School of Engineering and Computing Sciences, University of Durham
NS Distributed Systems – Summative Assessment

Coursework Description

You are required to implement a fault-tolerant movie information system based on the passive replication architecture. The system comprises one primary and two backup servers, where each server maintains a list of movie information. Each record of the list is formulated by 1) movie name, 2) IMDb URL of the movie, and 3) a very short sentence describing the movie (or the plot). It is assumed that the system may have at most two servers failing at any time. Data loss possibly happens when a server fails. The movie information system has a front-end (FE) server working as the only interface for a client to connect the system for movie information retrieval or update. In other words, the FE server makes the system architecture transparent to the client. In addition, the FE server also maintains the status of the servers. When the primary server cannot find the requested movie information locally, it will connect to external Web services to acquire such information.

References:

- Passive replication architecture (Lecture 3)
- Web services for acquiring movie information: http://www.omdbapi.com/

Requirements:

1. System Design:

   Write a one-page design document, showing the design of two main aspects: 1) movie information maintenance and 2) system recovery procedures for server failure situations. Movie information maintenance refers to data propagation among servers or information updates due to user requests. To illustrate your design, you should itemize the different situations that your system can handle under the two main aspects, and specify the actions to perform for each situation.

   The design document should also show the **main program modules** in your implementation and briefly explain the functionalities of these program modules.

2. Implementation:

   Implement your system design using **socket API** and the techniques learnt in practical 1. You may choose Java or Python as the programming language. Since distributed systems is the main focus of this coursework, you **should not** spend too much time on issues other than those of distributed systems, e.g. implementing a database system to support movie information retrieval or update. Instead, you may use a simple list or array data structure to maintain a movie information list. Note that it is fine if you want to develop the implementation by extending your work of the term 1 Networks assignment.

3. Perform tests and show test case results:

   Run and test your implementation. Show sufficient test cases with their inputs and results by displaying relevant messages and information from the servers and the client, to demonstrate the completeness and correctness of your implementation.

Assessment:

The assignment will be assessed by the following items, and the level of achievement in each item will be marked against the university's marking criteria:

https://www.dur.ac.uk/resources/university.calendar/volumeii/2014.2015/coreregsug.pdf

- System design (15%)
- Completeness and correctness of the implementation of the server program (45%)
- Completeness and correctness of the implementation of the client program (15%)
- Robustness of the implementation (10%)
- Completeness and correctness of test case results (10%)
- Extra distributed systems related features introduced [* Brief written justification required] (5%)

Deliverables:

1. All program source codes of the *server program* and the *client program* and any necessary files or libraries that are required to execute your programs [requirement 2].
2. A two-page report (in MS word or PDF format) which includes the following items:
   - The one-page design document [requirement 1];
   - Limitations of your design and implementation;
   - Extra distributed systems related features introduced (if any);
   - A brief guideline showing how to set up, install and use your programs;
   - A statement describing what else you would add into your implementation to improve its quality if you had more time;
   - A list of any public domain source code that you have used in your coursework.
3. A set of test case results [requirement 3].

Hand-in Date / Time: 6 March 2015 at 2:00pm

All coursework files should be compressed into a single zip file and submitted through DUO. You are responsible to make sure that there is no missing item in your submission.

Note: Be aware of plagiarism rules (https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/).