マルチメディア通信と分散処理　97－12
コンピュータセキュリティ　8－12
（2000．3．21）

Javaによる組み込み用Webサーバの試作と評価

山口智久[†]　峯村治実[†]　大野次彦[†]　久山和宏[‡]　下間芳樹[†]

[†]三菱電機株式会社　情報技術総合研究所

[‡]三菱電機株式会社　電力・産業システム事業所

インターネットの普及に伴い、Webブラウザをインタフェースとするアプリケーションシステムが増えてきている。また、システム全体のコストダウンが図れるという利点から、組み込み機器をWebブラウザで監視・制御を行いたいというニーズが高まってきている。これを実現するために、われわれはWebによる遠隔監視・制御機能とJavaによるアプリケーション構築の容易性を提供する組み込み用のコンパクトなWebサーバ（開発コード名TSUBASA）の開発を行ってきた。TSUBASAでは、サーブレットによる動的なコンテンツの送信、動的なモジュール交換、遠隔からのさまざまな管理を行うことができ、またこれらの機能をコンパクトなサイズで実現できた。

A prototype of embedded Web server with Java and its evaluation

Tomohisa YAMAGUCHI[†], Harumi MINEMURA[†], Tsugihiko OHNO[†],
Kazuhiro KUYAMA[‡], Yoshiki SHIMOTSUMA[†]

[†]Information Technology R & D Center, Mitsubishi Electric Corporation

[‡]Energy & Industrial Systems Center, Mitsubishi Electric Corporation

Increase of application systems using Web browser interface brought by the spread of the Internet has roused the needs for control and monitoring of network-connected devices with Web browsers. Reduction in costs of whole the systems enabled by Web-based control and monitoring is the most important readn for the needs. To meet the needs, we have developed a compact embedded Web server program, named "TSUBASA". TSUBASA realize functions of contents sending and dynamic module exchanging and remote administration with very compact size.

1.はじめに

インターネットやイントラネットの普及に伴い、ほとんどのPCにWebブラウザがインストールされるようになってきている。またアプレットを利用することにより、Webブラウザ上で様々なアプリケーションに対応することが可能になってきた。このような背景から、Webブラウザの統一的なインタフェースを使用して、種々のデバイスの監視・制御を行いたいという要求が高まってきている。このような要求に対し、われわれは種々のデバイスへの組み込みを可能とするコンパクトなWebサーバ（開発コード名TSUBASA）の開発を行ってきた[1]。組み込み用Webサーバは、Webサーバ機能の中からそのシステムに必要な機能のみを組み込み、Webブラウザからそのシステムの

情報を見たり、システム内の各デバイスの制御を行えるようにするものである。今回、この組み込み用Webサーバを Java によって作成した。Java で作成することにより、JavaVM が動作するプラットフォーム上でならどこでも動作させることができる。また、この組み込み用 Web サーバの機能拡張用にサーブレット実行環境を搭載した。このことにより組み込み用 Web サーバ上でのアプリケーションを ServletAPI を利用した Java アプリケーションとして作成できるため、各デバイスに対応したアプリケーションを容易に作成できるようになる。以下に Java による組み込み用 Web サーバの実現方式とその評価結果について述べる。

2. 組み込み用Webサーバ実現方式の検討

組み込み用Webサーバでの用件として以下の6つがあげられる。

・コンパクトなサイズ
・各機能のモジュール化
・動的なモジュールの交換
・機能拡張モジュール用標準 I/F の提供
・セキュリティ
・リモート管理

2.1.コンパクトなサイズ

組み込みデバイスは、通常コストなどの面から、搭載されるメモリが少ないため、小さなメモリに格納でき、かつ実行できる必要がある。

2.2.各機能のモジュール化

デバイスには様々なものがあるため、用途や各デバイスの能力などに応じて適切なモジュールのみから構成される必要がある。

2.3.動的なモジュールの交換

組み込みデバイスは、一度起動して、運用が開始されてしまうと再起動することが難しい場合があり、モジュールの交換も動的に行える必要がある。

2.4.機能拡張モジュール用標準I/Fの提供

機能拡張モジュールはデバイスごと、アプリケーションごとに開発することになるため、開発者が容易に作成できるようにするための標準的なインタフェースが必要である。

2.5.セキュリティ

アプリケーションによっては、監視システムなど機密性の高い情報を送受信することもあり、また不正な制御を行うと大きな被害が発生するデバイスの制御に用いることも考えられるため、第三者からの不正アクセスやデータの改竄を防止する必要がある。

2.6.リモート管理

組み込み用 Web サーバはデバイスに組み込まれるため、外部からのアクセス手段がネットワークに限られてしまうような場合に、ネットワークを通してのリモートからのサーバの設定を行える必要がある。

