Assignment 2:

Start with 1 user, then move up to 5, then 10 then 20 etc.

The tests where preformed on https://www.pythonanywhere.com/

Test Number:

1

Number of users:

5

Ramp up time:

1 user per second

Observations:

The server could handle this fine no real issues as it was only 5 users.

The response times in the chart are a bit all over the place at the start but even out as it stables out with the number of users and steady requests being received.

Did any failures happen?

There where no fails at this point but that is expected with only 5 users and a very short testing time.

What did the load times look like?

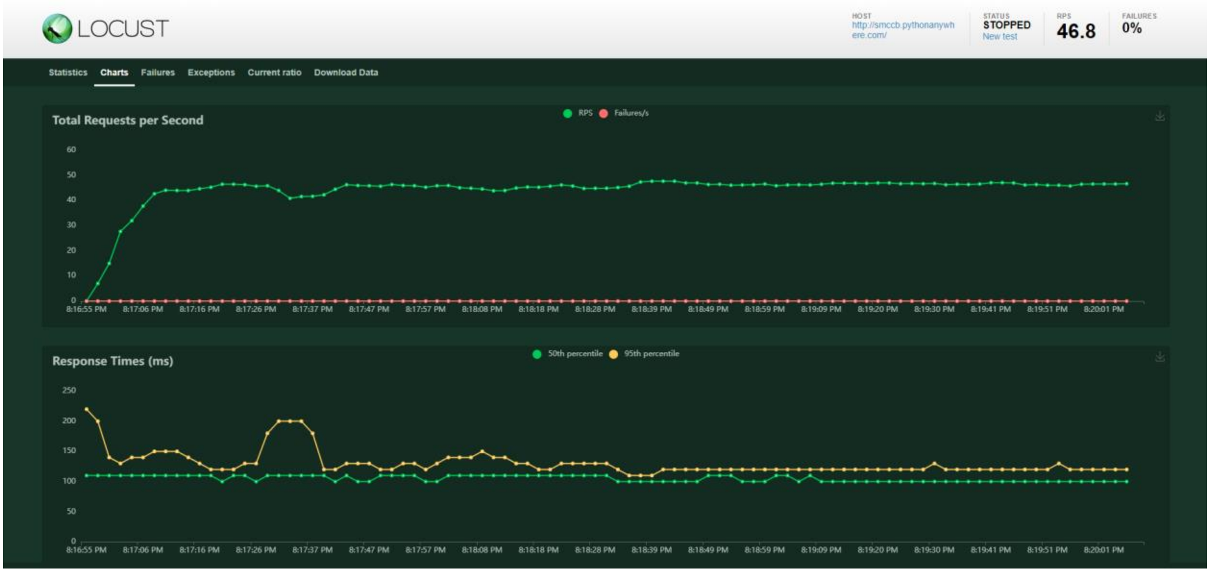
The average was 109 with a min of 93 and a max of 438. The response times took a few seconds at the start to even out explained in the observations I made.

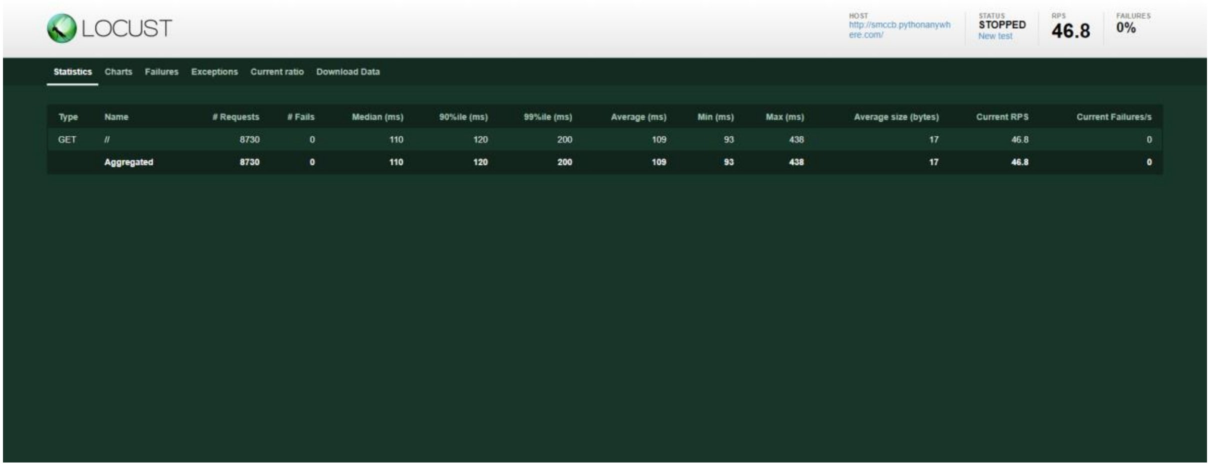
What was the average load time?

The average load time was 109.

How are these results different from the previous tests?

This test was the only one I didn’t run for 5 minutes so I can’t see if anything would have happened had it ran for longer. Though with only 5 users there probably wouldn’t have been any significant changes.





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	//	8730	0	110	120	200	109	93	438	17	46.8	0
	Aggregated	**8730**	**0**	**110**	**120**	**200**	**109**	**93**	**438**	**17**	**46.8**	**0**

Test Number:

2

Number of users:

10

Ramp up time:

1 user per second

Observations:

The load times are fairly consistent. However, does peek a lot more than throughout the test than the first test did.

Did any failures happen?

There where no failures. That is to be expected with only 10 users. And only a 5 min runtime along with the very little content on the flask app.

What did the load times look like?

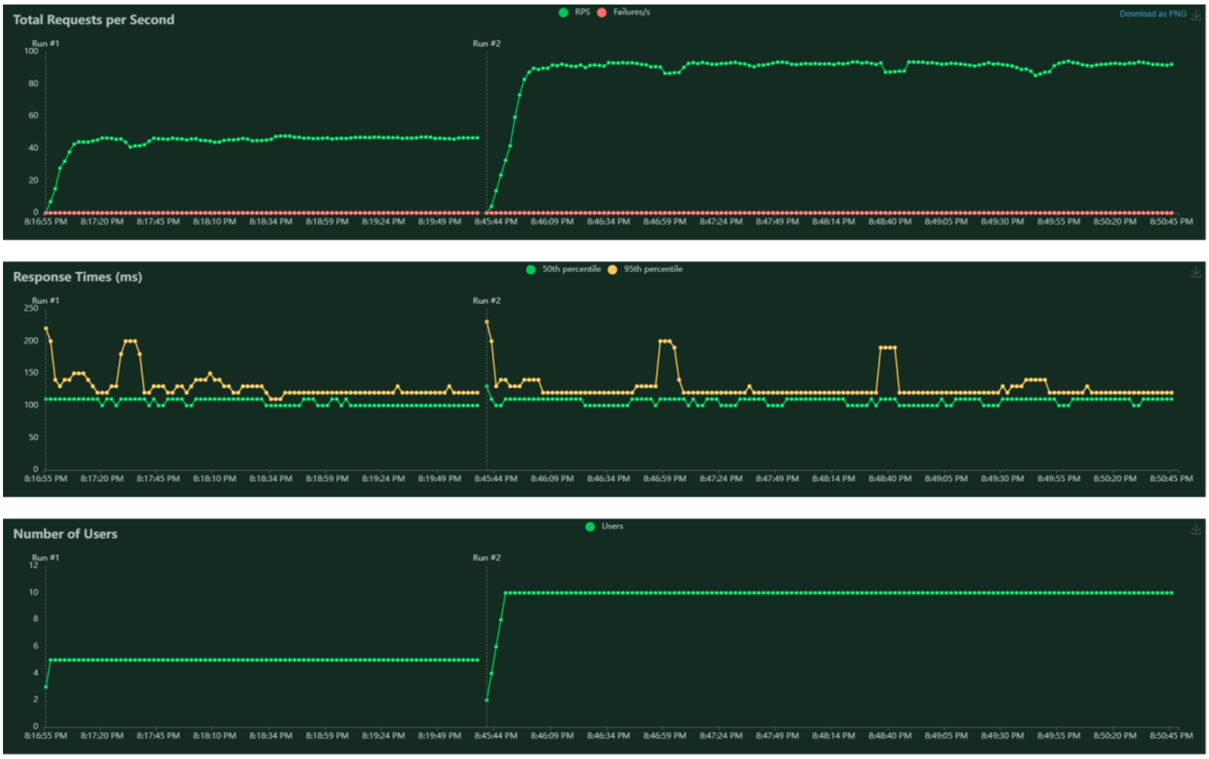
Load times averaged 109, had a minimum of 92 and a max of 422. So very similar to test 1 and show that flask app is well able to handle 10 users especially with the very little content on the page being tested.

What was the average load time?

