卒業論文

PostgreSQL と Servlet による Web アプリケーション開発

高知工科大学
電子・光システム工学科

小松　勇貴

平成 13 年 2 月 9 日

要旨

インターネットが広がり、ウェブショッピングをはじめとする、Web アプリケーションの利用が増えてきた昨今、Web アプリケーションを設計する上で、パフォーマンスは重要な要素になっている。そのパフォーマンスを大きく左右するのが、サーバ層におけるアプローチの選択である。ここでは、PHP4 、Servlet 、CGI-Perl 、などを用いた場合の、パフォーマンスの比較を行う。

目次

第 1 章

はじめに

1.1　Web アプリケーションとは

Web アプリケーションは、動的なウェブサイトまたはウェブページ全般を指す。動的なウェブサイトとは、ユーザからの要求に応じてページを生成するサイトのことで、例をあげると、大学の図書館での蔵書検索、オンラインショッピング、掲示板、ネットオークションなどたくさんある。

上であげたサイト以外にも、今まではローカルのパソコンの中で動いていたアプリケーションを、オンライン上で利用できるようにしたサイトも、最近では多く。ウェブサイト上でメールの送信や受信を行うことができたり、スケジュール管理を提供しているところもある。またインターネット以外では、イントラネット上に、配置したサーバを利用して、会社のグループウェアを運営したり、企業間の取引を、Web アプリケーションの技術を用いて、行っている例もある。

1.2　Web アプリケーションの利点

従来のシステムでは、サーバとのやり取りに、専用のクライアントアプリケーションを使ってきた。この方法はサーバサービスごとにクライアントアプリケーションが必要で、さらにユーザが使う OS が違えば、OS ごとにクライアントアプリケーションを用意する必要があった。これに対し、Web アプリケーションは、サーバ・クライアント間でのデータのやりとりを HTML/HTTP プロトコルに準拠して行うので、インターネットが普及して、過程や会社のパソコンにウェブブラウザが標準に備えようになった今、クライアントを選ばずにサーバ・サービスを提供することができるのである。最近では、パソコンだけではなく携帯電話や、一般の電話にもウェブブラウザが搭載されており、ターゲットユーザはさらに広がっている。

第２章

Web アプリケーションの仕組み

2.1　３層モデル (3 tier model)

高度で複雑な機能を提供する Web アプリケーションは、３層モデル（3 tier model）に基づいて作られることが多い。３層は順に、ユーザにインターフェイスを提供するためのプレゼンテーション層、ユーザに各機能を提供しそこから発生する要求を処理するサーバ層、サーバ層によって処理されたデータを保存したり、保存しているデータを参照要求に応じて取り出したりするデータベース層と定義されている。

３層にすることで、負荷、リスクの分散、汎用性の向上などいくつかの利点が生まれる。Web アプリケーションにおいては、プレゼンテーション層にあたるのがウェブブラウザで、サーバ層に当たるものがウェブサーバとサーバプログラム、データベース層にあたるのが、データベースサーバである。