2.7.組み込み用Webサーバの構成

これらの要件に適合する組み込み用 Web サーバの方式検討を行った。この構成を図 1に示す。

構成としてはベースモジュール、コアモジュール、HTTP モジュール、リモート管理モジュール、機能拡張モジュールからなる。ベースモジュールはコアモジュールのロードと起動および再起動のみを行う非常にシンプルなモジュールでる。このようにベースモジュールは非常にシンプルな機能しか持たないため、バージョンアップを行う必要がない。このため、ベースモジュールのみをデバイスに組み込んでおき、他のバージョンアップを行う可能性のあるモジュールはリモートにおいておき、必要な時にネットワークを通してダウンロードし、実行させるという構成も可能となる。ベースモジュールを導入することにより、デバイス上のモジュールを全てバージョンアップすることができるようになり、またデバイスで必要とする記憶装置（メモリ、HD など）の量を小さくする

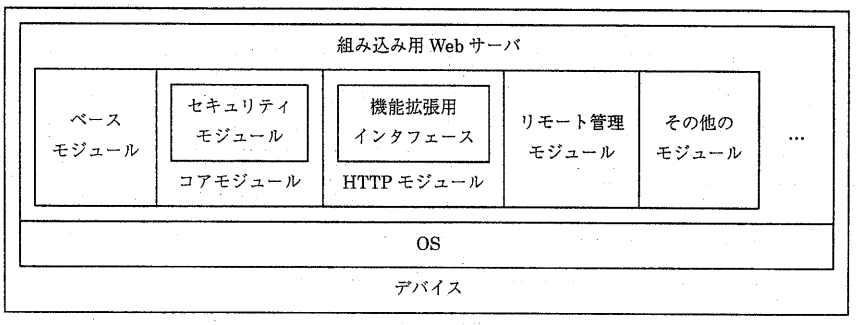


図 1　組み込み用 Web サーバの構成

ことが可能になる。 コアモジュールは HTTP モジュールやリモート管理モジュールなどの各モジュールを管理し、動的なモジュールの交換を行う。またコアモジュールにはセキュリティモジュールを組み込み不正アクセスや改竄を防止する。HTTP モジュールは Web サーバとしての機能を実現し、また機能拡張モジュールに対して標準的なインタフェースを提供する。リモート管理モジュールはリモートからの組み込み用 Web サーバの操作を提供する。機能拡張モジュールは各デバイスに応じたアプリケーションである。

3.構成

3.1.TSUBASAのシステム構成

今回Java で試作したTSUBASA のシステム構成を図 2に示す。

TSUBASA はインターネット／イントラネット等のネットワークに接続されたデバイス上の JavaVM 上で動作し、同じくネットワークに接続されている端末上の Web ブラウザからの HTTP 要求を受け取り、この要求に従い、HTML、アプレットなどの静的なコンテンツやサーブレットの実行によって得られる動的なコンテンツを HTTP 応答として送信する。通信プロトコルは HTTP/1.1 (RFC2616[2]) を使用する。端末側で用意するソフトウェアは Web ブラウザだけでよいため、i モード携帯電話のように Web ブラウザを搭載した全ての端末からデバイスへのアクセスが可能になる。

