732A54 Big Data Analytics

Help for written exam - Examples of exam question

Arian Barakat

2017-01-11

---

1. Databases for Big Data

1.1 NoSQL data stores and techniques

1. **Question:**
   Explain the main reasons for why NoSQL data stores appeared.

Answer:
There are several reasons why NoSQL appeared (mainly due to 'One Size does not fit all) and these are:

- Frequent schema changes, management of unstructured and semi-structured data
- Huge dataset - data volume increase exponentially
- Different applications have different requirements - new needs

---

2. **Question:**
   List and describe the main characteristics of NoSQL data stores.

Answer:

- Simple and flexible non-relational data models
- High availability and relax data consistency (CAP theorem. **Link to CAP**)
  - BASE vs. ACID
- Easy to distribute - horizontal scalability
- Data are replicated to multiple nodes
  - Down nodes easily replaced
  - No single point of failure
- Cheap and Easy (or not) to implement (opens source)

---

3. **Question:**
   Explain the difference between ACID and BASE properties.

Answer:

BASE:

- Basic Availability: An application works basically all the time (despite partial failures)
- Soft-state: Is in flux and non-deterministic (changes all the time)
- Eventual consistency: Will be in some consistent state (at some time in future)

ACID:

- Atomic: Everything in a transaction succeeds or the entire transaction is rolled back.
- Consistent: A transaction cannot leave the database in an inconsistent state.

- Isolated: Transactions cannot interfere with each other.
- Durable: Completed transactions persist, even when servers restart etc.

In other words, the BASE properties are more relaxed than the ACID properties

Link: Basic differences.

- ACID – properties needed to guarantee consistency and availabiltye.
- BASE- properties come into play if availability and partition tolerance are favored.

---

4. **Question:**
   Discuss the trade-off between consistency and availability in a distribute data store setting.

Answer:
By assuming that in a distribute data store setting there is no single point of failure we can be certain that a customer will be able to view and add to the shopping cart during various failure scenarios. However, due to the availability property, we can also assume some violation regarding the consistency property that there is a small probability that a transaction is not completely executed from beginning to end without interference from other transactions.

---

5. **Question:**
   Discuss different consistency models and why they are needed.

Answer:
There are several consistency models, some examples are:

- Strong Consistency
  - After the update completes, any subsequent access will return the updated value.
- Weak Consistency
  - The system does not guarantee that subsequent accesses will return the updated value. A number of conditions need to be met before the value will be returned. The period between the update and the moment when it is guaranteed that any observer will always see the updated value is dubbed the inconsistency window.
- Eventual Consistency
  - This is a specific form of weak consistency; the storage system guarantees that if no new updates are made to the object, eventually all accesses will return the last updated value. If no failures occur, the maximum size of the inconsistency window can be determined based on factors such as communication delays, the load on the system, and the num- ber of replicas involved in the replication scheme. The most popular system that implements eventual consistency is the domain name system (DNS). Updates to a name are distributed according to a con gured pattern and in combination with time-controlled caches; eventually, all clients will see the update.
- Other variations of the eventual consistency model

These consistency models are needed due to the ‘one size does not fit all’ term. In other words, for some applications the consistency criteria must be relaxed in order to obtains high availability.

Source:

---

6. **Question:**
   Explain how consistency between replicas is achieved in a distributed data store.

Answer: A replication factor r is introduced: not only for the next node but the next r nodes in clockwise direction become responsible for a key.

7.Question:
Explain the CAP theorem.

Answer:
The CAP theorem states that a distributed computer system can only have 2 out of the following three following properties:

- Consistency
- Availability
- Partition tolerance

---

8. **Question:**
Explain the differences between vertical and horizontal scalability.

Answer:

Overall Definition of Scalability:

- (Improvements of a) System in order to handle the growing amounts of data without losing performance Vertical Scalability (scale up):

- Adding resources (more CPUs, more memory) to a single node
- Using more threads to handle a local problem

Horizontal Scalability (scale out):

- Adding nodes (more computers, servers) to a distributed system - low costs for commodity hardware
- Often surpasses scalability of vertical approach

In other words, vertical scalability is increasing a system's capacity by increasing the power of one computer, while horizontal scalability is increasing a system's capacity by increasing by increasing the number of computers (and not necessary the power of those computers)

---

9. **Question:**
Explain how consistent hashing works and what are the problems it addresses.

Answers:

- **Arrange the nodes in a ring**
- **Include hash values of all nodes in hash structure**
- **Calculate hash values of the key to be added**
- **Choose node which occurs next clockwise in the ring**

The problem that consistent hashing tries to address is to minimize the number of nodes to be copied after a configuration change. This is done by incorporate hardware characteristics into hashing model and arranging the nodes in a ring and making sure that each node is in charge of the hash values in the range between its neighbor node. If a node is dropped or gets lost, missing data is redistributed to adjacent nodes. However, if a node is added, its hash value is added to the hash table and the hash realm is repartitioned, and hash data will be transferred to new neighbor.

10.Question:
Explain how vector clocks work and what are the problems they address?

Answers: problems that vector clock addresses:

- Error prone
- Insuffiency , as we cannot catch causalities (needed to detect conflict).

11. **Question:** List and describe dimensions that can be used to classify NoSQL data stores?

Answers:

- Data model – how the data is stored
- Storage model – in-memory vs persistent
- Consistency model – strict, eventual consistent, etc.
- Physical model – distributed vs single machine
- Read/Write performance – what is the proportion between reads and writes

….etc.

12. **Question:**

List and describe the main characteristics and applications of NoSQL data stores according to their data models?

Answers:

- Key-Value stores
- Column family stores
- Document stores
- Graph databases

HDFS

Explain what HDFS is and for what types of applications it is (not) good for?

??? defination

Is good for :

- Store very large files
- Streaming access
- Comodity hardware

Is not good for:

- Low latency data access
- Lots of small files(may affect the idle time for the processors)
- Multible writes and arbitrary file modification.

Explain the organization of HDFS.

Explain the process of reading and writing to HDFS.

Explain how high availability is achieved in HDFS.

- Dynamo
  - Explain the data model and list main applications of Dynamo?

    Answers:

    Key value store. Is the data model for dynamo.

    Applied on:

    - Best seller lists
    - Customer preferences
    - Product catalog.
  - Explain the Dynamo design considerations and what are the advantages of Dynamo in comparison to RDBMSs?

  - Explain how basic NoSQL techniques are applied in Dynamo.
  - Explain versioning and semantic reconciliation in Dynamo.