

第 8 章

PHPlot によるグラフ作成

サーバ側で動作する PHP スクリプトとデータベースを組み合わせることで、自由自在にアクセスしてきたユーザからの情報を保持し、加工することができるようになることは前章で確認してきた。しかし、情報を集めただけではハードディスクの肥やしになるだけである。集めた情報は加工し、見やすい形式でいつでも取り出すことができなければ意味がない。データの加工は表計算ソフトウェアで行うように、まず作表して集計し、その上でグラフ化することが多い。本章では PHPlot というグラフ描画ライブラリを用いたアクセス情報の収集とグラフ化を行うスクリプトを作成する。

8.1 アクセスリスト

Web サーバはアクセスしてきたブラウザ及びクライアントマシンの情報を保持している。これらのうち、一部は必ずログファイルに保存されており、一定期間保存され、アクセス解析や不正アクセスに対する操作のためのデータとして活用される。

PHP では Web サーバが保持しているアクセス情報は `$_SERVER['key値']` という配列を通して得ることができる。例として Apache が保持しているユーザのアクセス情報一覧を P.119, P.120 に示す。

これらのアクセス情報のうち、次の 4 つ

key 値 → フィールド名 (データ型)

HTTP_REFERER → referer (varchar(512))

USER_AGENT → user_agent (varchar(512))

REMOTE_ADDR → remote_address (varchar(512))

REQUEST_TIME → request_time (int)

のみをデータベース (unix10/access_list.db) に保存するスクリプト (access_list.php) を次の

順に作成しよう。

1. アクセス情報を出力する部分を作成する (図 8.1)。例えば次のようなスクリプトを書くと、上記 4 つの情報が取得できていることが確認できる。

```

1: // アクセス元の情報
2: // $_SERVER['HTTP_REFERER'] ... リンク元
3: // $_SERVER['HTTP_USER_AGENT'] ... ブラウザ名
4: // $_SERVER['REMOTE_ADDR'] ... アクセス元(クライアント)アドレス
5: // $_SERVER['REQUEST_TIME'] ... アクセス日時
6:
7: // $_SERVER定義チェック(未定義の場合もあり)
8: $referer = '';
9: $user_agent = '';
10: $remote_address = '';
11: $request_time = time(); // アクセス時間
12: if(isset($_SERVER['HTTP_REFERER']))
13:     $referer = $_SERVER['HTTP_REFERER'];
14: if(isset($_SERVER['HTTP_USER_AGENT']))
15:     $user_agent = $_SERVER['HTTP_USER_AGENT'];
16: if(isset($_SERVER['REMOTE_ADDRESS']))
17:     $remote_address = $_SERVER['REMOTE_ADDRESS'];
18: if(isset($_SERVER['HTTP_USER_AGENT']))
19:     $request_time = $_SERVER['REQUEST_TIME'];
20:
21: echo '$_SERVER[' . '¥' . 'HTTP_REFERER' . '¥' . ']' => ' . $refer
er . "<br />¥n";
22: echo '$_SERVER[' . '¥' . 'HTTP_USER_AGENT' . '¥' . ']' => ' . $us
er_agent . "<br />¥n";
23: echo '$_SERVER[' . '¥' . 'REMOTE_ADDR' . '¥' . ']' => ' . $remote
_address . "<br />¥n";
24: echo '$_SERVER[' . '¥' . 'REQUEST_TIME' . '¥' . ']' => ' . $reque
st_time . ' => ' . date('Y-m-d H:i:s', $request_time) . "<br />¥n";

```

アクセスリスト

```

$_SERVER['HTTP_REFERER'] => http://133.88.120.197/~tkouya/webhpclinux/index.html
$_SERVER['HTTP_USER_AGENT'] => Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko)
Chrome/9.0.597.98 Safari/534.13
$_SERVER['REMOTE_ADDR'] => 133.88.121.79
$_SERVER['REQUEST_TIME'] => 1297435414 => 2011-02-11 23:43:34

```

[トップに戻る](#)

\$_SERVERの値を出力

図 8.1 アクセス情報表示のみ

2. SQLite で、`unix10/access_list.db` を作成し、`id` フィールド (`int` 型) と上記 4

- つのフィールドの合計 5 フィールドを持つテーブル (access_list_all) を作って、access_list.php から読み書き可能であるようにパーミッションを変更しておく。
3. ブラウザから読み込む度に、上記 4 つのアクセス情報を access_list.db に保存するよう access_list.php を手直しし、すべてのアクセス履歴が閲覧できるようにする (図 8.2)。

アクセスリスト

アクセス情報(データベース格納分のみ)

```

$ _SERVER['HTTP_REFERER'] => http://133.88.120.197/~tkouya/webhpclinux/index.html
$ _SERVER['HTTP_USER_AGENT'] => Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.94 Safari/534.13
$ _SERVER['REMOTE_ADDR'] => 133.88.121.79
$ _SERVER['REQUEST_TIME'] => 1297343184 => 2011-02-10 22:06:24

```

アクセス一覧

アクセス総数: 7

ID	8
REFERFER	http://133.88.120.197/~tkouya/webhpclinux/index.html
USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.94 Safari/534.13
REMOTE_ADDR	133.88.121.79
REQUEST_TIME	2011-Feb-10 22:06:24

ID	7
REFERFER	http://133.88.120.197/~tkouya/webhpclinux/index.html
USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.94 Safari/534.13
REMOTE_ADDR	133.88.121.79
REQUEST_TIME	2011-Feb-10 22:06:18

ID	6
REFERFER	http://133.88.120.197/~tkouya/webhpclinux/index.html
USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.94 Safari/534.13
REMOTE_ADDR	133.88.121.79
REQUEST_TIME	2011-Feb-10 22:00:34

[トップに戻る](#)

データベースからアクセス履歴一覧を取り出して表示

図 8.2 アクセス履歴表示

8.2 PHPlot の使い方

次の段階ではアクセス履歴から必要なデータを抜き出して集計し、それをグラフ化する。そのためにはグラフ描画のためのライブラリが必要となる。PHP からグラフを描画する方法は複数あるが、ここでは PHPlot[7] を使用する。

使用に当たっては幾つか条件があるので、まずそれを確認してから、PHPlot を用いたグラフ描画方法を見ていくことにする。

ファイルが必要となる。無料で使用できるものとしては IPA フォント [8] が提供されているので、ここではそれを使うことにする。使用するのは ipaexg.ttf(ゴシック体) のみなので、これを phplot.php, rgb.inc.php と同じディレクトリに置いて使うことにする。

8.2.2 棒グラフの描画スクリプト

以上の環境が整ったら、早速グラフを描いてみよう。PHPlot の特徴は、グラフをビットマップ形式(デフォルトは PNG ファイル)で描画するところにある。従って、グラフ描画のための PHP スクリプトを呼び出す時には PNG ファイルを読み込むように扱わなければならない。

