Conjunction of Majorities Protocol for Scalable Transactions in Distributed (Graph) Databases

Jim Webber (jim.webber@neo4j.com), Chief Scientist, Neo4j

Introduction

Neo4j is a graph database. which safely stores and performantly queries highly associative networks of data. Neo4j uses Raft to provide multi-server fault-tolerance, but at the expense of being a single-leader system which limits write throughput and suffers latency jitter during leader re-elections.

Problem statement

Scaling a graph database is a hard problem. In some graph databases, scale-out is solved by building a graph API atop an scalable eventually consistent database (e.g. Janus Graph over Apache Cassandra). However, Ezhilchelvan et al[1] show that such systems will corrupt data under normal operation because they do not have a coordination protocol that can enforce *reciprocal consistency*, a fundamental property for correctness in graph databases – if I follow you, you know that you are followed by me irrespective on which servers (and redundant replicas) those records are stored. Ezhilchelvan shows it is trivial to cause an inconsistency in the data when an edge connecting vertices across two servers is updated or deleted in the presence of contention.

Our solution

For a scalable solution, any server in the system must be capable of accepting reads and writes and uphold reciprocal consistency in order to be correct. We propose a protocol called ComTx (Conjunction of Majorities for Transactions) for sharded graph databases that can support many writers, help with concurrency control, and in the absence of pathological contention, provide high throughput.

Servers are arranged into shards which are replicas of some part of the graph the user wishes to store. Each server contains a query engine, a storage engine, a resource manager, and a coordinator. The coordinators' task is to take users' updates and apply them to the relevant underlying shards. The resource managers' task is to determine whether requests from a coordinator are compatible with their underlying data as it has been previously committed and locally adjudicate on any schema constraints (e.g. uniqueness or cardinality).

When asking a resource manager to commit a transaction, the coordinator transmits some metadata called the Transaction Directed Acyclic Graph (TxDAG). The TxDAG contains the immediate committed ancestors of the current transaction, taken from the database on which the coordinator happens to be situated. The receiving resource managers can compare the incoming TxDAG with their own and determine whether it is *compatible* (including but not limited to being identical) and will vote accordingly. If a majority of RMs in a shard vote to commit, and all shards have majorities, then the transaction will be committed. This upholds reciprocal consistency since nodes and edges on "both sides" of a shard will commit (or abort) together.

Determining a compatible transaction is intricate, but important to allow good throughput (since relying on strict equality leads to many spurious aborts and so wasted effort). In one of the simpler cases, if a transaction arriving at an RM has committed ancestors in its TxDAG, while the RM understands those transactions to be currently in the prepared state, then the RM can safely make those transactions committed, knowing they must be committed by a majority elsewhere. Now the current transaction *may* have its dependencies satisfied and be able to commit.

For high write throughput the solution permits many coordinators to manage many active transactions concurrently. An individual transaction's progress will be stored in each RM's local TxDAG, which enables them to reason about contention and incompatibility, and vote accordingly.

The future

Our prototype for this system is underway and to-date the experience is encouraging. We have theoretical proofs of the properties the system can and cannot uphold. We hope to gain critical expert feedback (or at least generate some amusement or controversy) by presenting this at the workshop.

---

[1] https://doi.org/10.1007/978-3-030-02227-3_1