Exam 1 - Requires Respondus LockDown Browser - R...

Attempt 1 of 1

Written Feb 19, 2025 3:01 PM - Feb 19, 2025 4:48 PM

Attempt Score **95.5 / 100 - 95.5 %**

Overall Grade (Highest Attempt) **95.5 / 100 - 95.5 %**

Firebase [20 points]

Consider writing Python code to manage the user information stored under a given node in a Firebase database located at db_url. Assume the code starts with:

```
import requests
db_url = "https://dsci551-default-rtdb.firebaseio.com/"
node = “users”
# That is, https://dsci551-default-rtdb.firebaseio.com/users.json contains all user information

#step 1: fill in code

#step 2:  fill in code

#step 3: fill in code

#step 4: fill in code
```

Question 1 — 5 / 5 points

Fill in the code (step 1) to add a new user whose id is 100, name is “John”, and age is 25.

```
requests.put('https://dsci551-default-rtdb.firebaseio.com/users/100.json', json = {"name": "John", "age": 25})
```

Question 2 — 5 / 5 points

Fill in the code (step 2) to increase the age of user 200 by 1. Assume that the database already has information about some users including user 200.

```
age = requests.get('https://dsci551-default-
rtdb.firebaseio.com/users/200/age.json')
new_age = int(new_age) +1

requests.patch('https://dsci551-default-
rtdb.firebaseio.com/users/200.json', json = {"age": new_age})
```

Question 3 5 / 5 points

Fill in the code (step 3) to add a new attribute “gender” for user 200 and set its value to “male”.

```
requests.patch('https://dsci551-default-
rtdb.firebaseio.com/users/200.json', json = {"gender": "male"})
```

Question 4 5 / 5 points

Fill in the code (step 4) to find all users who are more than 25 years old. You can assume that a proper index has been created for you. Return user information in a dictionary with user ids as keys.

```
import json

response = requests.get('https://dsci551-default-
rtdb.firebaseio.com/users.json?orderBy="age"&startAt=25')

result = json.dumps(response.text())
print(result)
```

Storage System [15 points]

Consider a hard disk drive with the following characteristics:

- 10,000 RPM
- 12ms maximum seek time
- Block size = 4KB
- Transmission bandwidth (maximum) = 100MB/sec

Question 5 2 / 2 points

What is the average rotational latency of the device? Show your derivation.

10000 RPM = 1 min/10000 rotations = 60 sec/10000 rotations = 60000ms/10000 rotations = 6 ms/1 rotation

The average rotational latency = 6ms/2 = 3ms

Question 6 2 / 2 points

What is the average seek time? Explain your answer.

The average seek time = (the maximum seek time)/ 3
The maximum seek time = 12ms, so the average seek time is 12ms/3 = 4ms

Question 7 2 / 2 points

Suppose there is 4MB of data stored sequentially on the same track of the device. How many blocks does the data occupy? Show your derivation. Note you should not take an approximation.

4MB = 4*2^10KB

4*2^10KB/4KB = 2^10 = 1024

It will take 1024 blocks to store 4MB data.

Question 8 4 / 4 points

Suppose the disk head is at a random position. Compute the completion time of reading the above 4MB sequential data. Show your derivation.

Completion time = seek time + rotation latency + transfer time

Transmission bandwidth (maximum) = 100MB/sec = 100MB/1000ms

Transfer time of 4MB sequential data = 4MB/(100MB/1000ms) = 40ms

The total number of blocks = 1024
The average seek time is 12ms/3 = 4ms
The average rotational latency = 6ms/2 = 3ms

The completion time = 4+3+40 = 47ms

Question 9 2 / 2 points

In the completion time obtained above, which factor, latency or transmission, dominates? Explain your answer.

Transmission dominates the completion time.
Since the whole 4MB data is stored in sequential layout on a same track, the disk only has to seek the right track and wait the disk head rotate to the start point of the track one time.
So, the completion time mostly depends on transfer time, which is transmission.

Question 10 3 / 3 points

Now suppose the above 4MB data are now randomly distributed over the device. Will it take longer to finish the read? Explain your answer.

Yes, it will take longer to finish the read.
Since data are randomly distributed over the device, the disk has to do seeking and waiting for the disk head rotate to the start point of the sector multiple times, so that it will take longer to finish the read.
In this case, transmission will no longer dominate the completion time, instead latency will be.

File System [15 points]

Question 11 8 / 8 points

Explain what each of the following commands does.

a. ls -l
b. ls -a
c. mkdir
d. rm
e. cat
f. nano
g. cd

ls -l: list the details of the files and directories under the current path.
ls -a: list all files and directories under the current path, including hidden files.
mkdir: means make directory, create a directory.

rm: means remove, remove a file.
cat: print out the content of the file.
nano: open an edit interface inside the terminal, allowing you to edit the content of the file.
cd: means change directory, which allows you to switch from current directory the others.

Question 12 3 / 3 points

Explain what “chmod 400 hello.txt” does. Will this command change the “modify” timestamp of hello.txt? Explain your answer.

"chmod 400 hello.txt" will change the permission of the file to read only, and this command will not change the "modify" timestamp of hello.txt, because the "modify" timestamp only changes when the content of the file is updated. The "chmod 400 " command only changes the metadata of the file, the content of the file doesn't change, so the "modify" timestamp won't change.

Question 13 4 / 4 points

Give at least four examples on the information stored in the inode of a file.

The name of the file
Inumber
Size of the file
Access time
Modify time
Change time
The permission of the file

HDFS [20 points]

Suppose you have just started an HDFS (using command: sbin/start-dfs.sh). Suppose the file system is currently empty (with no files or directories) and you are now under the Hadoop home directory (e.g., hadoop-3.4.1).