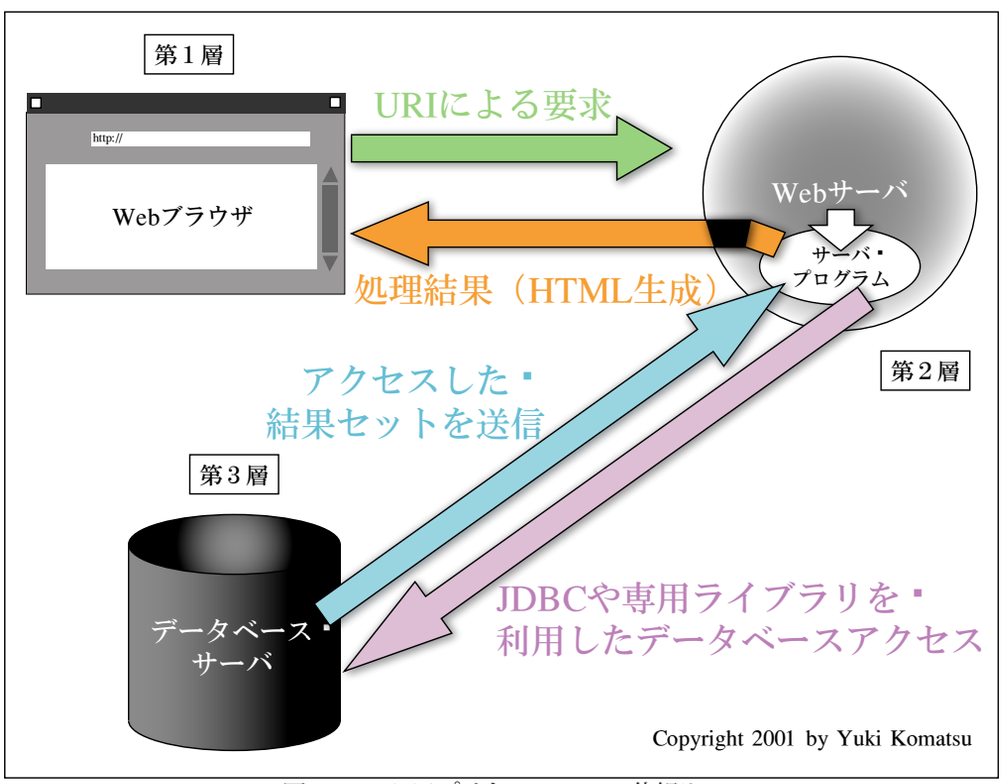


図１．Web アプリケーションの仕組み

2.2　サーバ層の構成

2.2.1　CGI

現在、サーバ層を構築する手段として最もポピュラーなものは、ウェブサーバと CGI（Common Gateway Interface）の組み合わせである。CGI とはウェブサーバが、外部プログラムを使うための手法で、Perl をはじめとして、C 、C++ や Ruby といったローカルでも使われている言語を､用いたプログラムを呼び出して実行することができる｡サーバプログラムはウェブサーバから URI による要求を受けて、ユーザに送る HTML を動的に生成する。

この方法は一般的で事例も多く導入が簡単なため、よく用いられるが、外部プログラムを呼び出すという仕組み上、プロセスを新たに一つ作るとになり、サーバに対する負荷が高い。プロセスが増えるとプログラムの起動コストが処理の大半を占めるようになり、同時に複数のユーザが利用することの多いサービスには不向きである。

CGI ではないが、CGI で培った資産を利用し、弱点を補う方法として、サーバプログラムをモジュール化しあらかじめウェブサーバの起動時にロードし、高速に動作させる方法もある。モジュールとして動作させることで､サーバ側で新たにプロセスを作る必要が無く､ウェブサーバのプロセス内で処理する事ができるので、複数のリクエストにも余裕を持って対処できる。

CGI を使う場合、ユーザはあらかじめ、決められたディレクトリにプログラムを、実行可能な状態で用意しておく必要がある。そして、クライアントが、そのプログラムの実行を要求する URI をウェブサーバに送って、はじめて実行される。このプログラムは URI に含まれる引数などもとに、HTML を生成するように作られていたり、引数を元にサーバ内のファイルを加工するように作られていたりする。サーバのプログラムを呼び出して実行する仕組みであるため、悪意のあるユーザに意図しないサーバのプログラムを実行される危険性があり、サーバ管理者はセキュリティ対策を強化する必要がある。

2.2.3　PHP モジュール

PHP（PHP: Hypertext Preprocessor）は HTML への埋め込み型のスクリプト言語で、ウェブサーバにモジュールとして組み込むことによって、高速に動作する上に、多機能である。また特徴として、各種データベースとの連携に優れており、実行時にエラー発生があった場合、エラーの起こった行番号などが表示される、インタプリンタ型であることも手伝ってデバッグが楽である。

