# Quiz2 (6/9)

**Question 1**(1 point)

Which HTTP verb is used to create a resource in REST?

Question 1 options:

|  |  |
| --- | --- |
|  | POST |
|  | GET |
|  | PUT |
|  | HEAD |

**Question 2**(1 point)

Compared to the BitTorrent architecture, Chord makes it possible to download different pieces of a file from multiple nodes in parallel.

Question 2 options:

|  |  |
| --- | --- |
|  | True |
|  | False |

**Question 3**(1 point)

Module 2b presents an SOA example (see slide 5). The interaction between providers and databases is reminiscent of which architectural style?

Question 3 options:

|  |  |
| --- | --- |
|  | Intel Core microarchitecture |
|  | Event-driven architecture |
|  | Data-centered architecture |
|  | Object-based architecture |

**Question 4**(1 point)

Select the correct statement(s) regarding vertical and horizontal distribution.

Question 4 options:

|  |  |
| --- | --- |
|  | Vertical distribution places logically different components (i.e., belonging to different layers) on different hosts. |
|  | Vertical distribution relies on hashing to distribute load across components. |
|  | Both horizontal and vertical distribution can improve throughput by harnessing together the hardware resources of multiple machines. |
|  | Horizontal distribution places logically equivalent components (i.e., belonging to the same layer) on different hosts. |

**\*Question 5**(1 point)

In a service-oriented architecture (SOA), dividing the implementation of a single service into multiple services can hurt performance in which of the following ways?

Question 5 options:

|  |  |
| --- | --- |
|  | Higher context switching overhead due to additional inter-process communication. |
|  | Slower transaction processing due to a higher proportion of distributed transactions (i.e., ones that interact with multiple services) versus ordinary transactions (i.e., ones that interact with only one service) . |
|  | Higher serialization/deserialization overhead due additional inter-process communication. |
|  | Higher throughput due to additional vertical and horizontal distribution. |

**Question 6**(1 point)

Select the **correct**statement(s) about RESTful architectures.

Question 6 options:

|  |  |
| --- | --- |
|  | A RESTful service can store the state of a resource using a database.  For example, it can store profile photos for users of a social network. |
|  | According to the textbook, all services offer the same interface, consisting of at most four operations (GET, PUT, POST, DELETE). |
|  | RESTful services do not keep track of users and sessions explicitly. |
|  | REST messages specify the content encoding (e.g., text/html) explicitly. |

**Question 7**(1 point)

Select the correct statement(s) regarding services and architectures.

Question 7 options:

|  |  |
| --- | --- |
|  | In SOA, functionality is divided into a collection of services. |
|  | In a monolithic architecture, the services and the database are always combined. |
|  | Services that use Web protocols for transport are called Web services. |
|  | A service hides its implementation from clients. |

**Question 8**(1 point)

Select the correct statement(s) regarding distributed software architectures.

Question 8 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | A) | Peer-to-peer architectures rely primarily on vertical distribution. | |
|  | |  |  | | --- | --- | | B) | Component are modular units with well-defined interfaces. | |
|  | |  |  | | --- | --- | | C) | Connectors are mechanisms that mediate communication among components. | |
|  | |  |  | | --- | --- | | D) | In a layered architecture, each layer communicates with only one adjacent layer. | |

**Question 9**(1 point)

Select the advantage(s) of REST over SOAP.

Question 9 options:

|  |  |
| --- | --- |
|  | Superior compatibility with existing firewalls. |
|  | Multiple options for transport protocol. |
|  | REST allows different data encodings. |
|  | Responses of GET requests can be cached using existing HTTP mechanisms. |

# Quiz 3(5.5/9)

## ****Question 1**** (1 point)

Which socket option should be used to prevent disconnection due to network inactivity?

Question 1 options:

|  |  |
| --- | --- |
|  | SetReuseAddress |
|  | SetSoTimeout |
|  | SetTcpNoDelay |
|  | SetKeepAlive |

## ****Question 2**** (1 point)

Select method(s) by which inter-process communication can be achieved given that processes are executed on the same computer.

Question 2 options:

|  |  |
| --- | --- |
|  | Process stack |
|  | Named pipes |
|  | Socket |
|  | Thread-local variables |

## ****Question 3**** (1 point)

Select all of the benefits offered by VMMs.

Question 3 options:

|  |  |
| --- | --- |
|  | Insulates applications from the hardware, for example by allowing binary code compiled for the ARM architecture to run on an x86 host. |
|  | Reducing the operating cost of commodity servers |
|  | VM replication |
|  | Load balancing through live migration of VMs |

## ****\*Question 4**** (1 point)

Suppose that a client C on port 1024 connects to a server S on port 8080, and then closes its connection, which causes the client's socket to enter the TIME\_WAIT state.  Then while the client's socket is in TIME\_WAIT, no connection by any client can be established to S on port 8080.

Question 4 options:

|  |  |
| --- | --- |
|  | True |
|  | False |

## ****\*Question 5**** (1 point)

Select correct statement(s) regarding the Java Virtual Machine.

Question 5 options:

|  |  |
| --- | --- |
|  | It insulates Java applications from the underlying operating system. |
|  | It improves portability |
|  | It enables automatic load balancing |
|  | It allows hardware to be commoditized. |

## ****Question 6**** (1 point)

Select the correct statement(s) regarding processes.

Question 6 options:

|  |  |
| --- | --- |
|  | In a typical dispatcher/worker design, the dispatcher thread(s) feeds requests to a pool of worker threads. |
|  | IPC is costly in the sense that it requires context switching. |
|  | Threads in the same process can communicate directly using shared memory. |
|  | An LWP shares its address space with other LWPs within the same process. |

## ****\*Question 7**** (1 point)

The ServerSocket is a Java class implementing server sockets. When instantiating an object from this class, which one should be provided so that the ServerSocket can communicate with the clients?

Question 7 options:

|  |  |
| --- | --- |
|  | Client IP address only |
|  | Client port number only |
|  | Client IP address and port number |
|  | None of the above |

## ****Question 8**** (1 point)

In application-independent protocols, which of the following decides details like the message format, header, payload, etc?

Question 8 options:

|  |  |
| --- | --- |
|  | Operating System |
|  | Middleware |
|  | Application |
|  | Network Interface |

## ****Question 9**** (1 point)

Source and destination port numbers are included in which of the following?

Question 9 options:

|  |  |
| --- | --- |
|  | HTTP header |
|  | IP header |
|  | TCP header |
|  | UDP header |

# Quiz 4(5/8)

## ****\*Question 1**** (1 point)

An RPC server crashes.  A client then tries to execute an RPC and fails with an exception because the server is down.  What does the client's failure exemplify?

Question 1 options:

|  |  |
| --- | --- |
|  | Tight referential coupling |
|  | Tight temporal coupling |
|  | Loose referential coupling |
|  | Loose temporal coupling |

## ****\*Question 2**** (1 point)

Select correct statement(s) regarding referential and temporal coupling.

Question 2 options:

|  |  |
| --- | --- |
|  | Message queuing with multiple senders and multiple receivers is loosely coupled referentially |
|  | RPCs are tightly coupled both referentially and temporally |
|  | A pub/sub system is loosely coupled referentially |
|  | Message queuing with multiple senders and multiple receivers is tightly coupled temporally |

## ****Question 3**** (1 point)

Consider a message queueing system with the primitives introduced in the lecture module. Choose the primitive used to check whether a specified queue is empty or not.

Question 3 options:

|  |  |
| --- | --- |
|  | Put |
|  | Notify |
|  | Poll |
|  | Get |

## ****\*Question 4**** (1 point)

Select all of the responsibilities of the server stub.

Question 4 options:

|  |  |
| --- | --- |
|  | Marshalling the arguments of the RPC |
|  | Adding TCP/IP header to the messages that the server wants to send to the client |
|  | Serializing the result of invoking the procedure |
|  | Transforming requests coming in over the network into local procedure calls |

## ****\*Question 5**** (1 point)

Consider a typical system in which the operating system is responsible for passing messages between user-level processes and network interface card (NIC). In such a system, how many systems calls in total are required to fully execute one synchronous RPC?  Count system calls at both the client host and the server host.

Question 5 options:

|  |  |
| --- | --- |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |

## ****Question 6**** (1 point)

Select correct statement(s) regarding marshaling and unmarshalling.

Question 6 options:

|  |  |
| --- | --- |
|  | Marshaling handles differences in the representation of data (e.g., size and endianness of integers) between the caller and the callee |
|  | Marshaling is one of the basic functions that can be provided by communication middleware |
|  | Marshaling transforms data into a machine-independent format |
|  | Marshaling decides the order of parameters in a request message |

## ****Question 7**** (1 point)

The lecture module on Communication presents a layered network model that includes middleware at layer 5.  Which layers of the more common OSI model are replaced by the middleware layer in this lecture module?

Question 7 options:

|  |  |
| --- | --- |
|  | Session |
|  | Presentation |
|  | OS |
|  | Transport |

## ****Question 8**** (1 point)

Communication in an electronic mail system is

Question 8 options:

|  |  |
| --- | --- |
|  | One-way synchronous |
|  | Persistent |
|  | Transient |
|  | Tightly coupled temporally |

# \*Quiz 5

## ****Question 1**** (1 point)

Select the correct statement(s) regarding the remote access and upload/download models.

Question 1 options:

|  |  |
| --- | --- |
|  | In the remote access model, the client must interact with the server to read or write a file. |
|  | In the upload/download model, a client can access a file locally after downloading it. |
|  | In the upload/download model, clients can modify a file locally and return the revised file to the server upon closing it. |
|  | In the remote access model, an RPC interface can be used to interact with files. |

## ****Question 2**** (1 point)

Select correct statement(s) regarding updating files in GFS.

Question 2 options:

|  |  |
| --- | --- |
|  | Data transfer between different replicas is parallelized. |
|  | The master provides clients the address of replicas. |
|  | The primary replica confirms that all the replicas have received the new data by communicating with them. |
|  | The client first sends data to the primary replica. |

## ****Question 3**** (1 point)

Consider the following two statements:  
(i) An NFS server can also act as an NFS client, and mount directories exported by other NFS servers.  
(ii) An NFS server is able to export such remote directories, and serve files in these directories directly to clients.

Question 3 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (i) is correct. (ii) is incorrect. |
|  | (i) is incorrect. (ii) is correct. |

## ****Question 4**** (1 point)

Select the advantage(s) of striping a file across different servers.

Question 4 options:

|  |  |
| --- | --- |
|  | Striping a file across N servers decreases the probability of permanent data loss by a factor of N. |
|  | In some circumstances, it provides parallel access to chunks of data for faster reading. |
|  | Compared to a typical NFS deployment, in which only one NFS server shares files with clients, striping a file across multiple servers reduces the total number of messages the client must send to read the entire file. |
|  | It improves scalability of the distributed file system, for example by allowing the system to store larger files. |

## ****Question 5**** (1 point)

Select correct statement(s) regarding the design of GFS.

Question 5 options:

|  |  |
| --- | --- |
|  | HDFS is a simplified open-source implementation of GFS. |
|  | Chunks of data are stored in the local file systems of the chunk servers. |
|  | In GFS, a layer of software (as opposed to a RAID controller) stripes data across multiple disks. |
|  | GFS performs replication and striping. |

## ****Question 6**** (1 point)

Select correct statement(s) regarding the semantics of file sharing.

Question 6 options:

|  |  |
| --- | --- |
|  | UNIX semantics are expensive to implement in a distributed system. |
|  | In UNIX semantics, different clients may have different views of a file's contents.  For example, following a write by one client, another client may read stale data. |
|  | Locks in NFSv4 have an associated lease, ensuring that the lock eventually releases. |
|  | In session semantics, one client update never overrides another client’s update. |

## ****\*Question 7**** (1 point)

Select the disadvantage(s) of client-side file caching.

Question 7 options:

|  |  |
| --- | --- |
|  | Achieving UNIX semantics is impractical with client-side caching. |
|  | Client-side caching has no known disadvantages. |
|  | Mechanisms are required to handle inconsistency between different copies of a file at different clients. |
|  | When a client host crashes, the file system will be corrupted. |

# Quiz 6(7/7)

## ****Question 1**** (1 point)

Assume we have a layer of clients and a layer of servers and clients use synchronous RPCs. The requests are CPU-intensive and there is no hyperthreading. The server layer has 6 cores and it takes 100ms to process one request in one thread at the server. The round trip time between the client and server layer is 20ms. There are 6 threads and 3 cores allocated at the client layer. What is the best upper bound on throughput?

Question 1 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | 60 requests/second | |
|  | |  |  | | --- | --- | | b) | 50 requests/second | |
|  | |  |  | | --- | --- | | c) | 43 requests/second | |
|  | |  |  | | --- | --- | | d) | 25 requests/second | |

## ****Question 2**** (1 point)

Which of the following Thrift Java server implementations execute requests serially?

Question 2 options:

|  |  |
| --- | --- |
|  | TThreadPoolServer |
|  | THsHaServer |
|  | TNonblockingServer |
|  | TSimpleServer |

## ****Question 3**** (1 point)

When using asynchronous Thrift RPC clients, which of the following Java synchronization primitives can we use to synchronize the callback methods and the code that makes the async RPC call?

Question 3 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | Condition | |
|  | |  |  | | --- | --- | | b) | CountDownLatch | |
|  | |  |  | | --- | --- | | c) | Both Condition and CountDownLatch | |
|  | |  |  | | --- | --- | | d) | None of the above | |

## ****Question 4**** (1 point)

Assume we have a layer of clients and a layer of servers and clients use synchronous RPCs. The requests are CPU-intensive and there is no hyperthreading. The server layer has 3 cores and it takes 100ms to process one request in one thread at the server. The round trip time between the client and server layer is 50ms. There are 6 threads and 3 cores allocated at the client layer. Is the throughput of the system likely to be server-limited or client-limited?

Question 4 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | server-limited | |
|  | |  |  | | --- | --- | | b) | client-limited | |

## ****Question 5**** (1 point)

Which of the following Thrift Java server implementations is the best option in terms of performance when there is a fixed number of clients (e.g., 64) that issue large requests over a small number of long-lived connections (e.g., ones managed using a connection pool)?

Question 5 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | TSimpleServer | |
|  | |  |  | | --- | --- | | b) | TNonblockingServer | |
|  | |  |  | | --- | --- | | c) | THsHaServer | |
|  | |  |  | | --- | --- | | d) | TThreadedSelectorServer | |

## ****Question 6**** (1 point)

Which of the following are good practices when programming using Thrift?

Question 6 options:

|  |  |
| --- | --- |
|  | In a multi-threaded client, carefully share Thrift transports, protocols, and client stubs, because they are not thread-safe. |
|  | In ecelinux environment, limit the maximum size of the worker thread pool when using TThreadPoolServer. |
|  | Reuse TSocket, TServerSocket, and their non-blocking counterparts when possible. |
|  | Do not hard-code the hostname and port number for simplicity. |

## ****\*Question 7**** (1 point)

Which of the following statements are correct?

Question 7 options:

|  |  |
| --- | --- |
|  | The set of data types supported by Apache Thrift includes int and long. |
|  | Apache Thrift is a software framework for scalable cross-language services development. |
|  | Apache Thrift enables distribution transparency by allowing clients to discover services automatically. |
|  | Apache Thrift specifies an interface definition language (IDL). |

# Quiz 7(2.6/6)

## ****Question 1**** (1 point)

Which of the following phases must run in a MapReduce job?

Question 1 options:

|  |  |
| --- | --- |
|  | Mapper |
|  | Reducer |
|  | Combiner |
|  | None of the above |

## ****Question 2**** (1 point)

Given input records R1 = "big frog" and R2 = "little frog", what is the output after the Map phase when using the Stripes method to calculate the cross-correlation of R1 and R2?  Assume that R1 and R2 are in separate input splits, and are handled by distinct map tasks on distinct servers.

