BDA 4-4

1

Hello everyone, I am Haiying Che, from Institute of Data Science and knowledge Engineering

School of Computer Science, in Beijing Institute of Technology,

in this session, we discuss about NoSQL Database

Core concepts CAP theorem and No SQL database transaction model BASE

2

The big data computing system can be summarized into three categories:

Data storing system, Data processing system, Data application system

Here we still in the scope of Data storing system.

In data storing system, there are 4 parts to accomplish different tasks,

which is Data collection and modeling, **Distributed file system, Distributed database/data warehouse and Unified Data Access Interface.**

Here we still in NoSQL Database.in No SQL database, there are two important concepts, CAP theorem and No SQL database transaction model BASE.

Let’s look at the CAP theorem first.

3

**CAP Theorem means A distributed system cannot meet 3 requirements of Consistency, Availability, and Partition tolerance at the same time.**

**At most, only two of them can be satisfied at the same time.**

C stands for Consistency, which means even in a distributed environment, the data of multiple storage nodes have the same data value at the same time, and all data backup updates should be synchronized

A stands for Availability, which means client can quickly read data, return operation results within a reasonable time, and ensure that every request has a response regardless of success or failure

P stands for partition tolerance, which means that when a partition in the system cannot communicate with other nodes, it does not affect the normal operation of the rest of the system, or the error or loss of part of the system data does not affect the overall operation of the system.

According to the CAP theorem , a distributed system can only meet two of the CAPs in the mid-run data read and write operations, but not three at the same time.

The choice of CAP also leads to the difference between relational databases and NoSQL databases.

Let’ s watch a video about **CAP Theorem** .

4

From the video we understand why the CAP **Consistency, Availability, and Partition tolerance cant be satisfied at the same time.**

**The diagram here shows different product choose different combination of C, A and P.**

**RDBMS choose C and A,**

**CA: That is to emphasize consistency (C) and availability (A), give up partition tolerance (P),**

the simplest way is to put all transaction-related content on the same machine.

Obviously, this approach will seriously affect the scalability of the system. Traditional relational databases (MySQL, SQL Server, and PostgreSQL) all adopt this design principle, so they have poor scalability.

**MongoDB, HBase and Redis choose C and P,**

**CP: That is to emphasize consistency (C) and partition tolerance (P), give up availability (A),**

when the network partition situation occurs, the affected services need to wait for the data to be consistent, so they cannot provide services to outside parties

**and CouchDB， Cassandra etc. choose A and P**

**AP: That is to emphasize availability (A) and partition tolerance (P), give up consistency (C),** which allow the system to return inconsistent data

5

Now let’s learn the NoSQL characters BASE

Speaking of BASE (Basically Available, Soft-state, Eventual consistency), I have to talk about ACID.

A database transaction has four **ACID properties**:

* A (Atomicity): Atomicity means that the transaction must be an atomic unit of work. For data modification, **either all or none of them are performed.**
* C (Consistency): Consistency means that when the transaction is completed, all data must be kept in a consistent state
* I (Isolation): Isolation means that changes made by concurrent firms must be isolated from changes made by any other concurrent firms
* D (Durability): Persistence means that after the transaction is completed, its impact on the system is permanent. This modification will be maintained even if a fatal system failure occurs

The basic meaning of BASE is Basically Available, Soft-state, and Eventual consistency:

**Basically, available** means that when a part of a distributed system becomes unavailable due to a problem, the other parts can still be used normally, that is, the situation where the partition fails

**"Soft-state"** is a term corresponding to "hard-state". When the data stored in the database is "hard state", data consistency can be guaranteed, that is, the data is always correct. "Soft state" means that the state can be out of sync for a period, with a certain lag

**Eventual consistency** means subsequent access operations temporarily can fail to read the updated data, but after a period, the updated data must eventually be read.

6

 According to whether the subsequent operations can obtain the latest data under high-concurrency data access operations.   The types of consistency include strong consistency and weak consistency

The main difference between the two is whether the subsequent operations can obtain the latest data under high-concurrency data access operations.

For strong consistency, after performing an update operation, other subsequent read operations can ensure that the latest updated data is read; otherwise, if it cannot be guaranteed that all subsequent reads will read the latest updated data, then it is Just weak consistency.

**Eventual consistency** is just a special case of weak consistency, allowing subsequent access operations to temporarily fail to read the updated data, but after a period, the updated data must eventually be read.

 7

In this session we learned two important concepts, CAP theorem and No SQL database transaction model BASE-Basically Available, Soft-state, Eventual consistency.

thank you for your attention, if you have any question, feel free to contact me.