対応データベースサーバ（2001 年 3 月調べ）

Adabas D 、Ingres 、Oracle (OCI7,OCI8) 、dBase 、InterBase 、Ovrimos 、Empress 、FrontBase 、PostgreSQL 、FilePro (読込みのみ) 、mSQL 、Solid 、Hyperwave 、Direct MS-SQL 、Sybase 、IBM DB2 、MySQL 、Velocis 、Informix 、ODBC 、Unix dbm

2.2.3　Servlet

Java サーブレット（Servlet）は、Java 言語で書かれたサーバプログラムを、サーバ側で専

用のサーブレットエンジンを用いて動作させる仕組みである。サーブレットエンジンはウェブサーバとは独立して動いており、ウェブサーバはクライアントから、プログラムへのリクエストがあると、モジュールを使って、その要求をそっくりそのまま、サーブレットエンジンに渡し、サーブレットが処理した結果を受け取り、ユーザに返す。

Java で記述する Servlet の特徴は、Java 言語から引き継ぐものが多く、プラットフォームの非依存性や、スレッドによるプロセス管理、中間コードの存在による速度の向上、セキュリティの高さなどが上げられる。

第３章

パフォーマンスの比較

3.1　応答速度

　Web アプリケーションは、複数のユーザが同時に使うことのできるシステムで、大規模なものでは、一度に数十〜数百というユーザーが利用する場合もある。そのときに、ユーザにある程度の応答速度を保てなければならない。そのためには、ハードウェアや回線などの設備も十分なものを用意する必要があるが、ソフトウェアの選定も大きな要素のひとつとなる。ここでは、先ほどあげた CGI 、PHP 、Servlet を用いてサンプル Web アプリケーションを作成し、パフォーマンスの比較テストをする。

3.2　テスト環境

ハードウェア
CPU: Celeron 708 MHz (1412.30 BogoMIPS)
L2 Cache: 128KB
Memory: 128MB
Mother Board: ASUSTek P3B-F
HDD: Maxtor 98196H8
ATA Interface: Ultara100

ソフトウェア (すべて Debian Package 形式を利用しています)
OS:
Debian GNU/Linux unstable
Linux Kernel 2.4.6

データベースサーバ:
PostgreSQL 7.1.2-1 - Object-relational SQL database, descended from POSTGRES

ウェブサーバ:
Apache 1.3.20-1 - Versatile, high-performance HTTP server

CGI
Perl 5.6.1-5 - Larry Wall’s Practical Extraction and Report Language
libpgperl 7.1.2-1 - Perl modules for PostgreSQL

PHP:
php4 3:4.0.6-1 - A server-side, HTML-embedded scripting language
php4-pgsql 3:4.0.6-1 - PostgreSQL module for php4

Servlet:
Tomcat 3.2.2-1 - Java Servlet 2.2 engine with JSP 1.1 support
libpgjava 7.0.2-3 - Java database (JDBC) driver for PostgreSQL
j2sdk1.3 1.3.1-1 - Java 2 SDK: Blackdown Java(TM) 2 SDK, Standard Edition

3.3 計測方法

あらかじめデータベースにサンプルデータとして、高知工科大学の一期生の名簿情報を登録しておき、その中から、CGI(Perl) 、PHP4 、Servlet を用いて姓と名を取り出し、ユーザに返すプログラムを作成した。このプログラムを、Apache 付属のベンチマークソフト ApacheBench, Version 1.3c-deb <Revision: 1.45 >を用いて仮想的に、サーバに対して複数のユーザが同時に接続する状況を作り出す。そのときの、サーバのレスポンス速度を計測することによって、各ユーザへの応答速度を調べる。この計測方法はローカルより実行されるため、回線速度などには影響されず、純粋にソフトウェアによる速度の差だけが検証できる。

表１．サンプルデータ

1010001	飯塚　宏平	イイヅカ　コウヘイ	男	物質・環境システム工学科
1010002	池川　佳鈴	イケガワ　カリン	女	物質・環境システム工学科
1010003	石川　香織	イシカワ　カオリ	女	物質・環境システム工学科
1010004	伊藤　隆文	イトウ　タカフミ	男	物質・環境システム工学科

