーバ : マザーボードの情報を提供するデータベースシステム  
MEM情報サーバ : メモリの情報を提供するデータベースシステム  
Report サーバ : 作成したレポートが置かれるデータベース

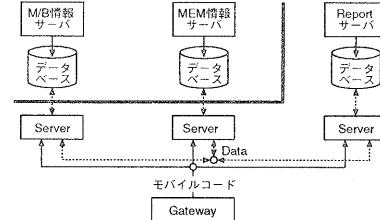


図 6: PC 構成案レポートシステム

表 5: データベースが提供する情報

マザーボードの情報	メーカ、型番、ソケットの型 チップセット、メモリの型 etc、ベンダ、価格
メモリの情報	型、容量、CL、ECC, etc, ベンダ、価格

CPU の仕様 (FC-PGA) とチップセット (i820) を指定して適合するマザーボードを検索し、更にマザーボードのメモリの仕様情報を用いて、適合するメモリを検索する。検索したマザーボードとメモリの情報から、価格評価レポートを作成する。

マザーボードとメモリのデータベースを、実験用 MRC システムにより協調動作させ、付録 A の図 7 に示すモバイルコードを実行する。付録 A の図 8 に示す実行結果では、ノード「result」で join を取って得られた、PC 構成案の価格評価レポートを表示している。この例では、FC-PGA 対応 i820 のマザーボードに 128M サイズのメモリを組み合わせる場合には、370SCD と PC100SDRAM(CL=2)、もしくは PC133SDRAM(CL=3) の組み合わせが、一番安価なことが分かる。MRC システムを利用すれば、各ベンダが独自に使用しているデータベースをそのまま利用して、B2B 市場に随时参加する形態を実現できる。

## 7 考察

提案した MRC システムでは、ネットワーク上に散在するデータベース群をコモディティとみなす、自在に協調動作させるために、1) 広域性と規模への対応、2) データベースを協調動作させるコミュニティへの参加と脱退が容易、3) 柔軟なデータ

タベースのアクセス権の設定, 4) 各データベースの独自仕様が利用可能, 5) 統一的なインターフェースの提供という課題を, 次のように解決している.

1)について, DFG クエリの実行と制御が複数の Server と Gateway に分散しているため, データベース数の増加には, データベースと組で動作する Server の追加, 利用者数の増加には, DFG クエリを受け付ける Gateway の追加という形で, 容易に対応できる. 2)について, 前記のように分散処理をしているため, 参加のための作業は, データベースと組で動作する Server の立ち上げとリソース情報の登録であり, 脱退のための作業は, リソース情報の登録の削除と Server の停止である. 作業コストは小さく, 作業の影響は他の Server や Gateway に及ばない. 3)について, “メンバ”, “コミュニティ”, “ユーザ”という3階層からなるアカウントモデルにより, データベースを協調動作させるコミュニティを構成する, 利用者やデータベースの変更作業の影響を, 局所化している. 4)について, DFG クエリのノードに, 各データベースの仕様に従ったクエリ文をそのまま記述できるため, データベースの独自機能が利用できる. 5)について, 利用者に対しては, 4)の特長を利用し, データベースの運用主体側で, “コミュニティ”が行う定型的な処理に対して, クエリ文のマクロを用意できる. データベースに対しては, データ駆動方式により実行制御することにより, クエリ文間のデータ依存性という, 本質的な依存関係に基づく, 簡明で適応範囲の広いインターフェースを提供する. また, DFG クエリの記述クラスは, 関係データベースにおける表など, 永続的データを介したクエリ文の依存関係が表現可能という意味で, 従来のクエリ文の記述クラスと基本的に同等であるといえる.

## 8 おわりに

本稿では, ネットワーク上に散在するデータベース群をコモディティとみなし, 自在に協調動作させるための, MRC システムについて示した. 本システムの特徴は, 実行と制御の分散と, 独自仕様の局所化である. 今後の課題として, DFG クエリのプログラミング環境などの利用者の支援機

構, 認証や暗号化通信などのセキュリティ機構, また, DFG クエリのトポロジを動的に変更可能にすることによる, 適用範囲拡大の検討がある.

## 謝 辞

日頃御指導いただく皆様方に深謝いたします.

