



CORBAを用いた3階層クライアント／サーバ方式によるデータベースシステムについて

メタデータ	言語: jpn 出版者: 室蘭工業大学 公開日: 2007-06-07 キーワード (Ja): キーワード (En): CORBA, ORB, CGI, HTTP, applet 作成者: 越野, 克弥, 畑中, 雅彦, 川原, 徳代, 大岩, 高雄 メールアドレス: 所属:
URL	http://hdl.handle.net/10258/146

CORBAを用いた3階層クライアント/サーバ方式によるデータベースシステムについて

その他（別言語等） のタイトル	Database System by Three-tier Client/Server System with CORBA
著者	越野 克弥, 畑中 雅彦, 川原 徳代, 大岩 高雄
雑誌名	室蘭工業大学紀要
巻	50
ページ	175-182
発行年	2000-11-30
URL	http://hdl.handle.net/10258/146

CORBAを用いた3階層クライアント/サーバ方式による データベースシステムについて

越野 克弥^{*1}, 畑中 雅彦^{*1}, 川原 徳代^{*2}, 大岩 高雄^{*2}

DateBase System by Three-tier Client/Server System with CORBA

Katsumi KOSHINO, Masahiko HATANAKA, Noriyo KAWAHARA and Takao OHIWA

(原稿受付日 平成12年4月28日 論文受理日 平成12年8月31日)

Abstract

Recently various database servers which can be utilized and managed using by web browsing tools are required. Since these servers are going to be connected each other through network, it becomes important how to keep their extensibility and easiness of maintenance. We have studied a system construction method based on the distributed object architecture, Java and CORBA, in considering of its extensibility. In this paper, we report some results of our prototype of three-layers client/server style database system implemented by Java applet and JavaIDL.

Keywords: CORBA, ORB, CGI, HTTP, Applet

1 はじめに

企業内の各種情報システムは、多くの場合データベースを中心にして構築されることが多くなってきている⁽¹⁾。

比較的小規模なシステムを構築する場合には、システム構築の容易さや見通しの良さから、2階層クライアント/サーバ方式を採用することがあ

る。しかし、この方式はシステムの拡張性、保守性が低いという問題を抱えており、規模の大きなネットワークを構築する場合には適していない。このことから、最近ではこれらの問題を解決することができる CGI (Common Gateway Interface)⁽²⁾を用いた3階層クライアント/サーバ方式が活用されるようになってきている。しかし、CGIを用いたデータベースへのアクセスは、非常にアクセス効率が悪く、また WWW (World Wide Web) サーバを必ず中継しなければならないので WWW サーバにも大きな負荷がかかるという問題がある。

^{*1} 室蘭工業大学

^{*2} 北海道エニコム(株)

そこで筆者らは、3 階層クライアント/サーバ方式のシステムを構築する手段として、CGI を用いる代わりに、Java の分散オブジェクトと、その分散オブジェクト間通信のための規格である CORBA(Common Object Request Broker Architecture)⁽³⁾ を用いたシステムの構築を行なったのでここに報告する。

2 従来のクライアント/サーバシステムについて

2.1 2 階層クライアント/サーバシステムについて

2 階層のクライアント/サーバ方式を取り入れたシステム (以降、2 階層クライアント/サーバシステムと呼ぶ) はネットワークを介したデータベースシステムを構築する際の、最もシンプルな形態である (図 1)。

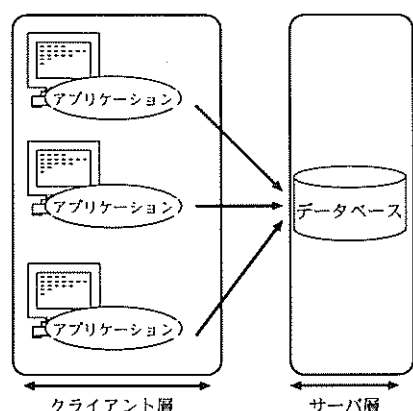


図 1: 2 階層クライアント/サーバシステム

このシステムは、データベースを持つサーバ層と、データベースに対しアクセスを行うクライアント層の 2 階層から構成されている。システムの構成を 2 階層にすることによって、ユーザインターフェース部分とデータベース部分を分離でき、ネットワークを介したデータベース操作が可能となる。