〜〜

省略

〜〜

1010584	米田　周平	ヨネダ　シユウヘイ	男	社会システム工学科
1031013	浅野　裕史	アサノ　ヒロシ	男	社会システム工学科
1031014	影山　幸治	カゲヤマ　コウジ	男	社会システム工学科
1031015	白鳥　秀彦	シラトリ　ヒデヒコ	男	社会システム工学科
1031016	田中　孝恒	タナカ　タカツネ	男	社会システム工学科
1031017	谷　　治孝	タニ　ハルタカ	男	社会システム工学科
1031018	松本　康夫	マツモト　ヤスオ	男	社会システム工学科

3.4　各プログラム

データベースのオーナーユーザーは、ウェブサーバの実行ユーザである www-data とし、パスワードを qwerty とする。そして、データベース名を kut_sample 、名簿を登録してあるテーブル名を gakusei として、左から順に項目名を gakuseki 、sei 、mei 、seikana 、meikana 、sex 、shozoku とする。

各プログラムの URI は下記のとおりとする。
CGI Perl: http://192.168.1.1/cgi-bin/StudentList.cgi
PHP4: http://192.168.1.1/php/StudentList.php
Servlet: http://192.168.1.1/servlet/StudentList

１〜５０コネクションを想定して、計測するための計測用シェルスクリプトを最後に掲載する。このシェルスクリプトは、要求を出して、５秒間休み、次の要求を出すという処理を１から５０まで繰り返すスクリプトで、計測が終わると、必要項目を抽出するようになっている。参考のため、出力結果を html 化したものも計測サンプルに加えてある。

3.4.1 CGI-Perl StudentList.cgi

```
#!/usr/bin/perl

print “Content-type: text/html; charset=EUC-JP ” , “¥n¥n ” ;

print “<html>¥n<head>¥n<title>Student List</title>¥n</head>¥n<body bgcolor=¥ ” #FFFFFF¥” text=¥”
#000000¥” >¥n” ;

use Pg;

$conn = Pg::connectdb( “dbname=kut_sample” );

$sql = “select sei,mei from gakusei ” ;

$result = $conn->exec($sql);

$numOfRows = $result->ntuples;
$numOfFields = $result->nfields;

for( $r=0; $r < $numOfRows; $r++ ) {
   for( $f=0; $f < $numOfFields; $f++ ) {
          print $result->getvalue($r,$f);
   }
   print “<BR>¥n ”
}

print “</body>¥n</html> ” , “¥n ” ;
```

3.4.2　php4 StudentList.php

```
<?php header (“Content-Type:text/html; charset=EUC-JP”); ?>
<html>
<head>
<title>Student List</title>
</head>
<body bgcolor=” #FFFFFF” text=” #000000” >
<?php
function pgout($cnt) {
  global $dh;
  $dh = pg_connect(“dbname=kut_sample”);
//  print“$cnt: $dh<br>¥n ”;
  if (!$dh){
    echo“ データベースへの接続に失敗しました。”;
  }
  $result = pg_exec($dh,” SELECT Sei, Mei FROM gakusei”);
  if (!$result){
    echo“ クエリーに失敗しました。”;
  }
  $row = pg_numrows($result);
  for ($i=0;$i<$row;$i++){
    $data = pg_fetch_array($result,$i);
    print $data[‘sei’].$data[‘mei’].”<br>¥n”;
  }
  if (!pg_close($dh)){
    echo“ データベース接続のクローズに失敗しました。”;
  }
}
pgout($i);
?>
</body>
</html>
```

