Foreword: Special Issue on Repeatability and Sharing of Experimental Artifacts

Eric Eide

University of Utah, School of Computing
Salt Lake City, UT USA

eeide@cs.utah.edu

Welcome to this special issue of *Operating Systems Review* dedicated to the topics of repeatability and sharing of experimental artifacts in systems research. The twelve papers in this issue present a snapshot of activities, results, and viewpoints on this theme. As the guest editor of this issue, I hope that you find these papers to be both inspiring and useful. I hope that by reading them, you will be motivated to carefully think about repeatability and artifact sharing as you pursue your own systems experiments. Several of the papers in this special issue present tools or testbeds that support repeatability and sharing, so I expect that you will discover one or more “new tricks” that will make it easier for you to create and share high-quality, sound, and repeatable experiments!

Motivation

As highlighted by this special issue, there is increasing interest in promoting repeatable and reproducible research in computer science. A community gains confidence in scientific claims when the experiments supporting those claims are published, examined, and repeated. Repeating research also helps a community expand on previous results and discover new directions for investigation. Repetition requires that the artifacts used in an experiment be made available to, and made usable by, investigators. In other words, artifacts must be shared. Possible artifacts include not only binaries, source code, and datasets but also computing and networking infrastructure such as testbeds and clouds.

Systems researchers face special challenges in the areas of repeatability and artifact sharing. Some of these challenges arise from the nature of systems software, which is typically complex and rife with internal and external dependencies. Other issues arise from execution environments. For example, to reproduce performance and scalability results, one may need access to actual and specific hardware. Still more systems-specific challenges arise from the difficulty of measuring properties of interest, such as time and energy consumption.

Preparation

Jeanna Matthews, chair of ACM SIGOPS and editor of *Operating Systems Review*, suggested the idea of a special *OSR* issue that would address the challenges of repeatable systems research. She recruited me to serve as the guest editor, and in June 2014, we posted the call for contributions.

The stated goal was to present a selection of high-quality papers about the state of repeatable research in computer systems and the sharing of experimental artifacts related to systems research. The call solicited both technical papers and position papers related to these ideas. The suggested topics for submissions included:

- novel ways to manage systems research artifacts, dependencies, workflows, and data collection;
- testbeds for repeatable research and/or publishing executable artifacts;
- “rich publications” that connect papers to software and data;
- repeatability in distributed and nondeterministic systems;
- mitigating obstacles to artifact sharing, repeatability, and reproduction;
- repeatability challenges specific to systems;
- measurement bias in systems experiments;
- experiences and outcomes from systems repetition and reproduction studies;
- improving scientific practice in systems research; and
- incentives for producing and sharing high-quality research artifacts.

The review committee received 21 submissions, and each was assigned to three committee members for review. The guest editor (yours truly) declared a conflict of interest with two of the submissions, and for those, another member of the committee volunteered to chair the review process. That person chose the reviewers of those papers, and the reviewers’ identities were not revealed to the guest editor. In total, 63 reviews were written, many with long and detailed comments.

The review committee discussed the papers at length via the online paper-review site and email. Submissions were evaluated on technical quality, originality, relevance, and presentation. Ultimately, the committee selected twelve for publication in this special issue.

Selected Papers

The first paper, by Dror G. Feitelson, introduces the topics of this special issue and clarifies the meanings of terms such as “repeatability” and “reproducibility.” Although these words are often used interchangeably and inconsistently, Feitelson suggests that we can communicate more effectively if we distinguish between them.

The second paper, by Heidi Howard et al., details the authors’ experience in replicating a previous study of Raft, a distributed consensus protocol. A new Raft implementation allowed Howard and her colleagues to not only validate the results of the original Raft study, but also recommend several optimizations to the protocol.

In the third paper, Mayank Varia et al. detail their experience with repeatable research and the automated assessment of secure search systems. They describe the framework they created for performing rigorous test and evaluation (T&E) of multiple implementations of a secure search system, and they distill their experience into a set of mantras for software T&E systems in general.

Abraham N. Aldaco et al. then describe “locating arrays,” which are an approach to designing screening experiments for complex

engineered systems. It is often infeasible to test every combination of a system's input values to find the variables (factors) that most affect the system's behavior. Locating arrays provide a solution: they allow a researcher to design an experiment, containing only a small number of design points, that can find the factors and interactions that most influence the system's behavior.

Ali Abedi et al. continue the discussion of experiment methodology by describing the use of "multiple interleaved trials" within their study of 802.11n MIMO networks. They investigate the extent to which 802.11n MIMO network experiments can be repeated. When using existing methodologies, Abedi et al. found that they were not able to obtain repeatable results even under controlled WiFi conditions. Using multiple interleaved trials, they were able to obtain repeatable results.

Next, Christian Dietrich and Daniel Lohmann consider the issue of "internal repeatability"—being able to repeat your own experimental results—and present two tools they developed to improve their own research practices. They describe the benefits they obtained from these tools within the VAMOS project, which examined compile-time configurable features in Linux, and within the DanceOS project, which investigates software-based protections against transient hardware faults.

Luka Stanisic et al. then present a workflow for conducting experiments in a reproducible way using two well-known tools: Git and Org-mode. The authors describe how their methodology helped them to perform two studies in the high-performance computing (HPC) domain.

Continuing the discussion of tools, Carl Boettiger details the usefulness of Docker for performing reproducible experiments in systems. Docker is a popular, open-source framework for defining and creating lightweight "containers" based on OS virtualization. Boettiger describes how Docker can help researchers overcome barriers to reproducible experiments including "dependency hell," poor documentation, code rot, and barriers to adoption.

Next, Cristian Ruiz et al. report their experience in developing Kameleon, a system for building software appliances. Kameleon is designed to ease the construction—and reconstruction!—of software stacks for research in HPC, Grids, cloud computing, and distributed systems. Notably, Kameleon implements a modular "recipe" language for defining software appliances, which is meant to encourage collaboration and reuse.

The tenth paper, by Sarah Edwards et al., turns our attention to testbeds. Edwards and her colleagues present a methodology for designing and deploying repeatable experiments on network testbeds, with particular focus on the GENI testbed. The paper illustrates best practices through a case study; by following the best practices, a researcher can reduce the effort and time required to deploy effective experiments.

The eleventh paper, by Robert Ricci et al., continues the discussion of testbeds for repeatable research. Ricci et al. argue that network testbeds can ease the "packaging" of many research artifacts by recreating the environments in which those artifacts are deployed, executed, and evaluated. Apt, the Adaptable Profile-driven Testbed, is their facility that embodies this approach toward enabling repeatability and sharing.

Finally, Bruce R. Childers et al. conclude this issue with a paper that presents recommendations for creating and sharing interoperable simulators and modeling artifacts for research in computer architecture. The recommendations are the result of a workshop, including participants from academia, industry, and government, that sought to improve research in this area through greater artifact sharing and collaboration.

Acknowledgments

This special issue would not have been possible without the strong support of Jeanna Matthews, the meticulous work of the review committee, and the great efforts of the authors who submitted their work to this *OSR*. Thank you all!

Review Committee

- Jeannie Albrecht, Williams College
- Tim Brecht, University of Waterloo
- Amer Diwan, Google
- Eric Eide, University of Utah (*chair*)
- Sebastian Fischmeister, University of Waterloo
- Jelena Mirkovic, USC Information Sciences Institute
- Kiran-Kumar Muniswamy-Reddy, Amazon.com
- Victoria Stodden, University of Illinois at Urbana-Champaign
- Ryan Stutsman, Microsoft Research
- Peter F. Sweeney, IBM Research
- Michael Zhivich, MIT Lincoln Laboratory

Additional Reviewers

- Amir Ghods, University of Waterloo
- Augusto Born de Oliveira, University of Waterloo
- Jean-Christophe Petkovich, University of Waterloo