

Abstract: Streaming APIs allow for big data processing of native data structures by providing MapReduce-like operations over these structures. However, unlike traditional big data systems, these data structures typically reside in shared memory accessed by multiple cores. Although popular, this emerging hybrid paradigm opens the door to possibly detrimental behavior, such as thread contention and bugs related to non-execution and non-determinism. This study explores the use and misuse of a popular streaming API, namely, Java 8 Streams. The focus is on how developers decide whether or not to run these operations sequentially or in parallel and bugs both specific and tangential to this paradigm. Our study involved analyzing 34 Java projects and 5.53 million lines of code, along with 719 manually examined code patches. Various automated, including interprocedural static analysis, and manual methodologies were employed. The results indicate that streams are pervasive, stream parallelization is not widely used, and performance is a crosscutting concern that accounted for the majority of fixes. We also present coincidences that both confirm and contradict the results of related studies. The study advances our understanding of streams, as well as benefits practitioners, programming language and API designers, tool developers, and educators alike.

Raffi Khatchadourian: is an Assistant Professor in the Department of Computer Science at Hunter College and a member of the Doctoral Faculty of The Graduate School and University Center's Ph.D. Program in Computer Science of the City University of New York (CUNY). His current research interests lie in software engineering and programming languages, particularly in regards to automated software evolution, with the goal of easing the burden associated with correctly and efficiently evolving large and complex software. He has served as a program committee member for ECOOP, OOPSLA's Onward!, ASE's LBR, and ICSE's and ESEC/FSE's demo tracks. He has also co-organized the ACM Symposium for the Foundations of Software Engineering and the European Conference on Object-Oriented Programming. He received his MS and PhD in Computer Science from Ohio State University and his BS in Computer Science from Monmouth University. Prior to joining CUNY, he was a software engineer at Apple, Inc. in Cupertino, California, where he worked on Digital Rights Management (DRM) for iTunes, iBooks, and the App store. He also helped develop distributed software that tested various features of iPhones, iPads, and iPods.