3.4.3 Servlet StudentList.java

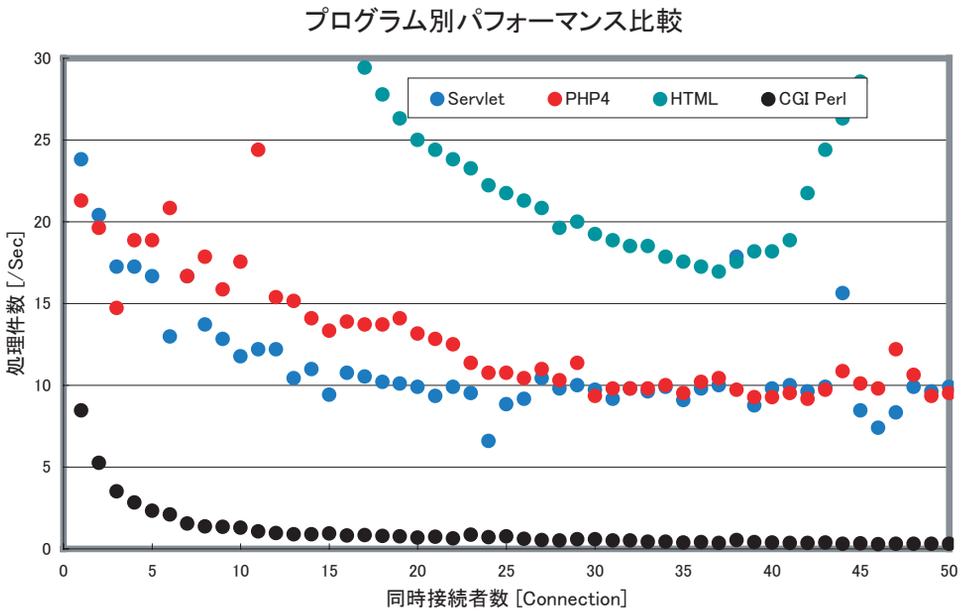
```
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class StudentList extends HttpServlet{
   public void doGet(HttpServletRequest req, HttpServletResponse res)
   throws ServletException, java.io.IOException {
      res.setContentType( “text/html; charset=EUC-JP” );
      PrintWriter toClient = res.getWriter();
      toClient.println( “<html>” );
      toClient.println( “<head>” );
      toClient.println( “<title>Student List</title>” );
      toClient.println( “</head>” );
      toClient.println( “<body bgcolor=¥” #FFFFFF¥” text=¥” #000000¥” >” );
      try {
         Class.forName( “org.postgresql.Driver” );
         Connection con = DriverManager.getConnection( “jdbc:postgresql:kut_sample?encoding=EUC_JP” ,”
www-data” ,” qwerty” );
         Statement st = con.createStatement();
         ResultSet rs = st.executeQuery( “select sei,mei from gakusei” );
         while(rs.next()) {
            String s = rs.getString(1);
            String m = rs.getString(2);
            toClient.println(s + m + “<br> ” );
         }
         rs.close();
         st.close();
         con.close();
         toClient.println( “</body>¥n</html>” );
          } catch(Exception e) {
         e.printStackTrace();
         toClient.println( “<h1>A problem occurred.<br>”
         + “Please try again.</h1> ” );
         toClient.println( “</body></html>” );
         toClient.close();
      }
      toClient.close();
   }
}
```

