Jonathan Roland

WEB-335 Introduction to NoSQL

11/1/2020

Instructor: Prof. Krasso

Discussion 3.1

Document databases are key-value pair stores that store and retrieve various types of documents such as XML and JSON. In this structure the key acts as the document identifier and the value stores the document. The stored documents can share similarities but are not bound to strict column adherence. Instead they allow different attributes and data structures from document to document. Popular document database types include MongoDB, CouchDB, RavenDB and others.

In document databases like MongoDB the terminology differs from relational database counterparts. A schema in a relational database is called a database in MongoDB. A grouping of documents in a database is called a collection. This level in the organization chart is comparable to a table in a relational database. Likewise, documents in a document database are similar to rows in a relational database.

Document databases offer customizable write parameters to allow the user to weigh performance against consistency concerns. Document databases like MongoDB concern themselves with availability by allowing variable settings for consistency while replicating data to nodes. While forming write commands, the user can choose how many nodes will need to replicate the write before the operation returns a success, along with other similar concerns. This customizability also allows data to be split up among nodes to keep traffic contained to specific request types, such as read-only.

Transactions work differently in document databases. In relational databases data is modified with INSERT, UPDATE, and DELETE commands which can affect numerous tables and rows per transaction. The user can utilize COMMIT and ROLLBACK commands on these operations as well. In document databases, transactions only occur to a specified document when the user requests to write to it.

The principle disadvantage of document databases is in complex data operations. Because documents do not follow an enforced structure (even within individual documents), complex queries like join operations will never reliably return the same results. Because of the single document transaction model (also called an atomic transaction), document databases are also poor candidates for complex transactions. If these are common cases, a relational database will be a better solution. For cases where data is stored and retrieved on a more individual basis, and where speed and scalability are concerned, document databases are a great choice.

Reference:

*NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence by Pramod J. Sadalage Martin Fowler(2012-08-18)*. (2020). Addison-Wesley Professional.