Question 2 options:

|  |  |
| --- | --- |
|  | <[big], {frog:1}> , <[little], {frog:1}> |
|  | <[big, frog],1> , <[frog, little],1> |
|  | <[frog], [{big:1}, {little:1}]> |
|  | <[big], {frog:1}> , <[frog], {little:1}> |

## ****Question 3**** (1 point)

The Hadoop Distributed Cache can be used in which phase?

Question 3 options:

|  |  |
| --- | --- |
|  | Mapper |
|  | Reducer |
|  | Both of the above |
|  | None of the above |

## ****Question 4**** (1 point)

In the Projection example in module 7b without duplicate elimination, let N be the number of input tuples and let M be the number of output tuples. What can we say about N and M?

Question 4 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | N >= M | |
|  | |  |  | | --- | --- | | b) | N = M | |
|  | |  |  | | --- | --- | | c) | N > M | |
|  | |  |  | | --- | --- | | d) | N <= M | |

## ****\*Question 5**** (1 point)

Which of the following statements is true about the Reducer and Combiner in MapReduce?

Question 5 options:

|  |  |
| --- | --- |
|  | Combiners must only be used with functions that are both commutative and associative. |
|  | Each combiner performs local aggregation of the intermediate outputs with the same key to help cut down the amount of data transferred from the Mapper to the Reducer. |
|  | Both Reducers and Combiners get the input values from multiple Mappers.  (Assume one map task per node.) |
|  | They are always identical. |
|  | When you add a Combiner class to your code, it is guaranteed to reduce the number of bytes shuffled across the network. |

## ****\*Question 6**** (1 point)

Which of the following statements is true about the Pairs and Stripes approaches in the cross-correlation problem?

Question 6 options:

|  |  |
| --- | --- |
|  | The Stripes approach is less complex and always faster than Pairs. |
|  | Combiners can be used more effectively in the Stripes approach because the keys output by mappers are shorter. |
|  | The Stripes approach computes an associative array for each key. |
|  | The Pairs approach uses a nested for loop to output every possible tuple. |

# Quiz 8

## ****Question 1**** (1 point)

Which of the following are the examples of transformations with narrow dependencies?

Question 1 options:

|  |  |
| --- | --- |
|  | union |
|  | groupByKey |
|  | filter |
|  | map |
|  | join operations whose inputs are co-partitioned |

## ****Question 2**** (1 point)

Consider the PageRank computation. Let the damping factor be 0.5, the total number of documents N = 10, the initial rank for each document be 0.1. Let contribs denote an RDD that stores PageRank contributions of the form (targetURL, float). What is the correct way to compute the new ranks for each URL?

Question 2 options:

|  |  |
| --- | --- |
|  | rank = contribs.reduceByKey(\_+\_).mapValues(sum => 0.05 + 0.5\*sum) |
|  | rank = contribs.reduceByKey(\_+\_).mapValues(sum => 0.5 + 0.5\*sum) |
|  | rank = contribs.mapValues(sum => 0.05 + 0.5\*sum) |
|  | rank = contribs.mapValues(sum => 0.5 + 0.5\*sum) |

## ****Question 3**** (1 point)

Which of the following is a transformation in Spark?

Question 3 options:

|  |  |
| --- | --- |
|  | Union |
|  | CountByValue |
|  | Reduce |
|  | Filter |
|  | Map |

## ****Question 4**** (1 point)

Which of the following is true for RDDs in Spark?

Question 4 options:

|  |  |
| --- | --- |
|  | RDDs have a schema with named attributes, similarly to relations in a database. |
|  | The lineage graph in Spark is a Directed Acyclic Graph (DAG) of RDDs and transformations. It is used to both optimize execution and enable recovery from failures. |
|  | RDDs are fault-tolerant and immutable. |
|  | RDDs are partitioned collections of elements that can be operated on in parallel. |

## ****Question 5**** (1 point)

Given the following PageRank example. Let the input graph be url1—> url4, url2 —> url3, url2 —> url1, url3 —> url1, url4 —> url2, url4 —> url3. The initial rank for each URL is 1 and the damping factor is 0.50. What are the ranks of the R=URLs after 1 iteration?

Question 5 options:

|  |  |
| --- | --- |
|  | url1 = 1.25, url2 = 0.75, url3 = 1.0, url4 = 1.0 |
|  | url1 = 1.1875, url2 = 0.75, url3 = 0.9375, url4 = 1.125 |
|  | url1 = 1.244, url2 = 0.575, url3 = 0.819, url4 = 1.361 |
|  | url1 = 1.425, url2 = 0.575, url3 = 1.0, url4 = 1.0 |

## ****Question 6**** (1 point)

Which of the following is true when you call the function persist() on a RDD in Spark?

Question 6 options:

|  |  |
| --- | --- |
|  | Calling persist() guarantees to improve the execution speed of all actions performed on an RDD. |
|  | Calling persist() gives a hint to the Spark framework that partitions of an RDD should be retained, preferably in main memory, so that they can be reused efficiently later on. |
|  | You must call persist() to prevent permanent loss of data if there is a crash failure. |
|  | RDD partitions can be persisted at multiple levels of the memory hierarchy. |

# Quiz 9(3/6)

## ****\*Question 1**** (1 point)

Why are scalable and fault-tolerant systems needed for graph processing?

Question 1 options:

|  |  |
| --- | --- |
|  | Many important data sets are very large and look like graphs.  Specialized tools can be better for such data sets than traditional solutions like relational databases. |
|  | It is difficult to parallelize certain graph algorithms across multiple computers.  Graph processing frameworks like Pregel provide a programming model that simplifies the implementation of such algorithms, which reduces developer effort. |
|  | Relational databases cannot be used to process graph data. |
|  | A distributed graph requires real-time computation. |

## ****Question 2**** (1 point)

In Pregel, how are the inputs assigned to each worker in the initialization phase?

Question 2 options:

|  |  |
| --- | --- |
|  | The master first finds all graph components or dense subgraphs and then decides the optimal placement of vertices for each worker. |
|  | A hash function is used to distribute vertices across workers. |
|  | Inputs are assigned in a first-come-first-serve manner. |
|  | Each worker decides its partition to maximize locality. |

## ****\*Question 3**** (1 point)

In the SSSP example given in module 8b, what does the Combiner return?

Question 3 options:

|  |  |
| --- | --- |
|  | For each strongly connected component C, it returns the minimum distance value (mindist) sent to a vertex in C |
|  | For each vertex v, it returns the minimum distance value (mindist) sent by v |
|  | For each strongly connected component C, it returns the minimum distance value (mindist) sent by a vertex in C |
|  | For each vertex v, it returns the minimum distance value (mindist) sent to v |

## ****Question 4**** (1 point)

In the Bulk Synchronous Parallel (BSP) model of computation, how is a superstep defined?

Question 4 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | Receive messages (except in the initial superstep), then send messages, then compute. | |
|  | |  |  | | --- | --- | | b) | Send messages, then receive messages, then compute. | |
|  | |  |  | | --- | --- | | c) | Compute, then receive messages (except in the initial superstep), then send messages. | |
|  | |  |  | | --- | --- | | d) | Receive messages (except in the initial superstep), then compute, then send messages. | |

## ****Question 5**** (1 point)

In slide 10 in lecture module 8b, if there is no edge between the vertex with initial label 6 and the vertex with initial label 2, in which superstep do all the vertices vote to halt?

Question 5 options:

|  |  |
| --- | --- |
|  | Superstep 2 |
|  | Superstep 3 |
|  | Superstep 4 |
|  | Superstep 5 |

## ****Question 6**** (1 point)

What is the primary reason why Hadoop MapReduce not a good fit for large-scale graph processing?

