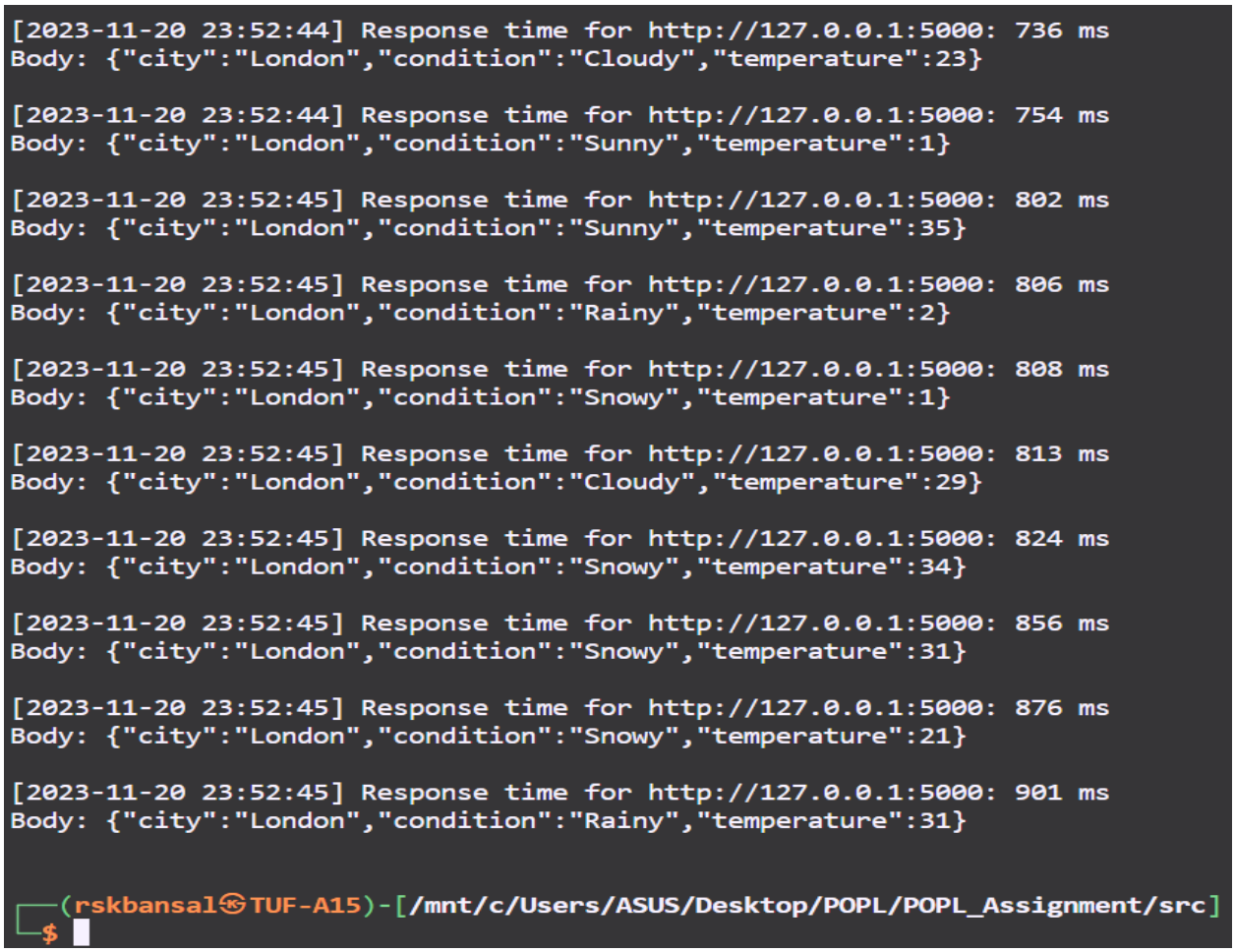
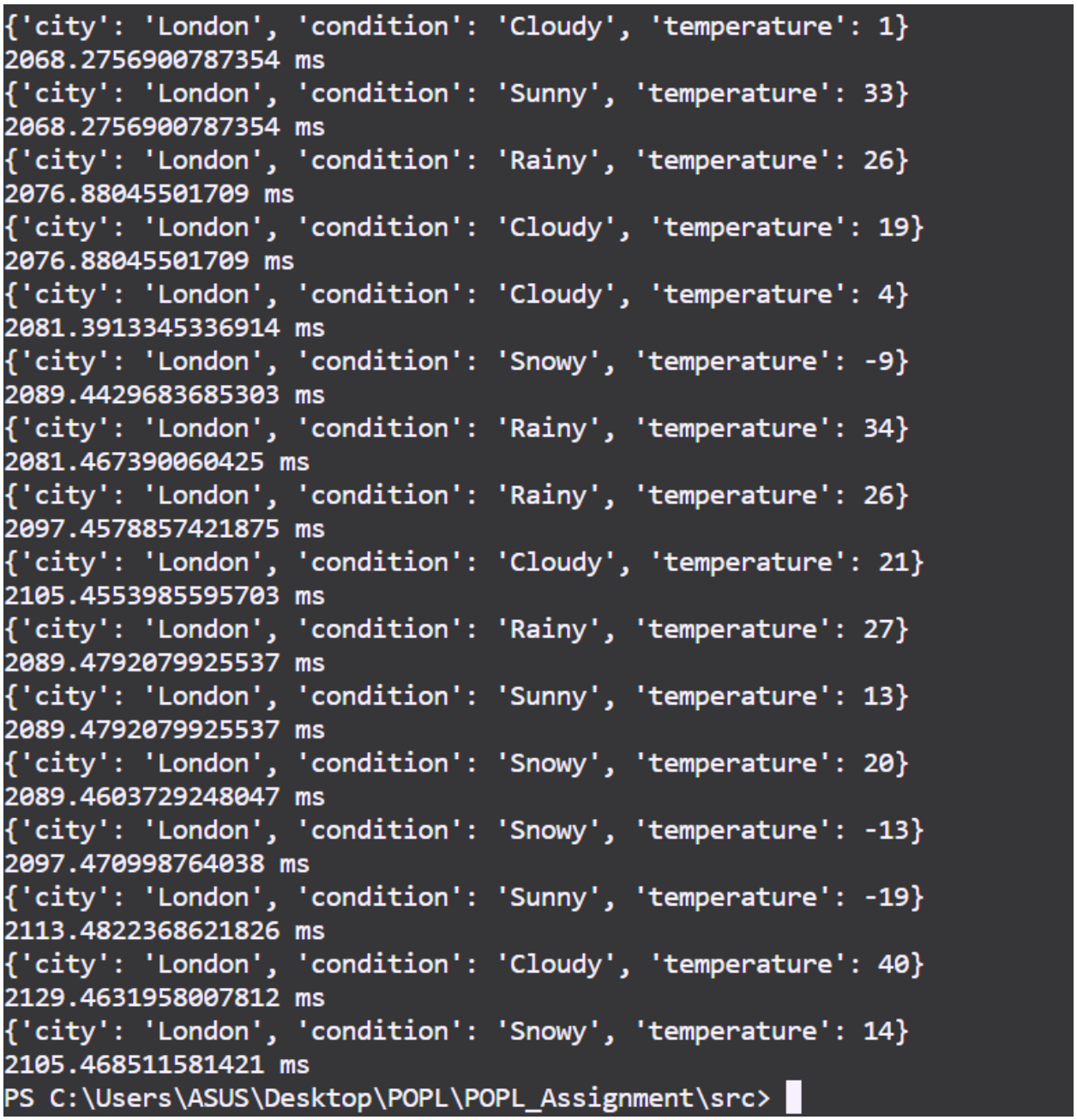
# Results

```
┌──(rskbansal㉿TUF-A15)-[/mnt/c/Users/ASUS/Desktop/POPL/POPL_Assignment]
└─$ ab -n 20 -c 20 "http://localhost:5000/get_weather?city=Goa"
This is ApacheBench, Version 2.3 <$Revision: 1903618 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done


Server Software:        gunicorn/19.10.0
Server Hostname:        localhost
Server Port:            5000

Document Path:          /get_weather?city=Goa
Document Length:        31 bytes

Concurrency Level:      20
Time taken for tests:   0.064 seconds
Complete requests:      20
Failed requests:        0
Non-2xx responses:      20
Total transferred:      3880 bytes
HTML transferred:       620 bytes
Requests per second:    312.46 [#/sec] (mean)
Time per request:       64.008 [ms] (mean)
Time per request:       3.200 [ms] (mean, across all concurrent requests)
Transfer rate:          59.20 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    1   0.4      1       1
Processing:     4   31  18.2     35      60
Waiting:        2   30  18.3     32      59
Total:          4   32  18.0     36      60

Percentage of the requests served within a certain time (ms)
  50%     36
  66%     44
  75%     47
  80%     50
  90%     57
  95%     60
  98%     60
  99%     60
 100%     60 (longest request)
```