3.4.4　計測用スクリプト

```
#/bin/bash

url1=http://192.168.1.1/StudentList.htm
url2=http://192.168.1.1/php/StudentList.php
url3=http://192.168.1.1/servlet/StudentList
url4=http://192.168.1.1/cgi-bin/StudentList.cgi

task1=html
task2=php4
task3=servlet
task4=cgi-perl

rm -f per_$task1.txt
rm -f per_$task2.txt
rm -f per_$task3.txt
rm -f per_$task4.txt
rm -f req_$task1.txt
rm -f req_$task2.txt
rm -f req_$task3.txt
rm -f req_$task4.txt
rm -f size_$task1.txt
rm -f size_$task2.txt
rm -f size_$task3.txt
rm -f size_$task4.txt

let request=$1+1

counter=1
while [ $counter -lt $request ]; do
  /usr/sbin/ab -c$counter $url1 >>per_$task1.txt
  echo $counter url1
  let counter=$counter+1
  sleep 5
done

counter=1
while [ $counter -lt $request ]; do
  /usr/sbin/ab -c$counter $url2 >>per_$task2.txt
  echo $counter url2
  let counter=counter+1
  sleep 5
done
```

```
counter=1
while [ $counter -lt $request ]; do
  /usr/sbin/ab -c$counter $url3 >>per_$task3.txt
  echo $counter url3
  let counter=counter+1
  sleep 5
done

counter=1
while [ $counter -lt $request ]; do
  /usr/sbin/ab -c$counter $url4 >>per_$task4.txt
  echo $counter url4
  let counter=counter+1
  sleep 5
done

cat per_$task1.txt |egrep second: >>req_$task1.txt
cat per_$task1.txt |egrep HTML >>size_$task1.txt
cat per_$task2.txt |egrep second: >>req_$task2.txt
cat per_$task2.txt |egrep HTML >>size_$task2.txt
cat per_$task3.txt |egrep second: >>req_$task3.txt
cat per_$task3.txt |egrep HTML >>size_$task3.txt
cat per_$task4.txt |egrep second: >>req_$task4.txt
cat per_$task4.txt |egrep HTML >>size_$task4.txt
```

3.5　計測結果



今回の計測でわかったことは、Servlet と PHP4 のパフォーマンスが高いということと、CGI は複数同時にアクセスがあると極端に速度が落ちるということである。

追加実験を行うとすれば、今回は計測環境が限られているためできなかったが、サーバのハードウェア性能とパフォーマンスが、どのような関わりにあるかを調べたい。また、上記のプログラムではデータベースへのセッションがひとつだけなので、複数セッションになったときに接続を維持した状態で処理を行う持続的接続処理についても未検証である。あるサイトによると、持続的接続を使うと PHP3 と比べて Servlet のパフォーマンスが勝るという結果が出ている。PHP4 は PHP3 に比べて劇的に速くなっているので、PHP4 と Servlet だとどのような結果になるのかきになるところだ。

謝辞

本卒業論文を完成させるに当たり、指導教官である原央教授には大変お世話になりました。深く感謝しております。また、矢野政顕教授、河津哲教授には、同研究室の学生と変わらぬご指導を頂けたことも、忘れようがございません。ありがとうございました。
そして、同じく机を並べて勉強した友人たちにも、おつかれさまと、ありがとうを送らせていただきます。

参考文献

[1] 糸魚川茂夫 . FreeBSD/Linux/Windows 2000 で使う PostgreSQL 詳解

[2] 小西　敏 . サーバーサイド Java でグループウェアに挑戦

[3] 石井達夫 . 改訂版 PC UNIX ユーザのための PostgreSQL 攻略完全ガイド

[4] 丸山　宏、田村健人、浦本直彦 .XML と Java による Web アプリケーション開発

[5] The Ja-Jakarta Project
http://www.ingrid.org/jajakarta/index.html

[6] プログラミング特論発表資料
http://www.padc.mmpc.is.tsukuba.ac.jp//member/bandaxay/php/php.html

[7] Fermata
http://www.asahi-net.or.jp/~WW6K-MK/

[8] VineUsers-ML 全文検索システム
http://vine.ic.sci.yamaguchi-u.ac.jp/search.html

[9] Java 活用ビジョン本命は 3 層システム
http://java.nikkeibp.co.jp/Java/Report/970401Tokusyu1.shtml

[10] Enterprise JavaBeans コンポーネントとは？
http://www.ibm.co.jp/developerworks/java/001020/j_part1.html

[11] Performance Comparison Of Alternative Solutions
For Web-To-Database Applications
http://rain.vislab.olemiss.edu/~ww1/homepage/project/mypaper.htm

[12] Postgre SQL mailing list in Japan
http://www.sra.co.jp/people/t-ishii/PostgreSQL/

[13] @IT
http://www.atmarkit.co.jp/