copernic: awesome data store

data

Abstract

copernic is web application that is (mostly) implemented with Python programming language. It is supported by a database that is a quad store versioned in a direct-acyclic-graph. It is possible to do time traveling queries at any point in history while still being efficient to query and modify the latest version. The versioned quad store is implemented using a novel approach dubbed generic tuple store. copernic goal is to demonstrate that versioned databases allow to implement workflows that ease cooperation.

Keywords

- data management system
- data science
- distributed version control system
- knowledge base
- open data
- quality assurance
- reproducible science
- python programming language

Introduction

Versioning in production systems is a trick everybody knows about whether it is through backup, logging systems and ad-hoc audit trails. It allows to inspect, debug and in worst cases rollback to previous states. There is not need to explain the great importance of versioning in software management as tools like git, mercurial, and fossil have shaped modern computing.

Having the power of multiple branch versioning open the door to manyfold applications. Like, it allows to implement a mechanic similar to github's pull requests and gitlab's merge requests in many products. That very mechanic is explicit about the actual human workflow in entreprise settings, in particular, when a person validates a change made by another person.

The *versioned quad store* make the implementation of such mechanics more systematic and less error prone as the implementation can be shared across various tools and organisations.

copernic takes the path of versioning data and apply the change-request mechanic to collaborate around the making of a knowledge base, similar in spirit to WikiData and inspired from existing data management systems like CKAN.

The use of a version control system to store open data is a good thing as it draws a clear path for reproducible science. But none, meets all the expectations. **copernic aims to replace the use of git and make practical cooperation around the creation, publication, storage, re-use and maintenance of knowledge bases that are possibly bigger than memory.** Resource Description Framework (RDF) offers a good canvas for cooperation around open data but there is no solution that is good enough according to Collaborative Open Data versioning: a pragmatic approach using Linked Data, by Canova *et al.*

copernic use a novel approach to store quads in an ordered key-value store. It use FoundationDB database storage engine to deliver a pragmatic versatile ACID-compliant versioned quad store where people can cooperate around the making of the future of knowledge. It also rely on a new algorithm to query versioned tuples based on a topological graph ordering of changes. copernic only stores changes between versions. copernic does not rely on the theory of patches introduced by Darcs but re-use some its vocabulary. copernic is the future.