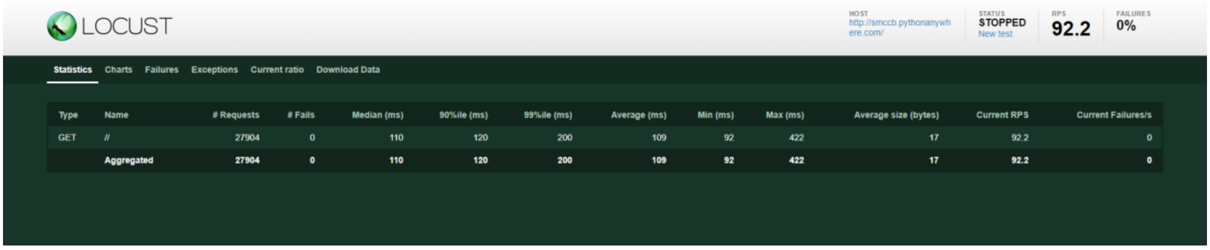
Average load time was 109 which is the same as test 1. These load times probably couldn't get much lower as an average because the web flask app runs on a free server.

How are these results different from the previous tests?

Not really much to say this test was very similar to test1.



This test was done right after the first and so shows both charts in the above image. The right side is test2. The response time peeks more frequently however is still overall quite consistent.



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	//	27904	0	110	120	200	109	92	422	17	92.2	0
	Aggregated	**27904**	**0**	**110**	**120**	**200**	**109**	**92**	**422**	**17**	**92.2**	**0**

Test Number:

3

Number of users:

20

Ramp up time:

1 user per second

Observations:

Vey flat load times seems like the server is not struggling and well able to manage only 20 users.

Did any failures happen?

No failures happened in this test.

What did the load times look like?

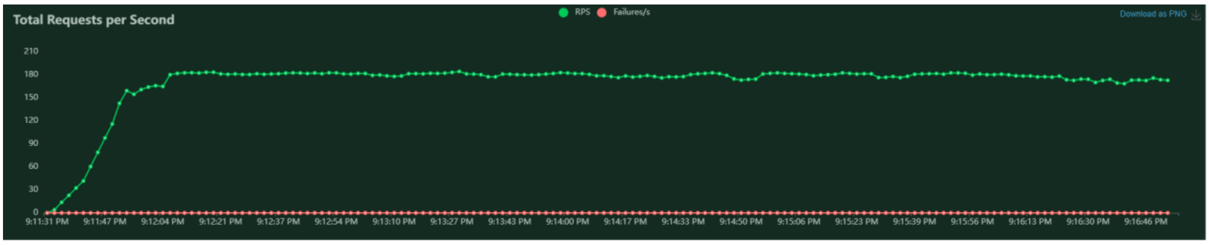
The load times look fine and seem within an acceptable range. They are fairly consistent.

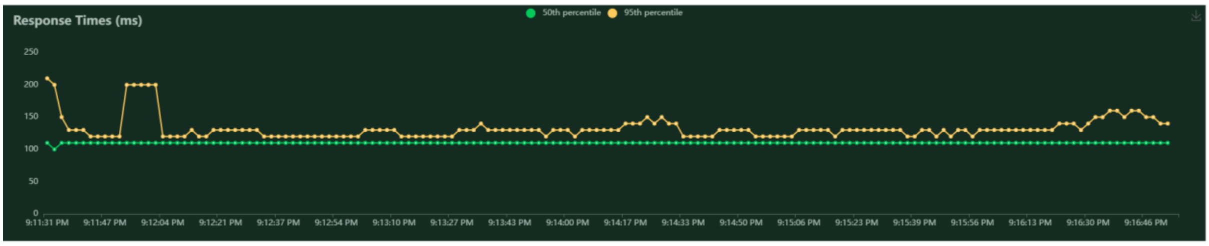
What was the average load time?

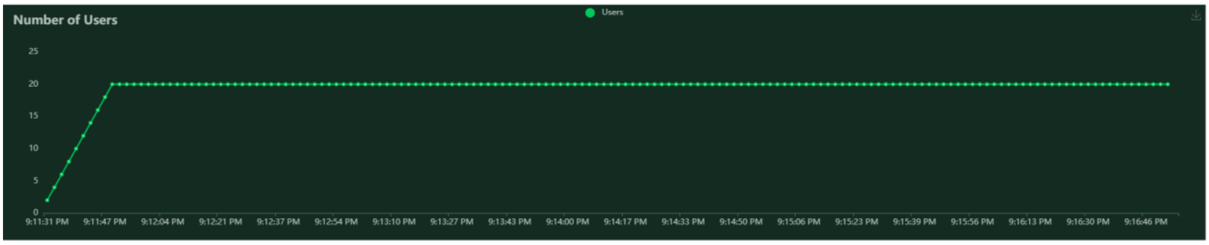
Average load times where 111 which is only slightly more than test 1 and 2 but still well within an acceptable range.