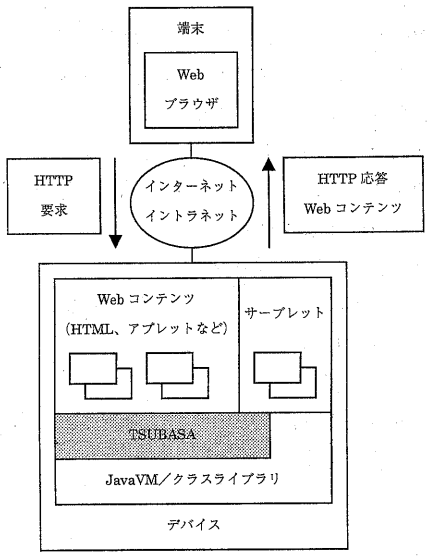


図 2　システム構成

3.2.TSUBASAのソフトウェア構成

今回試作した TSUBASA のソフトウェア構成を図 3に示す。

今回試作した TSUBASA を構成するソフトウェアは大別してコアモジュール、HTTP モジュール、サーブレット実行モジュール、リモート管理

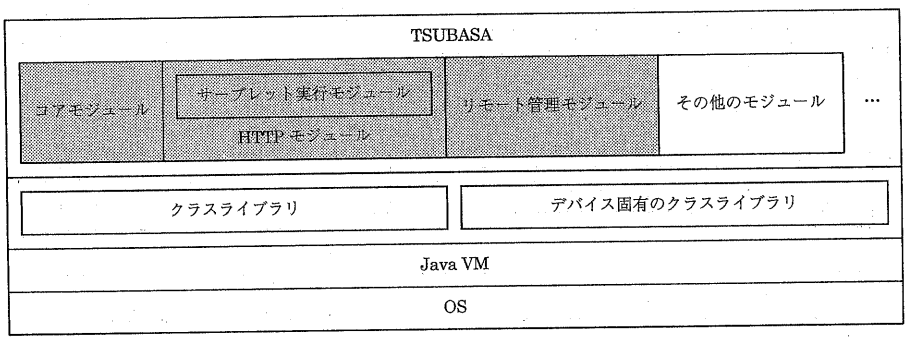


図 3　ソフトウェア構成

モジュールからなる。各モジュールの概要を以下に示す。

・コアモジュール：
HTTP モジュール、リモート管理モジュールなどのモジュールの管理を行い、モジュールの動的な交換をサポートする。
・HTTP モジュール：
Web サーバとしての機能を実現し、サーブレット実行環境も提供する。
・サーブレット実行モジュール：
サーブレットをロードし、実行する。Servlet API を提供する。
・リモート管理モジュール：
組み込み用 Web サーバのリモートからの操作を可能にする。

4. 機能

今回試作した TSUBASA の機能を以下に示す。機能は大きく分けて、動的モジュール交換機能、HTTP 機能、サーブレット実行機能、リモート管理機能に分けられる。

4.1.動的モジュール交換機能

動的モジュール交換機能は、HTTP モジュール、リモート管理モジュールなどのモジュールを、TSUBASA を再起動することなく交換する機能である。動的モジュール交換機能はコアモジュールで実現している。コアモジュールでは各モジュールの状態や依存関係などを管理しており、モジュールを交換する際には、以下の手順で行う。

①モジュールの停止
②モジュールの交換
③モジュールの開始

①のモジュールの停止では、もし交換しようとしているモジュールの機能が他のモジュールから利用されているのであれば、まずその交換しようとしているモジュールの機能を利用しているモジュールを停止してから交換しようとしているモジュールを停止する。

②のモジュールの交換では、モジュールを構成するクラスファイルのダウンロードを行う。ダウンロード元は URL で指定できるところであればローカルでもリモートでもかまわない。

③のモジュールの開始では、もし①のモジュールの停止手順で、交換しようとしているモジュールの機能を利用していたために停止させられたモジュールがあった場合には、交換したモジュールを開始した後、開始される。

以上のような手順により、モジュールの交換後も交換前と同様に動作することができる。

4.2.HTTP機能

HTTP 機能は、Web ブラウザからの HTTP 要求に従い、HTTP 要求で指定されたリソースに対するコンテンツを HTTP 応答として返す機能である。HTTP 機能は HTTP モジュールで実現している。HTTP モジュールでは、HTTP の処理として HTTP/1.1 で必須 (must) と定義されている部分の実装を行っており、これに加えて、コンテンツの送信を効率良く行うためのパーシステント・コネクションもサポートしている。また、HTTP モジュールでは、Web ブラウザとの同時接続数の管理、Web ブラウザからの HTTP 要求での対象であるリソースの管理 (リソース名と実際のファイルとの関連付けやリソースのアクセス権の管理など) を行う。リソースとしては、HTML ファイルや画像ファイル (GIF や JPEG) などの静的なファイルとサーブレットを扱う。

4.3.サーブレット実行機能

サーブレットは CGI (Common Gateway Interface) のように Web サーバの機能拡張を行うためのモジュールで、アプレットと同様に専用の実行環境にロードされて実行される。また Servlet API はこのサーブレットを作成するための API で、Java の拡張 API (javax) として提供されている[3]。今回試作した TSUBASA でのサーブレット実行機能では、Servlet API 2.0 をサポートする。このサーブレット実行機能は HTTP モジュール内のサーブレット実行モジュールで実現している。サーブレットは HTTP モジュールに登録することによって、Web ブラウザからアクセス可能となり、登録名 (エイリアス名) によってアクセスすることができるようになる。

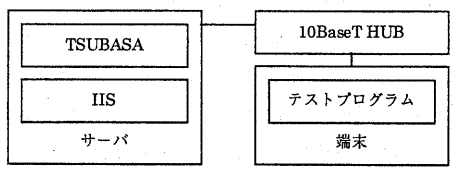
4.4.リモート管理機能

リモート管理機能は、端末上の Web ブラウザから、遠隔地にある TSUBASA を操作するための機能である。リモート管理機能はリモート管理モジュールで実現している。リモート管理モジュールでは、TSUBASA 操作用ホームページを提供して、TSUBASA を操作したいユーザはこのホームページにある操作コマンドのリストから操作コマンドを選択したり、パラメータ入力フィールドにパラメータを入力することによって操作を行う。このホームページはリモート管理モジュールで提供するサーブレットが動的に作成する HTML で記述されたページであり、コマンドやパラメータの入力には FORM を使用する。

5.TSUBASAの評価

5.1.応答性能

今回試作した TSUBASA の評価として、端末上のテストプログラムからコンテンツを要求してからコンテンツが返されるまでの応答時間の測定を行った。測定環境の構成を図 4に示す。



CPU:MMX Pentium 200MHz
Memory:64Mbyte
OS:WindowsNT4.0 Server
Java:JDK1.1.6

CPU:PentiumIII 550MHz
Memory:256Mbyte
OS:Windows98
Java:JDK1.1.7

図 4　応答性能測定の測定環境

テストプログラムは Java で記述されたプログラムで、java.net.Socket を使用して Web サーバに接続し、HTTP の GET メソッドを使用してコンテンツの要求を行い、コンテンツの要求からコンテンツが全て返されるまでの時間を測定するプログラムである。使用したコンテンツデータは5K、10K、20K、50K、100K、200K のサイズのものを使用した。また参考として同じ環境で IIS[4] (WindowsNT 上の標準的な Web サーバ) でも同様の測定を行った。この測定結果を図 5に示す。測定では 3 回測定を行い、その平均をとったものである。

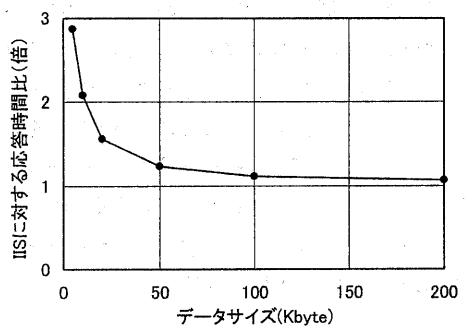


図 5　応答性能の測定結果

図 5より、応答時間に関してはIISに比較して最大で約3倍であるが、組み込み用に使用されるため、実用上問題ない範囲であることがわかった。

5.2.モジュールサイズ

今回試作したTSUBASAのモジュールサイズはトータルで約100Kbyteであった。このサイズはクラスファイルでのサイズであり、実際の運用時に使用されるjarファイルでは約1/2程度になる。今回の試作ではモジュールサイズのチューニングを行っていないが、非常にコンパクトなサイズで実現できることを確認した。

5.3.動的モジュール交換

今回試作したTSUBASAの動的モジュール交換の動作確認として、リモート管理モジュールおよびHTTPモジュールの交換を行った。

5.3.1.リモート管理モジュールの交換

TSUBASAではリモート管理モジュールを利用している他のモジュールはないため、リモート管理モジュールの交換では以下に示す手順で交換が行われる。

①リモート管理モジュールの停止
②リモート管理モジュールの交換
③リモート管理モジュールの開始

5.3.2.HTTPモジュールの交換

TSUBASAではHTTPモジュールはリモート管理モジュールから利用されているため、HTTPモジュールの交換んでは以下に示す手順で交換が行われる。

①リモート管理モジュールの停止
②HTTPモジュールの停止
③HTTPモジュールの交換
④HTTPモジュールの開始
⑤リモート管理モジュールの開始

以上のような手順でリモート管理モジュール、HTTPモジュールの交換が行われることを確認できた。

6. おわりに

今回、Javaによる組み込み用のWebサーバの方式検討を行い、これにもとづき試作を行い、評価を行った。今回試作したTSUBASAはJavaで記述されているため、JavaVMがプラットフォームである環境ではどこでも動作することができる。また、通常のHTMLコンテンツはもちろん、サーブレットを使用することによってデバイスの状態などの動的な情報をブラウザに返すことができる。またモジュールの動的交換処理の確認も行うことができた。今後は今回の試作では組み込まなかったベースモジュールおよびセキュリティモジュールの組み込みや機能拡張および性能の向上を図っていく予定である。

参考文献

[1] 山口他:組み込み用Webサーバの試作と評価,情報処理学会論文誌,Vol40,No11,pp.4147-4150(1999)
[2] R.Fielding他:RFC2616、Hypertext Transfer Protocol--HTTP/1.1、June、1999
[3] http://java.sun.com/products/servlet/
[4] http://www.microsoft.com/japan/products/iis/