Position paper presented at *Archive '10, the NSF Workshop on Archiving Experiments to Raise Scientific Standards*, Salt Lake City, UT, May 2010.

Toward Replayable Research in Networking and Systems

Eric Eide

University of Utah, School of Computing

eeide@cs.utah.edu

Abstract

The increasing use of data repositories, testbeds, and experiment-management systems shows that the networking and systems research communities are moving in the direction of repeatability. We assert, however, that the goal of these communities should not be repeatable research, but *"replayable" research.* Beyond encapsulating the definition and history of an experiment, a replayable experiment is associated with a mechanism for actually re-executing a system under test. In this paper, we outline the challenges to be overcome in building an archive of replayable experiments in computer networking and systems research.

1. Introduction

Although the benefits of repeatable research have long been known to the scientific community, it is only recently that computer scientists have started to articulate the importance and value of repeatable research in their domain. Consider, for example, recent activities within the networking and distributed systems communities:

- In the networking community, researchers have moved toward the creation of **data repositories** (e.g., [2, 5, 13]): facilities built for publishing and archiving the data sets that support scientific conclusions. These allow researchers to repeat experimental analyses, but they do not help in validating or repeating data-collection processes.
- Concurrently, the community has been creating a variety **network testbeds** (e.g., [11, 12, 15]): environments that provide resources for scalable and "real-world" experimentation. In contrast to data repositories, testbeds have focused on supporting control and therefore repeatability (e.g., Emulab) or on deployment of applications on production networks (e.g., PlanetLab). Actually *achieving* a high degree of control or a large-scale deployment, however, typically requires significant human effort. Testbed interfaces have been designed around resource access, and mostly leave users responsible for packaging, automating, and publishing their experiments and results.
- Researchers in distributed and networked systems are creating specialized **experiment-management systems** (e.g., [1, 6, 14]). that help experimenters use testbeds more effectively. An experiment-management system may help users to create experiments quickly through templates or code generation; automate important steps, such as software deployment, data collection, and data analysis; provide interactive experiment monitoring and steering features; and/or help users keep track of their activities over long periods of time.

Although each of these areas is an active topic of research in and of itself, many instances of these types of systems are now in common and growing use. Data repositories, network testbeds, and experiment-management systems are moving the systems community forward in terms of supporting repeatable research.

We assert, however, that the standard of the systems community should not be repeatable research. We claim instead that our goal should be to support what we call **"replayable" research**. A replayable experiment not only encapsulates the definition of a computer-based activity, but in addition, is associated with a facility for actually re-executing that activity.

2. Repeatable and Replayable

In a scientific community, experiments are repeated in exact or modified form, and usually by many people, to develop increasingly accurate hypotheses that predict the behavior of systems under study. At a minimum, the repeatability of an analysis requires the collection, dissemination, and preservation of the "raw" input data, and also a thorough definition of the analysis itself. The repeatability of a data-collection activity—i.e., reproducing raw data—requires that the activity be defined rigorously and that the system under observation be repeatable. In the networking and systems communities, the systems under study may or may not be inherently repeatable. Often, a complex computer system under study will combine parts that are deterministic (e.g., ordinary computer programs) with parts that are not (e.g., live Internet traffic).

Capturing the activity. In the computer science field, repeatable research can be a surprisingly difficult challenge even for individual researchers. Vern Paxson described the situation faced by someone who needs to reproduce his or her own work after a break [10]:

> "It is at this point—we know personally from repeated, painful experience—that trouble can begin, because the reality is that for a complex measurement study, the researcher will often discover that they cannot reproduce the original findings precisely! The main reason this happens is that the researcher has now lost the rich mental context they developed during the earlier intense data-analysis period."

These problems are compounded when an experiment is to be shared among many researchers, or formally published to an even wider audience. In these cases, the "rich mental context" must be explicitly communicated in order for the experiment to be comprehended and repeated. The context that must be captured and saved includes not only the artifacts that are part of the experiment—e.g., the source code and/or binaries of the software under test, input files, and so on—but also information about the purposes of these things, the results of previous experiments, and the history of the artifacts and the trials that have been performed.

Thus, as Paxson points out, it can take a significant work to ensure that an analysis is repeatable. That work is compounded when the work is to be shared. In the computer-science community, which lacks a long history of valuing repeated research, this work has often not been seen as worthwhile.

However, there are signs that attitudes in computer science are changing. In the areas of networked systems and operating systems, for example, there is an increasing awareness of the potential bene-

fits of repeatable research [4, 8]. Another sign of changing attitudes is found in recent computer-science research conferences, which have instituted repeatability requirements [7]. A third sign is found in the recent NSF announcement that future research proposals will need to include explicit data-management plans [9].

The networking community is already shifting to experiments based on testbeds, driven by experiment-management systems, and interacting with repositories. All of these provide features for promoting repeatable research, but "packaging" such research still requires too much human effort. A primary goal of these systems in the current decade should be to lessen the manual effort needed to package research artifacts for inspection and reuse—by the public, students, colleagues, and by authors themselves.

From repeatable to replayable. When the initial configurations, inputs, and results of a computer network can be reliably captured as described above, then it becomes possible to create *open, community archives* of experiments and results in networking (and other fields as well). Given suitable infrastructure, researchers and educators could package their experiments easily and make them available to the community at large. This activity would enable new research and education activities by making well-defined "units" of research available to a broad community.