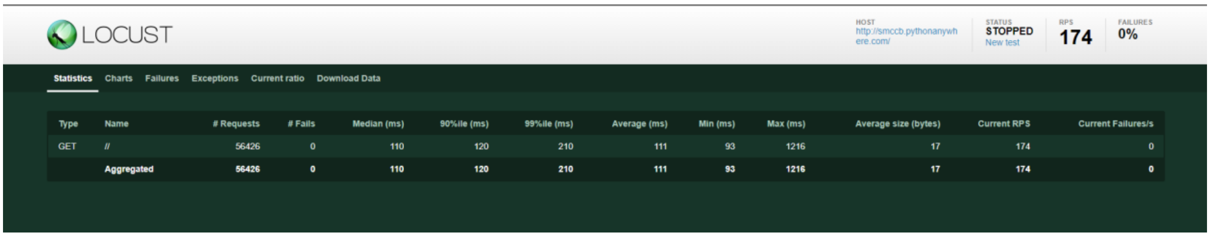
How are these results different from the previous tests?

This test was very similar to first two, it probably needed to have a larger user amount jump between tests here.









HOST http://smccb.pythonanywhere.com/

STATUS STOPPED

New test

RPS 174

FAILURES 0%

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	//	56426	0	110	120	210	111	93	1216	17	174	0
	Aggregated	**56426**	**0**	**110**	**120**	**210**	**111**	**93**	**1216**	**17**	**174**	**0**

Test Number:

4

Number of users:

50

Ramp up time:

1

Observations:

The total requests per second is less consistent and bumpier that the other tests 1-3. This is probably due to using a larger number of users and slowly ramping up from 1 causing a longer adjustment period.

Did any failures happen?

No failures during this test.

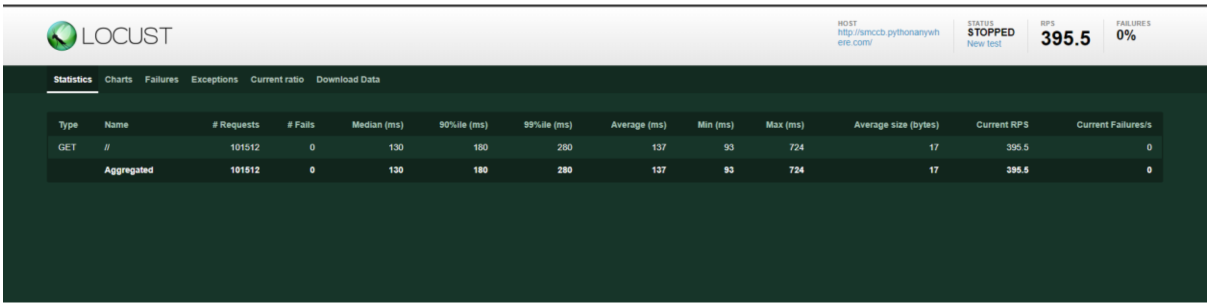
What did the load times look like?

Load times are fairly consistent but do throttle a little.

What was the average load time?

Is 137.

How are these results different from the previous tests?



