Corruption in Distributed Graph Databases

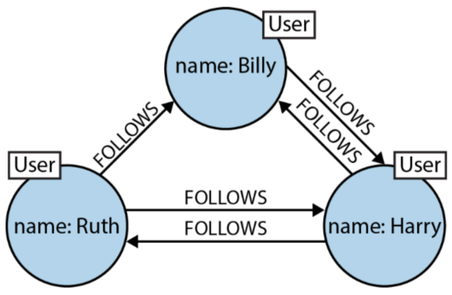
Jack Waudby

Newcastle University

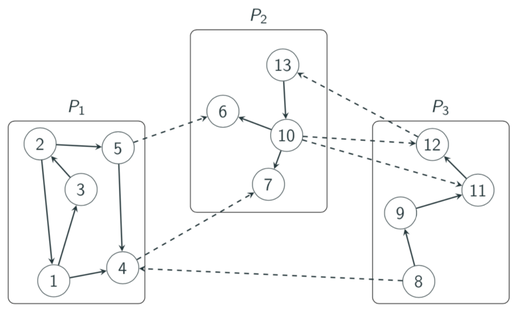
EPSRC Engineering and Physical Sciences Research Council

Common System Architecture

Labeled Property Graph



Partitioned Graph



NoSQL Storage Backend

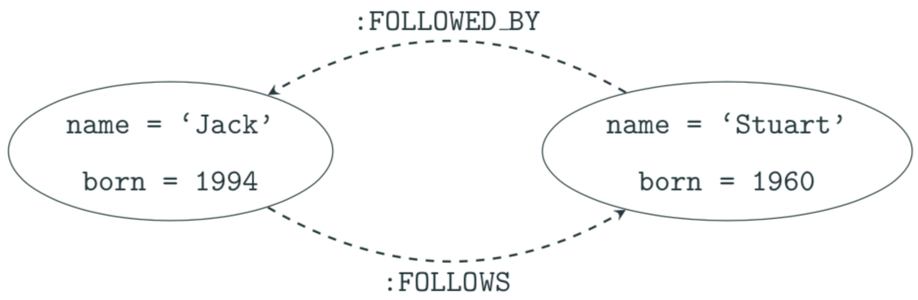
cassandra

Graph Processing Layer

JanusGraph

Reciprocal Consistency

An edge is represented by two physical objects that must remain **reciprocally consistent**



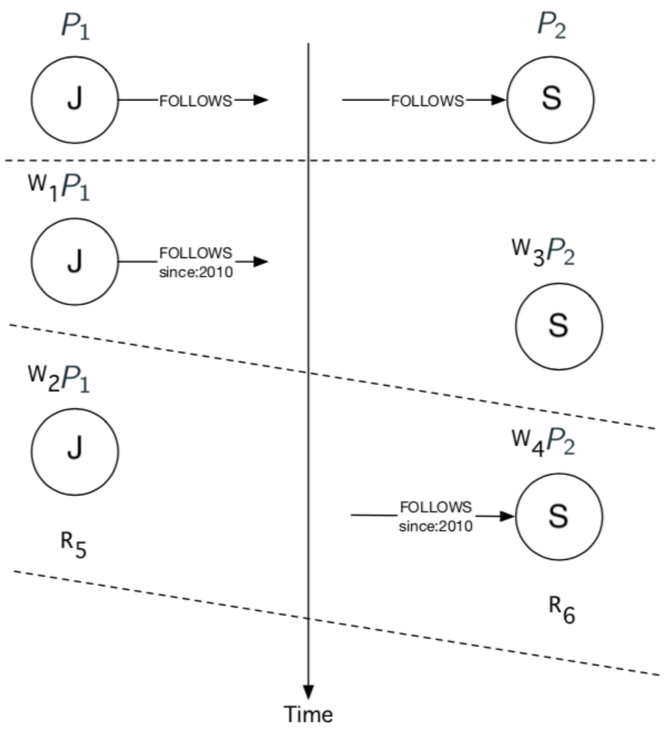
This allows bi-directional edge traversal

Edge Half-Corruption

Maintaining reciprocal consistency is a challenge for distributed edges

NoSQL storage backends do not provide ACID transactions

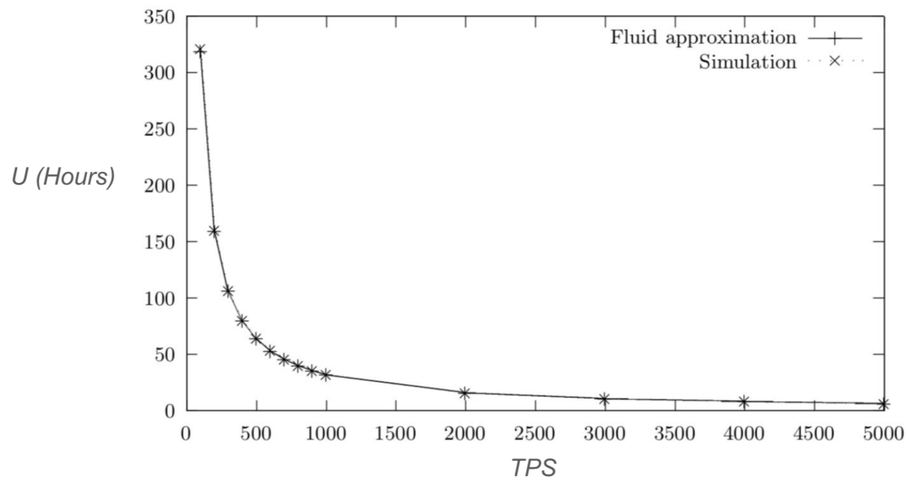
Concurrent updates can interleave and edges become **half-corrupted**



Database Corruption

Simulated the rate of corruption:

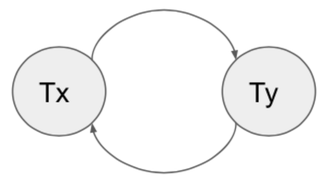
- Scale-free Graph Database with 1 billion nodes, 11 billion edges (30% distributed)
- 10% queries read-write transactions
- Reads-per-query geometrically distributed (mean 15)



Solution

Three-step approach:

1. Identified baseline isolation requirement for distributed graph databases
   (a) Half-corruption of distributed edge occurs due to write cycles between transactions



2. Developed lightweight edge concurrency control mechanism: Interference free updates on a single edge
   (a) Optimistic, non-blocking approach
   (b) Transactions aborted when possibility of half-corruption detected
   (c) Some transactions can be "innocently" aborted
3. Performance evaluation of mechanism via simulation