

Fast HTTP server

제출일	2022.10.04	전공	컴퓨터공학과
과목	컴퓨터네트워크(02)	학번	201602037
담당교수	이영석 교수님	이름	이규정

① 외부컴퓨터 환경

외부 컴퓨터: 라즈베리 파이3, IP주소: 192.168.1.165

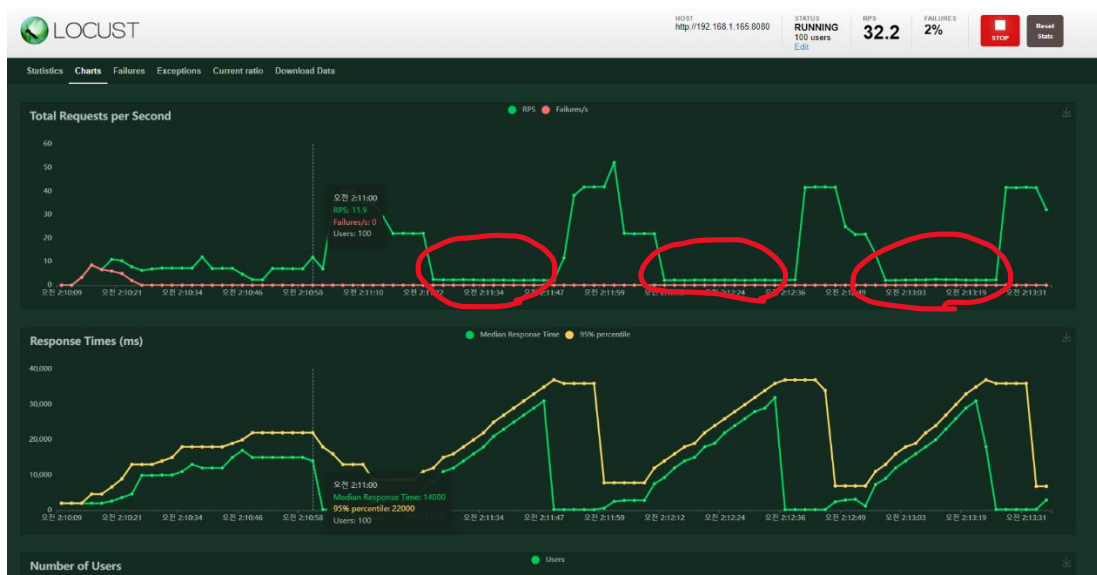
노트북: 무선 랜 연결

라즈베리 파이: 유선 랜 연결

Backlog: 1024

② Single process server

저번주차 과제코드에서 locust.py 코드에 wait_time만 수정해 준 결과입니다.



Single process server에서는 요청 데이터가 들어오면 데이터가 처리되고 다시 send 될 때까지 다른 요청이 들어오면 그 요청을 처리하지 못합니다. 즉, 요청들을 동시에 처리하지 못합니다. 이와 같은 현상이 발생하여 서버의 성능이 떨어지게 됩니다.

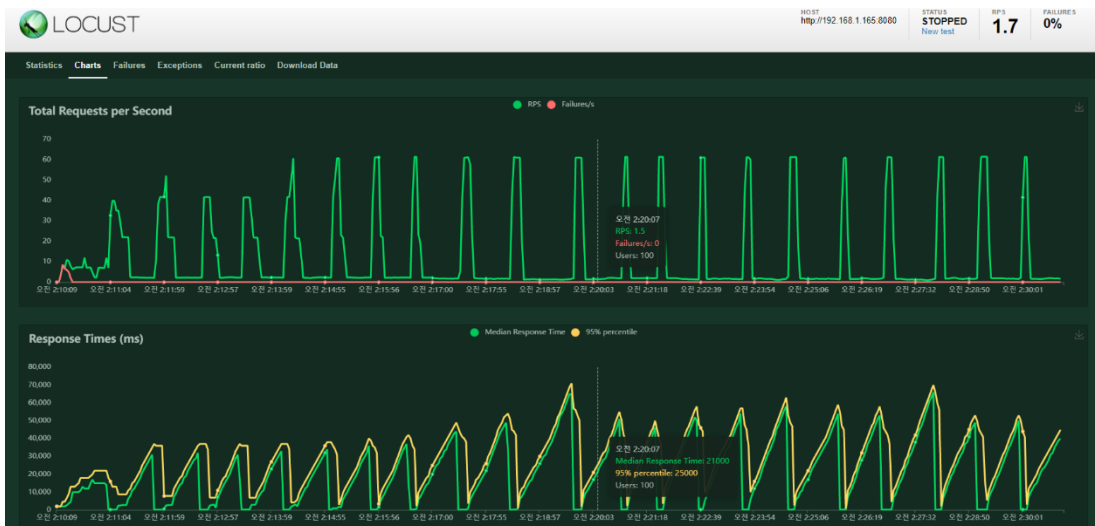
위의 차트에서 빨간색 표시가 된 곳에서 RPS가 굉장히 떨어지게 되는데, 이 부분에서 server의 Received protocol을 살펴본 결과 용량이 큰 이미지 파일을 요청하고 보내는 부분에서 RPS 드랍이 발생하였습니다.

