TY B.Tech. (CSE) – II [2022-23]
5CS372 : Advanced Database System Lab.
Assignment No. 5

PRN: 2020BTECS00033
Name: Prathamesh Raje

Title: Performance tuning for Assignment No.3 & 4.

Introduction:

When we are developing any application we have to test it under various conditions like how it behaves under very high load conditions, etc. Performance testing in software helps to analyse the performance of the software. Performance testing is important alongside unit testing, system testing, and integration testing. Performance testing is used to analyse the performance, server response time, and throughput under different load conditions.

Locust:

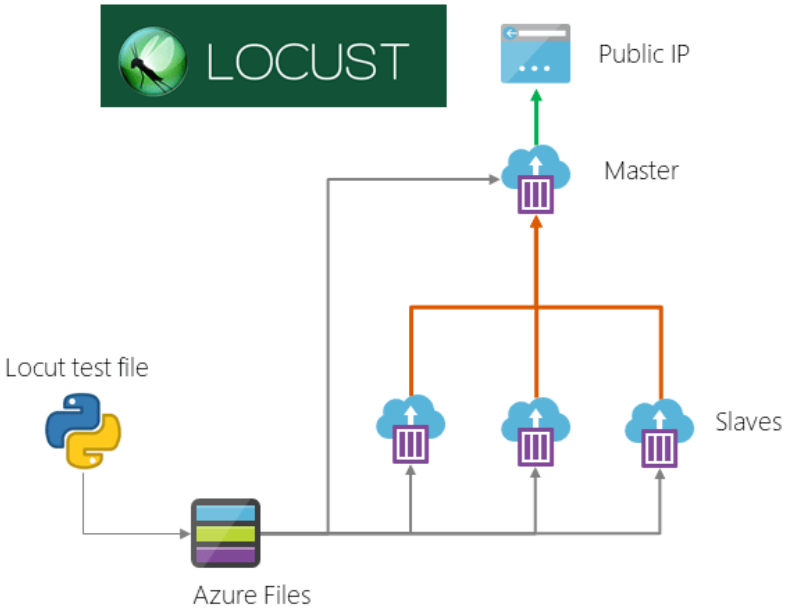
- Locust is an easy to use, scriptable and scalable performance testing tool.
- You define the behaviour of your users in regular Python code, instead of being stuck in a UI or restrictive domain specific language.
- This makes Locust infinitely expandable and very developer friendly.
- Locust has a user-friendly web interface that shows the progress of your test in real-time.
- We can even change the load while the test is running. It can also be run without the UI, making it easy to use for testing.

Importance of Performance Testing:

Performance testing is a crucial step in determining how your web app will perform under heavy loads. Locust is a valuable tool for performance testing, as it can discover the maximum number of users that your web app can handle.

Functional Block Diagram:

The diagram below shows how Locust models users and simulates a heavy load:



