Documentation – Excercise 3
Distributed Systems Project

DISCLAIMER: At the time of writing this document and finishing up the source code (From 19:30 onwards on Sunday the 20th) the department servers were down, making the assignment guidelines unretrieveable. The documentation and the caching part of the program were made based on what I could remember.

Deploying the program:

Provided in the submission folder is the git file. One can try different stages of the program by checking out different branches with the command ”git checkout <branch name>

Branches are as follows:

- Master: Includes the step 0 of the program: A simple calculator to compute equations with one operator.

- Step1: Calculator is extended to accept more complex equations. Basic operands are supported, a real calculating order is not. Negative numbers are not supported.

- Step2: Extends the calculator with three different ways to plot sin(x) in the range -π to π.

Server side calculation generates a list of values with PHP, passes them in a file to gnuplot which produces a .png image to display on page.

Client side calculation plots the values with JQuery and draws the graph to a html-canvas.

Mixed version does the plotting and drawing on client side but every atomic calculation is sent to server. This of course causes a lot of traffic exchange between the client and the server, but to alleviate this, there exists a javascript based cache which keeps recent calculations in memory and if a cached calculation is requested, the solution is returned from client side instead of contacting the server.

Step2 version of the program can be tried at http://laursuom.users.cs.helsinki.fi/calc/ . Some helpful prints for the mixed sine plotting can be seen in the javascript console.

The effect of caching

The main takeaway from the excercise was to inspect the effects of caching in web applications.

When plotting sine, there are several repeating atomic calculations when using the approximate formula $x-x^3/3! + x^5/5! - x^7/7!$. The formula itself has 26 atomic operations. This means 26 different GET requests to the server per formula repeated 63 times for the given range with 0.1 increases. With the use of caching the requests can be reduced to 18. So instead of 26*63=1638 requests, only 1134 are needed. The performance gain is ~36.4%. The actual increase is not as big, as the exercise was designed so that only x number of most recent calculations (0 to 50 in my program) are kept in memory. This means, that repeating calculations are occasionally replaced and have to be calculated again.

Operating the application

Operating the program is easy. You can calculate by inputting the equation to the submission box and pressing submit. The calculation history will also be shown underneath. Calculator allows positive integers and operators +, -, * and /.

To plot the sine on server, client or mixed, click the corresponding button. Cache size for mixed version can be changed with the slider element.

Sine plotting images can be discarded with a button titled 'Clear pictures'.

Author: Lauri Suomalainen, 013791364