LOCUST

HOST: http://192.168.1.165:8080 STATUS: STOPPED New test RPS: 1.7 FAILURES: 0%

Statistics Charts Failures Exceptions Current ratio Download Data

Type	Name	# Requests	# Fails	Median (ms)	50%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	2200	40	390	6800	10000	2004	10	17707	343	0	0
GET	/201602037.html	2100	0	180	220	260	182	12	1013	283	0	0
GET	/index.html	2200	20	210	1400	9900	713	11	14696	346	0	0
GET	/myimage.jpg	2181	0	23000	49000	66000	24910	511	72062	249924	1.7	0
GET	/myscript.js	2100	0	190	260	39000	2155	111	71161	115	0	0
GET	/mystyle.css	2101	0	200	240	16000	824	141	41819	84	0	0
GET	/notfound.html	2101	0	23000	49000	64000	24558	142	71493	121	0	0
Aggregated		14983	60	240	32000	57000	7912	10	72062	363956	1.7	0



실습시간에 말씀하셨던 failure 가 single process에서 몇 개의 요청을 받으면 발생하는지 관찰하기 위해서 20분 동안 관찰해보았지만 나타나지 않았습니다.

대략 14000개의 요청을 받았고, 특이한 점은 Current RPS 부분에서 이미 지 파일에는 항상 1.0 ~ 2.0 사이의 RPS가 계속 요청되고 있었습니다.

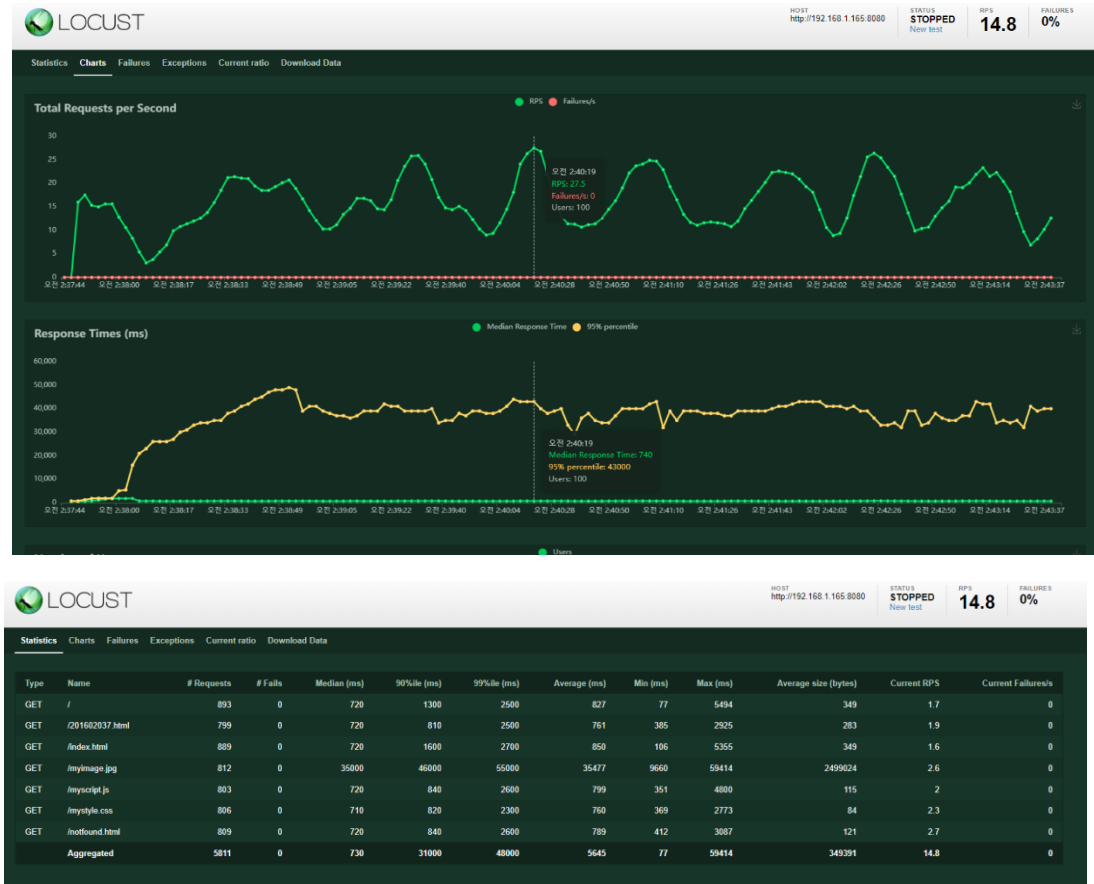
이 부분에 대한 의문점은 뒤에 multi process/thread server를 돌려 본 결과와 실습시간에 들었던 single process 동시에 여러 요청들을 처리하지 못한다는 부분에서 해결할 수 있었습니다.