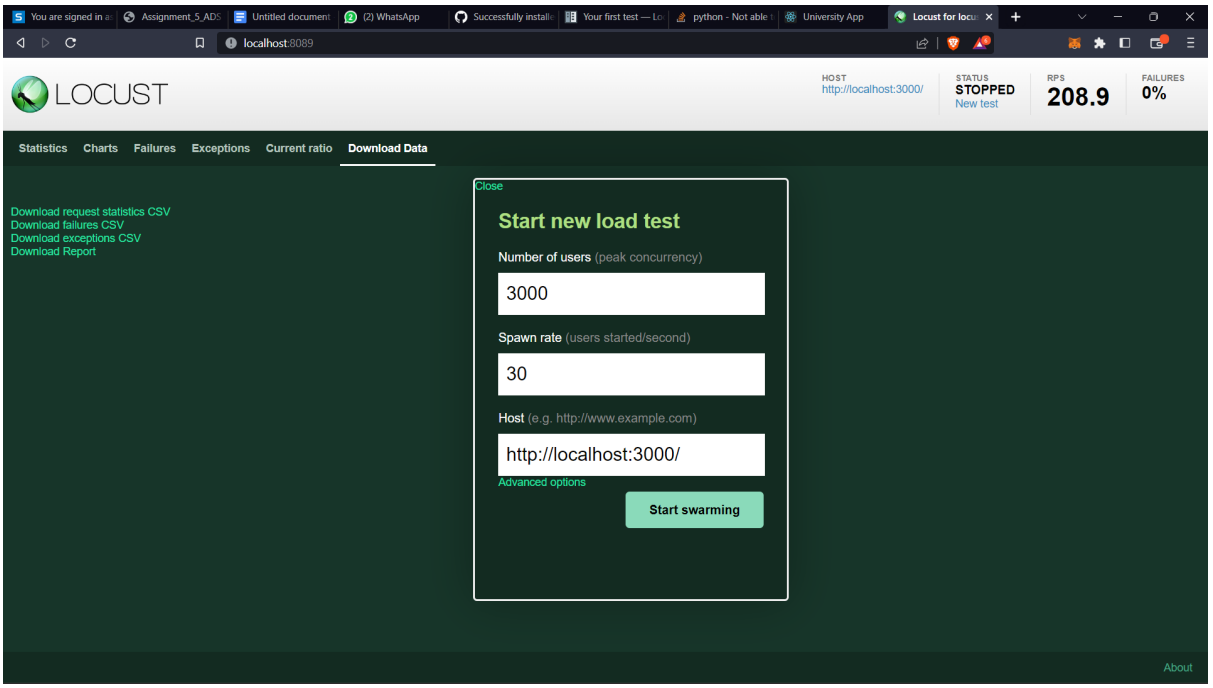
Performance testing on Locus For Assignment 3:

Paths for Testing:

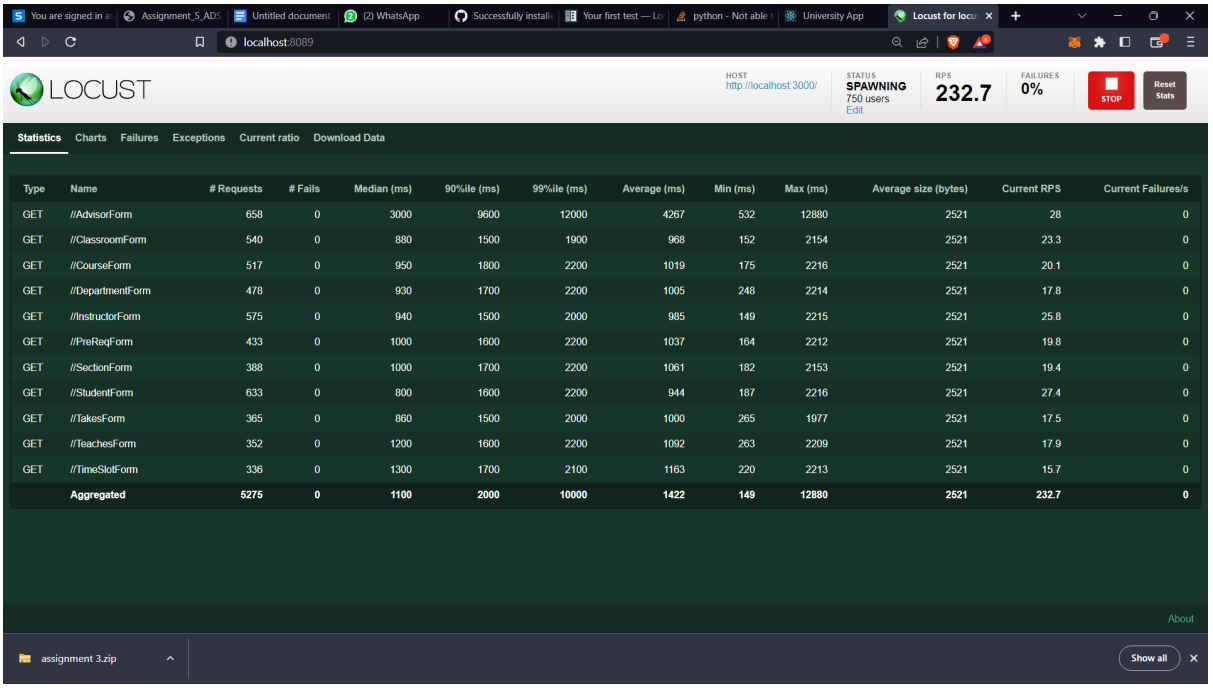
```python
locust.py > ...
from locust import HttpUser, task

class HelloWorldUser(HttpUser):
    @task
    def hello_world(self):
        self.client.get("/AdvisorForm")
        self.client.get("/StudentForm")
        self.client.get("/InstructorForm")
        self.client.get("/ClassroomForm")
        self.client.get("/CourseForm")
        self.client.get("/DepartmentForm")
        self.client.get("/PreReqForm")
        self.client.get("/SectionForm")
        self.client.get("/TakesForm")
        self.client.get("/TeachesForm")
        self.client.get("/TimeSlotForm")
```

