**Report**

Name : Preet Jain

Nsid : pvj636

Student ID : 11270494

**Experimental Setup:**

The main aim of this experiment is to compare effectiveness of SOAP and REST performances in similar situations. SOAP is a lightweight communications protocol for exchange of information in a decentralized distributed environment. It usually provides an XML format for communication between client and server. Whereas REST is more like an architectural style than a protocol. In REST state and functionality are divided in distributed resources which are accessible through normal HTTP commands like PUT, POST, GET and DELETE. In the end the aim of both REST and SOAP is to enable web applications build on different programming languages to communicate with each other while coming scalability, efficiency in mind.

To compare SOAP and REST against each other I have chosen a simple chat application setup. In this application user can be added, removed, and can message other users individually in the application. For SOAP to create a web service I have used C# and .NET framework, and to create a RESTFul web service I have used Python with flask RESTFul. Flask RESTFul makes it a bit easier to code a lightweight RESTFul service. In SOAP I have a simple arraylist which acts as a Queue for incoming messages, the client’s thread will always check in with for any new messages directed to them, the server will also keep track of users connected to the service. For REST I have a similar setup but the users, messages are treated as resources. The setup is kept similar so that the comparison is more accurate.

**Experiment:**

The experiment is to perform different kind of tests on both the web services, compare the results and try to conclude about which service is better. The tests that I have performed include some data driven tests and load tests. For testing purposes, I have used readyAPI. This software will essentially act as a client and give us results in the form of numbers and graph which are shown below. ReadyAPI lets me add data sources for each feature/method to perform data driven testing and with load testing I have decided on different number of users, increasing time the services are being used for and plotting them on a graph.

As you can see from the table below

The tabe for REST: Table foe SOAP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Method* | *VUs* | *Amt. of time test ran* | *Max/Min requests time* | *Avg Requests time* |
| addUser  (POST) | 10 | 1 min | 28/2 | 13 |
|  | *15* | 2 min | 28/3 | 8 |
|  | 25 | 4 min | 58/3 | 11 |
| Message  (Post) | 10 | 1min | 30/2 | 10 |
|  | 15 | 2min | 32/3 | 13 |
|  | 25 | 4 min | 55/3 | 15 |
| getUSer  (Get) | *10* | 1 min | 49/3 | 12 |
|  | *15* | 2min | 62/3 | 15 |
|  | *25* | 4min | 84/4 | *17* |
| userDelete  (Delete) | *10* | 1 min | 16/1 | *4* |
|  | *15* | 2min | 29/1 | 5 |
|  | *25* | 4min | 43/1 | *7* |
|  |  |  |  |  |
|  |  |  |  |  |

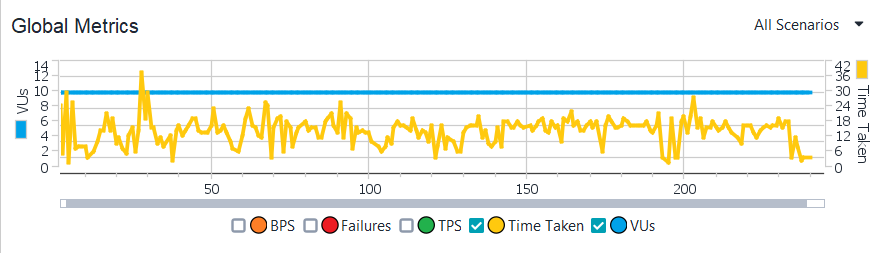
As we can see form the tables for SOAP and REST, the time taken per request is lower overall for the REST server, than the SOAP.

By analizing the graphs and some other metric shown in readyAPI data volume it is clear that average request are pretty consistent in REST meanwhile it drops its inconsistent in SOAP. But the data volume for both remains pretty consistent.

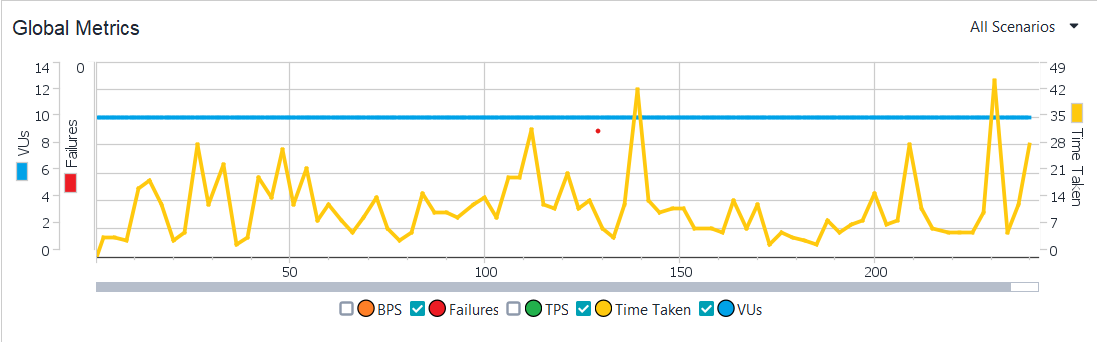
Hence, based on our experiment we can easily conclude that REST is a better service when it comes to scalability and expansion of a software.When launching it to a larger user set, REST will give a smooth and uninterrupted experience.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Method* | *VUs* | *Amt. of time test ran* | *Max/Min requests time* | *Avg Requests time* |
| addUser  (POST) | 10 | 1 min | 88/4 | 8 |
|  | *15* | 2 min | 99/4 | 15 |
|  | 25 | 4 min | 144/3 | 17 |
| getUser  (GET) | 10 | 1min | (201)/8 | 24 |
|  | 15 | 2min | (664)/24 | 160 |
|  | 25 | 4 min | 2888/178 | 991 |
| SendMessage  (POST) | *10* | 1 min | 33/3 | 9 |
|  | *15* | 2min | 80/4 | 08 |
|  | *25* | 4min | 189/3 | *9* |
| userDelete  (Delete) | *10* | 1 min | 35/4 | *9* |
|  | *15* | 2min | 37/4 | *7* |
|  | *25* | 4min | 48/4 | *9* |
|  |  |  |  |  |
|  |  |  |  |  |

