DOS homework

Mohammad Ghulmi
Shokry Yahya

11/11/2020

1 program design

to build this program we first created a python virtual environment to manage all the dependencies of the program. then we split the program into 3 pieces that communicate with each other

1. Catalog server : it manages the database(we used a file) and it's responsible for searching and editing the database
2. order server It manages the process of buying books by first asking the catalog server to see if any stock is available and then tells the catalog server to reduce the stock
3. front end server it acts as a middle man between the back end servers and the user

when the client enters that he wants to search in the front end a request is sent to the catalog server with category that the user searches for.
the catalog server then searches the database for any books with topics that match the requested and sends back the result in the form of json

```
request is get
{'books': 'title : How-to-get-a-good-grade-in-DOS-in-20-minutes-a-day , id : 1 , title : RPCs for Dummies , id : 2
```

when the front end receives the response it displays the result to the user
when the client enters that he wants to buy a request is sent to the order server which query and edits the catalog server and displays the result

```
request is post
{'result': 'success'}
```

2 trade offs

the main trade off we made was in the post and put requests by passing the value through the url it reduced security but improved performance and readability

3 improvements

the program can be improved by using an actual database instead of a file we can also add a gui instead of using the console and the order server can be made to manage a part of the database instead of just telling the catalog server

4 running the program

1. you need to start the virtual environment by using as depicted below

```
C:\Users\MSI\Documents\Flask\fenv>cd scripts

C:\Users\MSI\Documents\Flask\fenv\Scripts>activate

(fenv) C:\Users\MSI\Documents\Flask\fenv\Scripts>
```

2. check to make sure all the dependencies are installed

```
(fenv) C:\Users\MSI\Documents\Flask\fenv\Scripts>pip list
Package          Version
---------------- ---------
aniso8601        8.0.0
certifi          2020.11.8
chardet          3.0.4
click            7.1.2
Flask            1.1.2
Flask-RESTful    0.3.8
Flask-SQLAlchemy 2.4.3
idna             2.10
itsdangerous     1.1.0
Jinja2           2.11.2
MarkupSafe       1.1.1
pip              20.2.4
pytz             2020.1
requests         2.24.0
setuptools       49.2.1
six              1.15.0
SQLAlchemy       1.3.18
urllib3          1.25.11
Werkzeug         1.0.1
```

here is a list of all dependencies aniso8601==8.0.0 click==7.1.2 Flask==1.1.2 Flask-RESTful==0.3.8 Flask-SQLAlchemy==2.4.3 itsdangerous==1.1.0 Jinja2==2.11.2 MarkupSafe==1.1.1 pytz==2020.1 six==1.15.0 SQLAlchemy==1.3.18 Werkzeug==1.0.1

note you need to do the same for every cmd or virtual machine you're running on

3. catalog first then order then front end each on different cmd or virtual machine

4. you Need to repeat these steps for every replicated server

5 replication

we created 2 more copies of the server that runs on different ports or IPs then in the front end server we created a round robin algorithm to manage request to each of the servers

5.1 response time

we noticed that for non cached items the response time increases by 2 seconds fro each failed server since the timeout on requests is 2 seconds but the cache reduces the time of any request