Question 6 options:

|  |  |
| --- | --- |
|  | It is not sufficiently fault-tolerant |
|  | It does not have enough functionalities to solve graph problems |
|  | It requires too much I/O between stages |
|  | Stragglers can create load imbalances |

# Quiz 10(7/10)

## ****Question 1**** (1 point)

Consider the following execution. This execution is

表格

描述已自动生成

Question 1 options:

|  |  |
| --- | --- |
|  | Both causally and sequentially consistent. |
|  | Causally consistent, not sequentially consistent. |
|  | Not causally consistent, sequentially consistent. |
|  | Not causally consistent, not sequentially consistent. |

## ****Question 2**** (1 point)

Consider the following two statements:  
(i) If each read quorum overlaps in at least one replica with each write quorum, then read-write conflicts are detectable.  
(ii) If each write quorum overlaps in at least one replica with each write quorum, then write-write conflicts are detectable.

Question 2 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (i) is correct. (ii) is incorrect. |
|  | (i) is incorrect. (ii) is correct. |

## ****Question 3**** (1 point)

Select the correct statement(s) regarding the anti-entropy mechanism discussed in lecture,

Question 3 options:

|  |  |
| --- | --- |
|  | To detect the latest update, updates can be timestamped. |
|  | Different replicas periodically exchange hash trees because exchanging the entire dataset is expensive in terms of network communication. |
|  | The discrepancy between different replicas is detected using Merkle trees. |
|  | In the absence of collisions in the hash function, if the roots of two replicas’ hash trees are identical then the data sets stored by those two replicas are identical as well. (Note that collision happens when two distinct inputs have the same hash value.) |

## ****Question 4**** (1 point)

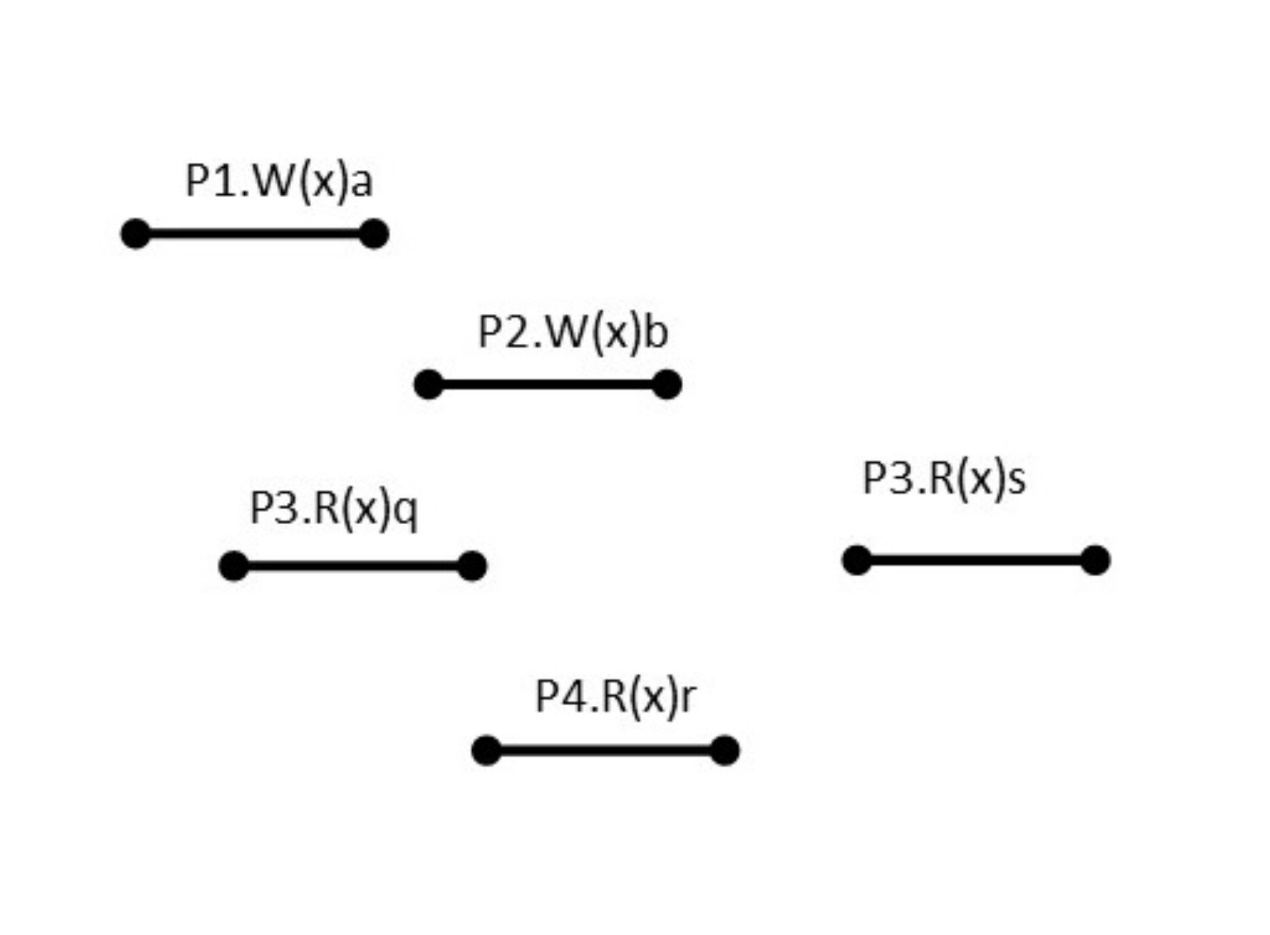
Select correct statement(s) regarding eventual consistency.

Question 4 options:

|  |  |
| --- | --- |
|  | Eventual consistency refers to a consistency scheme in which if no update happens for a long time; all replicas will gradually become consistent. |
|  | Eventual consistency refers to a consistency scheme in which if clients do not write anything, all servers will eventually hold the same data. |
|  | Session guarantees restrict the behavior of operations applied by a single process in a single session. |
|  | If a data store is not sequentially consistent, then it is not eventually consistent either. |

## ****Question 5**** (1 point)

Select all possible values for q, r, s so that the following execution stays linearizable.



Question 5 options:

|  |  |
| --- | --- |
|  | q=a, r=b, s=b |
|  | q=b, r=b, s=b |
|  | q=a, r=a, s=a |
|  | q=a, r=b, s=a |
|  | q=a, r=a, s=b |
|  | q=b, r=b, s=a |

## ****Question 6**** (1 point)

Select correct statement(s) regarding primary-based replication protocols.

Question 6 options:

|  |  |
| --- | --- |
|  | The split-brain situation is a situation in which all parts of the system agree about the primary replica server. |
|  | In the remote-write primary-based replication protocol, writes are always executed on the local replica. |
|  | In a primary based replication protocol, updates are executed by a designated primary replica and pushed to other replicas. |
|  | In the local-write primary-based replication protocol, updates are always executed on the remote replica. |

## ****Question 7**** (1 point)

Select correct statement(s) regarding full and partial replication.

Question 7 options:

|  |  |
| --- | --- |
|  | In full replication, the effective capacity of the system grows with the number of servers. |
|  | In partial replication, when the number of servers exceeds the replication factor, each server stores only a subset of data objects. |
|  | In full replication, each server keeps a copy of each data object. |
|  | In a partial replication scheme, when the number of servers exceeds the replication factor, the system has more effective storage than a full replication scheme. |

## ****Question 8**** (1 point)

Consider the following execution.

表格

描述已自动生成

Which of the following read responses are possible in process P3 in order to preserve causal consistency?

