

Spanner: Becoming a SQL System

Summary

Spanner is produced by Google. It is a globally distributed data management system, providing an abstraction that lets you query distributed data like it is all on one machine. Spanner scales very well horizontally. Spanner was developed because of the limitations of a key-value store. Spanner is distinct because it allows the user to make a transaction in a system where the data is stored on different machines. I think this is a helpful sentence from the paper: “A database is horizontally row-range sharded”.

Spanner exposes two different types of APIs. One is for making queries and the other is for reading queries. Spanner uses location hints to map queries to the correct machines.

It uses a filter tree to track to make the range row lookup faster. This data structure is maintained during run time. The following are ways that Spanner handles failures. The system hides all transient failures. A transient failure would be complete if the action was simply tried again. This means that the developer writing a program with this call would not have to make a retry loop. This is particularly good because writing a good retry loop is particularly challenging. This is because it is hard to make it with the correct back-off.

Spanner can be upgraded even if all of the machines running Spanner do not upgrade. In this way, new changes are always able to be made.

Spanner uses a common SQL language.

Spanner uses a log-structured merge tree to store data this is similar this using a memtable and sstables.

Pros

- I thought that the paper did a good job of addressing how a developer might use the system.
- I thought that the conclusion did a good job wrapping up the paper and hitting the big design points of the paper.

Cons

- I would like to see documentation for setting up Spanner.
- The author did not report any performance testing.
- I think it would be helpful to list the reason this is better than just a relational database.

Further Developed

It would be interesting to consider common applications of this software.

I find it interesting that so many applications use a SQL-like query language for querying data. It seems like a different way of doing it would be developed.

Other Comments

The developers focus a lot on scalability. Scalability is the utmost concern if you work at a company like Google.

The talk about random data creation and random query generation. This is something that I have thought would be helpful before. It would be interesting to see how they would do this.