|  |  |
| --- | --- |
|  | When a computing can be called as data intensive computing? What is the challenges in data intensive computing and how these challenges can be addressed? Explain. |
| 1. | In what ways does Google Bigtable's schema-less, column-family-based design differ from traditional relational databases, and how does this impact its suitability for workloads like analytics, IoT, and real-time data streaming? |
| 2. | How does Bigtable achieve a balance between strong data consistency within a single row and eventual consistency across distributed systems, and what are the implications of this design for real-time applications? |
| 3. | How does CouchDB's schema-free JSON document storage and support for MapReduce queries make it suitable for specific use cases like offline-first applications, and what are its limitations compared to other NoSQL databases like MongoDB or Cassandra? |
| 4. | How does AWS enable scalability and elasticity for dynamic workloads through services like EC2 Auto Scaling and AWS Lambda, and what strategies can organizations use to optimize costs while maintaining performance in a cloud-native environment? |
| 5. | How does AWS architecture leverage services like EC2, S3, RDS, and Lambda to achieve scalability, fault tolerance, and high availability, and what are the key considerations for designing cost-efficient and secure solutions within this ecosystem? |
| 6. | How does Apache Cassandra's distributed architecture, based on a peer-to-peer model and consistent hashing, ensure scalability and fault tolerance for handling large-scale, real-time data workloads, and what trade-offs exist in terms of complexity and consistency guarantees? |
|  |  |
|  |  |