# ID2221 - Data-Intensive Computing (Oct. 25, 2024)

**Examiner:** Amir H. Payberah (phone num: 072-55 44 011)
**No aids are allowed**
The exam consists of two parts. Part 1 contains 15 multiple-choice questions, each with only one correct answer, and each question has one point. Part 2 is a written question with five points. The final grading for the exam is as follows:
**A:** $x \geq 18$, **B:** $18 < x \leq 16$, **C:** $16 < x \leq 14$, **D:** $14 < x \leq 12$, **E:** $12 < x \leq 10$, **F:** $x < 10$

---

1. Which of the following scenarios could lead to the creation of a stale replica in GFS?.

   (a) A client writes to all replicas of a chunk simultaneously.
   (b) **A chunkserver fails and misses updates while it is down.**
   (c) The master server grants a new lease on a chunk to a different chunkserver.
   (d) A client reads data from multiple replicas of a chunk concurrently.

---

2. In the context of NoSQL databases, what is the key characteristic of the "eventual consistency" model?

   (a) **Updates are propagated to replicas asynchronously, potentially leading to temporary inconsistencies.**
   (b) Only a single designated replica holds the most up-to-date data, while others serve as backups.
   (c) Consistency is completely disregarded in favor of maximizing performance.
   (d) All database replicas are updated simultaneously, ensuring immediate consistency.

---

3. In the context of the CAP theorem, what does "partition tolerance" mean?

   (a) A system can continue operating even if data is unevenly distributed across nodes.
   (b) **A system can maintain functionality even when network communication between nodes is disrupted.**
   (c) A system can tolerate the loss of a certain number of nodes without losing data.
   (d) A system can adapt to changing data access patterns and query loads effectively.

---

4. When a new node is added to a Cassandra cluster that uses consistent hashing, what happens to the existing data?

   (a) All existing data is rehashed and redistributed across all nodes, including the new node.
   (b) The new node takes over responsibility for a portion of the data, but no data is moved from existing nodes.

(c) **Only the data that falls within the new node's hash range is moved from existing nodes to the new node.**

(d) The new node remains idle until a sufficient amount of new data is inserted to warrant its use.

---

5. If you aim to have MapReduce job output in total sort order and are using multiple reducers, what crucial factor must be considered when designing the partitioning function?

(a) The partitioning function should randomly distribute keys to reducers, ensuring an even workload.

(b) The partitioning function should group keys with similar characteristics together, optimizing data locality for reducers.

(c) The partitioning function should minimize the number of unique keys assigned to each reducer, reducing the memory footprint.

(d) **The partitioning function should ensure that keys assigned to different reducers maintain a consistent order, so that the final output remains sorted.**

---

6. What is the main advantage of using an In-Map Combiner in a MapReduce job?

(a) It reduces the number of map tasks required, decreasing overall job execution time.

(b) It allows map tasks to start processing data before all input splits have been fully loaded.

(c) It improves data locality by ensuring that related key-value pairs are processed on the same worker node.

(d) **It performs partial aggregation of data before it's sent to reducers, reducing network traffic.**

---

7. You have an RDD named words containing a collection of words. You want to count the frequency of each unique word using a key-value RDD. Which sequence of transformations will achieve this goal?

(a) `words.map(word => (word, 1)).reduceByKey(_ + _)`

(b) `words.flatMap(word => word.split("")).map(word => (word, 1)).reduceByKey(_ + _)`

(c) `words.reduceByKey(_ + _).map(word => (word, 1))`

(d) `words.groupBy(word => word).map((word, count) => (word, count.length))`

---

8. How does Spark manage fault tolerance using the lineage graph?

(a) By replicating the RDD data across multiple nodes in the cluster.

(b) By storing the results of all transformations to disk after execution.

(c) **By tracking the sequence of transformations and recomputing only the lost partitions.**

(d) By checkpointing all intermediate RDDs automatically during execution.

---

9. Which of the following best describes the relationship between a DataFrame and a Dataset?

(a) **A DataFrame is a Dataset that holds data as untyped rows, while a Dataset holds data as strongly-typed objects.**

(b) A DataFrame is an RDD, whereas a Dataset is only available in Spark SQL and cannot be converted to an RDD.

(c) A DataFrame provides compile-time type safety, while a Dataset does not.

