Computational Framework for Reproducibility and Generalizability

Juliana Freire and Dennis Shasha

June 28, 2010

Ever since Francis Bacon, a hallmark of the scientific method has been that experiments should be described in enough detail that they can be repeated and perhaps generalized. This implies the possibility of repeating results on nominally equal configurations and then generalizing the results by replaying them on new data sets, and seeing how they vary with different parameters. In principle this should be easier for computational experiments than for natural science experiments, because not only can computational processes be automated but also computational systems do not suffer from the ”biological variation” that plagues the life sciences. Unfortunately, the state of the art falls far short of this goal. Most computational experiments are specified only informally in papers, where experimental results are briefly described in figure captions; the code that produced the results is seldom available; and configuration parameters change results in unforeseen ways.

Recently, there has been a renewed interest in the publication of well-documented, reproducible results [6, 9, 12–14, 17, 21]. Since SIGMOD 2008 (when Shasha was program chair), the conference has encouraged authors of accepted papers to submit their code and tools for evaluation for both repeatability (can the experimental graphs in the paper be reproduced?) and workability (do slight variants of the experiments give reasonable results?). Participation rates have been high (a majority of papers that have no intellectual property conflicts) and a vote at the SIGMOD 2008 conference indicated that the vast majority of researchers would participate in a repeatability experiment if it were not too difficult and there were no intellectual property conflict.

However, a major roadblock to a more widespread adoption of this practice is the fact that it is hard both to derive a compendium that encapsulates all the components (e.g., data, code, parameter settings) needed to reproduce a result and to verify the results. For the latter, reviewers may need to replicate the environment used in the experiments, which can be challenging and sometimes impossible. The goal of this proposal is to greatly simplify the process of preparing and testing software for repeatability. We will create a system that helps *authors* create of reproducible results by allowing them to create workflows that encode the computational processes that derive the results (including data used, configuration parameters set) and connecting these to publication where the results are reported. In addition, the system will track software versions and dependencies, and support encapsulation of software into virtual machines as well as for remote access. We will provide tools for *testers* to enable them to repeat and validate results, ask questions anonymously, and modify experimental conditions. Those tools will carry over to subsequent uses in which a researcher wants to build on the work of another. As an active case study, we will apply these tools to the SIGMOD 2011 and SIGMOD 2012 repeatability and workability efforts. In the future, we foresee the use of these tools within other conferences and communities as well as providing a general mechanism underlying reproducible scientific publications.

1 Vision

Imagine a future where a reviewer of a computer science, bioinformatics, or computational physics paper reads about an algorithm and its accompanying zero-copyright software. The software promises a substantial improvement over the state of the art. The reviewer likes the paper, but thinks the experiments should have used more data or a different data distribution. The reviewer calls up a virtual machine encapsulating the software implementation of the algorithm and its system dependencies, varies the data input, and reruns the experiment. Observing surprisingly good performance, the reviewer rates the paper highly. A second reviewer tries the precise experiments reported in the paper on different generations of hardware and also finds results consistent with the paper's claims. A third reviewer changes the workflow provided by the authors to make the software applicable to a different problem and encounters an error. This third reviewer communicates with the author anonymously along with an error message and the author tells this reviewer how to change a configuration file to make the system work for the new problem. After the paper is published, a member of the scientific community reads it and and has an idea for a new algorithm. She subsequently publishes a paper that describes the new algorithm. In the new paper, the experimental comparisons between the original and new approaches link to the actual code, workflows, and data, and can be re-used and reproduced by others.

On a second paper for which the authors have provided code to be used by the community, some of the standard data sources (e.g., gene annotations) have been updated. A researcher using that code is able to incorporate the new version of the data sources easily by modifying the workflow. The researcher also modifies the workflow to use alternative algorithms over the old and new data sets.

2 Our Approach

For the last three years, ACM SIGMOD has engaged in a repeatability and workability initiative. Many of the lessons we have learned from that initiative (described in two issues of the SIGMOD Record) apply to computational analyses across computer science and even natural and social science. At a recent workshop (Archive 2010) in which the PIs participated, researchers from computing, physics, and biology discussed the issues of repeating and archiving computational processes. Virtually all affirmed that in their sub-specialties repeating computational experiments of other groups was not possible.

The universal technical challenge is that achieving repeatability takes a lot of work with current tools. Thus, the technical challenge is to make this easier for (i) the author of the software, (ii) the reviewer of the software, and, if the author is willing to disseminate code to the community, (iii) the eventual user of the software. Making this easier entails, at a minimum, conceiving of the outputs as the product of a computational process in which data inputs (and their versions) are made explicit. The process may evolve over the course of an investigation, so any tool we develop must be easy to deploy at different stages of a research project. In addition, to help reviewers, it is important that individual results reported in a paper (e.g., different plots) be linked to their provenance—the steps followed to derive a result and that can be used to reproduce the result.

