CodeDJ: Reproducible Queries over Large-Scale Software Repositories (Artifact)

Petr Maj[1]
Czech Technical University in Prague, Czech Republic

Konrad Siek[1]
Czech Technical University in Prague, Czech Republic

Alexander Kovalenko
Czech Technical University in Prague, Czech Republic

Jan Vitek
Czech Technical University in Prague, Czech Republic
Northeastern University, Boston, MA, USA

Abstract

Analyzing massive code bases is a staple of modern software engineering research – a welcome side-effect of the advent of large-scale software repositories such as GitHub. Selecting which projects one should analyze is a labor-intensive process, and a process that can lead to biased results if the selection is not representative of the population of interest. One issue faced by researchers is that the interface exposed by software repositories only allows the most basic of queries. CodeDJ is an infrastructure for querying repositories composed of a persistent datastore, constantly updated with data acquired from GitHub, and an in-memory database with a Rust query interface. CodeDJ supports reproducibility, historical queries are answered deterministically using past states of the datastore; thus researchers can reproduce published results. To illustrate the benefits of CodeDJ, we identify biases in the data of a published study and, by repeating the analysis with new data, we demonstrate that the study's conclusions were sensitive to the choice of projects.

2012 ACM Subject Classification Software and its engineering → Ultra-large-scale systems

Keywords and phrases Software, Mining Code Repositories, Source Code Analysis

Digital Object Identifier 10.4230/DARTS.7.2.13

Funding This work is supported by the Czech Ministry of Education, Youth and Sports from the Czech Operational Programme Research, Development, and Education, under grant agreement No.CZ.02.1.01/0.0/0.0/15_003/0000421 and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 695412).

Related Article Petr Maj, Konrad Siek¹, Alexander Kovalenko, and Jan Vitek, "CodeDJ: Reproducible Queries over Large-Scale Software Repositories", in 35th European Conference on Object-Oriented Programming (ECOOP 2021), LIPIcs, Vol. 194, pp. 6:1–6:24, 2021.
`https://doi.org/10.4230/LIPIcs.ECOOP.2021.6`

Related Conference 35th European Conference on Object-Oriented Programming (ECOOP 2021), July 12–16, 2021, Aarhus, Denmark (Virtual Conference)

1 Scope

The artifact produces all graphs, tables and numbers used in the paper using the Code DJ infrastructure presented in the paper. It also provides results of the case study presented in the paper.

[1] These authors contributed equally.

Dagstuhl Artifacts Series, Vol. 7, Issue 2, Artifact No. 13, pp. 13:1–13:4
Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
Dagstuhl Publishing, Germany

2 Content

The artifact package includes:

- Virtual machine with the artifact code and detailed instructions,
- Results of the queries performed for the paper,
- A toy dataset,
- The full dataset used for the paper.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `https://github.com/PRL-PRG/codedj-ecoop-artifact`.

4 Tested platforms

The VM is provided in the OVA format and has been tested in VirtualBox 6.1 on Linux (Ubuntu 20.04) and Windows (10 - 2104). Other platforms should be supported as well. The VM requires minimum of 8GB RAM. Username is `ecoop` and password is `ecoop`.

5 License

The artifact is available under the MIT license.

6 MD5 sum of the artifact

84f3bd083b289355e20112dd26cf5887

7 Size of the artifact

41 GiB

References

1 T. F. Bissyande, F. Thung, D. Lo, L. Jiang, and L. Reveillere. Orion: A software project search engine with integrated diverse software artifacts. In *International Conference on Engineering of Complex Computer Systems*, 2013. `doi:10.1109/ICECCS.2013.42`.

2 Roberto Di Cosmo and Stefano Zacchiroli. Software Heritage: Why and How to Preserve Software Source Code. *International Conference on Digital Preservation*, 2017. URL: `https://hal.archives-ouvertes.fr/hal-01590958`.

3 Robert Dyer, Hoan Anh Nguyen, Hridesh Rajan, and Tien N. Nguyen. Boa: A language and infrastructure for analyzing ultra-large-scale software repositories. In *International Conference on Software Engineering (ICSE)*, 2013. URL: `http://dl.acm.org/citation.cfm?id=2486788.2486844`.

4 Davide Falessi, Wyatt Smith, and Alexander Serebrenik. Stress: A semi-automated, fully replicable approach for project selection. In *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2017. `doi:10.1109/ESEM.2017.22`.

5 Jesus M. Gonzalez-Barahona, Gregorio Robles, and Santiago Dueñas. Collecting data about FLOSS development: The FLOSSMetrics experience. In *International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (FLOSS)*, 2010. `doi:10.1145/1833272.1833278`.

6 Georgios Gousios and Diomidis Spinellis. GHTorrent: GitHub's data from a firehose. In Michael W. Godfrey and Jim Whitehead, editors, *Working Conference on Mining Software Repositories (MSR)*, 2012. `doi:10.1109/MSR.2012.6224294`.

7 Crista Lopes, Petr Maj, Pedro Martins, Di Yang, Jakub Zitny, Hitesh Sajnani, and Jan Vitek. Déjà Vu: A map of code duplicates on GitHub. *Proc. ACM Program. Lang.*, (OOPSLA), 2017. `doi:10.1145/3133908`.

8 Hitesh Sajnani, Vaibhav Saini, Jeffrey Svajlenko, Chanchal K. Roy, and Cristina V. Lopes. Sourcerercc: scaling code clone detection to big-code. In *International Conference on Software Engineering (ICSE)*, 2016. doi:10.1145/2884781.2884877.

A Related Work

Our paper has been inspired and improves upon the following tools:

Stress: This system aims to help choose projects in a reproducible manner [4]. Its corpus consists of 211 projects which can be filtered on 100 pre-computed attributes such as bug tickets or lifetime. The corpus can be sorted and sampled randomly. Queries can be exported so they can be repeated later. Source code is not available for querying. Stress is inactive. `CodeDJ` scales to larger corpora and allows to specify richer queries. In terms of reproducibility, we support updates to the corpus.

Flossmetrics: This work analyzed 2800 open source projects and computed statistics about various aspects of their development process, such as number of commits and developers [5]. Information from additional sources such as project mailing lists and issue trackers was included. Queries could be formulated on metrics such as COCOMO effort, core team members, evolution and dynamics of bugs. Filtering based on these criteria was supported. The project is inactive and it did not support updates.

Orion: This system aimed to enable retrieving projects using complex search queries linking different artifacts of software development, such as source code, version control metadata, bug tracker tickets, developer activities and interactions extracted from the hosting platform [1]. The project is no longer maintained, it scaled to about 185K projects. `CodeDJ` is designed to scale to larger corpora and offers a more flexible query interface.

Boa: This system focuses on semantics queries over Java programs [3]. A corpus of 380K Java projects can be queried using a dedicated query language that supports automatic parallelization and pluggable mining functions. Source code can be queried in sophisticated ways as Boa is able to parse and analyze Java. A larger corpus of 7.5M projects can be queried on project summaries. Boa provides reproducibility by ensuring its queries are deterministic with respect to the dataset's version, which are created and archived infrequently (i.e. 2013, 2015, 2019, 2020). `CodeDJ` differs from Boa in that it is language agnostic and geared towards project selection, as opposed to project analysis. Furthermore, `CodeDJ` provides full reproducibility in the presence of a continuously evolving dataset.

Black Duck Open Hub: A public directory of open source software[2] that offers search services for discovering, evaluating, tracking, and comparing projects. It analyzes both the code's history and ongoing updates to provide reports about the composition and activity of code bases. `CodeDJ` allows researchers to write their own queries and supports reproducibility.

SourcererCC: The aim of this project is to detect code clones [8]. The tool scales to large datasets and can detect near-identical code at various granularities. It has been used to analyze cloning across large corpora of Java, JavaScript, Python, C and C++ projects on `GitHub` [7]. It

[2] https://www.openhub.net

can be used by researchers to detect duplication in their samples which is a source of bias. The project's web page appears to be inactive.

GHTorrent: This database of metadata about GitHub projects offers an SQL interface for queries [6]. It monitors GitHub events to constantly update the available data. The limitation of the approach is that GitHub's events do not have all commit details and file contents, thus these are not stored by GHTorrent. In our experience, the database is not always consistent, this may be due to missed events. We have attempted to upload queries through the public SQL interface but the queries timed out.

GitHub: This service provides two ways to query metadata and contents. A REST API can be used for requesting information about projects and listing them, its search queries provide filtering capabilities across a small set of fixed attributes. A web API provides extended filtering options such as searching within repositories written in a particular language. These interfaces are rate-limited and thus return partial results. The results are non-deterministic and non-reproducible as projects may be added and deleted at any time. CodeDJ provides a view of a subset of GitHub on which we support reproducibility and our queries are richer and deterministic.

We would be remiss if we failed to mention the Software Heritage Archive which aims to preserve all publicly available source code; currently upwards of 9.5B source files, 2B commits and 150M projects [2]. It only allows retrieval of single objects. The authors point to the fragility of current arrangements and the dynamic nature of source code repositories makes it difficult to reproduce studies that use them. We have encountered this ourselves: we see projects deleted from GitHub, changing names, or visibility. In the future, CodeDJ can be extended to query the heritage corpus as well as other repositories.