

Paper Evaluation

An Evaluation of Distributed Concurrency Control

Michael Lanthier

This paper evaluates many traditional concurrency control methods when applying them to distributed databases. They looked at common methods such as optimistic concurrency control, multi version concurrency control, and other two phase locking techniques in a distributed environment. The paper identified many bottlenecks when data is distributed such as the slow nature of 2 phase commit and hot records creating contention. They propose that network improvements as well as modifying ones data model can improve performance when dealing with distributed data.

Questions/Comments

1. This paper definitely goes well into depth on where each concurrency control method excels and struggles which is great but this may not be a good measurement for people to decide which technique to use in their database based on their workload. They did acknowledge that they assumed no faults but in large scale systems this is not an assumption that can be taken. In methods like Calvin, failure of the distributor node definitely will hinder the system performance. First you could have a single distributor but when it dies another system would have to identify it's failure and spin up another one. Otherwise you could use a quorum of nodes running a consensus protocol like PAXOS but that would decrease throughput as PAXOS is a very chatty and fairly expensive protocol to run. Additional research on how these concurrency control techniques perform when systems go down would be valuable information for those trying to design high performance systems.
2. This paper takes traditional methods of single system concurrency control and applies them to distributed databases. Like the previous paper on query optimization, are there any protocols that are designed specifically for distributed databases that take into account data distribution to increase throughput and decrease latency?