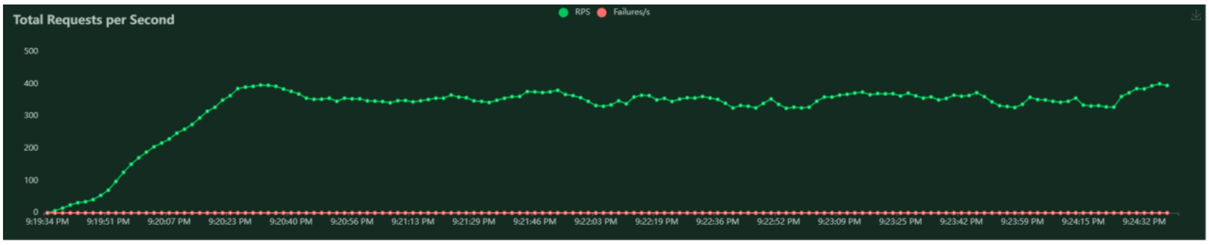
HOST
http://smccb.pythonanywhere.com/

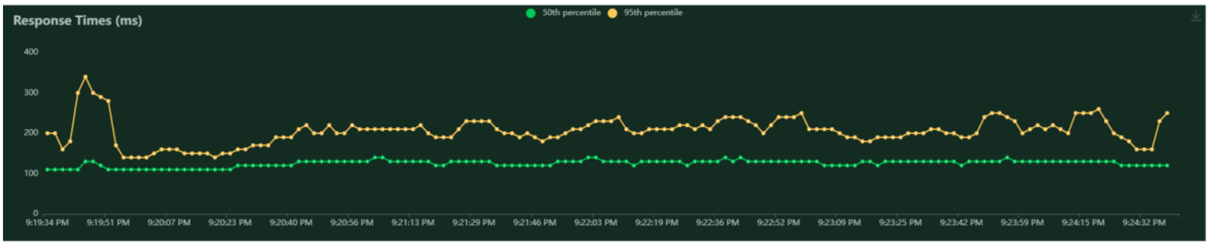
STATUS
STOPPED
New test

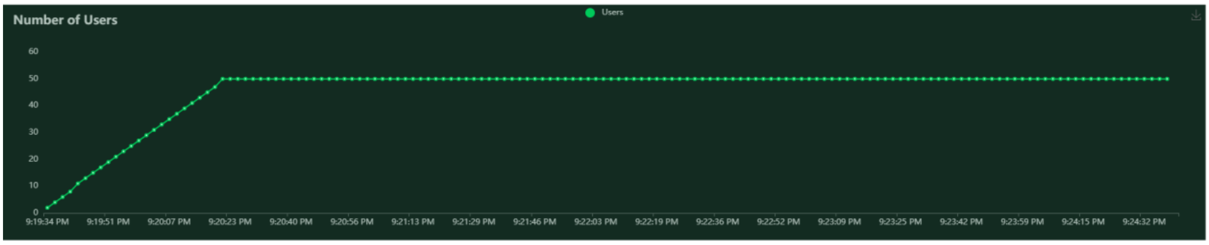
RPS
395.5

FAILURES
0%

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	//	101512	0	130	180	280	137	93	724	17	395.5	0
	Aggregated	**101512**	**0**	**130**	**180**	**280**	**137**	**93**	**724**	**17**	**395.5**	**0**







Test Number:

5

Number of users:

100

Ramp up time:

1 user second

Observations:

This test took a little bit to ramp up because I only did 1 user and should have probably done something more like 3 or 5.

This is the last test that shows the flask app being able to handle the requests. With an appropriate average load time and no fails. This means that the breakpoint of when the server can't handle that number of users reasonably.

Did any failures happen?

No

What did the load times look like?

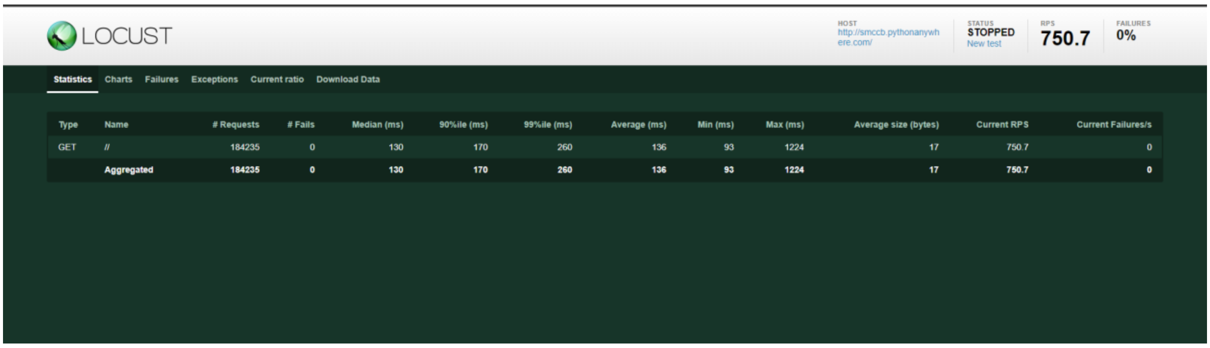
Load times for average seem fine at 136 while max is 1224 which users probably wouldn't wait for, and minimum is 93. This is one of the bumpier and shows that the highest and lowest load times are being switched between quite a lot and its not staying near a middle value, so it seems like the system is struggling a little. Or starting to.

What was the average load time?

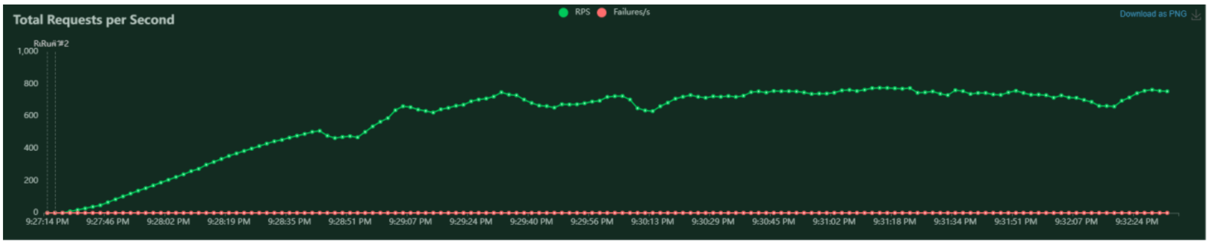
The average load time is 136.

