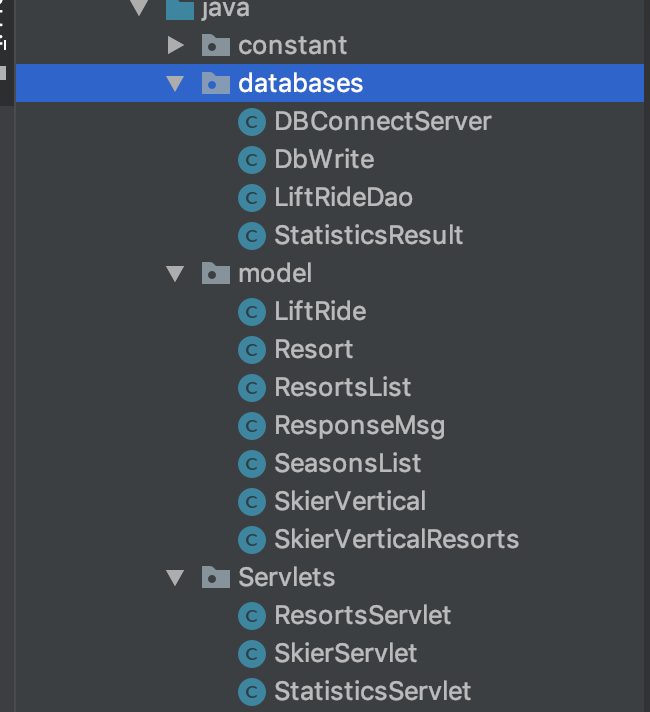
URL: <https://github.com/wushangzhen/Distribute-System>

Name: Shangzhen Wu

Server URL: http://34.219.46.253:8080/serve

**Design:**

I have these classes:



Database folder: all java class is database related code.

DBConnectServer class is the c3p0 connect pool I use to restrict the connections when connecting the database;

DbWrite class: I didn’t use it for batch insert. Just leave it.

LiftRideDao class: Use statement to insert and select through the database

Model folder: all java class for the body of response;

StatisticsResult: send the statistics result for /statistcs/\*

Servlets folder: contains all the servlet.

**My server implementation:**

I create two tables one for original data called skiers\_lift\_record

+----------+--------------+------+-----+---------+----------------+

| Field    | Type         | Null | Key | Default | Extra          |

+----------+--------------+------+-----+---------+----------------+

| id       | int(11)      | NO   | PRI | NULL    | auto\_increment |

| resortID | int(11)      | YES  |     | NULL    |                |

| seasonID | varchar(255) | YES  |     | NULL    |                |

| dayID    | varchar(255) | YES  |     | NULL    |                |

| skierID  | int(11)      | YES  |     | NULL    |                |

| time     | smallint(6)  | YES  |     | NULL    |                |

| liftID   | smallint(6)  | YES  |     | NULL    |                |

+----------+--------------+------+-----+---------+----------------+

one for sum of all skiers vertical called skiers\_vertical\_record

+----------+--------------+------+-----+---------+-------+

| Field    | Type         | Null | Key | Default | Extra |

+----------+--------------+------+-----+---------+-------+

| resortID | int(11)      | NO   | PRI | NULL    |       |

| seasonID | varchar(255) | NO   | PRI | NULL    |       |

| dayID    | varchar(255) | NO   | PRI | NULL    |       |

| skierID  | int(11)      | NO   | PRI | NULL    |       |

| vertical | bigint(20)   | YES  |     | NULL    |       |

+----------+--------------+------+-----+---------+-------+

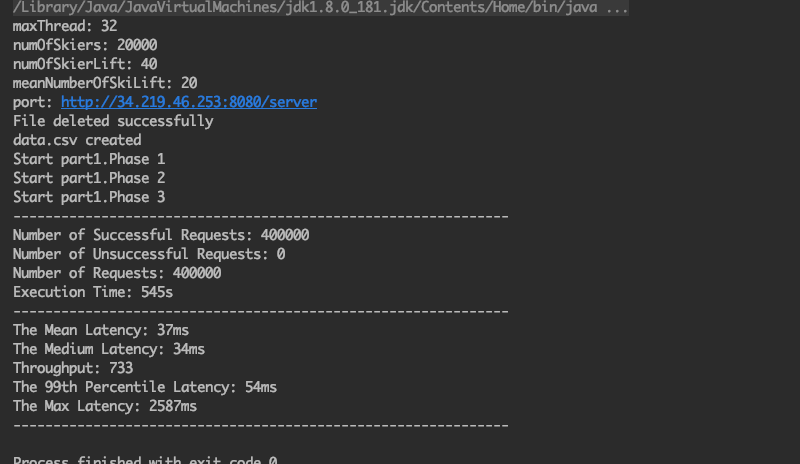
Every post I will insert the data into the first table and update the second table.

