

C/Sシステム開発におけるJava利用方式の評価

5H-1

萬木優子 原田道明 鈴木由美子 北畠重信 熊井秀憲* 阿波道雄* 森俊治*

三菱電機(株)情報技術総合研究所 *同 情報システム製作所

1. はじめに

Javaはプラットフォーム中立性やアプリケーション配布コストの削減効果をもち、広く普及しつつある。本論文では、Javaによるビジネス系C/Sシステム構築法の検証を目的として、Javaで可能な種々のアプリケーション形態に対する生産性・性能の比較を行った結果を報告する。

Java上での代表的なC/S間通信方式にはSocket, RMI, HTTP, IIOP等がある。今回は開発言語の混合利用を考慮し、2層形態(JDBCによるDB直接アクセス)、2種類の3層形態に絞って比較した。

通信方式の違いはアプリケーションの設計・部品化などの生産性に大きく影響する。通信方式の違いが性能に与える影響は大幅なものではなく、むしろデータベースアクセスの多重度やサーバプロセスの適切化に考慮が必要であると考察される。

2. アプリケーションの違いと特徴

比較対象としたアプリケーション形態の特徴と構成(図1)を以下に示す。

(1) JDBC 2層

アプリケーションをクライアント・サーバに分離せず、クライアントからJDBCドライバを経由して直接DBにアクセスする。

(2) HTTP 3層

アプリケーションの機能をクライアント・サーバに分離する。クライアントはHTMLページとアプレットで構成し、C/S間の通信はWWWサーバを経由して行う。

サーバ実装形態には幾つかの変種が存在するが、性能測定を目的として①、生産性評価を目的として②の2種類の形態を試みた。

① Javaで記述したCGIコード…WWWサーバの標準的なインタフェースであるCGIを選択し、Javaでサーバを構築した。

② Web拡張製品の利用…CGIを拡張してサーバアプリケーションの部品化や機能拡張を図った製品が幾つか発表されている。データベースアクセス

の容易性・汎用性の高い製品を選択し、生産性向上を評価した。

(3) IIOP 3層

アプリケーションをクライアント・サーバに分離し、C/S間通信はCORBA IIOPを用いて行う。

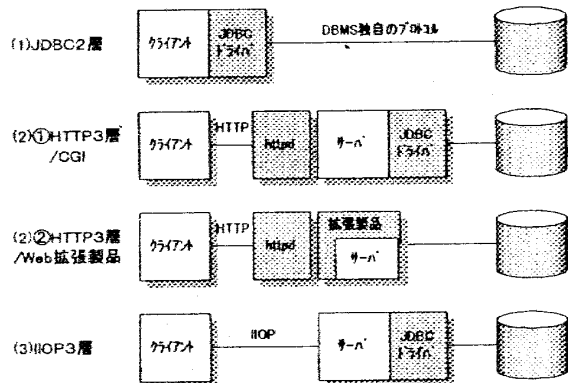


図1. 評価対象C/Sシステムの構成

3. 生産性の評価

業務システムのプロトタイプ開発を通して実装工程の生産性を評価した。作成した業務は定型的DB処理(新規登録、更新、検索)と帳票・印刷処理を含み、画面数7、DBテーブル数7(フィールド数86)である。表1に、検索業務を例に全ライン数と部品化の図られたライン数を示す。

表1: 検索業務の通信方式別ライン数

	JDBC	HTTP (CGI)	HTTP (拡張製品)	IIOP
ライン数	83	136	207	159
うち部品化ライン数			143	78

(1) JDBC 2層

画面と処理の分離が図られず部品化が進まない傾向にある。構築ツールによりSQLや通信処理を意識せずに構築できるため、再利用性や処理の共通化を無視した場合の工数は小さい。

(2) HTTP 3層

① CGI

実業務ではCGIを単独で用いることは少ないと考え、性能評価に特化したサンプル(検索処理)のみ作成した。CGIはセッション管理ができないため、更新処理では制約が多いと考える。

CGIをJavaで記述するには、Javaを起動するためのラッピングが必要である。CGIにJavaを利用した場合、Javaの通信用の機能^[2]を利用でき、通信処理の工数は比較的少ない。

② Web拡張製品利用

Evaluation of usage of Java in client-server system development
Yuko YURUGI, Michiaki HARADA, Yumiko SUZUKI, Shigenobu KITABATAKE,
Hidenori KUMAI, Michio AWA, Syunji MORI
INFORMATION TECHNOLOGY R & D CENTER

* INFORMATION SYSTEMS ENGINEERING CENTER
MITSUBISHI ELECTRIC CORPORATION

今回調査した製品は、SQLに制御構造を持たせた独自言語をもち、複雑なDB処理も合理的コストで記述できる。検索結果のHTMLへの変換を簡便化するライブラリが付属する。今回の業務では以下のような制約があった。