Question 8 options:

|  |  |
| --- | --- |
|  | t=d, r=c |
|  | t=c, r=b |
|  | t=b, r=c |
|  | t=b, r=d |

## ****Question 9**** (1 point)

Question 9 options:

文本

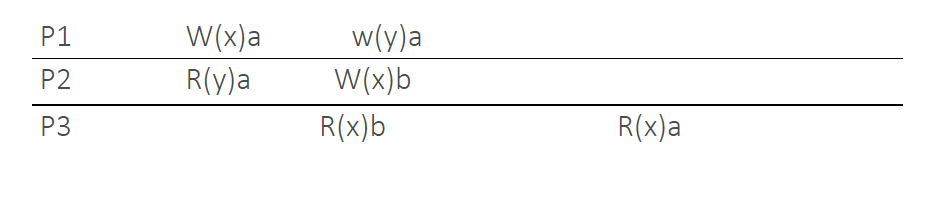
描述已自动生成 

What is the maximum number of replicas **N**  if providing a partial quorum with strong consistency is the goal?  Enter a number between 1 and 100 in the box below.



## ****Question 10**** (1 point)

Consider the following execution. This execution is

 Question 10 options:

|  |  |
| --- | --- |
|  | Both causally and sequentially consistent. |
|  | Causally consistent, not sequentially consistent. |
|  | Not causally consistent, sequentially consistent. |
|  | Not causally consistent, not sequentially consistent. |

# Quiz 11(3.5/6)

## ****Question 1**** (1 point)

Select correct statement(s) regarding strategies for dealing with RPC server crashes.

Question 1 options:

|  |  |
| --- | --- |
|  | Implementing exactly once semantics is difficult because there is no clear way of determining whether action was interrupted by a failure. |
|  | When the request is not idempotent, reissuing the request at the client can be unsafe. |
|  | At-most-once semantics guarantee the execution of each request.no |
|  | Achieving exactly once semantics is not always possible. |

## ****\*Question 2**** (1 point)

Select correct statement(s) regarding Triple Modular Redundancy, TMR.

Question 2 options:

|  |  |
| --- | --- |
|  | In TMR, a hardware component is replicated three ways. |
|  | TMR can mask the failure of one of the voters that are voting over copies of a component. |
|  | In TMR, voters are replicated three ways. |
|  | TMR can mask the failure of up to two copies of a component. |

## ****Question 3**** (1 point)

Consider the following two statements:  
(I) Omission failure can arise from a failure in the network or server.  
(II) In a Byzantine failure, a server may produce arbitrary responses at arbitrary times

Question 3 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****Question 4**** (1 point)

Select correct statement(s).

Question 4 options:

|  |  |
| --- | --- |
|  | If a system is highly reliable, then it is highly available as well. |
|  | Reliability can be measured using the MTBF metric. |
|  | If a system is highly reliable, the probability that the system performs correctly at any given time is always high. |
|  | If a system is highly available, it is highly unlikely to be interrupted during a relatively long time. |

## ****\*Question 5**** (1 point)

Consider an RPC service that prints a message to the screen upon receiving a request from the client. M indicates replying to the request (i.e., sending an ack), P indicates printing to the screen, and C indicates a server crash.  Select the correct statement(s) regarding this scenario.

Question 5 options:

|  |  |
| --- | --- |
|  | In MPC, MC(P), and C(MP), the client receives the acknowledgment. |
|  | When the server prints before replying to the request (i.e., P before M strategy), if the client does not receive the acknowledgment, it can tell whether PC(M) happened or C(PM). |
|  | When the client always reissues the request, if a crash happens after the server prints to the screen, the text is always printed more than once. |
|  | When the server prints after replying to the request (i.e., M before P strategy), if the client receives the acknowledgment, it cannot tell whether MPC happened or MC(P). |

## ****Question 6**** (1 point)

Select the correct statement(s) regarding RPC server crashes.

Question 6 options:

|  |  |
| --- | --- |
|  | The client cannot determine whether the server crashed before or after executing the request from the absence or presence of reply. |
|  | This course focuses mainly on Byzantine failures. |
|  | One of the examples of an RPC server crash is when the reply message from the server to the client is lost after it is sent by the server. |
|  | The server may crash before or after executing the client request. |

# Quiz 12(5/5)

## ****Question 1**** (1 point)

In the leader election example in module 8b, when do processes start the leader election process?

Question 1 options:

|  |  |
| --- | --- |
|  | When the leader dies, the znode on the path “.../workers/leader” is not deleted. The event triggers the processes that watch the znode to start the leader election process. |
|  | When processes start the leader election process, each process takes turns to update the existing znode on the path “.../workers/leader” with its hostname. The followers get the latest hostname from the znode to know the address of the leader. |
|  | When the leader dies, its ephemeral znode on the path “.../workers/leader” is deleted. That triggers a watch and the processes that watch the znode will start the leader election process. |
|  | When processes start the leader election process, each process votes for a leader.  A majority vote wins.  The leader creates a znode on the path “.../workers/leader” that contains its hostname. |

## ****Question 2**** (1 point)

Select the correct statement(s) regarding ZooKeeper.

Question 2 options:

|  |  |
| --- | --- |
|  | In ZooKeeper, data elements (i.e., the data payload of a znode) must be read and written in their entirety. |
|  | Watches provide the notification mechanism in ZooKeeper. |
|  | ZooKeeper does not allow conditional writes. 不确定对不对 |
|  | ZooKeeper is designed to store large objects. |

## ****Question 3**** (1 point)

In ZooKeeper, why are read operations allowed to return stale data?

Question 3 options:

|  |  |
| --- | --- |
|  | To improve the read throughput by sacrificing the real-time consistency of read operations. |
|  | Because doing so is necessary to circumvent the FLP impossibility result. |
|  | To make sure that clients always follow the FIFO ordering |
|  | To speed up failure detection via ephemeral znodes. |

## ****Question 4**** (1 point)

Select the correct statement(s) regarding ZooKeeper znodes.

Question 4 options:

|  |  |
| --- | --- |
|  | Ephemeral znodes are automatically deleted when the creator fails or explicitly deleted. |
|  | The numerical tags in a sequential znode can be reused, but it is not recommended. |
|  | Sequential znodes can be ephemeral. |
|  | Each znode is identified by a path, following a hierarchical namespace. |

## ****Question 5**** (1 point)

Workers should not use ephemeral znodes in which of the following applications?

Question 5 options:

|  |  |
| --- | --- |
|  | Group membership |
|  | Leader election |
|  | Dynamic configuration |
|  | Locks |

# Quiz 13(4/9)

## ****Question 1**** (1 point)

Raft ensures its **safety** properties despite which type(s) of failure?

Question 1 options:

|  |  |
| --- | --- |
|  | Crash of any number of servers |
|  | Byzantine failures |
|  | Delayed messages |
|  | Crash of a minority of servers |

## ****Question 2**** (1 point)

Select correct statement(s) regarding Raft normal operation.

Question 2 options:

|  |  |
| --- | --- |
|  | After committing a new entry to a quorum of replicas, the leader notifies followers of the committed entry in subsequent remote procedure calls. |
|  | In normal operation, there is exactly one leader, and all of the other servers are followers. Or no |
|  | A command is executed only after at least a quorum of servers reply to the leader’s AppendEntitries RPCs for this command. |
|  | Followers are passive, meaning that they issue no requests on their own but simply respond to requests from leaders and candidates. |

## ****Question 3**** (1 point)

Select correct statement(s) regarding the Log Matching Property.

Question 3 options:

|  |  |
| --- | --- |
|  | The log matching property implies that if a given entry is replicated at a majority of servers, then all the preceding entries are committed. |
|  | The empty log is the induction base for proving the log matching property. |
|  | The consistency check, which is the step of induction for proving log matching property, is done by including <index, term> of the entry preceding the new one(s) in AppendEntries RPC. |
|  | The log matching property implies that if two logs on different servers contain an entry with the same index and term, then the logs are identical in all preceding entries. |

## ****Question 4**** (1 point)

Select correct statement(s) regarding Raft decomposition.

Question 4 options:

|  |  |
| --- | --- |
|  | Leader election, log replication, and safety are three completely disjoint dimensions of Raft. |
|  | The leader accepts commands from clients, appends them to its log, and replicates its log to other servers. |
|  | Leader election is a procedure in which a server is selected to act as a leader. |
|  | The server with the most outdated log cannot become a leader. |

## ****Question 5**** (1 point)

Consider the following statements:

(I) The Raft leader election algorithm guarantees liveness deterministically.