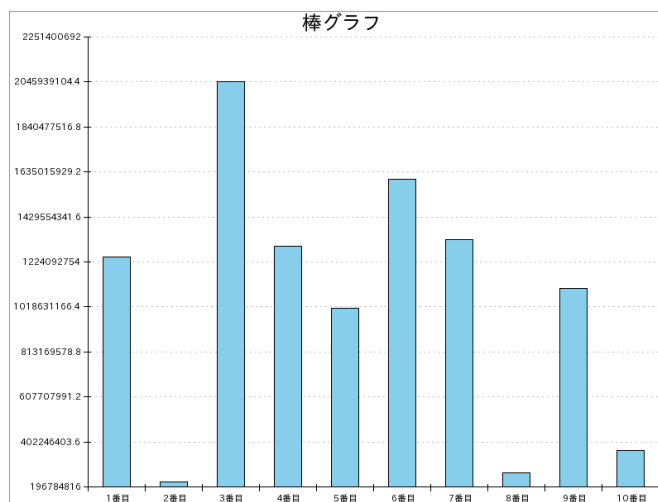


図 8.4 棒グラフ表示例

例えば、図 8.4 のような棒グラフを表示する PHP スクリプトは次のようになる。

```
1: require_once 'phplot.php'; // PHPlotクラス
2:
3: // 描画されるデータ(2次元配列)
4: $data = array();
5:
6: // $data = ('', $x, func($x))
7: for($i = 0; $i < 10; $i++)
8: {
9:     $x[$i] = ($i + 1) . "番目";
10:    $func_val[$i] = rand(); // 乱数
11:    $data[$i] = array($x[$i], $func_val[$i]);
12: }
13:
14: // グラフ描画(棒グラフ)
```

```
15: // 800 x 600 pixels
16: $graph = new PHPPlot(800, 600);
17: // グラフの枠線を直線で描く -> 'plain'
18: $graph->SetImageBorderType('plain');
19:
20: // 日本語フォント設定
21: $graph->SetTTFFPath('./'); // PHPPlotスクリプトと同じ位置
22: // IPAゴシック8~18ポイントに設定
23: $graph->SetFontTTF('title', 'ipaexg.ttf', 18); // タイトル
24: $graph->SetFontTTF('x_label', 'ipaexg.ttf', 8); // x軸ラベル
25: $graph->SetFontTTF('y_label', 'ipaexg.ttf', 8); // y軸ラベル
26: $graph->SetFontTTF('x_title', 'ipaexg.ttf', 10); // x軸タイトル
27: $graph->SetFontTTF('y_title', 'ipaexg.ttf', 10); // y軸タイトル
28:
29: // 棒グラフを描画 -> bars
30: $graph->SetPlotType('bars');
31: // データ形式 -> array("データラベル", 値)
32: $graph->SetDataType('text-data');
33: // 描画すべきデータ -> $data
34: $graph->SetDataValues($data);
35:
36: // グラフタイトル
37: $graph->SetTitle("棒グラフ");
38:
39: // 影なし:
40: $graph->SetShading(0);
41:
42: // グラフ描画実行
43: $graph->DrawGraph();
```

これを例えば `bar_graph.php` という名前で作成してエラーなしで実行できれば図 8.4 のグラフを PNG ファイルとして得られることになる。従って、このグラフを HTML ファイルに埋め込む時には画像同様 `IMG` タグを用いて

```

```

のように指定する。

8.2.3 折れ線グラフの場合

双曲線 $y = x^2$ を折れ線グラフで表現してみよう。描画すべき x, y の値は次のようにして 3次元配列 `$data` に格納しておく。

```
// データ
$data = array();

// [$x_start, $x_end]区間をx_div等分割
```

```
$x_start = -5;
$x_end = 5;
$x_div = 100;
$x_step = ($x_end - $x_start) / $x_div;

// $data = ('', $x, func($x))
for($i = 0; $i < $x_div; $i++)
{
    $x[$i] = ($x_start) + ($x_step) * $i;
    $func_val[$i] = $x[$i] * $x[$i]; // x^2
    $data[$i] = array('', $x[$i], $func_val[$i]);
}
}
```

折れ線グラフを描画する時の指定は下記のようになる。

```
// lines ->折れ線グラフ
$graph->SetPlotType('lines');

// X, Y軸タイトル
$graph->SetXTitle("x");
$graph->SetYTitle("y");

// データ形式 -> array(xの値, yの値)
$graph->SetDataType('data-data');
// 描画すべきデータ -> $data
$graph->SetDataValues($data);
```

