

フリーソフトウェアを使用した Web - DB連携システム構築法

2001.07.14
境田雅明

- Webサーバソフト
- データベース(DB)とインターフェイス
- Webアプリ開発方法
- Pgbashとは
- PEAK会員管理システムの構築例

Pgbash

Pgbashは、bash シェルの全ての機能を使いながらSQLを直接実行できる便利なシェルです

1

Web-DB連携システムの構成要素

Web サーバソフト	Apache
データベースソフト	PostgreSQL, MySQL, InterBase
Webアプリケーション (サーバ側で実行)	CGI.....Perl, C, Shell サーバAPI...mod_perl, PHP, JSP
DBインターフェイス	ODBC.....C言語, Windows APP JDBC.....Java言語 DB独自...C言語 (PostgreSQLではlibpq) 埋込SQL...C言語 (Pgbashではシェルスクリプト) 直接SQL...RDBMS付属ソフト, pgbash
ブラウザ	Internet Explorer, Netscape
クライアントサイドスクリプト	JavaScript, VBscript

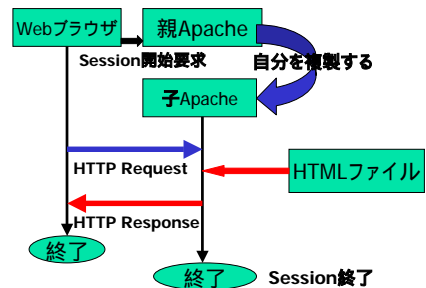
2

Apacheの特徴

- 1995年にNCSA版Webサーバの上位互換として誕生
- 現在、最も多く使用されている(シェア約65%)
- 各種プラットフォームに対応(Unix, Windows など)
- 軽快に動き、信頼性、安全性、堅牢性に優れている
- モジュールの追加により多機能化を実現できる
- CGIを標準装備, PHP, Servlet, JSPはモジュールを追加
- インストールが簡単 (configure, make, make install)

3

Apacheの動作



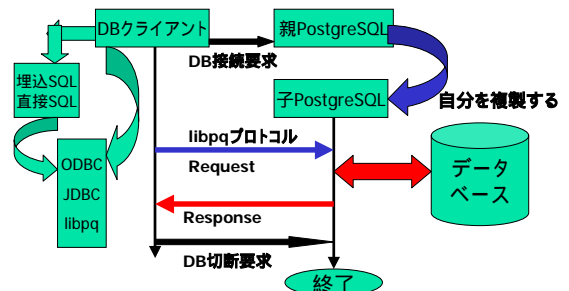
4

PostgreSQLの特徴

- 1995年にpostgres 9.5 誕生、1997年にPostgreSQLに改名
- 高機能、多機能により特に日本での人気が高まっている
- SQL 9.2、マルチバイト(国際化)に対応
- MVCCによってトランザクション性能が高い
- VACUUMによるゴミ掃除の必要性有り
- ODBC, JDBC, libpq, 埋め込みSQLなどインターフェイス有り
- 各種プラットフォームに対応(Unix系)
- ソースコード公開、Web使用無制限、無料
- アプリを開発して販売してもソースを非公開にできる

5

PostgreSQLの動作



6

フリーなデータベース

DB名	特徴	対応OS	license
MySQL 3.23	高速 (Webアプリ向き) トランザクション、副問合せ機能無し 小規模システム向き	Unix系 Windows	2000/6 より GPL
PostgreSQL 7.1.2	商用DB並みの機能 トランザクション性能が高い VACUUMが欠点(7.2改良予定)	Unix系	BSD
InterBase 6 (InterBase 5 は商用版)	商用版DBをオープンソースに 安定性、信頼性が高い.....? オープン化の方針が....	Linux (FreeBSD)	MPLを 元にし たIPL

7

DBインターフェイスの種類

インターフェイス	言語	特徴
ODBC	C言語	Driver Manager と Driver がある Windowsアプリでよく使用される
JDBC	Java言語	Driver Manager と Driver がある Java用アプリで使用される
RDBMS独自のデータベースアクセス関数	C言語	各RDBMSの性能を100%発揮 (PostgreSQLでは libpq)
埋め込みSQL	C, Fortran, Pgbash	SQLレベルのインターフェイス 親言語の中にSQLを埋め込む
直接SQL		SQLを入力すると結果を即座に表示 (PostgreSQLではPgbash, psql)