## 参 考 文 献

- 1) 高木, 金久: ゲノムネットのデータベースの利用法 [第2版], 共立出版, 2章, pp.11-42 (1998).
- 2) インターネット取引所, <http://nis.nikkeibp.co.jp/nis/ex/>
- 3) NETde 通販, <http://www.iijnet.or.jp/netde/index2.html>
- 4) A.P.Sheth, J.A.Larson :Federated database System for Managing Distributed, Heterogeneous, and Autonomous Databases, ACM Computing Surveys, Vol.22, No.3, pp.183-236 (1990).
- 5) W.Litwib, L.Mark, N.Roussopoulos :Interoperability of Multiple Autonomous Databases, ACM Computing Surveys, Vol.22, No.3, pp.267-293 (1990).
- 6) 細川, 清木: 関数型計算によるマルチデータベースシステムの問い合わせ処理方式, 情報処理学会論文誌, Vol.39 No.7, pp.2217-2230 (1998).
- 7) G.Wiederhold :Mediators in the Architecture of Future Information System, IEEE Computer, 25, pp.38-49 (1992).
- 8) R.Yerneni, C.Li, H.Garcia-Molina, J.Ullman :Computing Capabilities of Mediators, SIGMOD'99 (1999).
- 9) C.F.Goldfarb, P.Prescod :XML 技術大全, プレンティスホール出版, 9章, 29章 (1999).
- 10) M.W.Bright, A.R.Hurson, and Simin H.Pakzad :A Taxonomy and Current Issues in Multi-database System, IEEE COMPUTER, Vol.25, No.3, pp.50-59 (1992).
- 11) 八木, 高橋: モバイルコードを用いた協調型オープンデータベースシステムの構想, 2000-DPS-96-3, pp.13-18 (2000).
- 12) J.A.Sharp :データ・フロー・コンピューティング, サイエンス社 (1987).
- 13) D.H.Crocker :Standard for the format of ARPA internet text messages, RFC822 (1982).
- 14) C.J.Date, H.Darwen :標準 SQL ガイド改訂 第4版, アスキー (1999).

## A 適用例

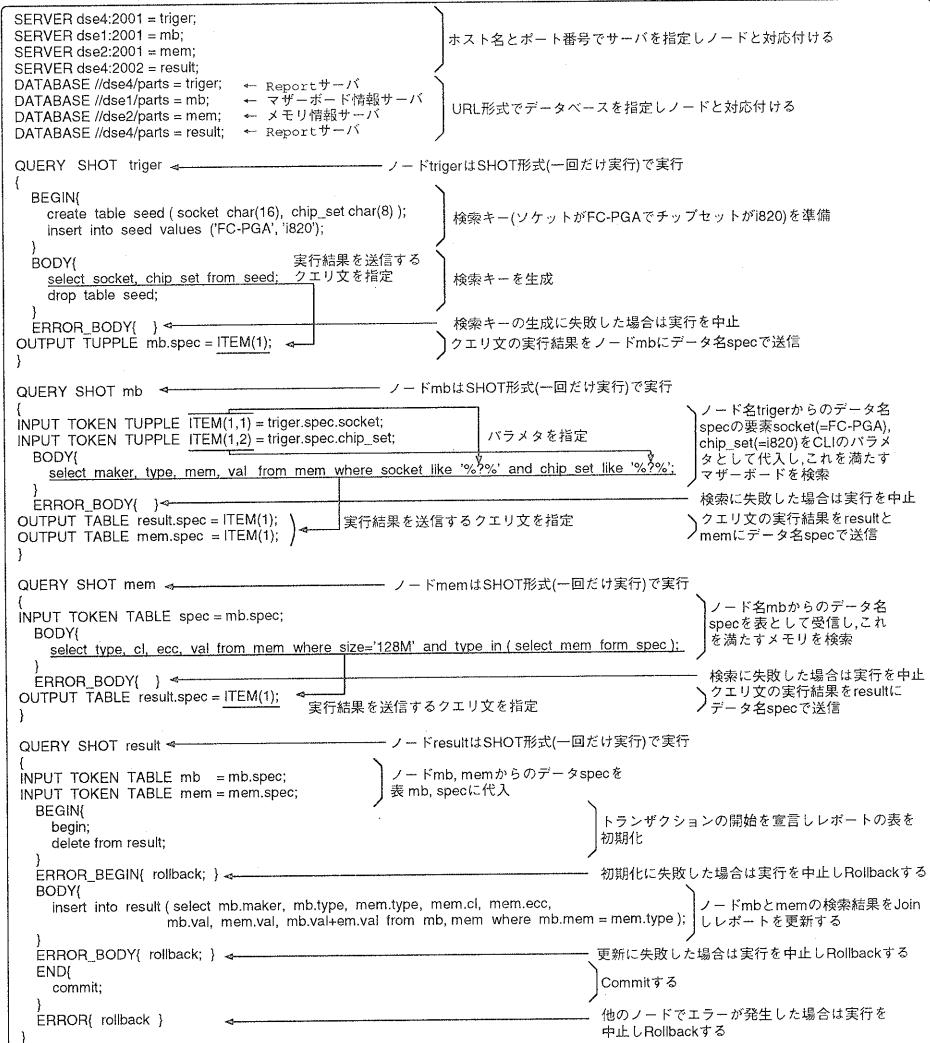


図 7: モバイルコード

maker	mb	mem	cl	ecc	mb_val	mem_val	total
SUPERMICRO	370SCD	PC100 SDRAM	2	-	19000	12000	31000
SUPERMICRO	370SCD	PC133 SDRAM	3	-	19000	12000	31000
Asus	CUC2000	PC100 SDRAM	2	-	20000	12000	32000
SUPERMICRO	370SCD	PC100 SDRAM	2	ECC	19000	14000	33000
Asus	CUC2000	PC100 SDRAM	2	ECC	20000	14000	34000
SUPERMICRO	370SCD	PC133 SDRAM	2	-	19000	19000	38000

図 8: 実行結果