(II) The Raft leader election algorithm ensures that at most one server wins the election (i.e., becomes a leader) in each term.

Question 5 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****Question 6**** (1 point)

Consider the following two statements:

(I) To keep the protocol understandable and straightforward, the Raft leader assumes that its log is correct.

(II) The Raft leader election algorithm ensures that the chosen leader has the best (i.e., most up to date) log among all replicas.

Question 6 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

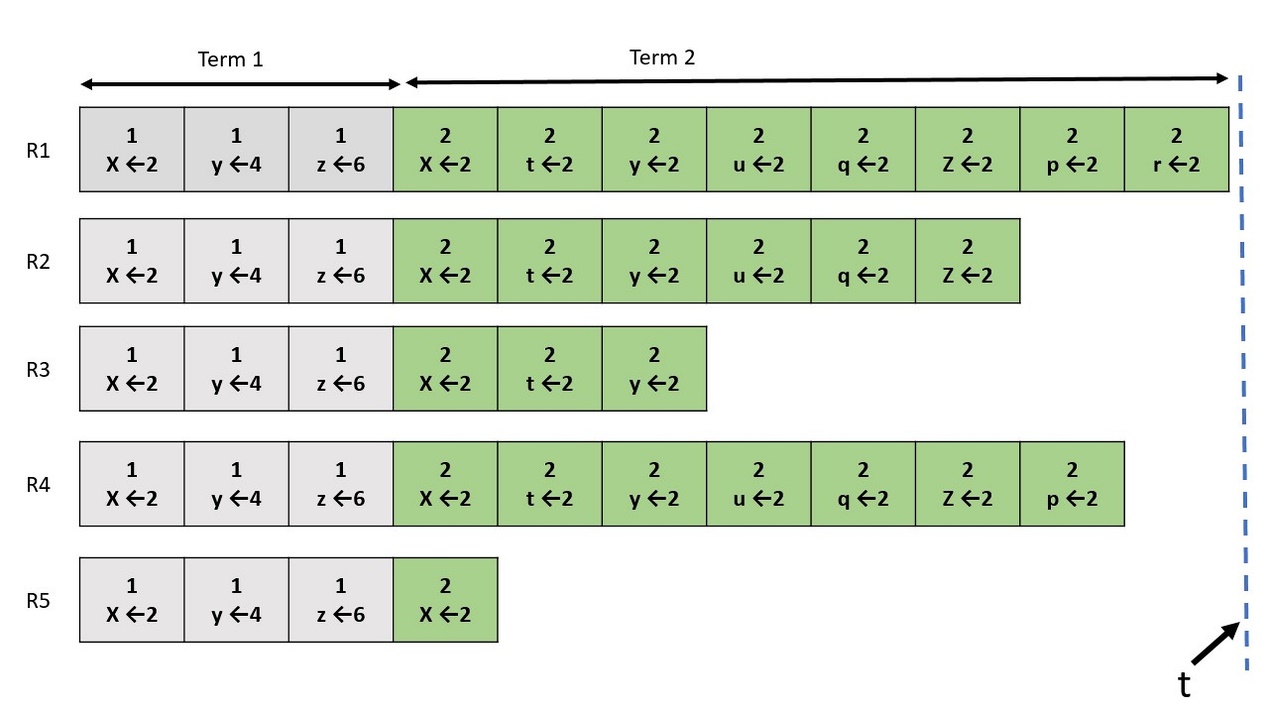
## ****Question 7**** (1 point)

Select correct statement(s) regarding Raft leader election.

Question 7 options:

|  |  |
| --- | --- |
|  | If a leader discovers another leader with a higher term, it crashes. |
|  | There cannot be more than one leader with the same term number. |
|  | A new round of leader election is started eventually if the previous leader crashes. |
|  | A candidate issues RequestVote RPCs to solicit votes in a leader election. |

## ****\*Question 8**** (1 point)

Consider the following log in a system implementing the Raft consensus algorithm.  We know that at time t, the leader crashes.  Which replica could be the leader at term two until t?  Which replica can be the next leader?

Question 8 options:

|  |  |
| --- | --- |
|  | Current leader: R5, Next leader: R4 |
|  | Current leader: R1, Next leader: R5 |
|  | Current leader: R1, Next leader: R2 |
|  | Current leader: R5, Next leader: R3 |

## ****Question 9**** (1 point)

Consider the following statements:

(I) The replicated log allows state machine replicas to execute the same commands in the same order.

(II) The consensus module in Raft must be able to reach multiple decisions rather than a single decision because there are multiple entries in the replicated log. Single mechanism

Question 9 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

# Quiz 14(3.5/5)

## ****Question 1**** (1 point)

Select correct statement(s) regarding windowed streams.

Question 1 options:

|  |  |
| --- | --- |
|  | In session windows, if the period of activity exceeds a given threshold, a new session will be created. |
|  | Hopping windows may overlap with each other, but there cannot be a gap between them. |
|  | Tumbling time windows are a special case of hopping time windows. |
|  | In tumbling time windows, the window size is greater than the hop.不一定 |

## ****Question 2**** (1 point)

Select correct statement(s) regarding Kafka producers.

Question 2 options:

|  |  |
| --- | --- |
|  | When a producer sets the "acks" property to “all,” only one replica must acknowledge a write for the write to be considered successful. |
|  | The topic name to which the record is being sent should always be provided when creating an instance of ProducerRecord object.不一定 |
|  | In a Kafka producer, both key/value serializers and deserializers must be specified. Consumer |
|  | A producer must know the address—the hostname and port number—of at least one broker. |

## ****Question 3**** (1 point)

Consider the following two statements:

(I) A record in the record stream semantic represents a concrete state.

(II) A record in the changelog stream semantic represents a state transition.

Question 3 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect.反了 |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****Question 4**** (1 point)

Select correct statement(s) regarding topics in Apache Kafka.

Question 4 options:

|  |  |
| --- | --- |
|  | Topics are stored as partitioned logs. |
|  | Topics all have the same size. |
|  | A record remains in a topic for a configurable period of time. |
|  | In Apache Kafka, consumers can maintain pointers to the next record to be read.  (Assume one partition per topic for simplicity.) offset |

## ****\*Question 5**** (1 point)

Select correct statement(s) regarding streaming APIs in Apache Kafka.

Question 5 options:

|  |  |
| --- | --- |
|  | The hostname and port number of at least one Kafka broker should be provided to the KafkaStreams. |
|  | In a KafkaStreams, the key/value serializers rather than deserializers should be specified. |
|  | The Application ID should be specified when using KafkaStreams. 不必 |
|  | When the KafkaStreams “start” method is invoked, the stream application starts executing, and it does not stop until the “close” method is invoked or the Java process is shut down (e.g., by a signal). |

# Quiz 15(6/12)

## ****Question 1**** (1 point)

If the clock skew is positive, what can we conclude about the clock under consideration?

Question 1 options:

|  |  |
| --- | --- |
|  | The clock is faster than the reference clock. |
|  | The clock is slower than the reference clock. |
|  | The clock is a perfect clock. |
|  | The given information is not enough to determine the speed of the clock compared to the reference clock. |

## ****Question 2**** (1 point)

Suppose that the actual clock offset between two computers is equal to 32ms. If NTP estimates δ = 24ms and θ = 36ms, select all possible values for the one-way network delay of request and response messages.

Question 2 options:

|  |  |
| --- | --- |
|  | dTreq = 26ms, dTres = 22ms |
|  | dTreq = 21ms, dTres = 27ms |
|  | dTreq = 28ms, dTres = 20ms |
|  | dTreq = 30ms, dTres = 18ms |

## ****Question 3**** (1 point)

Consider the following vector clocks representing event1 and event2:

VC1=[2, 4, 6, 1, 0, 3, 5] and VC2=[2, 3, 6, 0, 1, 3, 4]. Select correct statement.

