SCIENCES & BIOENGINEERING SCIENCES

The Research Group

Software Languages Lab

has the honor to invite you to the public defense of the PhD thesis of

Matteo Marra

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:
A live debugging approach for Big Data processing applications

Promotor:
Prof. dr. Elisa Gonzalez Boix (VUB)

Co-promotor:
Dr. Guillermo Polito (Université de Lille)

The defense will take place on
Tuesday, May 3, 2022 at 17h in auditorium I.2.03

Please contact Matteo Marra at mmarra@vub.be if you want to join the presentation through Microsoft Teams

Members of the jury

Prof. dr. Geraint Wiggins (VUB, chair)
Prof. dr. Bas Ketsman (VUB, secretary)
Prof. dr. Jan De Beule (VUB)
Prof. dr. Guido Salvaneschi (University of St. Gallen)
Prof. dr. Luc Fabresse (IMT Nord Europe)
Prof. dr. Miryung Kim (University of California)

Curriculum vitae

Matteo Marra obtained his BSc at the University of Bologna in 2015 and his master at the Vrije Universiteit Brussel (VUB) in 2017. He then started a PhD at the Software Languages Lab (SOFT) supported by an FWO-SB fellowship and cooperating closely with the RMoD research lab at INRIA Lille Nord Europe.

His research has mainly focused on advanced debugging techniques for Big Data processing applications to help developers solve errors in highly parallel and remotely executed applications. Matteo's research resulted in four publications in international peer-reviewed journals and conferences, and four contributions in international peer-reviewed workshops.

Abstract of the PhD research

The modern world heavily relies on data: in 2020, more than 64 ZBs of data were created, captured, copied, or consumed globally. As a result, novel software platforms have emerged to analyze large data sets from several domains in a parallel and scalable way. The two most prominent programming models for Big Data processing are Map/Reduce and Apache Spark. Both models envision the programming of complex problems through well-known functions and let the framework take care of the distribution aspects such as parallelization and fault tolerance to node failures.

Debugging Big Data applications is difficult due to their distributed and parallel nature, which increases the distance between the root cause of the bug and the observed failure. Furthermore, developers tend to use a large technology stack, which also complicates the debugging. A common debugging practice is to analyze log files, but they lack contextual information about which record(s) caused an error. Recently, Record \& Replay debuggers have been explored, but replaying Big Data applications can be very costly since they are normally long-lasting. Checkpoint-based debugging has been explored to lower the replay time, but still requires the creation of a checkpoint and a replay step.

In this dissertation, we explore a live debugging approach tailored to Map/Reduce and Spark-like programs. We first propose out-of-place debugging, a novel debugging architecture to debug remote and distributed applications. In this model, when there is an error in an application running remotely (e.g., in a cluster), the state of the computation is transferred to the developer's machine, in which the application can be debugged. This avoids replaying the execution while offering a full interactive debugging environment.

We then explore the applicability of out-of-place debugging to parallel distributed Map/Reduce and Spark-like applications, through two novel techniques for optimizing their debugging: composite debugging events, i.e., the grouping and centralized debugging of multiple similar debugging events, and dynamic local checkpoints, i.e., dynamic capturing of the execution state. Thus, we enable centralized debugging of remote Big Data applications and extend it with domain-specific debugging operations. Finally, we complement our debugging approach with a relaxed computational model that allows developers to instruct the runtime to automatically ignore a defined number of exceptions that happen at runtime. This feature is especially relevant for those data analytics applications that can accept a loss in accuracy (e.g., because of dirty data).

We implement our debugging techniques in Pharo Smalltalk on top of Port and Spa, our frameworks implementing the Map/Reduce and the Spark-like model, respectively. Furthermore, we generalized all the call-stack operations needed to enable our debugging approach in Sarto, a call-stack instrumentation layer for stack tailoring. The proposed out-of-place debugging approach applied to debug Map/Reduce and Spark-like programs, together with Sarto, represent the main contributions of this dissertation.

Our validation is two-fold: we validate our debugging approach quantitatively and qualitatively. For the quantitative study, we conducted performance benchmarks that show that our model scales to an increasing amount of both data and parallel exceptions. For the qualitative study, we conducted a user study to assess the usability of our solution for solving different debugging tasks and compare it to a reproduction of a state-of-the-art debugger for Spark applications. The results show that participants reported a better debugging experience using our debugger and validated positively the advanced features offered by our debugger.