Every get I will select the data from the second table to get the value.

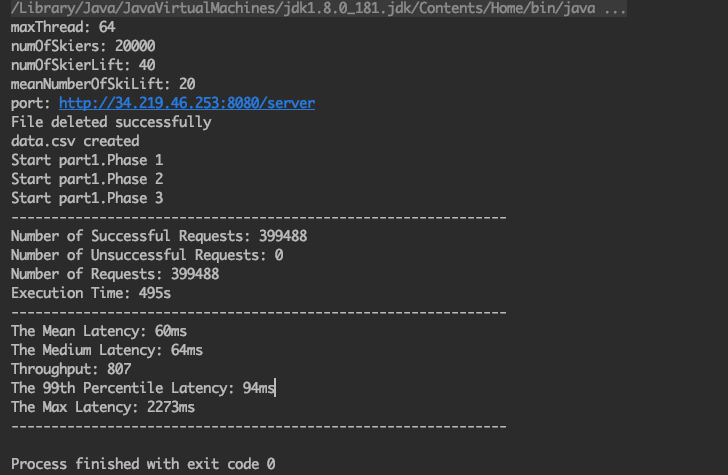
No cache, so the speed is not that fast.

**Single Server Tests:**

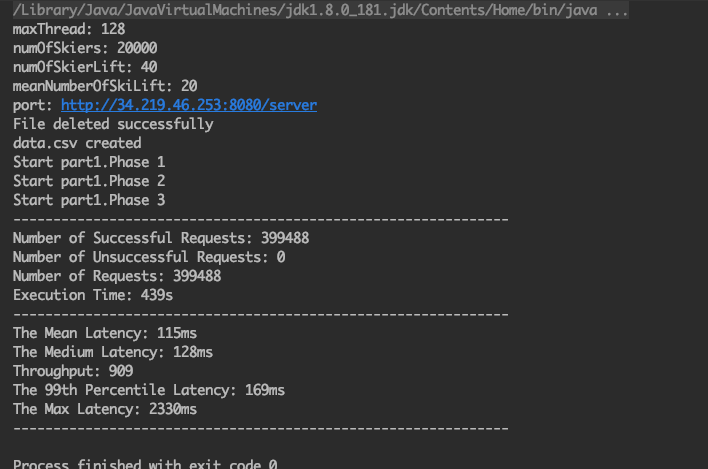
32:



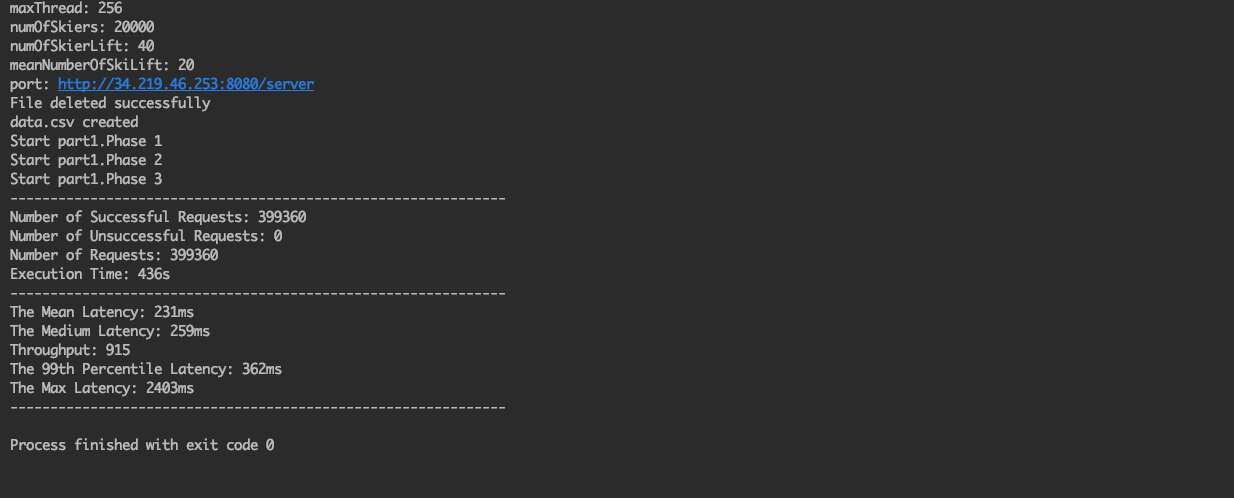
64:



128:

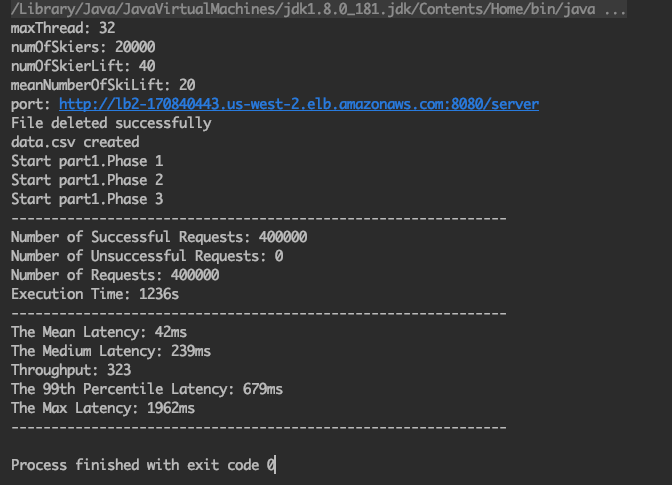


256:

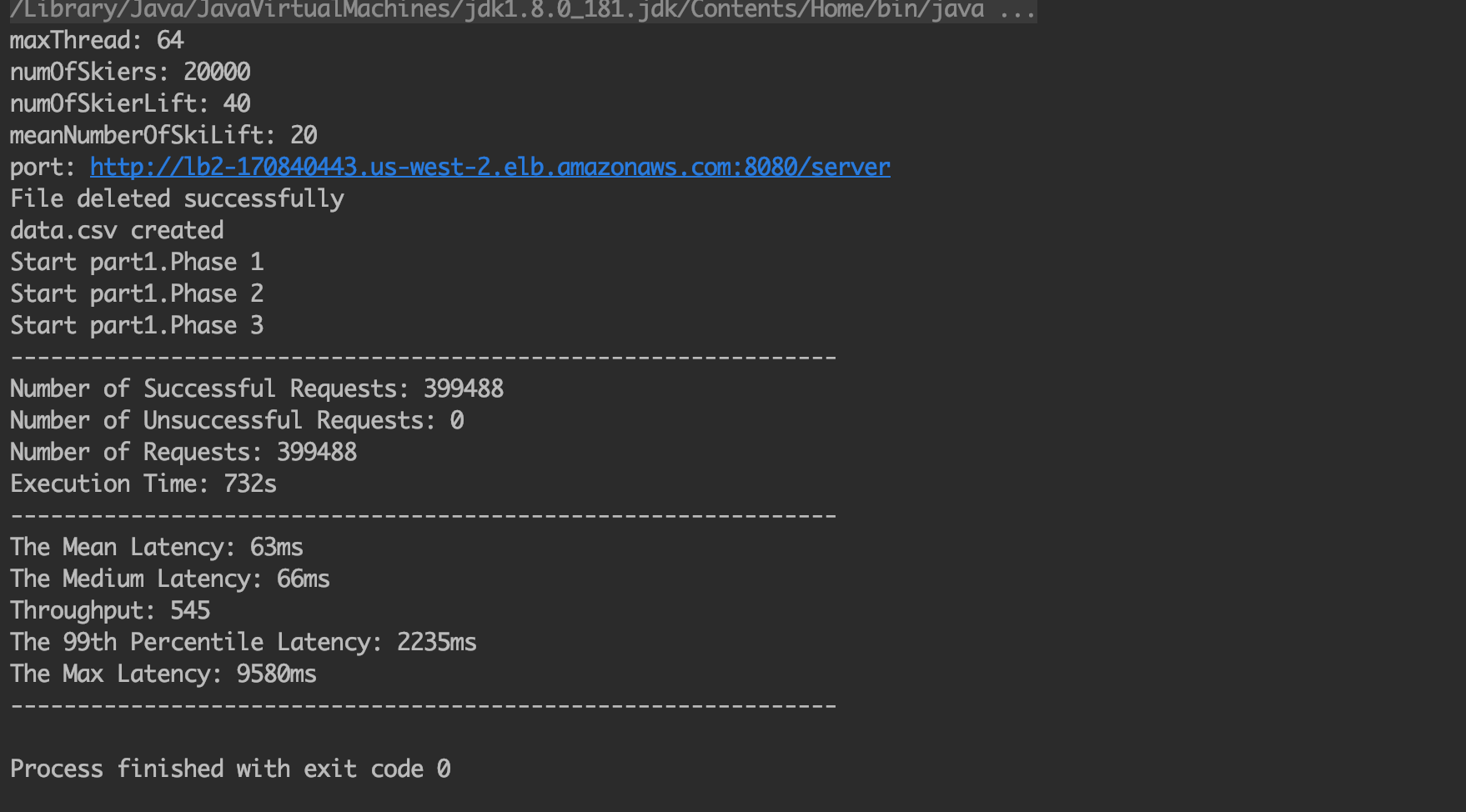