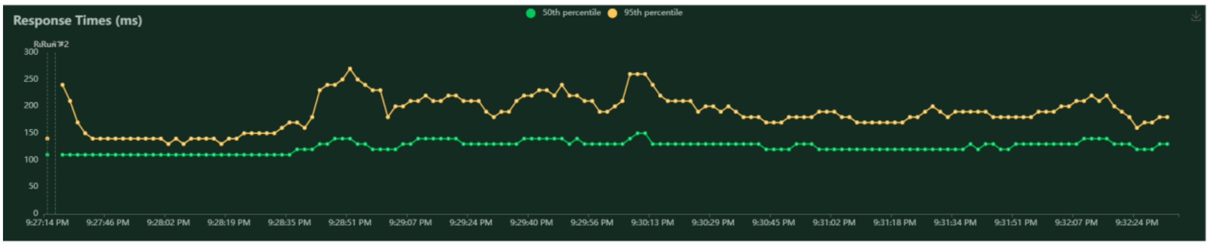
How are these results different from the previous tests?

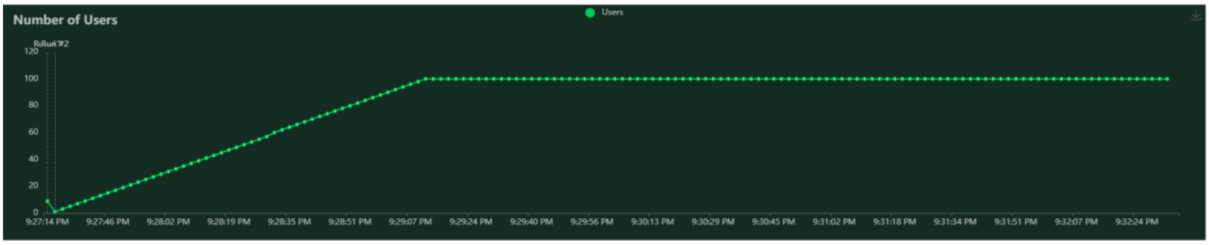
This is the last test that has no fails.



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	//	184235	0	130	170	260	136	93	1224	17	750.7	0
	Aggregated	**184235**	**0**	**130**	**170**	**260**	**136**	**93**	**1224**	**17**	**750.7**	**0**


Total Requests per Second


Response Times (ms)


Number of Users

Test Number:

6

Number of users:

500

Ramp up time:

5 users per second

Observations:

Probably should have ramped up by 10 users at a time as 5 seemed to show an adjustment in requests it was being sent per second.

This is the first test where fails had occurred and so the app should probably kept under 500 users as it doesn't seem manageable especially with the load times being quite high as well.

Did any failures happen?

Yes, but it does seem that python anywhere will try not fail requests but indeed serve them aver a very long load time. There where 242 fails but no current failures after running this test for 5 minutes. This was the first test with fails all others had no fails.

What did the load times look like?

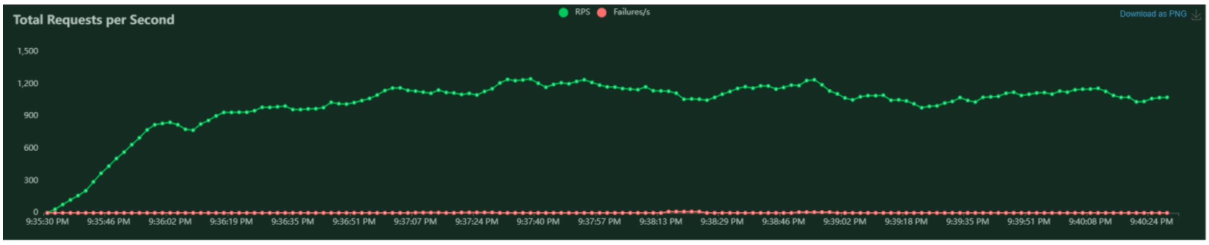
The average load times seemed quite consistent however it does seem to be getting longer the more time the test ran so if this where to have 500 users consistently it wouldn't not be manageable and wait time would be too long.

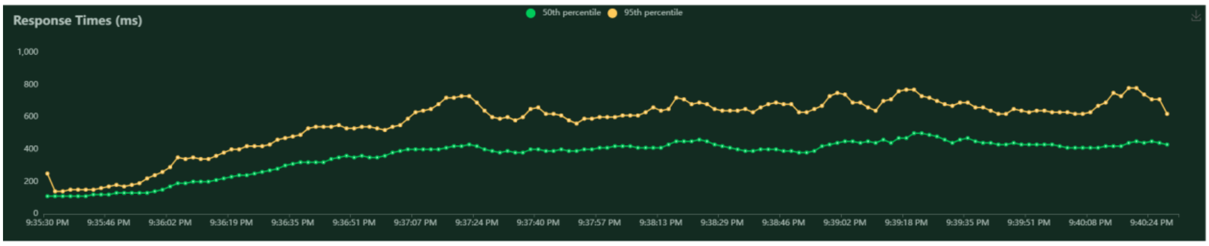
What was the average load time?