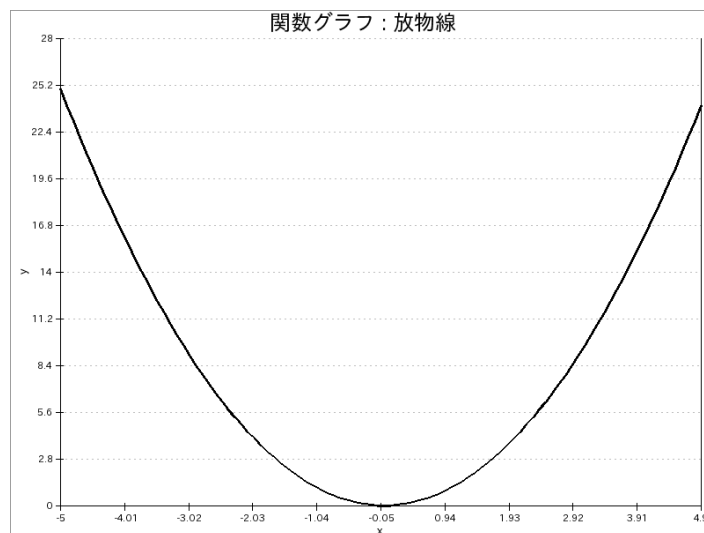


図 8.5 折れ線グラフ表示例

8.2.4 円グラフの場合

円グラフはデータの割合をパーセントで表示するためのものだが、PHPlot では元のデータだけ渡しておけばパーセントの計算を自動で行って円グラフを描画してくれる。実際、この例では棒グラフと同様、乱数として生成したデータをそのまま格納してあるだけである。

```
// データ
$data = array();

// $data = ('', $x, func($x))
for($i = 0; $i < 5; $i++)
{
    $x[$i] = rand();
    $data[$i] = array(($i + 1) . "番目", $x[$i]);
}
```

これを円グラフとして描画する時の指定は次のようになる。

```
// pie ->円グラフ
$graph->SetPlotType('pie');
// text-data-single -> array("データラベル", 値)
$graph->SetDataType('text-data-single');
// グラフに描画するデータ -> $data
$graph->SetDataValues($data);
```

円グラフの凡例 (legend) を付加する時には次の指定も行う。

```
// グラフタイトル & 凡例スタイル
$graph->SetTitle("円グラフ");
$graph->SetLegendStyle('left', 'left');

// プロットエリアのマージン設定(pixel単位)
//                               左 右 上 下
$graph->SetMarginsPixels(NULL, 50, 100, 50);

// ラベル付き(全てのデータラベルを表示)
foreach($data as $row)
{
    $graph->SetLegend(implode(":", $row));
}
// グラフ描画実行
$graph->DrawGraph();
```

こうして図 8.6 のような円グラフが描画できるようになる。

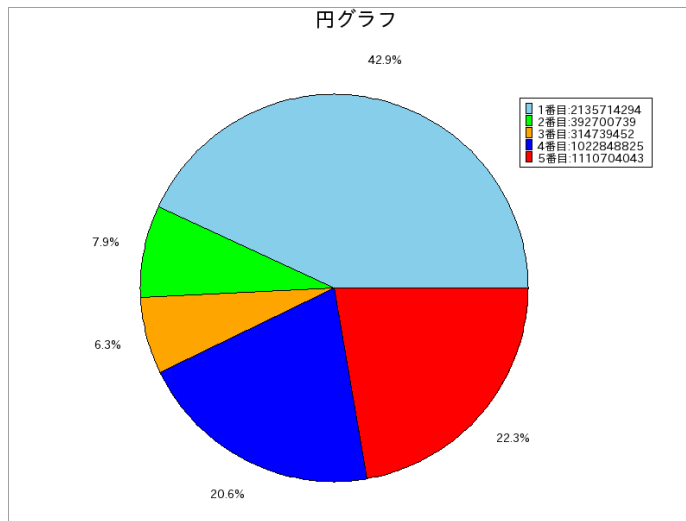


図 8.6 円グラフ表示例

8.3 アクセス解析ページの作成

最後に、PHPlot の機能を用いてアクセス履歴の集計結果をグラフ化してみよう。まず棒グラフとしてブラウザ数と時間ごとのアクセス数を描画する。

■ブラウザデータの取得とブラウザ数のカウント ブラウザ数は、`user_agent` の文字列を取り出してブラウザを特定できる文字列を探索してカウントする。Internet Explorer, Firefox, AppleWebkit[10](Google Chrome や Safari) をそれぞれ特定するためには、

Internet Explorer . . . “MSIE”を探索。

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
```

Firefox . . . “Firefox”を探索

```
Mozilla/5.0 (Windows; U; Windows NT 6.1; ja; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13
```

AppleWebkit . . . “Webkit” (“AppleWebkit”でも可)を探索

```
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.98 Safari/534.13
```

を取り出してカウントすればよい。例えば次のようになる。