8

Webアプリ開発方法の比較

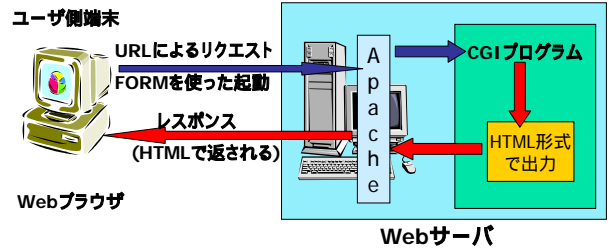
開発方法	F	S	D	W	備考
CGI (C言語)					計算処理を行なう場合は高速
CGI (Perl)					豊富な機能と使いやすさ
CGI (Pgbash)					小規模向け、最も簡単に作成できる
サーバAPI (mod_perl)					最も多く使用されている
サーバAPI (PHP)					高速で機能が高い、人気上昇中
サーバAPI (JSP)					言語体系が立派、これから人気がそう

F: プログラミング能力 S: Webにおける処理スピード
D: DBインターフェイスの簡単さ W: Webインターフェイスの簡単さ

9

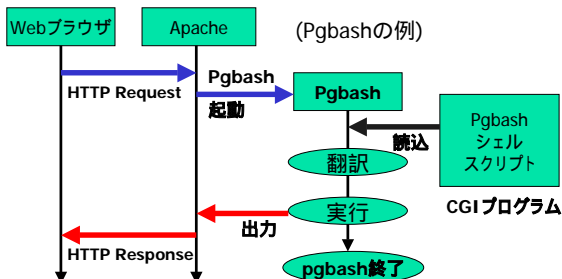
CGIの仕組み

(Common Gateway Interface)



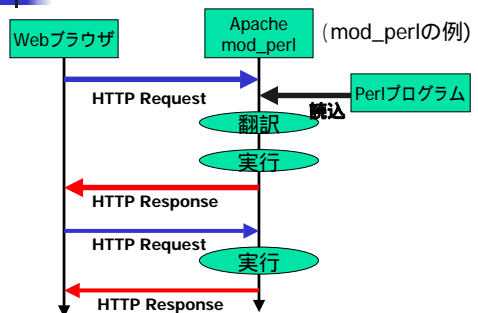
10

CGIの動作



11

サーバAPIの動作



12

Pgbashとは

- PostgreSQL MLの話の中から1999/08誕生
 - Bash-2.05シェルの全ての機能を使用できる
 - bashシェルの中にデータベースアクセス機能を搭載
 - 対話型では直接SQLインタ - フェイス
 - シェルスクリプトでは直接/埋込SQLインターフェイス
 - **簡単に小規模なWebアプリケーションを作成可能**
 - インストールが簡単(configure,make,make install)
- 但し、アクセス頻度が高いシステムには不向き

対話型環境における操作例

```
Prompt> /usr/local/bin/pgbash.....pgbashの起動
pgbash> CONNECT TO admin@xxx.com USER admin;.....DB接続
pgbash> name='鈴木一郎'
pgbash> SELECT * FROM test WHERE 氏名='${name}';.....検索
```

userid	氏名	電話	email
1201	鈴木一郎	111-222-3333	ichiro@xxx.ne.jp
1939	鈴木一郎	222-333-4444	suzuki@xxx.ac.jp

```
Pgbash> ls | more.....ファイル一覧表示などのシェル操作
pgbash> DISCONNECT all;.....全てのデータベースを切断
pgbash> exit.....pgbashの終了
```

Webアプリの起動方法

(入力データが必要な場合はFORMを使用)

```
<HTML>
<FORM METHOD=POST ACTION="WebアプリのURL">
<INPUT TYPE=text NAME="nam" SIZE=20>
<INPUT TYPE=password NAME="passwd" SIZE=24>
<INPUT TYPE=SUBMIT VALUE=submit>
</FORM>
</HTML>
```

CGIプログラムの例

```
#!/usr/local/bin/pgbash.....pgbashを指定
echo "Content-type: text/html".....必須
echo "".....必須
Set EXEC_SQL_OPTION CGI;.....CGIモードへ切替
.....$passwdのチェック.....
echo "<HTML>"
SELECT * FROM test WHERE 氏名='${name}';
echo "</HTML>"
```

SELECTの結果はHTMLテーブル形式で出力

userid	氏名	電話	email
1201	鈴木一郎	111-222-3333	ichiro@xxx.ne.jp
1939	鈴木一郎	222-333-4444	suzuki@xxx.ac.jp

