

Effects of Centralized and Distributed Version Control on Commit Granularity

Jochen Wuttke^{*}, Ivan Beschastnikh^{*}, and Yuriy Brun[†]

^{*}University of Washington

[†]University of Massachusetts

{wuttke, ivan}@cs.washington.edu, brun@cs.umass.edu

ABSTRACT

Version control systems are critical for coordinating work in large software engineering teams. Recently, distributed version control (DVC) systems have become popular, as they have many advantages over their centralized (CVC) counterparts. DVC allows for more frequent commits, and simplifies branching and merging. These features encourage developers to make smaller, finer-grained commits that do not interleave changes related to different development tasks. Such commits improve accountability and ease certain tasks, such as reverting changes that later cause problems.

DVC systems are also better suited for repository mining techniques, making available more useful information about the development process [2]. For example, approaches that infer collaboration patterns can benefit from the more detailed attribution of data in DVC. This can be used by an integration server to send email about failed test cases to just the subset of developers who authored the relevant code. DVC may also lead to smaller and more focused commits, which could benefit mining techniques that identify changes relevant to specific development tasks, such as refactorings [3].

However, to date, there has been no explicit evaluation of the practical differences in mining DVC over CVC, though some work acknowledges that developers might use DVC and CVC differently [1]. We report on such an evaluation with one counterintuitive finding that raises doubts about certain DVC promises and opens research questions into what causes DVC and CVC differences. Further, our finding indicates that repository type should be controlled for in repository mining experiments.

BODY

In a study of six CVC and six DVC repositories we found that the median size of code commits in DVC is 38% larger than in CVC.

REFERENCES

- [1] E. T. Barr, C. Bird, P. C. Rigby, A. Hindle, D. M. German, and P. Devanbu. Cohesive and Isolated Development with Branches. In *FASE*, 2012.
- [2] C. Bird, P. Rigby, E. Barr, D. Hamilton, D. German, and P. Devanbu. The Promises and Perils of Mining Git. In *MSR*, pages 1–10, 2009.
- [3] M. Kim, D. Cai, and S. Kim. An Empirical Investigation into the Role of API-Level Refactorings during Software Evolution. In *ICSE*, pages 151–160, 2011.

Volume 1 of Tiny Transactions on Computer Science

This content is released under the Creative Commons Attribution-NonCommercial ShareAlike License. Permission to make digital or hard copies of all or part of this work is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

CC BY-NC-SA 3.0: <http://creativecommons.org/licenses/by-nc-sa/3.0/>.