Question 3 options:

|  |  |
| --- | --- |
|  | event1 happens before event2. |
|  | event2 happens before event1. |
|  | event2 and event1 are concurrent. |
|  | The given information is not enough to determine the order of events |

## ****\*Question 4**** (1 point)

Select the correct statement(s) regarding clock synchronization.

Question 4 options:

|  |  |
| --- | --- |
|  | When clock skew is not equal to zero, clocks diverge over time. |
|  | The maximum drift rate determines the absolute value of the maximum possible clock offset. skew |
|  | Clock offset determines the absolute time difference between two clocks. |
|  | A perfect clock (i.e., clock skew = 0) always has the same value as the reference clock. |

## ****Question 5**** (1 point)

Question 5 options:

The following three measurements have been obtained using NTP.  
▪ (T1, T2, T3, T4) = (1,10,15,25)  
▪ (T1, T2, T3, T4) = (10,17,23,32)  
▪ (T1, T2, T3, T4) = (8,19,20,29)  
What is the best estimate for the value of the offset of the NTP server relative to the NTP client?



## ****\*Question 6**** (1 point)

Select the correct statement(s) regarding Lamport’s happens-before relation.

Question 6 options:

|  |  |
| --- | --- |
|  | If neither a→b nor b→ a, then a and b are concurrent. |
|  | In one process, if a occurs before b in program order, then a→ b is true. |
|  | If a and b happen in two different processes that do not send or receive any message (to each other or to any other processes), then neither a→b nor b→a are true. |
|  | If a→b and b→c, then a→c. can’t imply |

## ****\*Question 7**** (1 point)

Select the correct statement(s) regarding vector clocks.

Question 7 options:

|  |  |
| --- | --- |
|  | VCi [ i ] is represents the logical time at process Pi. |
|  | Processes learn the vector clock by communicating directly with other processes, or indirectly via third parties. |
|  | When Pi receives a message from Pj, it sets VCi[ k ] = VCj [ k ] for each k != i. |
|  | If VCi[ j ] = k then Pi knows that at least k events have happened at Pj |

## ****\*Question 8**** (1 point)

Select the correct statement(s) regarding NTP.

Question 8 options:

|  |  |
| --- | --- |
|  | Measuring the exact value of the one-way network delay between two computers with different clocks is not always possible using NTP. |
|  | NTP requires four measurements to estimate the clock offset. |
|  | NTP is a protocol used for estimating the clock offset between two computers. |
|  | The maximum estimated value of the one-way network delay is used to choose the most reliable estimate of the offset. |

## ****Question 9**** (1 point)

Consider the following statements regarding a Lamport logical clock:  
(I) If one event's logical time is less than another event’s, then the former happens before the latter.  
(II) if an event happens before another event, then the former event's logical time is smaller than the logical time of the latter event.

Question 9 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****\*Question 10**** (1 point)

Consider the following statements regarding the execution of NTP with client A and server B.  
(I) A always adjusts its time.  
(II) If B is a stratum-k server, then A will become a stratum-(k + 1) server if its original stratum level was previously larger than k+1. k

Question 10 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

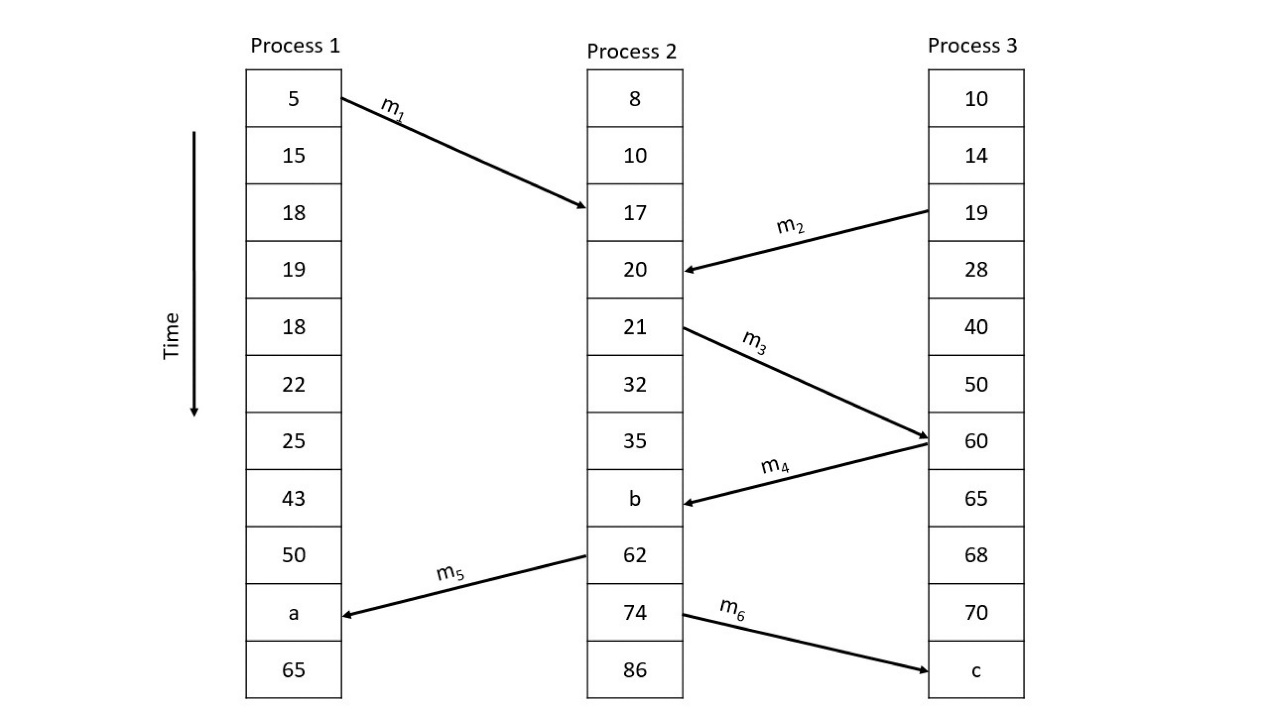
## ****\*Question 11**** (1 point)

Consider the following statements regarding the NTP offset and delay formulas.  
(I) δ is equal to the average of the one-way network delay of request and response messages. min  
(II) The error in the estimated offset is proportional to the difference between the one-way network delays of the request and response messages.

Question 11 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****Question 12**** (1 point)

Consider the following three processes, each of which has its own clock, and they are using a Lamport logical clock for synchronization. Select all possible values for the numbers a,b,c. Note that arrows are representing communication between two processes.

Question 12 options:

|  |  |
| --- | --- |
|  | a=63, b=61, c=71 |
|  | a=63, b=61, c=75 |
|  | a=63, b=42, c=75 |
|  | a=64, b=61, c=75 |
|  | a=63, b=61, c=84 |
|  | a=64, b=38, c=72 |

# Quiz 16(5/8)

## ****Question 1**** (1 point)

Select the correct statement(s) regarding scalability.

Question 1 options:

|  |  |
| --- | --- |
|  | Although scaling conventional databases is not impossible, it comes at a cost (e.g., of higher transaction latency). |
|  | Scaling out a database primarily improves the throughput of write queries.read |
|  | In a database with multi-master replication, resolving write/write conflicts is generally more difficult than in a database with primary-based replication. |
|  | By adding storage, memory, and processing cores, it is possible to scale up conventional databases. |

## ****Question 2**** (1 point)

Consider the following two statements regarding CAP:

(I) Causal consistency is achievable in AP systems.

(II) CP systems are appropriate for inconsistency-tolerant and latency-sensitive applications. AP

Question 2 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****Question 3**** (1 point)

Select the correct statement(s) regarding CQL operations and their execution in Cassandra.

