Database Systems (DB) – Theory

Homework:01

CS363 Spring 2022 Homework 1

Date: 25th February, 2022

PROBLEM 01:

You are designing a system to crowdsource student evaluations of college courses (very much like Carta). As part of this system, you want to store a table of reviews containing the following information (for each review):

• Review Date - Date (3 bytes)

• Academic Year - int32

• Academic Quarter - char[10]

• Course ID - char[5]

• Rating (0.0 to 5.0) - float32

• Grade in the course - char[2]

• Estimated Hours Per Week - int32

• Review (text) - char[224]

1.1:

What is the size of each row in bytes?

Answer:

Size of each row=**256 bytes**

1.2:

Assume that this data is stored on a hard disk in disk blocks and the disk blocks are grouped in DB blocks. How many rows can be stored per disk block?

Answer:

Disk Block= 64KB= $64x10^3$ bytes Number

of rows= $\frac{64x10^3}{256}$ **=250 rows**

1.3:

How many rows can be stored per DB block?

Answer:

DB block= 64MB= $64x10^6$ Bytes

Rows per DB Block= $\frac{64x10}{256}$^6**=250,000 rows**

1.4:

After 10 years (40 quarters), how large in MB will the table of course reviews be? Round your answer to 1 decimal place.

Answer:

Since we have to calculate for 50% students so,

Number of students=7500

Total classes of 7500 students (3 classes each) = 7500*3=22500 classes

Total number of rows in 40 quarters= 22500*40= 9,00,000 rows

Total number of bytes= 900000*256=230400000 Bytes

Total size of table in MB's= **230.4 MBs**

1.5:

How many DB blocks would be needed to store the table of course reviews?

Answer:

Each DB block= 64MB

Total DB Blocks= $\frac{230.4}{64}$**=3.6 DB Blocks**

1.6:

How long would it take in hours to retrieve an evaluation (row) if the table rows are stored randomly on disk (we must seek and scan every row)? Round your answer to 1 decimal place.

Answer:

Seek time of disk=10ms

Transfer rate of disk= 100MB/s

Total time= seek time* number of rows+ scan time

= seek time* number of rows+ (size of table/transfer rate)

$=10*10^{-3}*900000+\frac{230.4}{100}$

=9000+2.304

=9002.304seconds

Time in hours= $\frac{9002.304}{3600}$ = **2.5hrs**

1.7:

How long would it take in seconds if the rows are grouped in disk blocks (which are randomly stored on disk)? Round your answer to 3 decimal places.

Answer: if rows are grouped into disk blocks here is what will happen. Number of disk blocks= $\frac{230}{64KB}.4MB = 3.6*10^3$ blocks= 3600 blocks

Total time= seek time* number of disk blocks+ scan time

= 10ms*3600+$\frac{230.4MB}{100MB/s}$=36+2.304

= 38.304s

PROBLEM 02:

You're the leader of the infrastructure team of a popular-enough startup. You're concerned with performance metrics for a particular table that is queried frequently. The table has the following specs,

• Row size: 64KB

• Number of rows: $5 * 107$

• Total Data: Number of rows * Row size = 3200GB = 3.2TB The table is stored on a system with the following specs,

• RAM: 64GB

• Hard Disk space: 10TB

The system receives numerous queries each second; each query consists of fetching some random row of the table. For the purpose of this problem, let's assume parsing and transferring queries take zero time. We'll also assume seeks in RAM take zero time. **Note**: Remember that the average time for finding a record during a full scan is half of the maximum time.

2.1:

What is the average response time in secs for a query, i.e., time to fetch a row? Assume that all rows are equally likely to be queried.

Answer:

Rows= $5*10^7$

Every row has size 64KB

Total table size= 3.2TB

Disk transfer speed=100MB/s

Since we are again using a disk so we will calculate time with seek and scan time.

Total time= seek time* number of rows+ scan time

$= 10*10^{-3} * 5*10^7 + 3.2TB$

$100MB/s$

= **532000s**

2.2:

Would you suggest any change to the current architecture given this information? State your suggestion concisely. What's the average response time in secs after your suggestion?

Answer:

Rather than using disk blocks we can move to DB blocks too but here it is not required so we can move to RAM here.

In case of RAM following time will be taken.

Rows= $5*10^7$

Every row has size 64KB

Total table size= 3.2TB

RAM transfer speed=100GB/s

Total time= 32s

PROBLEM 03:

Imagine you are designing a table to store recent transactions for an online shopping platform and there are 1 trillion transactions. You want to record the following information:

• user id

• user name

• item id

• item name

• transaction id

• amount of money ($) for the transaction (e.g. $4.11, $670.50, etc)

Assume there are 1 billion users, and 1 billion items for sale on the platform. The longest string for user and item names contain 64 characters. You should consider proper data types listed below: byte, short, int, long, float, double, Boolean, char.

3.1:

What is the size of each row in bytes? Think about the size of each column by selecting proper data types. You need to select the most suitable data type for each column by considering efficiency.

Answer:

• user id **---- int32 = 4 Bytes**

• user name ---- **char [64] = 64 Bytes**

• item id ---- **int 32 = 4 Bytes**

• item name ---- **char [64] = 64 Bytes**

• transaction id ---- **int32 = 4 Bytes**

• amount of money ($) for the transaction ---- **double = 4 Bytes**

Total number of bytes with assuming data types=**144bytes**

3.2:

What is the most appropriate data type for the following column: User ID?

Answer:

Int

3.3:

What is the most appropriate data type for the following column: User Name?

Answer:

Char

3.4:

What is the most appropriate data type for the following column: Item ID?

Answer:

Char

3.5:

What is the most appropriate data type for the following column: Item Name?

Answer:

Char

3.6:

What is the most appropriate data type for the following column: Transaction ID?