**Load Balanced Server Tests:**

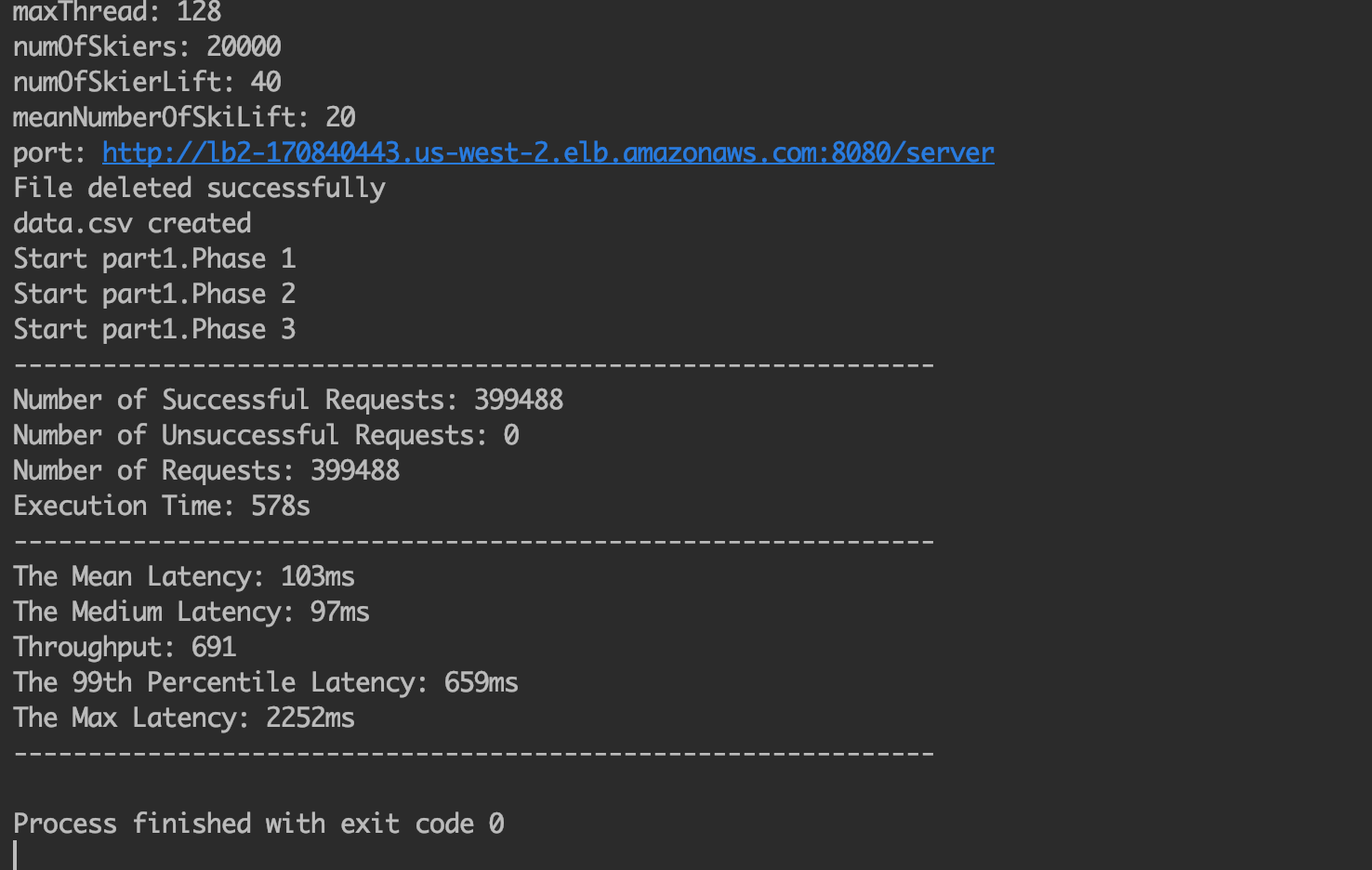
32:



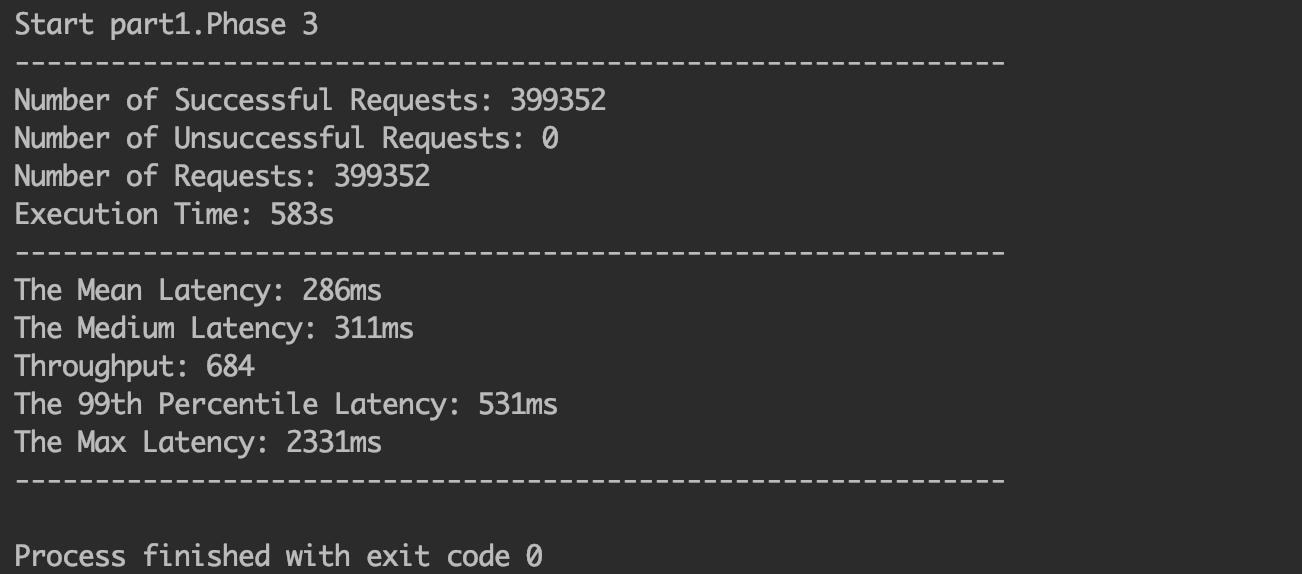
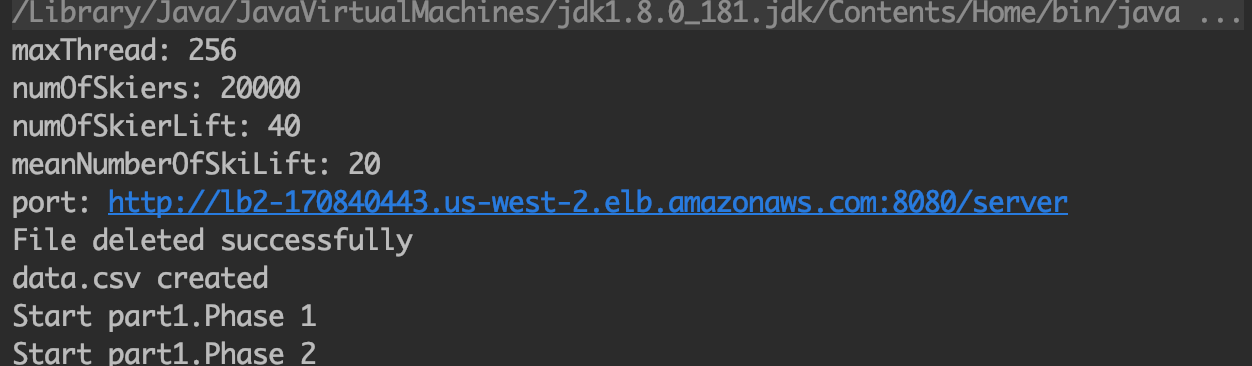
64:



128:



256:



**Runtime Statistics Collection:**

I use the ConcurrentHashMap to record every request(post, get)’s last 100 call. Record the latency in the server side. When I call the statistics endpoints. It will get the min, max, mean from the last 100 latency in the cache.