2 階層クライアント/サーバシステムが採用される理由として、以下の点が挙げられる。

- システムの構成が非常にシンプルなので、構築が容易である。
- クライアントアプリケーションがデータベー

スに直接アクセスしているので動作が非常に高速である。

このような特徴がある一方、問題点として保守性、拡張性が低いことが挙げられる。これは、システム内のクライアント全てにデータベースへアクセスするためのアプリケーションを持たせなければならない、システムの機能拡張を行なう場合、全てのクライアントのアプリケーションを更新しなければならないためである。

また、クライアントからのデータベースに対する直接的なアクセスにより、データベースに多大な負荷がかってしまう。

この種の問題はネットワークの規模が小さい場合には無視できるが、ネットワークの規模が大きくなるにつれ問題は深刻になる。

2.2 3 階層クライアント/サーバシステムの特徴

前節で述べた問題を克服するシステムとして、2 階層クライアント/サーバのシステム間にもう一層、中間層を加えた 3 階層クライアント/サーバ方式によるシステム (以降、3 階層クライアント/サーバシステムと呼ぶ) がある (図 2)。

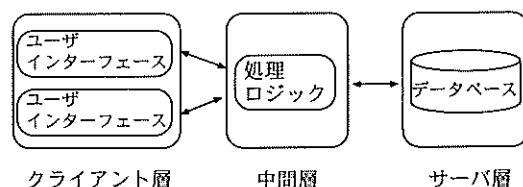


図 2: 3 階層クライアント/サーバシステム

3 階層のシステムの場合、中間層の部分にデータベースへのアクセスや、データの処理といった処理ロジックを持つので、クライアント側のアプリケーションはユーザインターフェース機能のみを提供するだけでよくなる。このことから、システム内の全てのクライアントの処理ロジックを共通化できるため、システムを拡張する際も、中間層にある処理ロジックのみを変更すれば良く、システムの拡張、保守が容易となる。

また、クライアントからのデータベースへの要求は中間層の処理ロジックが行う。このことから、定期的に複数のクライアントから同じ情報への検

索要求などがあるシステムなどでは、中間層が代表してデータベースから情報を引出し、複数のクライアントに結果を返すことができ、データベースへのアクセス集中を防ぐことができる。

問題点としては、階層を一層増やす事によって、システムの構築が2階層のシステムより格段に難しくなることが挙げられる。

2.3 CGIを用いた3階層クライアント/サーバシステムについて

現在、3階層クライアント/サーバの形態をとったシステムで最も多く実装されているものに CGI がある (図 3)。

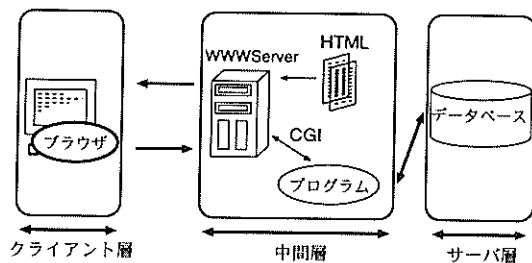


図 3: CGI を用いた 3 階層クライアント/サーバシステム

CGI とは WWW サーバと WWW サーバ上で動作するプログラムとのインタフェースである。クライアントは、WWW サーバから HTML ファイルをダウンロードし、その HTML ファイル内に埋め込まれている CGI フォームを使って、WWW サーバに要求を出す。通常 CGI フォーム内には、サーバに処理させたいデータを記入するテキストフォームやデータを送信する時に押すボタンなどが配置されている。WWW サーバは、クライアントからの処理要求と処理するデータを受けとり、CGI を通してデータを処理するためのプログラムを起動する⁽²⁾。サーバへのデータ送信には、HTML ファイルをダウンロードしてきた時と同様、HTTP(HyperText Transfer Protocol)を使う。CGI を用いたシステムでは、クライアントは HTML ファイル上からサーバ上のプログラムに要求を伝えるので、HTML ファイルが表示できるソフト、つまり WWW ブラウザを持っていればよく、特別なアプリケーションを必要としないという利点がある。

一方、問題点として、HTTP を用いることによるサーバへの負荷増大があげられる。HTTP は現状を保持したままの通信が行えず、毎回クライアントとの接続を切断してしまうという特徴を持っており⁽⁴⁾、以下のような理由によりサーバへの負荷を増大させてしまう。

- クライアントからの要求のたびにプロセスを起動するので、同時に稼働するプロセス数やプロセスの起動回数が増えれば増えるほど、システムのパフォーマンスは低下してしまう⁽⁵⁾。
- WWW サーバは HTML などのファイルの送信後、クライアントとの接続を切断し、データベースとの接続も切断してしまう。よって、クライアントの要求のたびにデータベースへ再接続しなければならない、システムのパフォーマンス低下につながる。

3 CORBA を用いた 3 階層クライアントサーバシステムについて

3.1 システム試作の目的

前述の通り、CGI を用いたシステムではサーバ側のパフォーマンスの低下の問題を抱えており、この結果、クライアントへのレスポンスの悪化、サーバの安定性の低下が引き起こされる。

しかしこれらの問題は 3 階層のクライアント/サーバシステムの形態に起因する問題ではなく、サーバとの通信を HTTP によって行っていることが原因となっている。

そこで筆者らはこれらの問題を解決するために、3階層クライアント/サーバシステムを構築する手段として CGI を用いないシステムの試作を行った。

本システムでは、クライアント、サーバを共に Java のアプリケーションとして作成した。また、クライアントアプリケーションとサーバアプリケーションとの通信には、分散オブジェクト間通信の規格である CORBA を用いた。

3.2 CORBA の特徴

CORBA とは、ORB (Object Request Broker) という分散オブジェクト通信機能を実装した、ソフトウェアの共通仕様を定めたものである。

ORB の主な役割として、ネットワーク上の CORBA によって協調できるクライアントとサーバの所在の管理を行っていることが挙げられる。このことにより、プログラマはオブジェクトの所在を意識しないアプリケーションの作成が可能となる。

もう一つの大きな特徴としては、異種言語、異種 OS 上で動作するオブジェクト同士の通信が可能であることが挙げられる。これは CORBA で協調するオブジェクトを作成する際、IDL (Interface Definition Language) というインターフェース記述言語を用いてオブジェクトのインターフェースを記述するので、言語や OS の差異はそのインターフェース部分で吸収することができるためである。

なお、これらの CORBA の特徴については、以前の論文で報告済みであるので詳しい説明等はそちらに譲る⁽⁶⁾⁽⁷⁾。

3.3 本システムの機能

今回試作したシステムは、3 階層クライアント/サーバ方式を用いたデータベースシステムの一例として、計算機情報の管理システムの構築を行った。データベースには、本研究室で所有する計算機の一覧と、計算機の詳細情報が登録されている。クライアントはクライアントアプリケーション上からデータベース内の情報の検索、更新を行う。

3.4 本システムの構成要素

この節では Java で作成したアプリケーションがクライアント層、中間層においてそれぞれどのように動作しているかを説明していく。

3.4.1 クライアント層

本システムでは、クライアントアプリケーションを Java のアプレットとして配布する方法をとっている。アプレットとは、WEB ページの一部として実行される形式の Java のアプリケーションである⁽⁸⁾。アプレットの本体は WWW サーバ上に

あり、必要に応じてクライアントがダウンロードするので、クライアントは常に最新のアプリケーションを使うことができ、またクライアントプログラム入手の手間も省くこともできる。アプレットは現行のほとんどのブラウザ上で実行することができるので、前節で CGI を用いたシステムの特徴として述べた「クライアントは特別なアプリケーションを必要としない」という利点は本システムでも有効であると言える。

3.4.2 中間層

中間層では、複数のオブジェクト (以降、サーバオブジェクトと呼ぶ) があらかじめデータベースとのコネクションを張り、クライアントからの要求の待ち受け状態となっている。クライアントからの要求に応じてデータベースとの通信を始め、クライアントのアプレットが破棄されると、再び待ち受け状態となる。サーバオブジェクトは常にデータベースとのコネクションを張った状態にあるので、クライアントの要求のたびにコネクションを張り直す必要が無い。また、Java のオブジェクトはすべてスレッドとして生成されるのでシステムのリソースの消費を最低限に抑えることができる。スレッドとはプロセスを細分化したもので、プロセスよりも高速に動作するものである⁽⁹⁾。

