COMP30220 Distributed Systems

Group Project Proposal

Group: Lamp+Gary
- Joe Duffin - 13738019
- Edwin Keville - 13718661
- Niamh Kavanagh - 12495522
- Gary Mac Elhinney - 13465572

Project Title:
A distributed voting and election system using Jax-WS, UDDI and Raft for concensus.

Summary:
We would like to implement a distributed voting system, where many clients can vote simultaniously on a winner from a set of candidates. Multiple servers will be used for data duplication to ensure reliabiltiy of our service. Our service will not fail if any of the servers dies as there will be no single point of failure.
We will use UDDI to publish and find the web services we implement.
The difficult part of the project will be reaching data concensus on the vote counts. We plan to research and implement our own version of the Raft algorithm to achieve this.

Team Member Responsibilities:
The initial responsibilities will be as follows:
Joe - Raft research and a toy prototype of the algortihm
Edwin - Implemention a Jax-WS framework with interfaces for the required method calls
Gary - Implementation the server and client classes defined in our frameworks interfaces
Niamh - Implementation UDDI for publishing our web service

We will only be able to work independantly for a very small amount of time and plan on working together and sharing sections of the above responsiblities between us once we start defining dependancies.

Proposed Achitecture:
We propose that each server will keep an SQLite database with 3 fields, candidate id (primary key), candidate name and quantity of votes. Each server will also maintain a log of transactions performed on the database.
It is this log that we will use Raft to come to concensus on. Once concensus is reached after a transaction, only then will the transaction be performed on the databases by their own respective servers. We have unanswered questions regarding batching of transactions and we plan to explore and report on this.
A key feature of our distributed service will be the isolation of each database. One server will have no knowledge of anothers database, only it’s transaction log. We believe this to be important to reduce the latency of the system.

References:
The raft paper: https://raft.github.io/raft.pdf
A visualisation of raft: http://thesecretlivesofdata.com/raft/