Multi process server는 요청이 여러 개 들어와도 다른 파일(HTML, CSS 등)의 RPS가 0으로 되지 않고 계속 요청을 처리하였습니다.

따라서 위와 같은 결과는 single process가 요청이 밀려 있으면 다른 요청을 처리하지 못한다는 뜻으로 해석이 됩니다.

평균 응답시간은 7912ms 로 측정되었습니다.

③ Multi process server

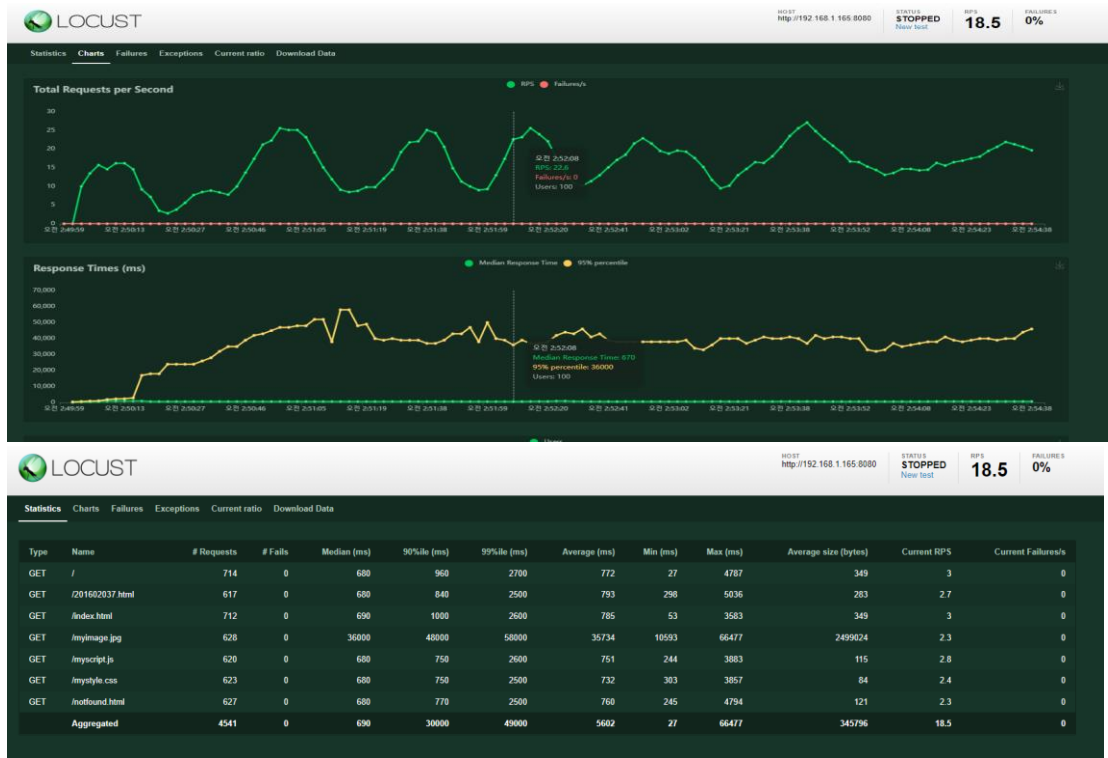


multi process server는 single process에 비해 전반적으로 우수한 성능을 보여주었습니다. RPS가 갑작스럽게 0에 수렴하는 부분도 보이지 않았고 안정적이게 요청이 수신되고 처리되었습니다.

또한 single process에서 관찰된 이미지파일이 수신되는 동안 다른 요청들의 RPS가 0이 되는 부분도 사라졌습니다. 이는 요청들을 다중 프로세스를 통해 동시적으로 요청들을 처리할 수 있음을 보여줍니다.

평균 응답시간은 5645ms로 측정되었습니다

④ Multi thread server



Multi thread server도 Multi process server와 유사하게 관찰됩니다. 마찬가지로 동시에 요청들을 처리하고 멀티 프로세스와 다른 점은 thread는 하나의 프로세스 안에서 요청들을 처리한다는 부분입니다.

평균 응답시간은 5602ms로 측정되었습니다.

⑤ 서버 성능 비교

성능적인 부분에서 single보다 multi process/thread 서버를 사용하는 것이 좋은가에 대한 부분은 single에서는 가장 높은 RPS가 30-40 까지 올라갔지만, multi 에서는 20-30에서 머무는 것을 볼 때 정확히 파악 할 수 없었지만 평균적인 RPS 수치로 보아 multi process/thread server가 좀 더 좋은 성능을 가지고 있다고 볼 수 있습니다.

평균적인 응답시간 부분에서는 multi thread(5602ms) > multi process(5645) > single process(7912ms)로 thread와 multi process는 거의 똑같은 응답시간이 오지만, single은 조금 낮은 응답시간을 보여줍니다.