There are 12 questions and ?? pages in this quiz booklet. To receive credit for a question, answer it according to the instructions given. *You can receive partial credit on questions.* You have 80 minutes to answer the questions.

Write your name on this cover sheet AND at the bottom of each page of this booklet. Some questions may be harder than others. Attack them in the order that allows you to make the most progress. If you find a question ambiguous, be sure to write down any assumptions you make. Be neat. If we can’t understand your answer, we can’t give you credit!

*THIS IS AN OPEN BOOK, OPEN NOTES QUIZ.*
*NO PHONES, NO LAPTOPS, NO PDAS, ETC.*
YOU MAY USE A CALCULATOR.

Do not write in the boxes below

<table>
<thead>
<tr>
<th>1-4 (xx/30)</th>
<th>5-7 (xx/26)</th>
<th>8-10 (xx/24)</th>
<th>11-12 (xx/20)</th>
<th>Total (xx/100)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name:
I Short Answer

1. [6 points]: Ben Bitdiddle is experimenting with shared nothing and shared disk parallel databases. Sometimes he finds that shared nothing systems perform better than shared disk systems, and sometimes he finds that shared disk systems perform better than shared nothing systems. Describe one setting, workload, or scenario in which you would expect a shared disk system to perform better than a shared nothing system.

   (Write your answer in the space below.)

2. [6 points]: Describe one setting, workload, or scenario in which you would expect a shared nothing system to perform better than a shared disk system.

   (Write your answer in the space below.)
3. [8 points]: In the paper “C-Store: A Column Oriented DBMS”, different compression schemes are proposed for different types of data. For each of the following columns indicate which compression scheme is likely to offer the best compression ratio? Choose from:

- Type 1: Run Length Encoding
- Type 2: Bitmap encoding
- Type 3: Delta encoding
- Type 4: Generic compression tool (e.g., gzip or zip) – assume that this option gets about 20% reduction in data size regardless of input data.

(Circle the best compression scheme in each case.)

A. An unsorted column of last names of students at MIT
   Type 1  Type 2  Type 3  Type 4

B. A sorted column of salaries of employees at MIT, assuming many employee salaries
   Type 1  Type 2  Type 3  Type 4

C. An unsorted column of genders of students at MIT (assume genders are represented as 1 character strings)
   Type 1  Type 2  Type 3  Type 4

D. A sorted column of birth-years (ages) of employees at MIT (MIT has thousands of employees)
   Type 1  Type 2  Type 3  Type 4
You’re running a PNUTS system (see the paper by Cooper et al.). Data items X and Y both start with value zero. Here are two functions that use the API described in Section 2.2 of the PNUTS paper:

\[
\text{fn1:} \\
\quad x_1 = \text{read-any}(X) \\
\quad x_1 = x_1 + 1 \\
\quad \text{write}(X, x_1) \quad // \quad X = x_1 \\
\quad \text{write}(Y, x_1) \quad // \quad Y = x_1
\]

\[
\text{fn2:} \\
\quad x_1 = \text{read-any}(X) \\
\quad x_2 = \text{read-latest}(X) \\
\quad y_1 = \text{read-any}(Y) \\
\quad \text{print } x_1, x_2, y_1
\]

You execute two calls to fn1, at different sites, at the same time. After both calls to fn1 have returned, you execute fn2 at a third site. There is no activity in the system other than described here, and no crashes or network failures.

4. [10 points]: What output is it possible to see from fn2, given the design of PNUTS and the above scenario? (Circle Yes for outputs that PNUTS could produce, and No for outputs that PNUTS could not produce.)

A. Yes / No 2, 2, 1
B. Yes / No 1, 2, 2
C. Yes / No 1, 1, 0
D. Yes / No 2, 1, 1
E. Yes / No 0, 0, 0

Name:
5. [10 points]: Which of the following are true of BigTable, as described in the paper by Chang et al.? (Circle True or False for each statement.)

A. True / False Adding a new column is expensive, since it requires moving existing columns to create space for the new column.

B. True / False BigTable is optimized for reads, and has comparatively slow writes.

C. True / False BigTable stores each tablet on the disks of two tablet servers, using primary/backup replication.

D. True / False If the BigTable master crashes, clients must wait for it to be restarted before they can do more work.

E. True / False BigTable stores tables in B+Trees on disk for fast primary-key lookups.

6. [8 points]: Which of the following statements about the Aurora system, as described in the paper by Abadi et al., are true? (Circle True or False for each statement.)

A. True / False The primary purpose of QoS curves is to figure out whether to shed load

B. True / False Input data streams are assumed to arrive in time order.

C. True / False The output of the BSORT operator is totally ordered according to the value of the attribute A in the order clause.

D. True / False Aurora uses an iterator-style model for passing tuples between operators.

Name:
II Parallel Databases

Dana Bass is trying to figure out how to best partition the tables in her 2-machine parallel database. The database she is using supports hash, range, and round-robin partitioning. When using hash and range partitioning she is required to specify the partitioning field, and when using range partitioning she must also specify the ranges of the partitioning attribute that are to be placed on each machine.

Her database consists of two large tables $A$ and $B$ of approximately the same size, each much larger than the sum of the two machines’ memories, with schemas $A(a, b)$ and $B(a, b)$. All fields are integers and are randomly and uniformly distributed. Each server stores a single heap file for each table in no particular order, and there are no indices.

