CS4225/CS5425 BIG DATA SYSTEMS FOR DATA SCIENCE

Tutorial 2: NoSQL

1. Explain the pros and cons of NoSQL systems in the form of the BASE properties which they (often) satisfy.

Answer:

As a review, the BASE properties state that:

- **B**asically **A**vailable**:** Reading and writing operations are available as much as possible, but without consistency guarantees (e.g. read may not get the latest updated value).
- **S**oft state**:** The state of the system is always 'soft' or changing with inputs, until it reaches 'eventual consistency'.
- **E**ventually consistent**:** The system will *eventually* become consistent (e.g. multiple reads *eventually* return the same value).

Considering these properties, some pros and cons of NoSQL system include:

Pros:

- **Performance**: NoSQL systems often sacrifice strong consistency while aiming for low latency and high availability (e.g., due to removing the need for locks and other concurrency mechanisms)
- **Scalability**: NoSQL systems are straightforwardly horizontally scalable (due to sacrificing strong consistency)

Cons:

- **Outdated data**: outdated data may cause mistakes or require additional work to handle on the application side

2. The following question relates to the paper (not required reading for the class, but still a useful summary if you are interested):

Rick Cattell. 2011. Scalable SQL and NoSQL data stores. SIGMOD Rec. 39, 4 (May 2011), 12-27.

In the paper, they have shared suitable applications for key-value stores and document stores:

Application of key-value store:

As an example, suppose you have a web application that does many RDBMS queries to create a tailored page when a user logs in. Suppose it takes several seconds to execute those queries, and the user's data is rarely changed, or you know when it changes because updates go through the same interface. Then you might want to store the user's tailored page as a single object in a key-value store, represented in a manner that's efficient to send in response to browser requests, and index these objects by user ID. If you store these objects persistently, then you may be able to avoid many RDBMS queries, reconstructing the objects only when a user's data is updated.

Application of document store:

A good example application for a document store would be one with multiple different kinds of objects (say, in a Department of Motor Vehicles application, with vehicles and drivers), where you need to look up objects based on multiple fields (say, a driver's name, license number, owned vehicle, or birth date).

Discuss some factors that make these applications suitable for key-value stores and document stores respectively.

Answer:

Key-value store:

- Improves scalability and efficiency – writing or reading user pages is faster.
- No need for complex queries or based on the content of user pages – just reads and writes.
- May be acceptable for user pages to be slightly stale – then eventual consistency is acceptable

Document store:

- Flexible schema may be beneficial (e.g. special types of vehicles may require different sets of fields)
- Unlike key-value stores, document stores are more suitable for queries based on fields of a document

3. Assume that your e-commerce company's webpage has a NoSQL document store database containing data about users visiting the webpage (e.g. IP address, country, time spent browsing the webpage, number of products they have bought, etc.) Your CEO suggests using the Range Partitioning scheme with the number of products they have bought as a partition key. State and explain a possible benefit of this choice, and a possible disadvantage of this choice.

Answer:

Benefits:

- If we need to run **filter or group by queries based on the number of products bought** (e.g., frequent buyers vs new buyers), this partitioning scheme is efficient (as information about buyers with similar number of products bought is **co-located**, allowing faster queries without needing to scan the entire database.

Disadvantages:

- May lead to highly **imbalanced** partitions, e.g. if a large number of users have only bought 1 product
- **Maintenance**: when a customer starts buying a lot of products, they need to be moved from one partition to another, leading to additional overhead.
- **Other queries**: other user characteristics may be more important, like age, geography etc. Queries grouping by such characteristics would need to scan multiple partitions.

4. Imagine you are developing a digital platform for NUS students to manage their courses and extracurricular activities they are taking, which also manages information about courses (e.g., professors, prerequisites, time-tables, etc), and aims to recommend courses for students.

Consider the NoSQL databases we have covered in the class, and choose one which is suitable for satisfying a specific use-case within the above digital platform. (You do not have to explain how to implement the entire system – you can choose a single use-case to focus on). Explain why your choice is appropriate.

Answer:

Document Store: could be useful for storing course information or student information. Document stores like MongoDB provide **flexible schema** to accommodate different course characteristics (e.g., some courses may have prerequisites / cross-listings, others may not). Similarly, some students may input various profile information, while others may not.

Graph DB: could be useful if certain use-cases crucially involve relationships: e.g. recommending courses to students, recommending study groups for students, managing complex prerequisites / time conflicts between courses to recommend a timetable, etc.