We propose to leverage and extend the infrastructure provided by provenance-enabled scientific workflow systems (SWS) as the basis for creating and evaluating repeatable results. SWS (see e.g., [4, 7, 10, 15, 16, 19, 20, 22–24]) have emerged in the scientific community as a means to automate repetitive processes and capture complex analysis processes at various levels of detail and systematically capture provenance information for the derived data products [1–3, 8]. Although existing SWS are used to automate repetitive processes, these processes are often repeated by the creators of the workflows and within the same environment. The shipping of a workflow to be run in an environment different from the one it has been designed at raises many challenges. From hard-coded locations for input data, to dependencies on specific version of software libraries and hardware, adapting these workflows to run on a new environment can be challenging and sometimes impossible. In this project, we will investigate different approaches to packaging workflows for publication.

Supporting the SIGMOD Repeatability Evaluation Process. As a first step toward our goal to build a general infrastructure to support software publication, evaluation and re-use, as a case study, we will implement a system to support the SIGMOD 2011 repeatability evaluation. We will divide the experiments into two classes: those using common settings (e.g., common operating systems running on single machines or small clusters) and those using uncommon settings (e.g., vast or unusual computational resources). The repeatability procedure will differ for the two classes.

For experiments in the common setting class, the author can make code and data available to a reviewer. For experiments in the unusual setting class, where it is not possible to reproduce the complete workflow in a different environment, the author will either make intermediate data (together traces of the primary execution) available to a reviewer along with post-processing tools or will offer a programmable interface to allow a reviewer to run experiments on the author's system. Regardless of the nature of the experiment, the author should also list which parameter settings can change and how they can change. This will allow the reviewer to vary the experiments in many ways.

To encode workflows, authors will be encouraged to use the NSF-funded, open-source VisTrails system (http://www.vistrails.org). After a preliminary evaluation of different SWS by Phillipe Bonnet, the chair of the SIGMOD 2011 repeatability program, VisTrails was selected for being easy to install, use, and its support of several features required for the repeatability evaluation, including: the ability to perform parameter sweeps, the support for provenance, and the simplicity to add new code and modules. Although VisTrails has been successfully used for simulation, data analysis and visualization, and it does have some support for adapting workflows to run in different environments [11], it lacks the necessary infrastructure for submitting software and testing for repeatability. Our goal in this project is to build such an infrastructure. We will address two key problems:

1. *Lower Barrier for Adoption:* An important goal for the proposed infrastructure is to help authors in the process of assembling their submissions. As such, we will design mechanisms that makes this as seamless as possible. For example, we will investigate the possibility of converting a makefile or other scripts used into a workflow either automatically or semi-automatically. Besides submitting workflows, to help in the reviewing process, it is useful for the authors to indicate the parameters that can vary (e.g., inputs provided certain formatting conventions are followed, numerical parameters, reporting options) as well as values that can be used. We plan to re-use the interface from VisMashups [18] support this step. In addition, if the authors use VisTrails to run their experiments, we will investigate techniques to mine the provenance to extract this information automatically. We will also prompt authors for the experimental setup for collecting primary data in case the reviewer or a future researcher wants to repeat the primary data collection as closely as possible.

2. *Package an Experiment:* We will provide support for packaging workflows corresponding to results reported in the paper together with the underlying code and library dependencies in a compendium that can be shipped to a reviewer. To accommodate for different requirements, we will provide a flexible mechanism that gives authors different choices as to how the packaging is done. For example, we would also like to provide support for authors to submit complete virtual machines when possible; and in cases where special resources are needed, we would like to support remote access to these resources (e.g., through a Web service invocation).

3. *Support Reviewing Process:* The reviewer should be able to unpack and run the experiment including perhaps calls to the author's hardware. Within VisTrails, he will run the workflows and perform the necessary modifications to the parameters to test the system in different ways and compare the results.

   We will use provenance as a means to guide the reviewer. While reviewing a submission, provenance of the steps followed by a reviewer will be automatically (and transparently) captured. We will investigate alternative mechanisms for visualizing this provenance and allow the reviewer to annotate the information with her findings and questions. The provenance information will also be used as a means of communication among reviewers—allowing them to collaborative evaluate an experiment [5], and between authors and reviewers, allowing authors to respond to questions raised by (anonymous) reviewers.

In the course of testing these tools in SIGMOD 2011 and SIGMOD 2012, we will mine the vistrails logs as well as set up questionnaires to determine which facilities need improvement and which new ones must be developed.

