

Spanner: Becoming a SQL System

Summary

Spanner is produced by Google. It is a globally distributed data management system, meaning that it provides 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 allow allows the user to make transactions in a system were 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 type 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 range row look up 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 is one that would 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 particular 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 is able to 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 use a common SQL language.

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

Pros

- I thought that the paper did a good job of addressing how a developer might use the system.

Cons

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

Further Developed

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

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

Other Comments