```
16: // ブラウザデータのみ全取得
17: $sql = "SELECT user_agent FROM " . $table;
18: $query_ret = $connect->query($sql);
19:
20: // $user_agent_name : ブラウザを特定するための文字列
21: $user_agent_name = array('MSIE', 'Firefox', 'WebKit');
22:
23: //初期化
24: $data = array(); // グラフデータ
25:
26: for($i = 0; $i < count($user_agent_name); $i++)
27:     $data[$i] = array($user_agent_name[$i], 0);
28:
29: while($str_user_agent = $query_ret->fetch())
30: {
31:     // ブラウザ数カウント
32:     for($i = 0; $i < count($user_agent_name); $i++)
33:     {
34:         if(strpos($str_user_agent[0], $user_agent_name[$i]) != false)
35:             $data[$i][1]++;
36:     }
37: }
```

これを棒グラフを描く `access_list_graph_browser.php` に埋め込んで棒グラフを描画する。

■時刻の取得と時間ごとのアクセス数のカウント 時間ごとのアクセス数は、アクセス時間をローカル時間に変換したのち、時刻部分のみを取り出してカウントすればよい。例えば次のようにしてグラフ描画のためのデータを作成し、グラフを出力する `access_list_graph_24hours.php` を作成する。

```
16: // アクセス時間のみ全取得
17: $sql = "SELECT request_time FROM " . $table;
18: $query_ret = $connect->query($sql);
19:
20: // $time_name : 0, 1, ..., 23
21: $hour_name = array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18, 19, 20, 21, 22, 23);
22:
23: //初期化
24: $data = array(); // グラフデータ
25:
26: for($i = 0; $i < count($hour_name); $i++)
27:     $data[$i] = array($hour_name[$i], 0);
28:
29: while($request_time = $query_ret->fetch())
30: {
31:     // 時間のみ取り出し
32:     $hour = date('H', $request_time[0]);
```

```

33:
34: // カウント
35: for($i = 0; $i < count($hour_name); $i++)
36: {
37:     if($hour == $hour_name[$i])
38:         $data[$i][1]++;
39: }
40: }

```

こうしてカウントされたブラウザ数と時間ごとのアクセス数をそれぞれ棒グラフとして描画する二つのスクリプトを `access_list.php` に `IMG` タグを用いて埋め込むと図 8.7 のような出力を得ることができる。

アクセスリスト

アクセス情報(データベース格納分のみ)

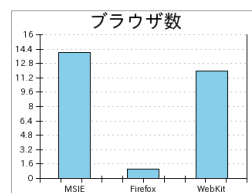
```

$SERVER[HTTP_REFERER] => http://133.88.120.197/~tkouya/webhpclinic/index.html
$SERVER[HTTP_USER_AGENT] => Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; AppleWebKit/534.13 (KHTML, like Gecko)
Chrome/9.0.597.94 Safari/534.13
$SERVER[REMOTE_ADDR] =>
$SERVER[REQUEST_TIME] => 1297345778 => 2011-02-10 22:49:38

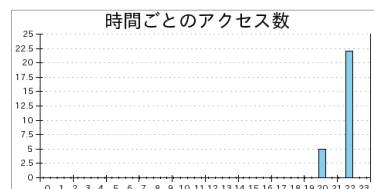
```

グラフ

ブラウザ数



時間ごとのアクセス数



← グラフを埋め込む

アクセス一覧

アクセス総数: 26

ID	27
REFERER	http://133.88.120.197/~tkouya/webhpclinic/index.html
USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; AppleWebKit/534.13 (KHTML, like Gecko) Chrome/9.0.597.94 Safari/534.13
REMOTE_ADDR	
REQUEST_TIME	2011-Feb-10 22:49:38

ID	26
REFERER	
USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 6.1; ja; rv:1.8.2.13) Gecko/20101203 Firefox/3.6.13
REMOTE_ADDR	
REQUEST_TIME	2011-Feb-10 22:49:19

ID	25
REFERER	
USER_AGENT	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
REMOTE_ADDR	
REQUEST_TIME	2011-Feb-10 22:38:58


[トップに戻る](#)

図 8.7 グラフ集計機能付きアクセス履歴表示

課題 B

PHPPlot サンプルスクリプトを参考にして、次の統計データをグラフ化せよ。

- ブラウザ名・・・円グラフ ("Internet Explorer", "Firefox", "Webkit", その他で分類)
- リモートアドレス・・・棒グラフ



第 8 章の学習チェックリスト

- PHP から PHPPlot を用いてグラフを描画できる。
- SQLite で作成したデータベースからデータを取り出して加工することができる。
- アクセス解析ページの全体構造を理解し、第三者に説明することができる。