3.5 追加機能

今回試作するシステムでは、データベースへのアクセス機能の他に、以前より我々の研究室で実験を進めてきた⁽¹⁰⁾、クライアントの自動識別機能も設けた。これは、アクセスしてきたクライアントをサーバ側で判断し、将来クライアントの表示特性に整合するように情報を整形して、送信する機能を付加するためである。

以前の実験では、HTTP ヘッダを解析する Proxy サーバを作成し、クライアントがその Proxy サーバを通過することによってクライアントの識別を行っていた。しかしこの方法では、全てのクライアントに Proxy を通過する設定を行わなければならないという問題があった。

そこで今回のシステムでは、クライアント側の設定を必要としない方法として、CGI によって環境変数を取得するプログラムを呼び出すシステム

を作成し、クライアントの識別を行った（以降、この CGI が動作する WWW サーバを識別情報取得サーバと呼ぶ）。呼び出されるプログラムは、perl を用いて作成した。

この追加機能部分については CGI に実装を行なったが、これは、本論文の主題である「3 階層クライアント/サーバ方式によるデータベースシステムの構築」という部分には直接関係しない部分であり、CGI は環境変数を取得するプログラムを呼び出す手法として一番容易な方法だと考えたためである。

環境変数とは、CGI が実行されたときにブラウザや HTTP サーバから CGI に渡される変数のことであり、クライアントを識別することが可能な情報も含まれている⁽¹¹⁾。

今回のシステムではクライアントにアプレットを用いているので、アプレット内で、クライアントにダウンロードされた後、自動的に識別情報取得サーバへアクセスするようにプログラミングしておけば、クライアント側での設定必要とすること無く識別情報の取得が行える。

今回の実験で識別情報として取得した情報は、IP、ホスト名である。

4 構成

4.1 開発環境

本システムの実装には Java 言語を使用し、開発環境には JDK(Java Developer Kit) のバージョン 1.2 を使用した。また CORBA 開発環境には、JDK1.2 で提供されている JavaIDL を使用した⁽¹²⁾。

4.2 システムの構成

今回作成したシステムの構成を図 4 に示す。また、このシステム内で使用したソフトウェア、計算機の OS を表 1 に示す。Linux のクライアントについては、Java1.2 が動作する WWW ブラウザの正式なリリースが無いので、アプレットを読み込むためのアプリケーションであるアプレットビューア⁽¹³⁾を使っている。

図 4 に示した本システムの動作の流れを以下に示す。

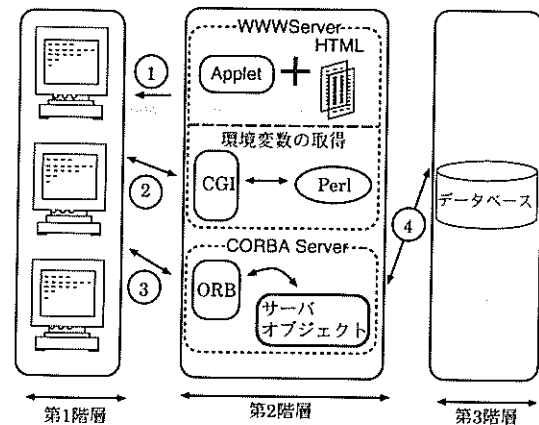


図 4: 今回作成したシステム

表1: 使用ソフトウェアと使用OS

	OS	アプレット実行ソフト
クライアント層	Windows98	Internet Explorer4.0
	Linux 2.2.14	アプレットビューア
	OS	WWWサーバ
中間層	Linux 2.2.14	Apache 1.3.12
	OS	データベース
サーバ層	Linux 2.2.10	PostgreSQL 6.3.1

1. クライアントは、WWW サーバから HTML ファイルとアプレットのダウンロードを行う。
2. ダウンロードされたアプレットは、識別情報取得サーバへアクセスし、IP とホスト名を取得する。
3. サーバオブジェクトへアクセスする。
4. 中間層では、計算機の一覧表をデータベースから取得しクライアントに返す。クライアント側では、識別情報が記されているウィンドウが表示され、その後、計算機一覧表が表示される。