Question 14 6 / 6 points

Now suppose you execute “bin/hdfs dfs -mkdir /user/john/dsci551”. Would this command succeed? If not, give commands to create the desired directory (i.e., /user/john/dsci551). For each command, indicate which server (namenode or datanode or both) should the HDFS client program (i.e., bin/hdfs) contact, and what actions should the server take?

It will not succeed.

Since /user and /user/john/ don't exist yet, you have to create nested directories.

1. bin/hdfs dfs -mkdir /user
HDFS client should contact namenode, update the hierarchy of directories.

2. bin/hdfs dfs -mkdir /user/john/
HDFS client should contact namenode, update the hierarchy of directories.

3. bin/hdfs dfs -mkdir /user/john/dsci551
HDFS client should contact namenode, update the hierarchy of directories.

Question 15 1.5 / 2 points

Now give a command to upload a text file “README.txt” located under the Hadoop home directory to the /user/john/dsci551 on HDFS.

bin/hdfs dfs -put "README.txt" "/user/john/dsci551"

Question 16 5 / 5 points

Explain which server(s) should the HDFS client contact for the above upload command and what actions should the server(s) take.

the HDFS client should contact both namenode and datanode.
First, contacting namenode to allcate a block and its replications for this file.
Second, contacting datanode to write the data into the blocks and its replications.
Third, contacting namenode to update the metadata for this file.

Question 17 3 / 3 points

The default block size in HDFS is 128MB. Explain why such a large block size (compared to block size in HDD, for example).

Because larger block size can reduce the storage size of metadata, and also it increases the speed of reading data(since data is stored in sequentail layout).

Question 18 4 / 4 points

For each of the following RPC calls, explain its purpose.

a. getBlockLocations
b. create
c. addBlock
d. writeBlock

getBlockLocations: To get the Located Blocks of the file, which contains datanodes, offsets, and the locations of its replication blocks.

create: To create a new block for storing the file.

addBlock: To allocate a block to the file, which contains the block and datanodes the block should be replicated to.

writeBlock: write data into the block and datanodes the block should be replicated to.

XML & XPath [15 points]

Consider the following XML file storing HDFS information.

```
<fsimage>
    <inodes>
        <inode>
            <id>100</id>
            <name/>
            <type>directory</type>
        </inode>
```

```
        <inode>
            <id>101</id>
            <name>user</name>
            <type>directory</type>
        </inode>
        …
    </inodes>
    <directories>
        <directory>
            <parent>100</parent>
            <child>101</child>
            <child>102</child>
        </directory>
        <directory>
            <parent>101</parent>
            <child>103</child>
        </directory>
        …
    </directories>
</fsimage>
```

Note inode type will be either “file” or “directory”.
Write an XPath expression for each of the following questions.

Question 19 — 2.5 / 3 points

Find name of a directory whose id = 102. Return actual name (not element).

```
'/fsimage/inodes/inode[id = 102 and type = "directory"]/name/text()'
```

Question 20 — 1.5 / 3 points

Find id of inode which has an empty name subelement. Return id value (not element).

```
'/fsimage/inodes/inode[not(name)]/id/text()'
```

Question 21 — 1.5 / 3 points

Find id of directory which has two children whose ids are 104 and 105. Return id value (not element).

'/fsimage/inodes/inode[/fsimage/directories/directory[child = 104 and child = 105] and type = "directory"]/id/text()'

Question 22 3 / 3 points

Find names of directories that start with “dsci”. Return names (not elements).

'/fsimage/inodes/inode[starts-with(name, "dsci") and type = "directory"]/name/text()'

Question 23 3 / 3 points

Find names of files whose name contains “hw” and id is between 100 and 200 inclusive. Return names (not elements).

'/fsimage/inodes/inode[contains(name, "hw") and type="file" and id >= 100 and id =< 200]/name/text()'

Miscellaneous [15 points]

Question 24 4 / 4 points

Which of the following JSON values are invalid? Explain your answer.

a. {‘name’: ‘john’}
b. {25: “age”}
c. [25, “age”]
d. [{}]

c and d are valid.

a is invalid since it uses single quote.
b is invalid since it doesn't use double quote on key.

c is valid since json accepts array, and array accepts numbers and strings in double quote.
d is valid since json accepts array, and array accepts empty object.

Question 25 2 / 2 points

Explain P/E cycle of SSD.

P/E cycle of SSD represents a iteration of Program(write) and Erase

For example, the initial state of the page is 1111(empty)

if we write some data into the page and change the state to 1001, this is called one program.
And the we erase the page (actually is the whole block), the state will return to 0000, this is called one erase.

we call one program + one erase a P/E cycle of SSD.

Question 26 2 / 2 points

Explain why Erase needs to be performed on block level in SSD.

Because when we out to erase a page(0 -> 1), we need to drive high negative voltage on the control gate to repel electrons out of the floating gate, this action will stress surrounding cells. So, it is dangerous to do on a single page, that's why erase needs to be performed on block level in SSD.

Question 27 2.5 / 3 points

Using Python's map, filter, and/or reduce functions, write a Python code to find even numbers in a given list (called "mylist") of integers and double their values. Note that odd numbers in the list will be discarded.

map(lambda x: x*2, filter(lambda x: x%2 = 0, mylist))

Question 28 4 / 4 points

Using Python's map, filter, and/or reduce functions, write a Python code to find the sum and count of values in a given list (called "mylist).

sum:
reduce(lambda x, y: x+y, mylist)

count:
reduce(lambda x, _: x+1, mylist, 0)

Done