server_test

May 5, 2023

[56]:
```
import os
import subprocess

import numpy as np
import requests
import time
import concurrent.futures
import matplotlib.pyplot as plt
```

[33]:
```
def send_rest(url, method, payload=None):
    try:
        if method == "GET":
            response = requests.get(url)
        elif method == "POST":
            response = requests.post(url, json=payload)
        elif method == "PUT":
            response = requests.put(url, json=payload)
        elif method == "DELETE":
            response = requests.delete(url)
        else:
            raise ValueError(f"Invalid method {method}")
    except:
        print(f"Request failed: {url}")
        return None
    return response


def rest_test(dns, request_list, num_requests):
    # print server name
    server = f"http://{dns}:{rest_port}"
    country = server.split('.')[1]
    print(f"Server: {country}")
    # test every request
    response_times_dict = {}
    for request in request_list:
        url, method, *args = request
        payload = args[0] if args else None
        # send requests concurrently
        with concurrent.futures.ThreadPoolExecutor() as executor:
            response_times = []
            for i in range(num_requests):
                start_time = time.perf_counter()
                response = send_rest(server + url, method, payload)
```

```
                end_time = time.perf_counter()
                response_time = end_time - start_time
                response_times.append(response_time)
            if url in response_times_dict:
                response_times_dict[url].extend(response_times)
            else:
                response_times_dict[url] = response_times
            # total time rounded to 3 decimal places
            total_time = round(sum(response_times), 3)

            print(f"URL: {url}, Method: {method}, Total Response Time: {total_time}")

    return response_times_dict
```

[57]:
```
server_list = [
    ('alnike.japaneast.cloudapp.azure.com', '20.210.110.130'),
    ('uswest-thiers.westus3.cloudapp.azure.com', '20.106.100.68'),
    ('useast-lennart.eastus2.cloudapp.azure.com', '20.1.139.66'),
    ('dapps.westeurope.cloudapp.azure.com', '98.71.185.120')
]

num_requests = 1
rest_port = 8081

rest_request_list = [
    ['/rest/order', 'POST', {'address': '123 Main St', 'meals': ['Portobello',␣
 ↪'Portobello']}],
    ['/rest/meals', 'GET'],
    ['/rest/largest-meal', 'GET'],
    ['/rest/cheapest-meal', 'GET']
]

response_times_dict = {}

# get this server's ip
this_ip = requests.get('https://api.ipify.org').text

for dns, ip in server_list:
    # if this is the current server, skip
    if ip == this_ip: continue

    # test REST
    response_times = rest_test(dns, rest_request_list, num_requests)

    for url, times in response_times.items():
        if url not in response_times_dict:
            response_times_dict[url] = {}
        response_times_dict[url][dns] = times
    print('\n')
```

```
Server: japaneast
URL: /rest/order, Method: POST, Total Response Time: 0.616
URL: /rest/meals, Method: GET, Total Response Time: 0.562
URL: /rest/largest-meal, Method: GET, Total Response Time: 0.486
```

```
URL: /rest/cheapest-meal, Method: GET, Total Response Time: 0.599


Server: westus3
URL: /rest/order, Method: POST, Total Response Time: 0.409
URL: /rest/meals, Method: GET, Total Response Time: 0.335
URL: /rest/largest-meal, Method: GET, Total Response Time: 0.31
URL: /rest/cheapest-meal, Method: GET, Total Response Time: 0.364


Server: eastus2
URL: /rest/order, Method: POST, Total Response Time: 0.665
URL: /rest/meals, Method: GET, Total Response Time: 0.568
URL: /rest/largest-meal, Method: GET, Total Response Time: 0.215
URL: /rest/cheapest-meal, Method: GET, Total Response Time: 0.25


Server: westeurope
URL: /rest/order, Method: POST, Total Response Time: 0.096
URL: /rest/meals, Method: GET, Total Response Time: 0.046
URL: /rest/largest-meal, Method: GET, Total Response Time: 0.029
URL: /rest/cheapest-meal, Method: GET, Total Response Time: 0.037
```