Constructing a Statistics Website with PHP/FI

Jan de Leeuw

UCLA Statistics

June 27, 2014

If we want to extend our websites from simple HTML (including frames and stylesheets) to something that moves, calculates, does graphics, and interfaces to databases, we have several options.

- Server Side
  - Server Side Includes
  - CGI
  - PHP and friends
  - Servlets
- Client Side
  - Helpers
  - Plugins
  - Java
  - JavaScript

Both the server side and the client side have certain advantages and disadvantages.

- On the server side the webmaster has complete control, what happens is independent of the client configuration. Every client action results in a download, there is no local state. Thus dynamic graphics, for example, is all but impossible.
- On the client side things are much more dependent on the local setup. Certain pieces of software have to be present. This is a problem if we want to serve our material to people all over the world. In the case of Java, quite a bit of initial downloading may be necessary, but once the downloading is done Java takes care of the local state.

In this presentation, I'll concentrate on the server side. I am a control freak.

SSI This is simple and efficient, but very limited. The HTML pages contains certain comments which are parsed by the httpd server (if the page is designated as a parsable page).

Servlets Server-side Java. Requires quite a bit of setup, and does not do graphics so far.

CGI The classical server-side approach. Specific programs/scripts are started by httpd using a URL. httpd also passes parameters from the HTML page (often a form) to the script. The program has to be started with each new action. If the program is a perl script, then all of perl has to be started with each mouse click, etc. Unless we use fastcgi, where the CGI scripts hangs around.

PHP In our implementation, PHP is a module, linked statically into httpd. Pages get parsed by the PHP interpreter in the server, and the commands get executed. The transformed page is send to the client as text/html or image/gif or whatever else.

PHP/FI is a C like language, with the usual control commands, and with many of the math, string, and system functions of the standard C library. It can be compiled as a CGI program, or as a module if we use the Apache httpd, in which case its gets linked into the server executable. PHP/FI comes with interfaces to the GD graphics library and to the mSQL database engine (and many others).

It can be extended in C by using a simple API. I added all of dcdflib, randlib, plus basic statistics (t, correlation, moments), plus basic graphics (boxplot, histogram, scatterplot). This means that the usual statistics commands can be given directly on the HTML page, and the command is then replaced by a numerical value or even a histogram. All of the UCLA Statistics website is written in PHP/FI. Unfortunately, the PHP people decided about a year ago that it was time for an (incompatible) update, to PHP3. This is faster and more stable, but translating the whole site is a major undertaking.

Examples

1. UCLA Statistics Homepage
2. The World of Statistics (uses imagemaps)
3. Bookmarks Database (uses mSQL to emulate a hierarchical file system).
4. Statistical Calculators
   - Power Calculator. CGI, pipes Xlisp-Stat.
   - SASculator. PHP, pipes SAS.
   - GLM Calculator. PHP, pipes Xlisp-Stat
   - CDF Calculator. PHP, no externals.
   - Two-sample Calculator, PHP, no externals.
   - Histogram. PHP, no externals.

Code for the histogram pages is in the handout.

Here are some of the relevant URL's. Take it from there. Remember: the PHP community has switched almost completely to PHP3, and if you start you should too.

UCLA Homepage `www.stat.ucla.edu`

PHP Homepage `www.php.net`

PHP/FI Manual
`php.iquest.net/phpfi/doc/doc.html`

mSQL Homepage `www.Hughes.com.au`

gd Homepage `www.boutell.com/gd/`