Locust interface:

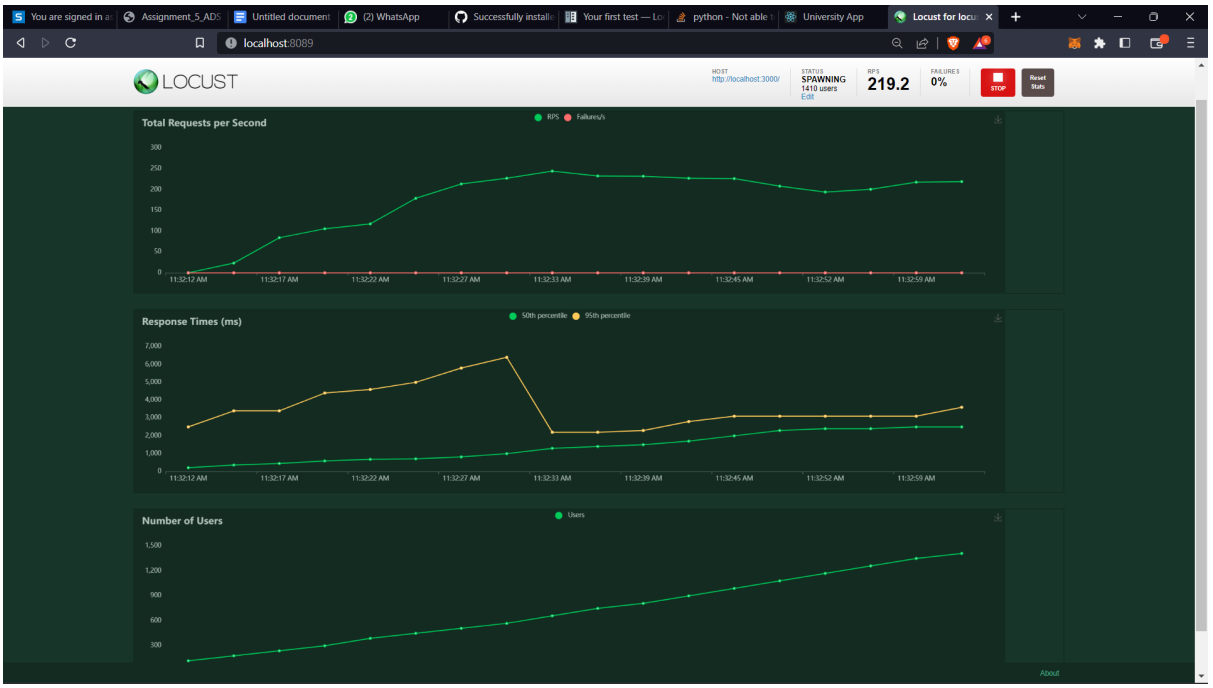


Statistics:

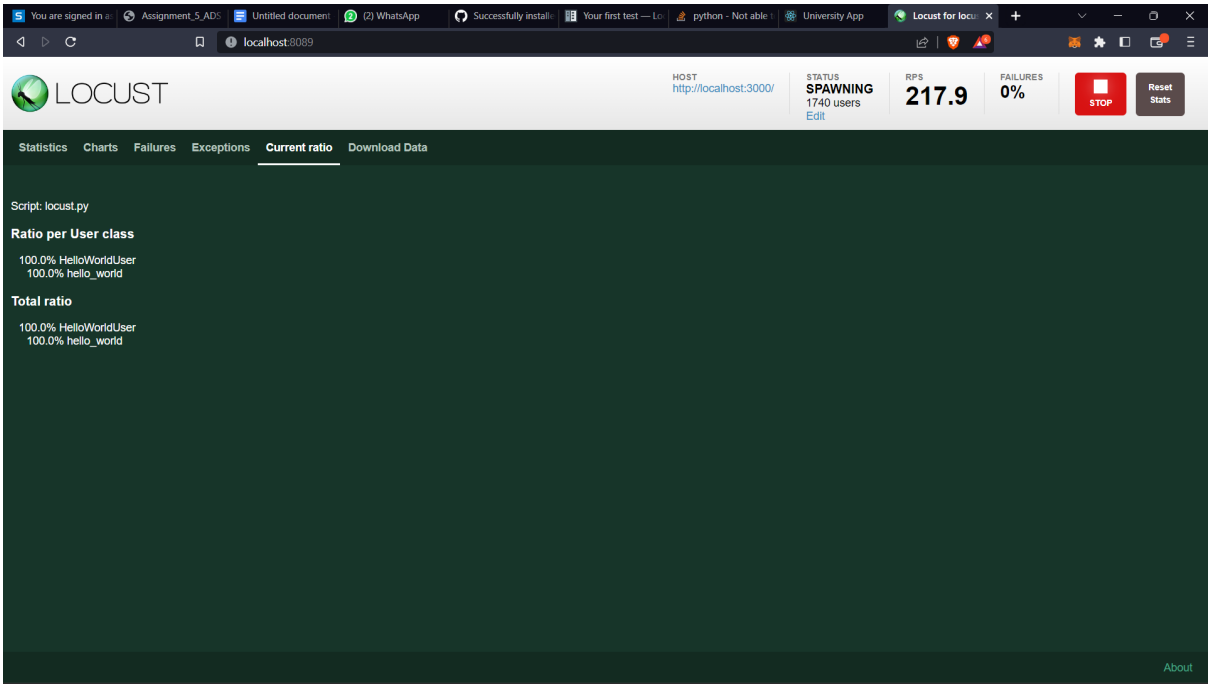


Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	//AdvisorForm	658	0	3000	9600	12000	4267	532	12880	2521	28	0
GET	//ClassroomForm	540	0	880	1500	1900	968	152	2154	2521	23.3	0
GET	//CourseForm	517	0	950	1800	2200	1019	175	2216	2521	20.1	0
GET	//DepartmentForm	478	0	930	1700	2200	1005	248	2214	2521	17.8	0
GET	//InstructorForm	575	0	940	1500	2000	985	149	2215	2521	25.8	0
GET	//PreReqForm	433	0	1000	1600	2200	1037	164	2212	2521	19.8	0
GET	//SectionForm	388	0	1000	1700	2200	1061	182	2153	2521	19.4	0
GET	//StudentForm	633	0	800	1600	2200	944	187	2216	2521	27.4	0
GET	//TakesForm	365	0	860	1500	2000	1000	265	1977	2521	17.5	0
GET	//TeachesForm	352	0	1200	1600	2200	1092	263	2209	2521	17.9	0
GET	//TimeSlotForm	336	0	1300	1700	2100	1163	220	2213	2521	15.7	0
	Aggregated	**5275**	**0**	**1100**	**2000**	**10000**	**1422**	**149**	**12880**	**2521**	**232.7**	**0**

