**Client Design**

**Chenxi Cai**

Guthub url:

https://github.com/xiaohaiguicc/DistributedSystem

The whole packages have two parts, Server and Client.

1. Server

The server uses lab1’s code and have two methods ‘doGet’ and ‘doPost’. These two methods will receive request and check if the url and body (for post) is valid, then return corresponding response.

According Swagger API, there will be two servlets, Skier and Resorts, so I wrote two similar servlets. The differences come from request parameter format and response body.

2. Client

Client has two part according to assignment 1

(1) Part1 have two parts, client and thread.

The thread class is to build a single thread and assign task to it. Every thread needs to set ip address and port, using swagger api to get the response. The api will call the servlets and return response.

The client class deal with three phases. I use countdown to count if 10% of threads finish executing. Every time, when first countdown is 0, the phase can begin and decrease second count down. Then the second countdown will be the first count down for the next phase. This logic will help to control three phases execute in order.

(2) Part2 is to calculate records and data

The record class is to record every POST request and response.

The Data calculate the mean, median, throughput, max, p99 response time…

and generate csv file.

256

A close up of text on a black background

Description automatically generated

128

A close up of text on a black background

Description automatically generated

64:

A screenshot of a cell phone

Description automatically generated

32:

A close up of text on a black background

Description automatically generated

Then we have the graph for the wall time by the number of threads:

The graph of throughput and mean response time against number of threads: