Assignment 1.1 - Web Service (WDSL/SOAP) - Lab Report

Katerina Chinnappan, Rohit Shaw, Kjell Zijlemaker

April 11, 2019

Question 3 - Proposed solution to a stateful calculator

Overview

A stateless web service does not keep track of information from one invocation to another one. In other words, it does not keep track of the state from the previous request. In general, a stateless web service is not a bad thing because many web services don't actually need to keep track of the previous state. However, some applications like Grid applications generally require statefullness so we will propose a solution on how to implement a stateful calculator service.

In order to make the calculator stateful, we need to add the state information as a variable in our web service. We are going to describe briefly the implementation and the process for the "add" method because this process & implementation logic will be exactly the same for other methods.

Process

Figure 1 illustrates the process of the "add" web method which keeps track of the previous state.

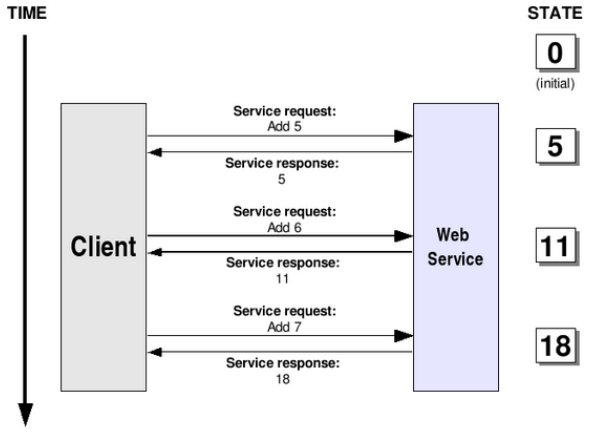

Figure 1: The communication between client and a stateful web service - taken from https://docs.huihoo.com/globus/gt4-tutorial/ch01s03.html

The client sends a service request to add the number 5. The web service responds by returning the number 5 back to the client. The client then wishes to add the number 6(since this is a stateful service, we keep track of the previous state (add number 5)). The service responds back to the client with the number 11 because it remembered the previous request and added 6 to 5. The next steps illustrated in Figure 1 are very intuitive.

Implementation

The simplest way to implement a stateful web service is to define a global `state` variable in the Java class. Each web method in the calculator service would then take one parameter `x`. This parameter would then be added or subtracted or whatever the mathematical operation is from `state` which tracks the state of the previous request. Another additional method `result` would be added which would return the final state of the variable `state`. Here is a sample pseudo-code(python)of how this would look:

```
Class Calculator:
   # initialize the state
   state = 0
   def add(x):
      return state = state + x
   def result:
      return state
```