Distributed Systems Project, Spring 2017

Jussi Kangasharju
Course Outline

1 programming exercise

Goal: Distributed architectures in practice

Individual work

Must demo solution at end of course
Course Schedule

17.1. Start of exercise
19.1. - 23.2. Q&A Sessions Tue, Thu 10-12
26.2. **Deadline** for exercise (hard deadline)
28.2. and 2.3. Demo sessions
People

Jussi Kangasharju
  Office hour: ask for appointment by email

Otto Waltari
Aleksandr Zavodovski
  Office hour: During meetings or ask appointment by email

Twitter: #UnivHelsinkiCS_DSP17
Slack: Sign up for slack team (info on course page)
  Use your real name!
Assignment:
Multitier and Web

Link to assignment will be posted to course website
Multitier Architectures

Alternative client-server organization methods
(Source: Tanenbaum: Distributed Systems)
Goal of the Assignment

Investigate practical effects of various multitier organizations
Where should functionality reside?
  Client, server, or network?
Concrete examples of good and bad choices
Three Tasks

Task 1: Implement server-based calculator
Task 2: Migrate some functionality on client side
Task 3: Implement caching on client side
Simple Calculator

Write simple calculator server
Two arguments, one operator, submit button
Possible operators +, -, *, and /
Input form looks like this:
Results from Server

Server returns result of calculation and new input form
Must keep track of all previous calculations and show their results on screen
  Just like old tape calculators
How you implement history is up to you
  Several possible solutions exist
Client

You need to write several versions of the client
Follow the three steps specified in the following slides
All three steps are required to pass the course
Client: Step 1

Modify form to have only 1 input field and submit
Write Javascript to parse input field
Send each “atomic” operation to server
Precedence left-to-right
Step 1: Example

Input field has expression: “1 + 2 * 3 / 4”
Three requests sent to server
Output:

1 + 2 = 3
3 * 3 = 9
9 / 4 = 2.25
Client: Step 2

Recognize sine function “\( \sin(x) \)” and plot it

Implement three variants of plotting:

1. All on server
2. All on client
3. Client uses atomic operations on server and plots locally
Client: Step 3

Implement caching of results on client side
Configurable number of results (0-N)
Use cached results whenever possible
Implement “Simplify” button to demonstrate cached results
Experiment with cache sizes and plot number of sent messages (in a separate document)
Guidelines

See assignment sheet for further information
Use users.cs.helsinki.fi for running your scripts
  Ask for help on Slack if needed
  Or come to Q&A sessions
No external libraries except as explicitly specified
Free selection of server-side code
What to Return?

Source code of all programs
Documentation on how to compile/run them
Describe implementation choices as well
**DEADLINE:** February 26th at 20:00
Q&A sessions on Tue and Thu until February 23rd
Demo Session

Demo session on February 28 and March 2
Need to demo solution in person to teachers
   No public demos
Dates for demos will be decided later
   Can indicate preferences
Questions?