(d) A Dataset is only used for streaming data, while a DataFrame is used for batch processing.

---

10. In Kafka, what is the purpose of partitioning a topic?

(a) It ensures that messages are processed in exactly the order they were produced.

(b) **It enables the distribution of messages across multiple brokers for scalability and fault tolerance.**

(c) It allows consumers to selectively subscribe to specific categories of messages within a topic.

(d) It compresses messages to reduce storage space and network bandwidth consumption.

---

11. What challenge arises when using event time for stream processing?

(a) Event timestamps are not always accurate, leading to inconsistencies in data analysis.

(b) Processing data in event-time order often results in significant processing delays.

(c) **Events can arrive out of order, making it difficult to determine the true sequence of events.**

(d) It is computationally expensive to maintain event timestamps for every data tuple in the stream.

---

12. How does Spark Streaming handle stateful operations, where computations depend on data from previous batches?

(a) It implicitly stores the state of all RDDs in memory, allowing access to historical data at any time.

(b) It relies on external databases to store and retrieve state information, ensuring durability.

(c) It automatically checkpoints RDDs to persistent storage at regular intervals, enabling state recovery.

(d) **It provides specialized stateful operations that allow developers to manage and update state explicitly.**

---

13. What is a key characteristic of the Gather-Apply-Scatter (GAS) model used in GraphLab?

(a) **It breaks down vertex computations into distinct phases for accumulating, processing, and distributing information.**

(b) It strictly enforces synchronous execution of vertex programs, ensuring deterministic results.

(c) It relies on message passing between vertices for communication, similar to Pregel.

(d) It prioritizes processing vertices with the highest degree (most connections) first to expedite convergence.

---

14. What is the primary challenge addressed by resource management systems like Mesos, YARN, and Borg?

(a) **Allocating and managing resources (CPU, memory, etc.) to diverse applications running on a shared cluster.**

(b) Efficiently storing and retrieving large datasets across a cluster of machines.

(c) Coordinating the execution of tasks in a distributed environment to ensure fault tolerance.

(d) Providing a secure and isolated environment for applications to prevent interference between them.

---

15. In Delta Lake, what is the purpose of the DeltaLog?

(a) It stores the schema definition for all tables within the Delta Lake environment.

(b) It acts as a cache for frequently accessed data, improving query performance.

(c) **It records all changes made to a Delta Lake table, enabling features like ACID transactions and time travel.**

(d) It manages the distribution of data files across the storage cluster, ensuring data availability and fault tolerance.

---

16. Assume you need to design a data processing pipeline for a social media platform that gathers and processes large volumes of unstructured data, such as text posts, images, and user interactions (likes, comments, shares). The goal is to:

- Ingest and store data in real-time.
- Process the data for multiple purposes, such as content recommendations, trend analysis, and user behavior analytics.
- Ensure that the system can handle millions of users and petabytes of data.
- Support batch processing for historical analytics, real-time processing for immediate insights, and ad-hoc querying.

The dataset includes:

- Unstructured data, e.g., posts, images, comments (text and metadata).
- User interactions, e.g., likes, shares, and real-time user behavior (clicks, views).
- Metadata, e.g., user profiles, device information, and timestamps.

Your task is to design a complete data pipeline that handles the following aspects:

1. **(1 point)** Data ingestion: propose tools and technologies to collect data in real-time from millions of users?
2. **(1 point)** Data storage: choose appropriate data storage systems for structured, semi-structured, and unstructured data. Explain your choice of storage solutions for (i) user profiles and structured metadata, and (ii) unstructured content like posts, images, and interactions. Explain the data model (e.g., relational, document-based, key-value) you would use for each type of data and why.
3. **(1 point)** Data processing: how will you process the data for real-time analytics (e.g., trending topics, recommendations) and batch analytics (e.g., historical user behavior)? Which tools will you use for streaming and batch processing, and why are they suitable for your application?
4. **(1 point)** Scalability and fault tolerance: discuss how your system will scale to handle millions of users and how it ensures fault tolerance. Describe any optimizations you would apply to improve performance and manage resource usage.
5. **(1 point)** Pipeline architecture: describe the full architecture pipeline and explain how the tools you have chosen work together to achieve a scalable, efficient, and reliable system.