Question 3 options:

|  |  |
| --- | --- |
|  | A put operation (UPDATE statement) is executed by a coordinator, which is the node to which the client connects. |
|  | SELECT is used to get one or more rows from a single Cassandra table. |
|  | CREATE KEYSPACE configures the replication factor for a keyspace. |
|  | In the get operation (SELECT statement), the coordinator must contact all replicas of a row. |

## ****Question 4**** (1 point)

Consider a partially replicated database system with ten servers and a replication factor of N = 4.  Suppose that the read and write quorums are configured as N\_R = 1 and N\_W = 1.  Is this configuration sufficient to guarantee full read availability?  Choose the best answer.

Question 4 options:

|  |  |
| --- | --- |
|  | No.  If all replicas of a data item reside in the same partition then clients who are not in that partition do not have access to those replicas and cannot read the data item. |
|  | No.  By the CAP principle, ensuring both high availability and strong consistency simultaneously is not possible. |
|  | Yes.  Hinted handoff ensures full read availability in this system. |
|  | Yes.   The use of small partial quorums ensures that this is an AP system with full read availability. |

## ****Question 5**** (1 point)

Select the correct statement(s) regarding Apache Cassandra.

Question 5 options:

|  |  |
| --- | --- |
|  | Columns in Cassandra can be sparse.  It has an efficient way to store such columns. |
|  | A row key uniquely identifies each row.  It is similar to a primary key. |
|  | In the Casandra data organization, each column has a name, rows of values, and each value has a timestamp. |
|  | Cassandra supports tunable consistency, and full write availability in some configurations. |

## ****Question 6**** (1 point)

Consider the following statements regarding hinted handoff:

(I) Hinted handoff does not ensure write availability.

(II) Hinted handoff is not necessary for storage systems using full replication.

Question 6 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****Question 7**** (1 point)

Consider the following two statements regarding PACELC:

(I) When a network partition happens, the system must choose between availability or consistency.

(II) If there is no network partition, the system must choose between low latency and consistency.

Question 7 options:

|  |  |
| --- | --- |
|  | Both statements are correct. |
|  | Both statements are incorrect. |
|  | (I) is correct. (II) is incorrect. |
|  | (I) is incorrect. (II) is correct. |

## ****\*Question 8**** (1 point)

Select the statement(s) implied by the CAP principle.

Question 8 options:

|  |  |
| --- | --- |
|  | A distributed system cannot guarantee both availability and consistency in any situation. |
|  | All distributed systems are either CP or AP. |
|  | A CA system guarantees both consistency and availability during a network partition. |
|  | When a network partition failure happens, the system must choose between availability or consistency. |

# Quiz 17(2.5/3)

## ****Question 1**** (1 point)

What is the correct order to containerize an application using Docker and run the container remotely in the cloud?

Question 1 options:

|  |  |
| --- | --- |
|  | Create a Dockerfile: Specify the parent image, set the working directory and provide commands to run your application. |
|  | Create an image: docker build. |
|  | Create the application (e.g.,: a Java program). |
|  | Save the image on the development host: docker save.  Copy the image to the remote host. |
|  | Run the container on the remote host: docker run. |
|  | Load the image on the remote host: docker load. |

## ****Question 2**** (1 point)

What are the differences between virtual machines and containers (such as Docker)?

Question 2 options:

|  |  |
| --- | --- |
|  | Virtual machines virtualize the hardware environment, whereas containers virtualize the software environment only. |
|  | When a container is terminated, any data saved in the container’s file system is lost. On the contrary, when a virtual machine stops, it persists any saved data. |
|  | Each virtual machine requires a separate guest OS, whereas each container requires a separate host OS. |
|  | Container images are generally much smaller than virtual machine images |

## ****Question 3**** (1 point)

The following is an example of which component in Docker?

# Set parent image.  
FROM openjdk:8-alpine  
# Set working directory.  
WORKDIR /usr/src/app  
# Copy Java file from the host to container's working directory.  
COPY Server.java .  
# Build the Java code.  
RUN javac Server.java  
# Port number the container is listening on at runtime.  
EXPOSE 8080  
# Command to run the server process on port 8080.  
CMD [ "java", "Server", "8080" ]

Question 3 options:

|  |  |
| --- | --- |
|  | Docker container image |
|  | Docker engine |
|  | Docker file |
|  | Docker command |

# Quiz 18(4.5/6)

## ****Question 1**** (1 point)

*Saved*

Which edition of the Distributed Systems textbook is used in this course?

Question 1 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | First | |
|  | |  |  | | --- | --- | | b) | Second | |
|  | |  |  | | --- | --- | | c) | Third | |
|  | |  |  | | --- | --- | | d) | Fourth | |

## ****Question 2**** (1 point)

*Saved*

The Hadoop framework for parallel data processing was initially developed at which company?

Question 2 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | Yahoo | |
|  | |  |  | | --- | --- | | b) | Facebook | |
|  | |  |  | | --- | --- | | c) | Netflix | |
|  | |  |  | | --- | --- | | d) | Sun Microsystems | |
|  | |  |  | | --- | --- | | e) | Google | |

## ****Question 3**** (1 point)

*Saved*

Which scientist or scientists defined a relation called "happens before"?

Question 3 options:

|  |  |
| --- | --- |
|  | Leslie Lamport |
|  | Maurice Herlihy and Jeanette Wing (in joint work) |
|  | John Ousterhout |
|  | Eric Brewer |

## ****Question 4**** (1 point)

*Saved*

Question 4 options:

In which year did the authors of the Spark framework publish a paper on Resilient Distributed Datasets in the NSDI conference? (Enter four decimal digits into the box.)



## ****Question 5**** (1 point)

*Saved*

Question 5 options:

Eric Brewer presented the CAP Principle in a keynote speech at the PODC conference. In which year did this happen? (Enter four decimal digits into the box.)



Hint: do a quick web search since this one is not in the lecture slides.

## ****Question 6**** (1 point)

*Saved*

The Curator client for ZooKeeper was initially developed at which company?

Question 6 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | Facebook | |
|  | |  |  | | --- | --- | | b) | Netflix | |
|  | |  |  | | --- | --- | | c) | Sun Microsystems | |
|  | |  |  | | --- | --- | | d) | Google | |
|  | |  |  | | --- | --- | | e) | Yahoo | |

# Quiz 19(2/4)

## ****Question 1**** (1 point)

What property is guaranteed between phase 1 and phase 2 of the coordinated checkpointing algorithm?

Question 1 options:

|  |  |
| --- | --- |
|  | No process can receive a message, except from the coordinator. |
|  | No process can send a message, except to the coordinator. |
|  | A distributed snapshot has been created in phase 1. |
|  | Each process has taken a local checkpoint in phase 1. |

## ****Question 2**** (1 point)

If one participant in the two-phase commit protocol is in the INIT state, what are the possible states of the other participants?

Question 2 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | INIT, READY, COMMIT | |
|  | |  |  | | --- | --- | | b) | INIT, READY, ABORT, COMMIT | |
|  | |  |  | | --- | --- | | c) | INIT, READY, ABORT | |
|  | |  |  | | --- | --- | | d) | INIT, READY | |

## ****Question 3**** (1 point)

What is the purpose of the coordinated checkpointing algorithm?  Choose the best answer below,

Question 3 options:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  | | --- | --- | | a) | To ensure that a new distributed snapshot is created. | |
|  | |  |  | | --- | --- | | b) | To ensure that a recovery line exists. | |
|  | |  |  | | --- | --- | | c) | To ensure that for every send event, the distributed snapshot also records the corresponding receive event. | |
|  | |  |  | | --- | --- | | d) | To ensure that each process produces a new local checkpoint. | |

## ****Question 4**** (1 point)

Which failure scenario(s) does the two-phase commit protocol (as described in the textbook) tolerate?

Question 4 options:

|  |  |
| --- | --- |
|  | Simultaneous failure of any minority of processes. |
|  | Failure of the coordinator in the INIT state. |
|  | Failure of any one process at any time. |
|  | Failure of any number of participants as long as the coordinator does not fail. |
|  | Simultaneous failure of the coordinator and at most one participant. |