However, we believe that "mere" archives of packaged experiments are not enough. A packaged experiment by itself is like an audio recording: for a recording to be useful, one must have a means for playing it. Similarly, a packaged experiment is only truly useful when it is accompanied by a mechanism for *actually* re-executing it, both in its packaged form and in modified forms.

A packaged experiment is **replayable** if it is accompanied by or associated with a mechanism for re-executing the activities described by the experiment. We believe the goal of supporting replayable research for computer networking and systems is now coming within reach, through a combination of the testbed, repository, and experiment-management technologies described above.

3. Challenges

Achieving replayable experimentation in networking and systems will require solutions to a number of problems, including issues such as the handling of intellectual property. (E.g., how can one publish a replayable experiment that includes commercially licensed software?) Below, we highlight three that will require new solutions to technical problems.

The first concerns the availability of replay infrastructure for nondeterministic orderings and external events. Networked systems typically involve a certain amount of nondeterminism, and some types of experiment data—such as live Internet traffic—are inherently unrepeatable. These issues are being addressed through system-level recording mechanisms that capture events in a lightweight fashion (e.g., [3]). These mechanisms reproduce the recorded events at the correct points during a replay. An ideal infrastructure for replayable research would allow an experiment not only to be perfectly replayed, but also *modified* and then executed. Thus, beyond ensuring fidelity during an exact replay, a significant challenge for an event-replay mechanism is to produce "reasonable behavior" during modified replays.

The second issue concerns long-term access to physical, experimental resources. Network testbeds like Emulab provide users with access to "bare metal" machines—not just virtual machines—for two reasons. First, many researchers explicitly or implicitly assume that the systems they study will ultimately be deployed to physical machines. Thus, the behavior that matters is the behavior on a network of physical machines. Second, Emulab does not offer high-fidelity virtual equivalents of specialized hardware such as network switches and embedded-system devices. Ultimately, however, physical devices wear out and are replaced by newer models—or are not replaced at all! To ensure that already-packaged experiments continue to be replayable, a possible long-term solution is for a testbed to offer virtual equivalents to all of its hardware. Unfortunately, this may be exceedingly effort-intensive.

The third issue is to more carefully define the notion of replayability, and to expose the attributes of replayability to the authors and consumers of packaged experiments. We believe that instead of seeking perfect replayability, the community should expect that experiments will be replayable *with respect to certain properties only*. One might say that this expectation exists today—nobody expects perfect replayability! Our point, however, is that today, reproducibility expectations are communicated implicitly at best. We claim that in the future, the authors of experiments should be able to explicitly choose properties of interest. Future experimental infrastructure should explicitly support such decision-making.

The explicit specification of reproducibility properties can guide solutions to the first two issues mentioned above. For the first, during the initial run of an experiment, a testbed can choose recording mechanisms to support a user-chosen level of replay fidelity, along with models for producing reasonable behavior in the face of changes during replay. For the second, a testbed can potentially determine when virtual machines may be an acceptable basis for an experiment: both initially, and during future replays.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. 0524096 and 0709430.

References

[1] J. Albrecht, C. Tuttle, A. C. Snoeren, and A. Vahdat. PlanetLab application management using Plush. *OSR*, 40(1):33–40, Jan. 2006.

[2] D. G. Andersen and N. Feamster. Challenges and opportunities in Internet data mining. Technical Report CMU–PDL–06–102, Carnegie Mellon University Parallel Data Laboratory, Jan. 2006.

[3] A. Burtsev et al. Transparent checkpoints of closed distributed systems in Emulab. In *EuroSys*, pages 173–186, Apr. 2009.

[4] B. Clark et al. Xen and the art of repeated research. In *FREENIX*, pages 135–144, June–July 2004.

[5] Cooperative Association for Internet Data Analysis (CAIDA). DatCat. `http://www.datcat.org/`.

[6] E. Eide, L. Stoller, and J. Lepreau. An experimentation workbench for replayable networking research. In *NSDI*, pages 215–228, Apr. 2007.

[7] I. Manolescu et al. The repeatability experiment of SIGMOD 2008. *SIGMOD Record*, 37(1):39–45, Mar. 2008.

[8] J. N. Matthews. The case for repeated research in operating systems. *OSR*, 38(2):5–7, Apr. 2004.

[9] NSF. Scientists seeking NSF funding will soon be required to submit data management plans. Press Release 10–077, May 10 2010.

[10] V. Paxson. Strategies for sound Internet measurement. In *IMC*, pages 263–271, Oct. 2004.

[11] L. Peterson et al. A blueprint for introducing disruptive technology into the Internet. In *HotNets*, Oct. 2002.

[12] D. Raychaudhuri et al. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *WCNC*, pages 1664–1669, Mar. 2005.

[13] C. Shannon et al. The Internet Measurement Data Catalog. *CCR*, 35 (5):97–100, Oct. 2005.

[14] Y. Wang et al. Automating experimentation on distributed testbeds. In *ASE*, pages 164–173, Nov. 2005.

[15] B. White et al. An integrated experimental environment for distributed systems and networks. In *OSDI*, pages 255–270, Dec. 2002.