2020-11-27 Meeting notes

Date

27 Nov 2020

Attendees

- Graham Wallis
- Chris Grote
- David Radley
- Jeremy Taylor (Product Manager)
- Jon Pither (Technical)
- Steve H. (Commercial Manager)

Goals

- Take a first look at the Crux OSS database and potential use related to Repository with history

Discussion items

Time	Item	Who	Notes
60mins	Crux and Egeria	All	• Open discussion, please see notes below

Notes

Crux is an open source project that implements a graph database, which supports efficient point-in-time queries. The implementation is in Clojure, and Crux can either be embedded as a library or run separately.

Crux offers a choice of backend stores - LMDB or RocksDB.

- LMDB is up to 3x faster than Rocks.
- A RocksDB can scale up to around 16TB - beyond that you would need to scale out by replicating. Crux actually started out built on Kafka with the ability to scale out the backend, for size or availability. The incorporation of an embedded Rocks implementation came later but is good for evaluation/development purposes.

Crux uses graph indexes which are more flexible than the indexes you would typically find in an RDBMS. The project actually started out trying to build a graph layer on top of Oracle, but the team discovered that was a non-starter and built Crux.

Compared to Datomic (https://www.datomic.com) if you perform a historical query in Datomic it will result in index scans. In contrast, Crux always incorporates history state into every query and doesn't rely on index scans.

In the Crux graph model each entity (vertex) or relationship (edge) is stored as a document. A document can include (untyped) references to other documents; so a relationship would be stored as a document with references to two other (entity) documents. A document can also have properties. A document is effectively the value of an entity (or relationship) at a point in time. The 'history' or 'evolution' of the object is stored as a time-series of documents. This schema avoids the need for JOINs.

Deletes are soft (reversible). Evictions are permanent. I'm not sure this is quite like our delete (soft) and purge - because I think Egeria may have separate requirements for permanent deletion (i.e. an instance is not recoverable) vs totally forgetting something ever existed, e.g. after 7 years. I'm not sure quite how we support these separately today.

The query language in Crux is called Datalog. This is a recursive language, similar to Prolog or SparQL. It is much lower level than Gremlin for example, and really seems to consist of a small number of logical operations (AND, OR, NOT and some predicates).

- It supports wildcard searches - so you should be able to search for all entities that have a particular substring in an attribute, but it does not support 'wild' wildcarding or 'wild' navigation - i.e. in both cases you need to know the attribute you are testing the predicate against. This would it impossible to find all paths between an arbitrary pair of vertices (entities, for example).
- Suggested initial learning via: http://www.learndatalogtoday.org (very simple tutorial, but the final chapter also illustrates how the very simple predicates can be combined into your own defined "rules" (think functions) that abstract commonly used combinations of predicates for your particular schema and scenarios)
- There will be a deeper dive on Datalog when Jeremy will be presenting at 'Reclojure' on Thursday 3rd December (15:00 - 18:00).
- Performance tests using the Waterloo Diversity SparQL benchmark, with 2 to 3GB of data and 10M 'triples', have shown that compared to neo4j or rdf4j, Crux gets to within 'an order of magnitude'.
- A graph query will result in the generation of tens of thousands of Rocks lookups but there is extensive caching.

In terms of mapping Egeria concepts to Crux our instance properties (core and type defined) would all be stored as document properties. Despite the time-series nature of Crux, Jeremy recommended that core properties such as createTime, updateTime are retained as first class properties of instances rather than being tempted to rely on the time series information.

In Crux, to find out when a document was added to the database you need to perform a scan.

Jeremy described Crux as having historical query capability but also important that it provides 'consistent queries'.

Action items

- [x] Graham Wallis to raise during Innovation section of next week's TSC (potential to start a small special interest group, eg. under innovation and adoption (?))
- [x] Chris Grote / Graham Wallis / David Radley to consider attendance to DataLog session at Reclojure (TBC)
- [ ] Chris Grote / Graham Wallis to follow-up with Crux on next steps, post TSC discussion