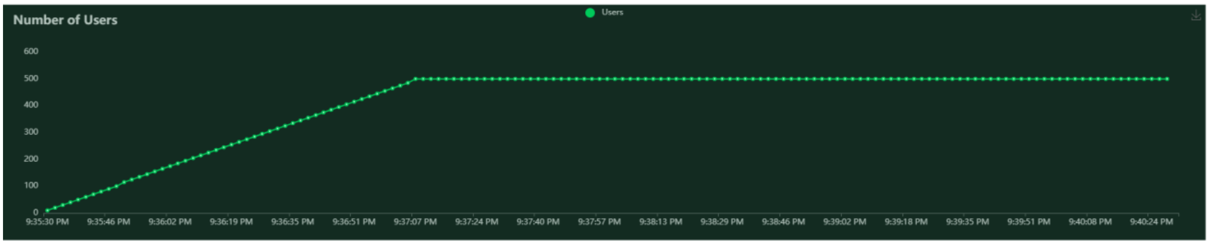
Average was 398 which is not a time user are going to wait. Some might but it is on the long side. Minimum is 93 while maximum reached 2637.

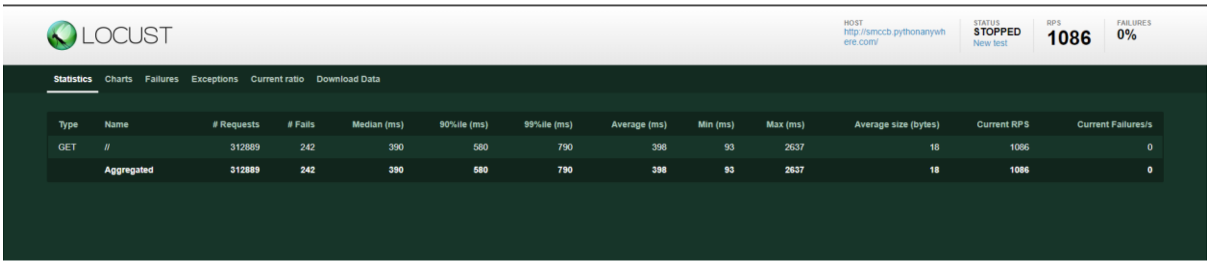
How are these results different from the previous tests?

This was the first test with any fails, this test also had poor response times, they were much higher than the 5th test however they where slightly more consistent not as bumpy, so they seemed to throttle the server less and just fail requests more.









HOST
http://smccb.pythonanywhere.com/

STATUS
STOPPED
New test

RPS
1086

FAILURES
0%

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	//	312889	242	390	580	790	398	93	2637	18	1086	0
	Aggregated	**312889**	**242**	**390**	**580**	**790**	**398**	**93**	**2637**	**18**	**1086**	**0**

Test Number:

7

Number of users:

1000

Ramp up time:

10 users per second

Observations:

The minimum response time is fairly consistent throughout all tests this shows minimum response time is not a good metric of how well a server is handling requests. The average and maximum seem to be a lot more accurate.

The number of fails for the 500 and 1000 users are nearly the same even with a lot more requests coming in.

The response times however seemed to not only throttle but also consistently go up the longer that the test ran showing that even if the server had managed to mostly handle the 1000 users for 5 minutes it would have just gotten worse and worse. So, the free server definitely is not suited to 1000 users.

Did any failures happen?

There were 247 total fails. This was quite close to the number of fails on the 500-user test (test6)

What did the load times look like?

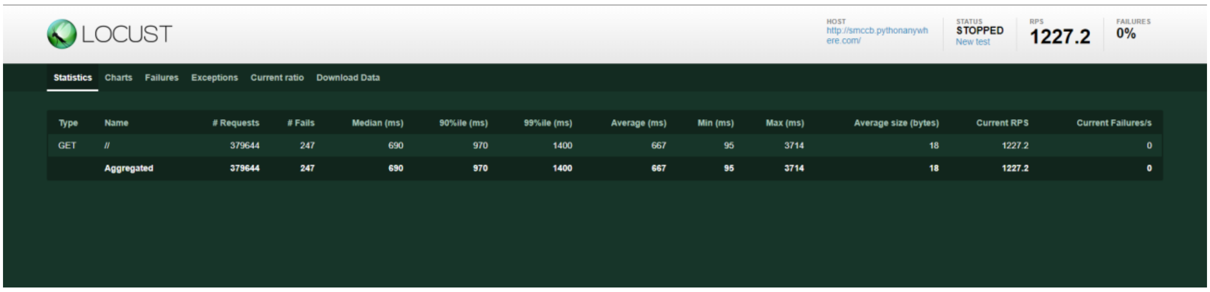
The average was 667, however the min was 95 and the max was 3714. So, with double the users from test 6 the average response time also doubled. The average seemed to be going up and can be seen through the green line going up in the second chart.