- ・HTTP POSTメソッド^[1]が利用できない。(クライアントからサーバへの大量データ転送が困難)
- ・基幹業務では、GUIの自由度に対する要求が高く、スプレッドシート等のアプレットに業務データを転送する必要が生じる。今回の製品は結果をHTMLページとして生成することを前提としたもので、実装に工夫を要する。データ転送のための部品を作成することにより工数を改善することができる(表1の部品化ライン数参照)。

なお、同製品は画面構築要求がHTMLページの生成のみで十分な場合には構築手順も単純で、頻出する画面レイアウト・画面要素を部品化することができたことを付記しておく。

(3) IIOP 3層

開発を始める前に、IDL等の基本的な概念についての学習が必要であった。通信処理はIDLから自動生成されるため通信を意識せずに構築できる。現状のIDLの制限として、基本型以外は参照渡しとなるため、転送データを基本型の配列に変換する必要がある。オブジェクト指向であり、部品化に適している。

4. 性能評価

性能評価はDB検索処理について、転送するデータ量と速度の相関、処理時間の内訳、同時接続数とサーバの負荷の相関を調べた。性能測定の際のHTTP 3層のサーバにはCGIを選択した(DBアクセスをJDBC経由としプログラムを統一するため)。

(1) データ量と速度

どの方式でもデータ量に比例してTAT(ターンアラウンドタイム)が延びる。方式間では、JDBC、IIOP、HTTPの順に速く、IIOPはJDBCの1.2倍、HTTPは2倍程度である。

内訳は、DBへの接続にかかる一定の時間+データ量に比例して増加する時間(検索結果からデータを取り出す時間、サーバからクライアントへデータを送信する時間、送られたデータをクライアントが読み込む時間)である。

IIOPはJDBCと比較すると通信+クライアント側読み込み分だけTATが長いがこの時間はTATの15%程度である。IIOPでの通信のオーバーヘッドはそれほど大きくないと言える。またIIOPではサーバでのデータの読み取りを一度に行わず、何回かに分けて取り出すようにし反応時間を短縮することができる。

HTTPでは、クライアントからのリクエスト送信、サーバ処理、サーバからのレスポンス送信をオー

バラップさせることができ、その分反応時間を短縮できる。しかし、結果を分けて返すことはできないため、それ以上の反応時間の短縮にはリクエスト自体を工夫する必要がある。

(2) 同時接続数とサーバの負荷

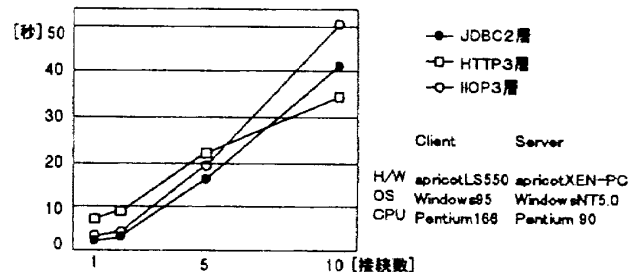


図2: 同時接続数と速度の相関

図2はクライアントをマルチスレッドにしてサーバへのクライアント接続数を擬似的に増やし、サーバの実行時間を調べたものである(比較のため、JDBC 2層もクライアントを他の2形態のサーバマシンに載せて測定)。

JDBC 2層とIIOP 3層では、同時接続数に比例して実行時間が長くなっている。HTTPがこの中で速いのは、CGIがリクエスト毎に独立プロセスで実行されるためと考えられる。一方IIOPが遅いのはサーバがマルチスレッドでもDBアクセスを行なうJDBCドライバが処理を直列化し性能が出ないためと考えられる。しかしIIOPのサーバも基本的にCGIと同じ処理を行っており、複数プロセスにすることによりシングルスレッドの場合の比率まで性能を改善できることが予想される。またIIOPではサーバを複数マシンに分散させることによる改善が容易である。

5. まとめ

Javaを利用したC/Sの3つの形態について、プロトタイプ作成を通して実装工程の生産性について評価した。また、検索処理について性能測定を行った。その結果、以下のことがわかった。

JDBC 2層は、部品化に適さないが、再利用性や処理の共通化を無視した場合の工数は小さい。

HTTP 3層では、サーバの実装形態の選択が重要であり、部品化により生産性を向上させることができる。性能的には今回の測定規模では問題ない。

IIOP 3層は、通信処理がIDLから自動生成されるため、通信に関する工数は少ない。また、オブジェクト指向であり、部品化に適している。性能的にはプロセス利用の適切化が必要である。

[参考文献]

- [1] エリック・ハーマン、「CGI入門」、プレントイスホール
- [2] 電通国際システム、「Javaイントラネット構築技法」、アスキー
- [3] ロバート・オーファリ他、「Java&CORBA C/Sプログラミング」、日経BP