一覧表の表示後は、一覧表より詳細を知りたい計算機、情報を変更したい計算機を選び、中間層へ要求を出す。中間層はその要求を SQL 文に直しデータベースへアクセスを行い結果をクライアントに返す。クライアントのアプレットが破棄された時点で接続が切断され、サーバオブジェクトは再び待機状態に入る。

計算機の一覧を表示する画面を図5に、情報の変更を行う画面を図6に示す。



図 5: 計算機の一覧表の表示画面

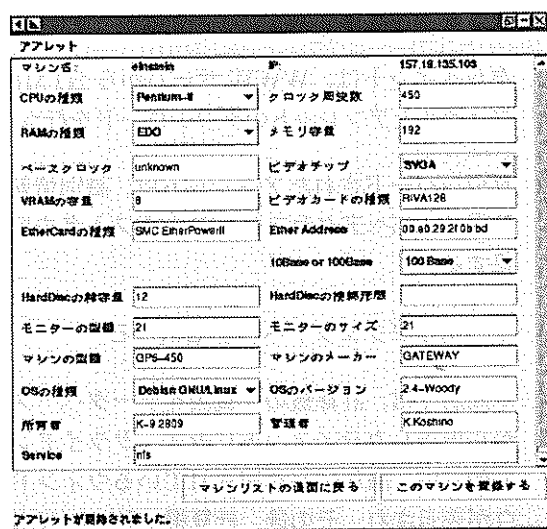


図 6: 情報変更画面

5 動作確認

本システムの動作確認として以下のことを確認した。

- アプレット上から情報を検索したい計算機を選び、データベースから情報を引き出し、表示する。
- アプレット上から、情報を更新、追加、削除したい計算機を選び、データベースから引き出してきた情報の変更を行う。変更後、正常にデータベースに登録された場合は、変更した旨をクライアントに伝える。

また、追加機能であるクライアントの自動識別機能についても正確に取得できているかの確認を行った。

6 結果

動作確認の結果、アプレット上からのデータベースの検索が正常に行なえることを確認できた。また、データベースに登録されている情報の変更も正常に行えることを確認した。すなわち、ユーザがアプレット上で行なった動作を、中間層でSQL文に直し、データベースへ正常にアクセスできたことを確認できたといえる。

また、クライアント自動認識機能についてもクライアントの計算機のIP、クライアント名が正常に取得できていることを確認できた。識別情報が表示されているウィンドウの一部を図7に示す。

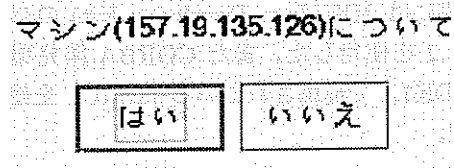


図 7: 識別情報の表示

7 考察

ここでは、実験の目的に対する評価、アプレットの配信における問題点、3 階層クライアント/サーバにおける CORBA の適応、この3つについて考察を行っていく。

7.1 実験の目的に対する評価

今回のシステムでは、中間層にあらかじめデータベースとのコネクションを張ったサーバオブジェクトを複数立ち上げておく方法をとったので、クライアントからの要求が連続して起きても、中間層における新しいプロセス、データベースとのコネクションが発生することが無く、システムのパフォーマンスは一定に保たれると考えられる。

このことから、CGIを用いて3階層クライアント/サーバをシステムを構築した際に発生する問題を解決する方法として、本システムは有効であると考えられる。

今回のシステムでは、最初にサーバオブジェクトを立ち上げた数以上のクライアントからの要求があった場合は対処することができない。しかし、このことによりサーバの処理性能に見合ったオブジェクトの数を設定し急激なクライアント数の増加を規制することによって、システムを安定動作させることができる。

7.2 アプレットの配信における問題

クライアントをアプレットとして配信する場合、考慮しなければならない重要な問題としてダウンロード時間が挙げられる。

今回のシステムでは、アプレットにはユーザインターフェース機能のみを持たせるようにした。これは、クライアントがダウンロードしなければならないアプレットサイズをなるべく小さくする目的と、中間層に処理ロジックを集中させて、拡張性を高める目的があるためである。

今回の実験では、クライアントがダウンロードしなければならないアプレットのサイズは18KBまで抑えることができた。この程度のサイズのアプレットならばダウンロードするのにもほとんど時間もかからなく、許容範囲内であると考えられる。

今回作成したシステムの例では、アプリケーション全体のサイズが38KBあり、アプレットのサイズの割合が半分程度占めていることになるが、これは今回作成したシステムはデータベースへアクセスするためのごく一般的な機能しかもたないため、システム全体のアプリケーションサイズが小さく、その結果、システム全体に対してのユーザインターフェースの割合が大きくなってしまっ

ているためである。今後、機能追加などをして中間層の処理ロジックが大きくなっていった場合でも、アプレットがユーザインターフェース機能しか持たない場合は、そのサイズが劇的に増える事は無いと考えられる。

7.3 3階層システムにおけるCORBAの適応

CORBAではORBがオブジェクトの管理を行っている点、言語やOSなどに依存しない点から、中間層における計算機の追加、サーバオブジェクトの追加が容易に行うことができる。つまり、拡張性の高いシステムの構築が可能であるといえる。また、オブジェクトの再配置が柔軟に行えることから、中間層で処理を行う計算機を複数台用意する事によって中間層への負荷集中も防ぐ事ができる考えられる。

これらのCORBAの特徴は、3階層クライアント/サーバシステムは階層が一層増える分だけ構築が難しいという問題がある程度軽減でき、3階層システムを構築する際、非常に有用であると考えられる。

8 まとめ

今回行なった実験から、3階層クライアント/サーバ方式によるデータベースシステムを構築する際にCORBA/Javaを用いることにより、従来のCGIを用いた3階層クライアント/サーバシステムが抱える問題を克服できることを示す事ができた。

今後の予定としては、本論文3章5節で述べたクライアントの識別情報を用いたコンテンツの自動変換機能の作成を検討している。

現在考えている機能として、クライアントの描画能力にあわせた画像の配信機能がある。

この機能は、サーバ側でアクセスしてきた計算機を識別し、データベースからクライアントの計算機の描画能力を求めて、画像の減色処理を行ったり、画像サイズの小さい物を配信する機能である。この機能によって、高品質な画像などを表示できないクライアントなどに対し、あらかじめ品質を落した画像を提供できるので、クライアントへの負荷の軽減、また、ネットワークトラフィックの軽減が期待できると考えられる。

参考文献

- (1) 好川 哲人, イントラネットとは何か?, Open-Design, CQ 出版社, No16(1996), p4-15
- (2) 佐々木望, 太田晶宏, 藤崎真美, 新・HTML and CGI 入門, エーアイ出版, (1997), p206
- (3) 松野良蔵, Java+CORBA 分散オブジェクトシステム構築, 翔泳社, (1999), p170-180
- (4) Robert Orfali, Dan Harkey (並河英二 訳), Java and CORBA C/S プログラミング, 日経 BP 社, (1997), p197-199
- (5) 渡辺操, 畑中雅彦, 田島和典, 青木貴, WWW における Java Applet の性能評価実験, 平成 10 年度室蘭工業大学紀要, (1998), p78-80
- (6) 越野克弥, 今野竜太, 畑中雅彦, 斉川宏明, 大岩高雄, CORBA 分散オブジェクト環境について (1), 平成 10 年度開発技術研究会, (1998), p15-16
- (7) 今野竜太, 越野克弥, 畑中雅彦, 斉川宏明, 大岩高雄, CORBA 分散オブジェクト環境について (2), 平成 10 年度開発技術研究会, (1998), p17-18
- (8) 有賀妙子, 竹岡尚三, Java プログラミング徹底マスター, ソフトバンク, (1998), p10
- (9) 学研編集部, 情報処理基本用語辞典, 学研, (1998), p136
- (10) 石井和幸, WWW クライアント自動評定のための情報取得法の検討, 平成 11 年度卒業論文, (1999), p21-25
- (11) 佐々木望, 太田晶宏, 藤崎真美, 新・HTML and CGI 入門, エーアイ出版, (1997), p190
- (12) Geoffery Lewis, Steven Barber, Ellen Siegel (日本ユニシス分散オブジェクト研究会訳), JavaIDL プログラミング, カットシステム, (1998), p1
- (13) 有賀妙子, 竹岡尚三, Java プログラミング徹底マスター, ソフトバンク, (1998), p5