7. [8 points]: For each of the following four workloads, specify which partitioning strategy from amongst the following will minimize the response time of the workload, assuming that CPU costs are negligible in comparison to network and disk I/O, that all disk I/O is sequential, and that disk bandwidth is comparable to network bandwidth.

Choose from these strategies:

a. Round robin partition both tables
b. Hash partition $A$ on $A.a$ and $B$ on $B.b$

For each workload, circle the best partitioning a, b, c, or d. If several partitionings are equivalent, circle one.

C. Many transactions issued serially, each of which runs a range query of the form $B.a > x$ ($x$ is a variable, chosen uniformly and at random from the range $[0 \ldots \max(B.a)]$).

D. Many transactions issued concurrently, each of which runs a query of the form $A.a = y$ ($y$ is a variable, chosen uniformly and at random from the range $[0 \ldots \max(A.a)]$).

Name:
Alyssa P. Hacker is building an instant messaging system called Chitter. In Chitter, messages can be addressed to one or more other users. Chitter is supposed to provide the following transactional guarantees about messages:

- **All or nothing**: When a message is sent, it is delivered to all of its recipients, or none of them; if one message cannot be delivered (because, for example, the machine that is supposed to store it is unavailable), none of the recipients should receive it.

- **In order**: If user U1 and U2 both receive messages M1 and M2, then if U1 sees M1 before M2, U2 should also see M1 before M2.

Alyssa implements Chitter as a three-node distributed database. Each user is given a separate table for his or her incoming messages, and each user’s messages table is stored on one of the nodes. When a message is sent, it is written into the messages table of each user who receives it.

8. **[8 points]**: Initially, to ensure that Chitter provides the all-or-nothing property, Alyssa uses a database system that uses two-phase commit (in the standard form, without presumed abort or presumed commit), and writes each message as a part of a single transaction. Even though her machines are on a fast LAN with network round trip times of only 100 ns, she finds that it takes about 20 ms to commit a transaction for a message, even when only one transaction is running at a time. Explain, in one sentence, what is likely going on.

(Write your answer in the space below.)
To improve the performance of Chitter, Ben Bitdiddle tells Alyssa that because messages are never deleted or updated, she can use a much simpler protocol than two-phase commit. Ben tells Alyssa about the Network Time Protocol (NTP), which can ensure that nodes in a cluster have clocks that are synchronized (i.e., agree with each other) to within 1 millisecond. He proposes Alyssa run NTP on her machines and then use the following protocol for sending a message:

- To send a message $m$, a client connects to one of the nodes $N$. The node assigns the message a 96-bit commit timestamp $TS_m$. To compute a timestamp, a node appends its unique 32-bit nodeid onto a 64-bit time in milliseconds since 1/1/2000 (the high order bits of the timestamp are the time); a node never issues the same timestamp for two messages.

- $N$ sends $m$ to each of the nodes that store the table for a user who should receive $m$, along with the value $TS_m$.

- When a node receives a message $m$ for a user $u$, it waits 2 milliseconds. If no messages arrive with a timestamp less than $TS_m$, the node “posts” $m$ by running a local transaction to write the message to $u$'s table. Otherwise, it posts the messages with timestamps less than $TS_m$ and then posts $m$.

9. [8 points]: Assuming that nodes can crash at any time, but that the network is never partitioned, delivers all messages, and never takes more than 100 ns to send messages between two nodes, does Ben’s protocol preserve the all-or-nothing property of Chitter? Why or why not?

   (Write your answer in the space below.)

The protocol above does preserve the in order property. However, if it is modified such that nodes only wait 0.5 ms instead of 2 ms, the in order property can be violated.

10. [8 points]: Given an example of a sequence of messages and timestamps that could result in a violation of the Chitter in order property when nodes only wait 0.5 ms.

   (Write your answer in the space below.)
IV Logging

You have a database with logging and recovery as described in the ARIES paper. Your database uses strict two-phase locking. Your database has just two items in it, X with starting value 10, and Y with starting value 100.

You start three transactions at the same time, with transaction IDs (TIDs) TA, TB, and TC:

**TA:**
```
BEGIN TRANSACTION
  X = X + 1
  Y = Y * 3
END
```

**TB:**
```
BEGIN TRANSACTION
  Y = Y * 2
  X = X + 5
END
```

**TC:**
```
BEGIN TRANSACTION
  X = X * 10
END
```

These three transactions are the only activity in the system. The system crashes due to a power failure soon after you start the transactions. You are not sure whether or not any of them completed. You look at the disk while the system is down and see that, in the heap file, Y has the value 200. You restart the system and let the database recovery procedure complete. You query the database for the value of X, and it returns the value 110.

Name:
11. [12 points]: Please write down a log, as it would have appeared on the disk while the system was down, that is compatible with the above story. You need only include Update (U), Commit (C), and Abort (A) records. Specify the TID (TA, TB, or TC) and record type for each record, and for Updates, the item being written (X or Y) and the new value being written.

(Write your answer in the empty log below.)

<table>
<thead>
<tr>
<th>TID</th>
<th>Type</th>
<th>Item</th>
<th>New Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12. [8 points]: ARIES stores both undo and redo information in log records. It is possible to change the design of ARIES to obtain a correct logging and recovery system that uses only redo information, so that an update record contains “after” value(s) but no “before” values (and no logical undo information). Describe:

A. What you would have to change about the rest of the system to get recoverability in such an ARIES-based “redo-only” system, and

B. The main reasons why the ARIES redo/undo approach is significantly better than this redo-only approach.

(Write your answer in the space below.)

End of Quiz II