HTML生成

埋め込みSQLの例

```
BEGIN;
DECLARE CUR CURSOR FOR
SELECT AA,BB FROM TEST WHERE USERID > 70;
While (( 1 )); do
  FETCH CUR INTO :aa, :bb ;....シェル変数に値を埋める
  if (( $SQLCODE != 0 )); then
    break
  fi
  echo "AA=$aa BB=$bb".....シェル変数の利用
Done
END;
SELECT COUNT(*) INTO :count FROM member;
Echo "Count=$count"
```

入力データ埋め込み機能

1. FORMのINPUT文でデータを入力

```
<INPUT TYPE=text NAME=資格 SIZE=8> ..... '士' を入力
<INPUT TYPE=text NAME=会員種別 SIZE=10> ..... '*' を入力
```

```
<INPUT TYPE=hidden NAME=EXEC_SQL_PREPARE_value
VALUE='%'':資格';:会員種別'...'>
```

2. Pgbash側でデータを埋め込む

```
EXEC_SQL_PREPARE_vau=" '士', '*' , ..."
```

3. シェルスクリプトのSQL文で利用する

```
INSERT into member values(..., $EXEC_SQL_PREPARE_value);
exec_sql "$EXEC_SQL_PREPARE"のような使い方も可能
```

PEAK会員名簿管理

- ・会員名簿の個別表示(氏名を指定して表示)
- ・会員名簿の表示(任意の検索条件を指定して表示)
- ・会員名簿の表示(全員全項目を表示)
- ・会員情報の新規登録
- ・会員情報の削除
- ・会員情報の更新(更新ログの表示)
- ・パスワードの変更

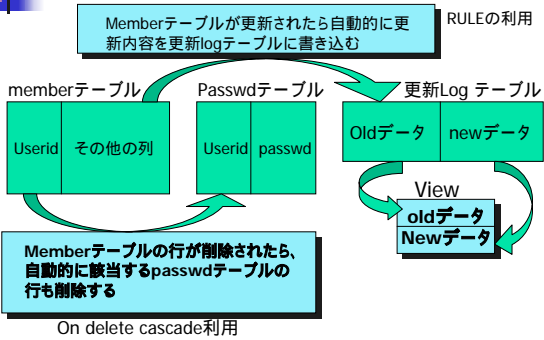
19

プログラム構成

メニュー	HTML	シェルスクリプト
氏名を指定して表示	Select_member_name.html	Select_name.cgi Select.cgi
検索条件指定で表示	Select_member.html	Select.cgi
全員全項目を表示	Select_all.html	Select.cgi
更新ログの表示	Select_log.html	Select.cgi
新規登録	Insert_member.html	Insert_member.cgi
会員情報の削除	Delete_member.html	Pre_delete.cgi Exec_sql.cgi
会員情報の更新	Update_member.html	Pre_update.cgi Exec_sql.cgi

20

スキーマ定義概念図



21

スキーマ定義

スキーマ定義内容	SQL92	内容
テーブルの作成	Create table	Member, passwd, log テーブル
制約の定義	Primary key	Useid を主キー
	Check	データの妥当な値の範囲を指定
	References ... On delete cascade	memberテーブルの行が削除されたとき、カスケードされたpasswdテーブルの行を自動的に削除
シーケンスの作成	Create sequence	自動的に userid を発行する
ルールの作成	Create rule	更新ログを作成するルール
ビューの作成	Create view	更新ログの表示用ビュー
権限の定義	Grant	更新権限、検索のみの権限

但し、シーケンスおよびルールはPostgreSQL独特の機能

22

新規登録(ユーザID自動発行)

1. シーケンステーブルの定義

```
CREATE SEQUENCE member_seq start 1301;
```

2. Webアプリケーションの処理

(1) ユーザIDを求める(連番)

```
select nextval('member_seq') into :sql_seq;
```

(2) データを書き込む

```
insert into member(userid, 資格, 会員種別, 役員,...)
values ($sql_seq, $EXEC_SQL_PREPARE_value);
```

23

更新ログを作成する機能

1. 更新ログ用テーブルの作成

保管したいoldデータとnewデータを定義

2. ルールを定義

```
create rule ルール名 as on UPDATE to member
DO insert into 更新ログテーブル values(
old.userid, old.氏名,
old.郵便番号, old.住所1, old.住所2,
new.郵便番号, new.住所1, new.住所2,...);
```

3. Updateで自動的に更新ログが書き込まれる

4. VIEWテーブルを作成して見やすくする

24