# APACHE BENCHMARK



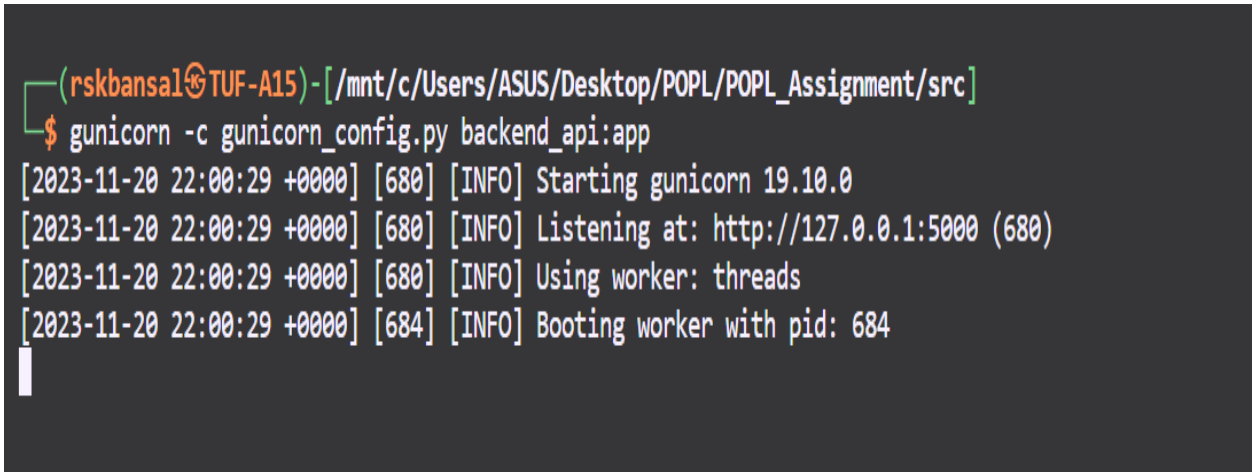

# RESULTS FOR C++




# RESULTS FOR PYTHON

```
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 338.7761ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Rainy\",\"temperature\":38}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 344.1674ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Cloudy\",\"temperature\":3}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 346.0162ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Cloudy\",\"temperature\":31}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 348.2016ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Rainy\",\"temperature\":12}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 350.5684ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Sunny\",\"temperature\":16}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 351.55ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Snowy\",\"temperature\":-9}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 353.2503ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Rainy\",\"temperature\":27}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 355.7076ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Sunny\",\"temperature\":37}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 358.7067ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Sunny\",\"temperature\":40}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 362.4549ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Sunny\",\"temperature\":13}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 364.2761ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Rainy\",\"temperature\":5}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 366.6395ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Rainy\",\"temperature\":35}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 368.9062ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Sunny\",\"temperature\":23}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 369.9165ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Cloudy\",\"temperature\":12}\n")
Response time for http://localhost:5000/get_weather?city=London&api_key=p0pl-15-fun: 373.988ms
body = Ok(b"{\"city\":\"London\",\"condition\":\"Sunny\",\"temperature\":-2}\n")
PS C:\Users\ASUS\Desktop\POPL\POPL_Assignment>
```

# RESULTS FOR RUST




# CODE FOR SERVER

## Conclusion for Results:

In conclusion, the observed data unequivocally demonstrates that the retrieval time for HTTP requests using a Rust client significantly outpaces its C++ counterpart. The Rust implementation exhibits an impressive average fetching time of approximately 350 milliseconds, notably surpassing the doubled duration observed in the corresponding C++ client. This compelling evidence underscores the superior performance of Rust in handling HTTP requests concurrently, reaffirming its prowess in efficient and expedient data retrieval.

Several factors could contribute to Rust outperforming C++ in time fetching for HTTP requests. Here are some potential reasons:

Async/Await Model in Rust:
1.) Rust's async/await syntax, especially when combined with the tokio runtime, allows for efficient asynchronous programming. This can result in better utilization of resources and improved performance when dealing with concurrent tasks, such as making multiple HTTP requests concurrently.

2.) Concurrency Without Threads Overhead:
Rust's ownership system enables concurrency without relying heavily on traditional threads. Asynchronous tasks in Rust can be multiplexed on a single thread, avoiding the overhead associated with thread creation and management. In contrast, C++ relies on threads for concurrency, which might introduce more overhead.