What was the average load time?

On average the load time was 667 and the chart seemed to indicate this number would grow the longer the test ran, the chart was also bumpy showing throttling.

How are these results different from the previous tests?

This test had much higher response times than the 6th test however the response times seemed to indicate that it was also throttling the server much more than 6 and response fails quite a lot more as well. Overall, this seemed like a drastic drop in usability although the 6th test was also not really a usable system. Like test 6 there where fails in requests.



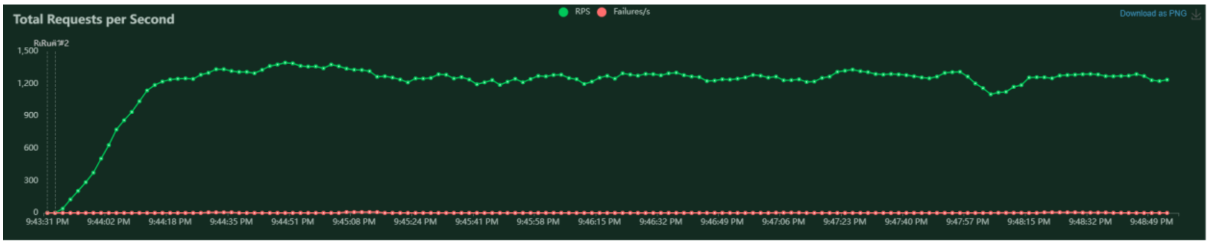
HOST http://smccb.pythonanywhere.com/

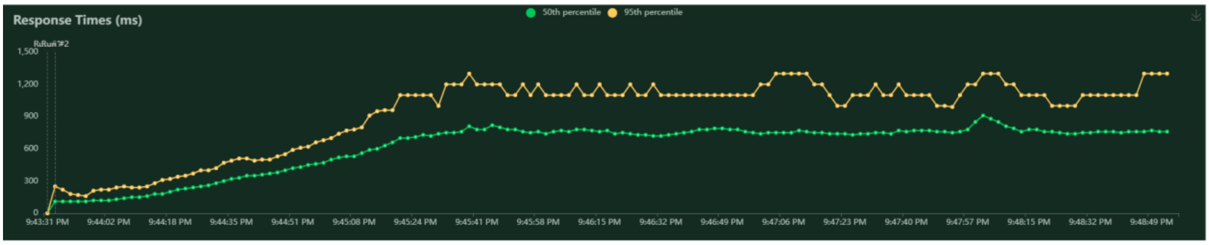
STATUS STOPPED New test

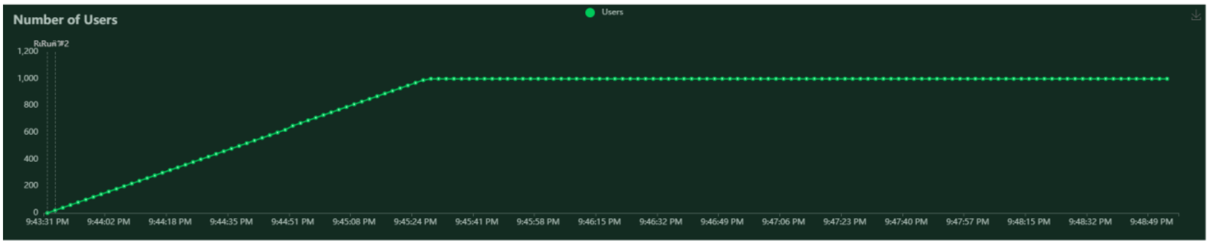
RPS 1227.2

FAILURES 0%

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	//	379644	247	690	970	1400	667	95	3714	18	1227.2	0
	Aggregated	**379644**	**247**	**690**	**970**	**1400**	**667**	**95**	**3714**	**18**	**1227.2**	**0**







For this assignment the python anywhere, flask was setup as part of the lab. The locust was downloaded, and the locust file was gotten from Moodle. Some details were changed such as how the fails where originally tracked with the requests. Then I just ran the 7 tests using the url of my web app no python anywhere and the different numbers of users being tested.