University of ZurichUZH

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, August 4, 2020

MSc Project
Topic: Prototype of a RCAS-based Search Engine for the Software Heritage Archive

The Software Heritage Archive (SWH-Archive) [2,3] is an attempt to collect all publicly accessible software source code and its revision (i.e., commit) history. It is the largest archive of its kind and archives more than 130 million repositories with 8 billion source code files. The SWH-Archive allows researchers to study the evolution of code over time at an unprecedented scale. Researchers often want to narrow down their analysis to the revisions of a single file or a set of files (e.g., all source files in the PostgreSQL project). Moreover, often only a limited portion of the revision history is relevant (e.g., only the revisions that occurred between May, 2020 and June, 2020 are required). Therefore, researchers need an efficient access method to locate the revisions that satisfy (i) a value predicate on an attribute of the revision and (ii) a path predicate on the file paths that are modified in a revision.

The Robust Content-And-Structure (RCAS) index [4] is a novel in-memory index for semi-structured hierarchical data. Unlike pure content indexes or pure structure indexes, the RCAS index is designed to answer Content-And-Structure (CAS) queries efficiently that consist of a path predicate and a value predicate. At the core of the RCAS index is a novel interleaving scheme, called dynamic interleaving, that interleaves paths and values at their discriminative bytes.

The goal of this project is to build the prototype of a search engine that connects the SWH-Archive with the RCAS index. Users of the search engine can enter their CAS queries and the system returns the revisions that match the query along with links to these revisions on the Software Heritage website where users can explore them in more detail. On the back-end the search engine integrates with the RCAS index to answer the CAS queries. To handle the scale of the SWH-Archive, a disk-based version of the RCAS index must be implemented [5].

Tasks:

- **Task 1: Literature Review**
  - Study the dynamic interleaving and the in-memory RCAS index proposed in [4], and the disk-based RCAS index in [5].
  - Read up on the relevant literature about the Software Heritage Archive [2, 3]. Familiarize yourself with the data model of the archive and its documentation [1].
- **Task 2: Disk-Based RCAS Index**
  - Implement the disk-based RCAS index from [5] in C++. The implementation must support the bulk-loading of the index and (b) the evaluation of CAS queries.
  - Feed the Software Heritage Archive dataset into your bulk-loading algorithm. Ingest as much data as possible into the index.
- **Task 3: Prototype of a Search Engine**
  - Develop a prototype of a search engine that allows users to answer CAS queries on the Software Heritage Archive.
  - The prototype must be web-based. On the server-side your prototype connects to your C++ implementation of the RCAS index to answer CAS queries.
  - The search results should link to the official Software Heritage Archive website where users can see the actual changes to file contents that were made in a revision.
- **Task 4: Writing the Final Report**
  - Summarize your work in a report. Describe, among others, the problem setting, the algorithms that you implemented, and the architecture of your prototype.

References

[1] Software Heritage. `https://docs.softwareheritage.org/devel/`. [Online; accessed July 2020].

[2] R. Di Cosmo and S. Zacchiroli. Software heritage: Why and how to preserve software source code. In *iPRES 2017: 14th International Conference on Digital Preservation*, 2017.

[3] A. Pietri, D. Spinellis, and S. Zacchiroli. The software heritage graph dataset: Large-scale analysis of public software development history. In *MSR 2020: The 17th International Conference on Mining Software Repositories*. IEEE, 2020.

[4] K. Wellenzohn, M. H. Böhlen, and S. Helmer. Dynamic interleaving of content and structure for robust indexing of semi-structured hierarchical data. *PVLDB*, 13(10), 2020.

[5] K. Wellenzohn, M. H. Böhlen, S. Helmer, and S. Zacchiroli. Robust content-and-structure indexing for large software archives. Work in progress.

Supervisor: Kevin Wellenzohn (wellenzohn@ifi.uzh.ch)

Start date: August 5, 2020

End date: August 4, 2021

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor