PRELIMINARY REPORT

Naveen Yamparala | RS09174
Team Mate: Akash Reddy Gurram | HR69338

System Specifications and design:

- We are going to develop a distributed web service framework by creating replicas of web services like Addition, Subtraction, Age Calculator, String Reversal in multiple sites/servers.
- To perform this, we plan on using RESTful web service architecture.
- On the client side we use AJAX to invoke a web service and for deploying the web services we plan on using the Spring Web Services.
- For service description we are going to use WSDL 2.0 as specified in the requirements document for the project.
- For service discovery we plan on using the Chord protocol.
- Load balancing can be done using Chord or using a replicated registry at each site and serve the request to the least loaded site.
- We will use GitHub for code management.
- Minimum System Requirements are:
    - 8 GB RAM
    - 100 GB Hard Disk
    - Linux/Windows Operating System

Assumptions:

- All the servers and load balancer have a mechanism to identify all the others.
- There should be a minimum bandwidth for the servers to accept the requests.
- There is a limit on the number of requests that a server can process.
- There is at-least one server available to accept the request at all times.
- The network is reliable and secure.
- There is no change in the topology.

Design Justifications:

- We are using RESTful for web services as it is the most used architecture as well as easy to implement. Moreover, RESTful supports wide variety of formats like XML, HTML, JSON etc.
- We are familiar with Chord protocol and it has an additional advantage of load balancing.
- Spring Web Services have been there for a long time and has good community support. Hence we are using it.

Plan of execution:

- **Step 1**: First, we plan on creating a single client and server with the services specified in the design.
- **Step 2**: We will replicate the services across different machines/servers.
- **Step 3**: We will implement service discovery

- **Step 4:** In the end, we will implement the load balancer.
- **Step 5**: Test whether service discovery and load balancing are working as expected under high loads (many requests)
- **Step 6**: We will try to make our sites participate in a block chain.

Proposed Timeline:

- Nov 12th – Step1
- Nov 15th – Step2
- Nov 24th – Step3
- Dec 5th – Step4
- Dec 8th – Step5
- Dec 10th –Step6

Testing Strategy:

- After replicating web services across different machines (Grad lab), we are going to do a functional test to test the entire functionality of the system.
- We will check whether service discovery and load balancing are working correctly by sending multiple requests for a service.
- We check the same after scaling up the system.