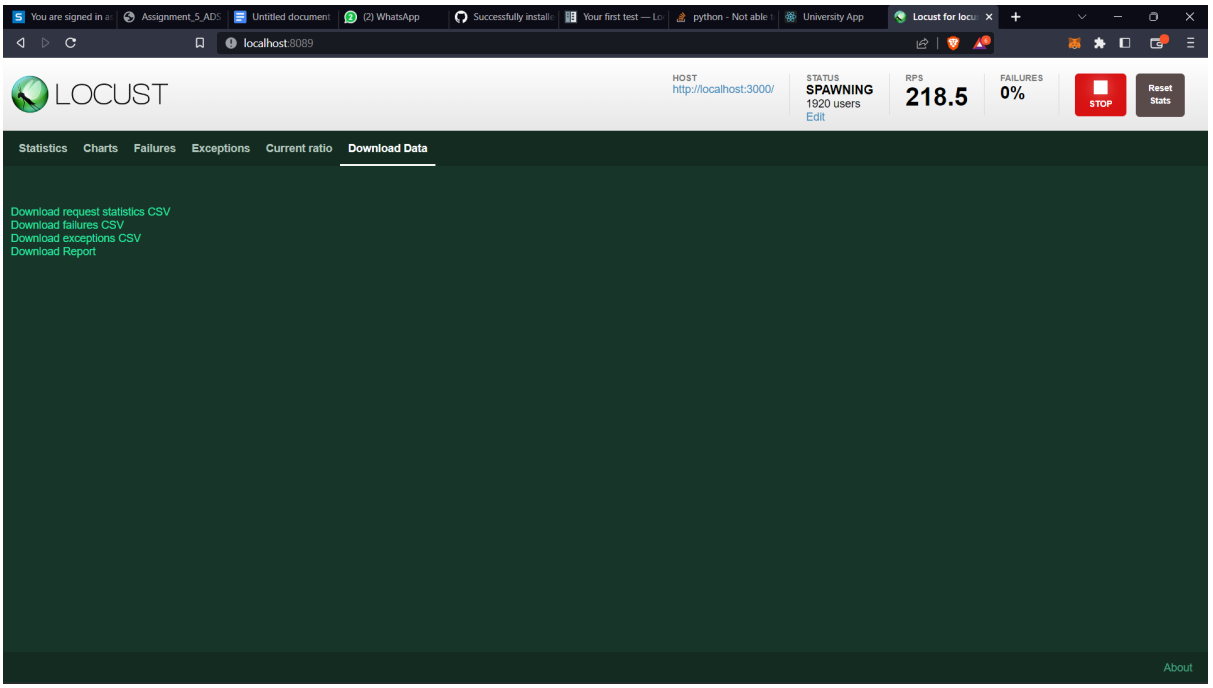
Charts:



Current Ratio:



Download Data:



Throughput:

- Here we have an average RPS of about 232.7% and failure is 0% for 3000 users.
- The measuring capacity of the server, or how much it can handle. Ideally, this number should be infinite, if not very high.
- It's important to note that throughput depends on other factors, such as internet speed, the Google server's current load, and CPU power.
- These factors continuously change, meaning you won't get the same results every time you run the test.

Deviation:

The variation from the average. This number should be zero, or very low.

Conclusion:

After analysing the graph, following are the conclusions:

- The throughput is 100% of input requests per minute.
- This means that the server handled all as 30 initially provided requests per minute.
- Sometimes, the throughput even got 1000 requests per minute.

Based on these numbers, we can deduce that site has very good throughput.

References:

http://docs.locust.io/en/stable/what-is-locust.html