Answer:

Int

3.7:

What is the most appropriate data type for the following column: Amount of money?

Answer:

Double

3.8:

What is the size of the table in TB? (1 point) **Answer:**

size of each row = 144 Byte total transactions =

1 trillion = 10^{12} Transactions

Size of table in TB = $144*10^{12}$ = **144TB PROBLEM 04:**

This question follows from question 3. For this question, assume that the size of the table is 200 TB.

4.1:

How long in seconds will it take to read the whole table from RAM?

Answer:

Transfer speed of RAM **= 100GB/s**

Total time to read whole table= $\frac{200TB}{100GB/s}$ = **2000s**

4.2:

How long in days (round to nearest integer) will it take to read the whole table from disk if each row of the table is stored randomly in the disk?

Answer:

Here we will calculate time using seek time and scan time.

Total time= seek time* number of rows+ scan time

$= 10*10^{-3} * 10^{12} + \frac{200TB}{100MB/s}$

= 115764 days

4.3:

How long in days (round to nearest integer) will it take to read the whole table from disk if the table is stored in DB blocks? (1 point) **Answer:**

Each DB block= 64MB Number of DB Blocks=

$\frac{200TB}{200MB}$ = 3125000 blocks

Total time= seek time* number of rows+ scan time

$= 10*10^{-3} * 3125000 + \frac{200TB}{100MB/s}$

= 24 days

4.4:

What is the cost in dollars for saving the table in RAM? Assume RAM costs $6000/TB.

Answer:

it will cost $1200000 to store 200TB sized table in RAM.

4.5:

What is the cost in dollars for saving the table in disk? Assume disk space costs $100/TB.

Answer:

It will cost $20,000 to store 200TB sized table in disk.

PROBLEM 05:

You have decided to start a new e-commerce site that you anticipate will host billions of products and you hope millions of users. You realize you will need a database system to keep track of all your data.

5.1:

What tables might you need for this? For example, a table to log each order a user placed for a product might be a good idea. List at least two other tables that you want to include in your design. (There are many correct answers).

Answer:

We can have two tables:

1. users that subscribe/create account on our e-commerce site
2. products that will be posted on site to keep track of how much items are sold, left on discount etc.

5.2:

Let's calculate the size of one of our tables. Consider the above example of a table to log orders. We would like to keep track of the following:

• Order ID: int64

• Product ID: see part 4.2

• User ID: see part 4.3

• Quantity: int32

• Timestamp: 4 bytes

• IP address: 4 bytes

• Mailing address: char [100]

We want the ability to host 5 billion products. How many bits should the Product ID be to store a unique ID for each product?

Given our answer above, what data type should we use?

• tiny int (1 byte)

• small int (2 byte)

• int (int32 – 4 bytes) • big int (int64 – 8 bytes) **Answer:**

We can use **big int (int64 – 8 bytes)** because it can store integers from -9223372036854775808 to 9223372036854775807 that can easily cover 5 billion products.

5.3:

We want the ability to account for 1 billion users. How many bits should the User ID be to store a unique ID for each user?

Given our answer above, what data type should we use?

• tiny int (1 byte)

• small lint (2 byte)

• int (int32 – 4 bytes) • big int (int64 – 8 bytes) **Answer:**

We can use **int (int32 – 4 bytes)** because it can store integers from -2147483648 to 2147483647 that can easily cover 1 billion users.

5.4:

Given the above and your answers for 5.2 and 5.3, how big is one row of our table (one record) in bytes? **Answer:**

According to the above information our row will have **132 bytes.**

5.5:

How big is the entire table in MB if we assume that we store the data for a week, and we receive 100 million orders in a day?

Answer:

Since there are 100 million orders a day so in a week total 700 million orders will be received that means total 700 million rows and each row has 132 bytes that gives total 92400000000 bytes and here total MBs will be **92400MBs.**

5.6:

Let's calculate the time it takes to perform operations on our database. Let's assume that our table is actually 10 GB (no matter what you got for an answer in part 5.4). Use the values given in the instructions for all calculations.

How much time does it take in milliseconds to look up a record if our table is in RAM?

Answer:

If our table is of size 10GB and transfer rate of RAM is 100GB/s so for each row it can be

$\frac{10GB}{100GB/s}$ = 0.1s = **100ms**

5.7:

How much time does it take in days (round to nearest day) to look up a record on disk if all records are in random locations?

Answer:

Here data is stored on a disk so we have to calculate its seek and scan time.

Total time= seek time* number of rows+ scan time

$= 10*10^{-3} *10*10^{9} + \frac{10GB}{100MB/s}$

= 100000100s

= **1157 days**

5.8:

Now let's say you divide your data into blocks so that each block is 64 MB. The blocks are still scattered randomly on disk. How long would it take in seconds (round to nearest second) to look up a record then?

Answer:

On dividing data into blocks, we have 156.25 blocks of 64MB each. Total time will be calculated as:

Total time= seek time* number of blocks+ scan

time $= 10*10^{-3} *156.25+ \frac{10GB}{100MB/s}$

$= 101.562s$

= **101s**

5.9:

Now, let's think about scale. Use the values given in the instructions for all calculations. If we had 10 machines, how would this impact the speed of looking up one record? How many times faster would look up be?

Answer:

Since there are 10 machines this will eventually increase the efficiency and this will be increased by 10 times.

5.10:

What if the data was stored on another machine? How long would it in milliseconds take to get that data if that other machine had the data readily available in RAM (round to nearest millisecond)?

Answer:

Whenever data is stored on any other machine its retrieval is speedy than on the same machine. Since data is available on another machine so here we need access time of RAM that is 20ns and latency of the network that is 1us so time taken for data to be received from another machine is 1us*20ns=1.02us= 0.00102ms