Next Steps.. The need for computational repeatability is pervasive and frankly urgent. It is pervasive because all fields of natural and even social science depend on data acquisition and analysis. It is urgent because very few such studies are repeatable and, so, for the vast majority of studies there is *no practical way to verify them.* To paraphrase one participant in the Archive 2010 workshop: "if a skeptical observer cannot test a scientific result, then are we doing science?" For this reason, an outgrowth of this EAGER funding will be a future proposal in which a comprehensive system and framework for repeatability and generalizability for computational analyses across science will be described. Starting with SIGMOD is appropriate because the community is willing, we are already involved in the repeatability effort, and many in the community are computationally sophisticated. Extending this work to the broader scientific community will require the knowledge and insight we gain from the SIGMOD effort to create an easy-to-use tool across disciplines.

References

[1] S. B. Davidson, S. C. Boulakia, A. Eyal, B. Ludäscher, T. M. McPhillips, S. Bowers, M. K. Anand, and J. Freire. Provenance in scientific workflow systems. *IEEE Data Eng. Bull.*, 30(4):44–50, 2007.

[2] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD*, pages 1345–1350, 2008.

[3] E. Deelman and Y. Gil. NSF Workshop on Challenges of Scientific Workflows. Technical report, NSF, 2006. http://vtcpc.isi.edu/wiki/index.php/Main_Page.

[4] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems. *Scientific Programming Journal*, 13(3):219–237, 2005.

[5] T. Ellkvist, D. Koop, E. W. Anderson, J. Freire, and C. T. Silva. Using provenance to support real-time collaborative design of workflows. In *IPAW*, pages 266–279, 2008.

[6] S. Fomel and J. Claerbout. Guest editors' introduction: Reproducible research. *Computing in Science & Engineering*, 11(1):5–7, Jan.-Feb. 2009.

[7] I. Foster, J. Voeckler, M. Wilde, and Y. Zhao. Chimera: A virtual data system for representing, querying and automating data derivation. In *Proceedings of SSDBM*, pages 37–46, 2002.

[8] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10(3):11–21, 2008.

[9] J. Freire and C. Silva. Towards enabling social analysis of scientific data. In *ACM CHI Social Data Analysis Workshop*, 2008.

[10] The Kepler Project. `http://kepler-project.org`.

[11] D. Koop, C. Scheidegger, J. Freire, and C. T. Silva. The provenance of workflow upgrades. In *IPAW*, 2010. To appear.

[12] S. Manegold, I. Manolescu, L. Afanasiev, J. Feng, G. Gou, M. Hadjieleftheriou, S. Harizopoulos, P. Kalnis, K. Karanasos, D. Laurent, M. Lupu, N. Onose, C. Ré, V. Sans, P. Senellart, T. Wu, and D. Shasha. Repeatability & workability evaluation of sigmod 2009. *SIGMOD Record*, 38(3):40–43, 2009.

[13] I. Manolescu, L. Afanasiev, A. Arion, J. Dittrich, S. Manegold, N. Polyzotis, K. Schnaitter, P. Senellart, S. Zoupanos, and D. Shasha. The repeatability experiment of sigmod 2008. *SIGMOD Record*, 37(1):39–45, 2008.

[14] J. P. Mesirov. Accessible reproducible research. *Science*, 327(5964):415–416, 2010.

[15] Microsoft Workflow Foundation.
`http://msdn2.microsoft.com/en-us/netframework/aa663322.aspx`.

[16] S. G. Parker and C. R. Johnson. SCIRun: a scientific programming environment for computational steering. In *Supercomputing*, page 52, 1995.

[17] E. Santos, J. Freire, and C. Silva. Information sharing in science 2.0: Challenges and opportunities. In *ACM CHI Workshop on The Changing Face of Digital Science: New Practices in Scientific Collaborations*, 2009.

[18] E. Santos, L. Lins, J. Ahrens, J. Freire, and C. T. Silva. Vismashup: Streamlining the creation of custom visualization applications. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1539–1546, 2009.

[19] Y. L. Simmhan, B. Plale, D. Gannon, and S. Marru. Performance evaluation of the karma provenance framework for scientific workflows. In L. Moreau and I. T. Foster, editors, *International Provenance and Annotation Workshop (IPAW), Chicago, IL*, volume 4145 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2006.

[20] The Taverna Project. `http://taverna.sourceforge.net`.

[21] J. E. Tohline and E. Santos. Visualizing a journal that serves the computational sciences community. *Computing in Science and Engineering*, 12:78–81, 2010.

[22] The Triana Project. `http://www.trianacode.org`.

[23] VDS - The GriPhyN Virtual Data System.
`http://www.ci.uchicago.edu/wiki/bin/view/VDS/VDSWeb/WebMain`.

[24] VisTrails. http://www.vistrails.org.