

Web アプリケーションサーバアーキテクチャへの Java の貢献

早稲田大学 理工学部 情報学科

深澤 良彰

fukazawa@fuka.info.waseda.ac.jp

大規模基幹アプリケーションを実現するための方法として、インタフェースとして Web をもち、実行機構として分散オブジェクト環境でのクライアントサーバ型アーキテクチャを採用している Web アプリケーションサーバ方式が注目を浴びている。本論文では、このような Web アプリケーションサーバ方式のソフトウェアを開発するための支援環境の動向について、特に Java 関連の技術に絞って述べる。さらに、このような形態のソフトウェアの普及が与える影響について分析する。

Contribution of Java to Web-Application Server Architectures

Waseda University

Yoshiaki Fukazawa

As a framework for implementing large-scale business applications, Web-application server architectures has been paid great attention. These Web-application servers have a Web-browser as their user interfaces and their basic architecture is the client/server style on distributed object-oriented environments. In this paper, the trends of development environments for Web-application servers are described focused on the Java related issues. Also the influences of popularization of these Web-application servers are analyzed.

1. 開発環境/ツールの分類と推移

ネットワーク技術の普及、柔軟なソフトウェアアーキテクチャへの要求などに伴い、クライアントサーバ型のアプリケーションが多く開発されるようになってきた。特に、Web をインタフェースとするものがその中心となってきている。

Web ベースのソフトウェア開発技術は、サーバ側の開発技術とクライアント側の開発技術の二つに分けて考えるべきである。この理由は、ソフトウェアの規模、性質、ハードウェアや OS などの環境がこの両者の間で大きく異なるためである。

クライアント側において注意すべき点は、プラットフォームが種々雑多であることである。Java が、これに対して果たした貢献としては、その処理系が JVM (Java Virtual Machine) を前提としたコンパイラとそのインタプリタから構成されていることがその中心である。

一方、サーバ側の技術は多岐に渡るが、これらを次の二種類に大別する (図1参照)。

- ・サーブレット、JavaScript、JSP (Java Server Page) と HTML 記述支援機能を組み合わせることにより、HTTP ベースの Web アプリケーションの開発を支援する技術

- ・JavaRMI、CORBA、EJB などの分散オブジェクト技術を用いたアプリケーション開発を支援する技術

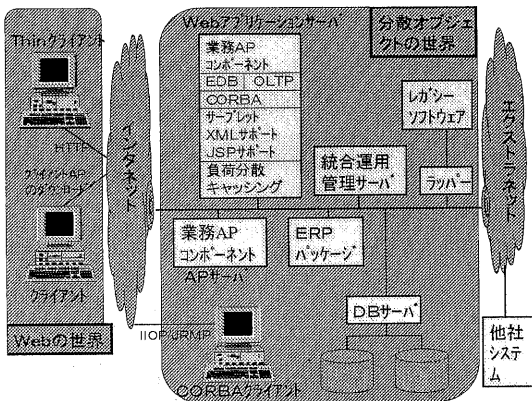


図1 Webアプリケーションサーバの構成

この両者におけるこれまでの問題点と解決への道筋をまとめてみる。

まず、前者を詳細化してみると、

- ・静的コンテンツ部分のオーサリング：FrontPage や Fusion など数多く市販されている HTML オーサリングツールを利用
- ・動的コンテンツ部分のスク립ティング：Perl なら通常のテキストエディタを利用、Java アプレットやサーブレットを作成する場合には、Java 開発ツールを利用
- ・コンテンツの管理と Web サーバへの配布：OS 提供の基本機能を利用

と、各局面で、さまざまなツールを使いこなすことが要求されてきている。そこで、これらを統一された一つのツールを用いて開発することが望まれてきている。たとえば、WebSphere Studio は、このような開発環境の一つである。

後者の問題点は、前者に対して、より複雑な手順が要求されることである。たとえば、CORBA を用いた開発手順例を次に示す。

- 1) IDL ファイルの作成
- 2) IDL からスタブとスケルトンを生成
- 3) サーバプログラムの実装
- 4) サーバプログラムのリポジトリへの登録
- 5) クライアントプログラムの実装
- 6) クライアントの起動

この過程を簡素化するために、言語処理系（たとえば、VisualAge、Jbuilder、VisualCafe など）にさまざまな工夫がなされてきている。最もよく利用される手法は、アプリケーション生成系を用意し、IDL から必要なファイル群を生成し、必要最小限の記述ですませることである。

この流れが最も強く反映されているのが、基幹業務を含めたビジネスアプリケーションの分野で Java を普及させるための技術である EJB である。Javabeans と同様に、開発ツールの役割りを重要視し、できるだけコーディング量を削減しようとしている。

以上見てきたように、Web アプリケーションサーバの開発環境/ツールは、雨後の筍のように生まれてきた各種技術を統合する方向へと進んできている。現時点では、一つのツールを使いこなすだけで、大規模基幹アプリケーションを開発できるところまでは達していないが、明らかにその方向へと進んできている。今後、ソフトウェア開発における諸概念の整理と、開発環境/ツールへのその反映が待たれる。

2. Web アプリケーションサーバの影響

PC・WS の機能・性能の向上により、徐々に規模が大きくなアプリケーションが PC・WS 上で実現されてきている。そして、ネットワーク性能の向上によるクライアントサーバアーキテクチャの実用化により、いよいよ大規模基幹アプリケーションをその範囲に収めようとしている。このような大規模基幹アプリケーションは、これまで一部のメインフレームメーカの寡占状態が続いていた。

これに対して、Web アプリケーションサーバの普及によって、必然的に、オープン化が進むことが期待される。たとえば、CORBA では、仕様しか規定されていないので、どんなソフトウェアメーカでも、CORBA ベースの分散ミドルウェアを開発・販売できる。ただし、最後に残っている信頼性の壁がある。たとえば、ソフトウェア的には、「枯れた」ソフトウェアが生まれにくいという現状がある。「枯れる」前に、次々と新しく発表される環境上に移植され、新たなバグが埋め込まれるからである。

Java に焦点を絞ると、更なる問題点も存在する。Java の大きな特徴は、各種の API が完全に公開されていることである。これが、オープン化を推進する原動力である。しかし、Java の各種の API を決定しているのは、特定の企業である。Web アプリケーションサーバの普及が、大規模基幹アプリケーションの特定企業による寡占状態から、別の特定企業の寡占状態へ移るだけでないことを祈りたい。