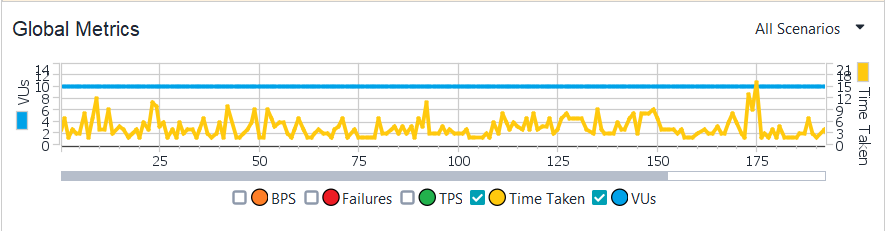
Graphs for Rest :

Graph for getUser(GET)

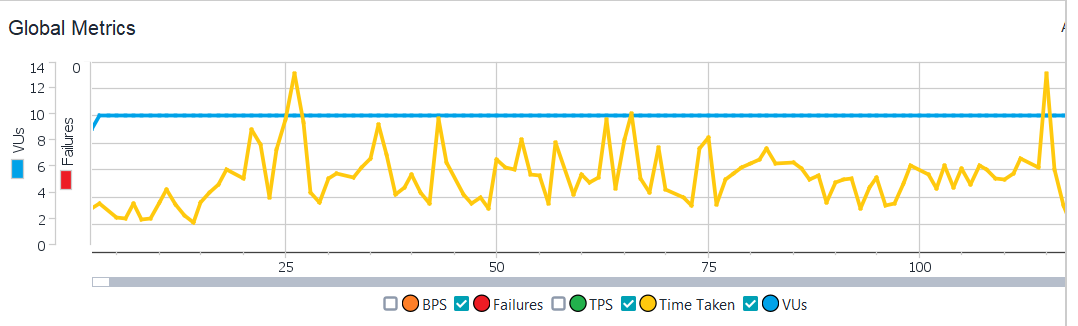
Grpah for sendMEssage(POST):



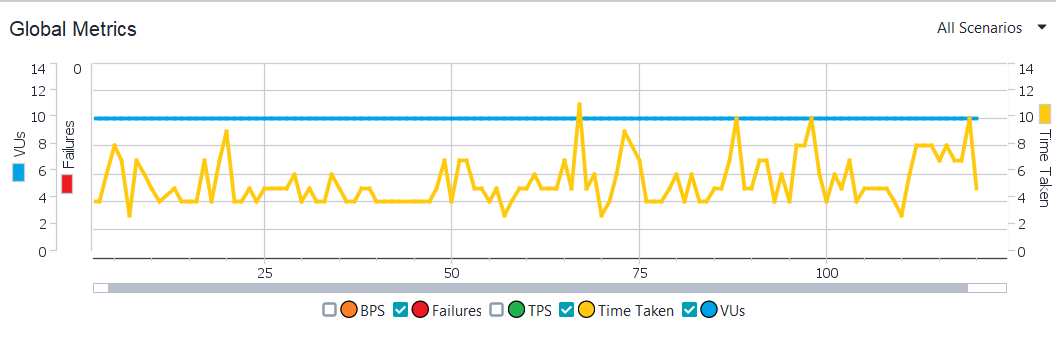
Graph for userDelete(DELETE):



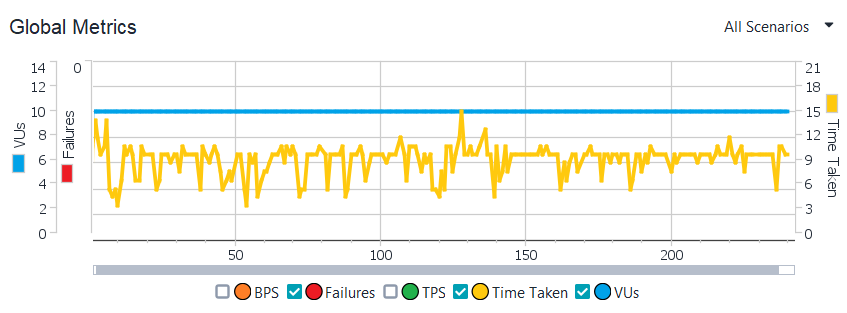
SOAP: getUSer(GET)



sendMessage(POST):



userDelete(DELETE)



Below is the google drive link to my presentation :

https://drive.google.com/file/d/1I2WgbVquuPGuf4G8\_GpgADB68eapmnD2/view?usp=sharing