INF5510 - Distributed Objects
Home Exam 2

Department of Informatics
University of Oslo

April 2014

Introduction

The assignment will count as 33% of your final grade for the course. The assignment is mandatory and has to be done before going up to the final exam.

Description

The assignment is about creating a framework for a simple replication of distributing objects with the help of Primary Copy Replication. The Framework should offer functionalities to replicate a given object to a given number of machines (nodes). To offer such function you need to create an operation called `operation replicateMe[X,N]` where **X** is an object that should replicate and **N** is the amount of numbers to be maintained including the original object.

It is therefore assumed that that the given object has an operation `cloneMe[]` which returns a replica of the original object. If **N** is greater than the number of available machines, the system will have to maintain as many as possible.

At the Primary Copy Replication, all updates must be performed on the Primary Copy. If a replica, which is not the Primary Copy has called an update, the update should be delegated to the primary replica. How these objects addresses the Primary Copy, is something you need to find out. Emerald has no built-in one-to-many reference mechanism.

In a classical Primary Copy Replication, when the primary replica has updated its condition, make sure that any other replica will receive the updated information. In the program you will be making the primary

replica orient framework, which further informs that an update has happened to all surviving replicas in the system. Tips here will be to look at Design Pattern, Observer. Furthermore, each replica will call the primary replica to get hold of updates. The framework will not involve in this operation.

Assumptions

When a replica has crashed or disappeared, thee framework should automatically generate a new replica. If the disappeared replica was the primary, the framework has to choose a new primary replica. How you exactly solve this is up to you, but it would be necessary to ask the surviving replicas for information.

The simplest solution would be to choose a random replica or the replica that has the latest updated data. The framework should be able to tell an object which second replica exists of the object so that each replica does not need to keep track of all others.

You must also test the framework with at least two tests:

1. One where you will maintain a name server with a lookup operation with a given name, returns the matching object.

2. One where you will maintain a time server as the same as you did during your assignment. The primary replica will maintain the time. If you use N that is larger than the available machines, then there should be a replica on each machine as far as it goes.

There will also be a requirement that the system be tested and run distributed using **PlanetLab** and run on **Emerald-Lite**.

It is important to note that a well-documented simple solution would be considered higher than an advanced solution that only partially works and / or is poorly documented.

Delivery

1. A short (3 - 6 pages) report that contains:

   - An analysis of the most significant decisions you made ??regarding the program. First and foremost design decisions
   - A description of the main classes/objects in the program

- A description of how you have tested the program and why you have chosen to test as you do. Eg you should mention if you have tested the program in several parts.
- Makefile if you have multiple files and like to write emx command of how you run your program

2. The source code

3. Testoutputs

The deadline for the assignment is on **Monday 12th of May 2014 at 23:59.** You will deliver your assignment to devilry with a tar.gz file that is named after your username.

Good luck! :)