A Case for DataForge: A SourceForge For Experimental Data

Prabal Dutta Archana Ganapathi David Molnar *

ABSTRACT

Computer systems research is awash in data, but we do not reap the fullest possible benefit. A variety of *data issues*, such as proprietary data sets, privacy concerns, uncertain data providence, priority questions, incompatible formats, and simply lost data make it difficult to build on experiments performed by previous systems researchers. None of these data issues are new, but the sheer scale of today's systems and variety of applications makes them more important than ever. Further, emerging areas such as sensor networks and recovery-oriented computing introduce regimes in which gathering data is itself a research goal, rather than as part of evaluation; these areas put further pressure on data issues in systems research. We outline data issues in systems research, challenges involved in addressing these issues, and we ask the question: has the time come for a *research data utility*, a "SourceForge for data?" We argue that answering this question will involve meaningful *systems* research, over and above policy questions such as whether a data set should count as a "publication." Finally, we sketch new kinds of systems research enabled by an infrastructure for sharing and manipulating research data sets.

1. INTRODUCTION

Huge amounts of data are measured for every systems project, but the field does not reap the full benefits of this data. After the paper is published, what happens to the data varies widely from researcher to researcher. The most common case is that once the data analysis is presented at a conference, the data set will not be reused for subsequent publications by the author. Thus it is backed up in a project's repository and forgotten. Occasionally, the same data set is used to compare the performance of two systems or algorithms; once this endeavor is complete, the data set is placed in storage and is not made widely available. Such practices make it inefficient for future generations to revisit the original research and build directly from previous work. Often, researchers spend much energy reconstructing previous studies and recreating data sets from the past.

Even so, there are numerous databases created for the sole purpose of making data accessible to researchers. For example, web caching data and http logs are available from several sources. More recently, traces of peer-to-peer file sharing data have become available [2]. In the architecture community, execution traces are gathered and made available. Unfortunately, these resources break with time as people move or projects finish. Regardless, such sites still provide us the benefit of knowing who is willing to share what data; this gives us an idea of who to track down.

Researchers often find to their dismay that nothing useful can be done with the data in hand. Non-standard formats, special-purpose tools, and other factors may impede the way to better analyses. Indeed, the problem is severe enough that Jacobs and Humphrey argue that we need a utility staffed by data professionals [3]. These people perform data *curation*, by which we mean they take on some of the work involved in making the data useful for future research.

Recently Gray and Szalay discussed their experiences in building a research data utility for use in astrophysics [1]. They argue that computer scientists can have a large impact on other fields by creating methods for storing and manipulating large amounts of data created by data-rich sciences. They further give concrete suggestions for how to build a data utility for serving other fields based on their experiences with astrophysics. While Gray and Szalay describe an architecture that requires "big iron," other researchers point the way towards less capital-intensive methods for data preservation. For example, the LOCKSS project uses a network of peers to aid libraries in their quest to preserve books and other materials [4].

We ask *why should other fields have all the fun*? Systems research is itself data-rich. Why can't we build a research data utility for our own use? We call this utility a DataForge, as in "a SourceForge

*E-mail: {prabal, archanag, dmolnar}@cs.berkeley.edu, Computer Science Division, Univ. of California, Berkeley, Berkeley, CA 94720

for data," and we believe that designing and building such a utility will both require and enable new research. In the next section, we lay out the *data issues* that motivate a DataForge, then lay out the benefits of a DataForge. We argue that pursuing a DataForge will involve meaningful systems research, over and above purely social or policy questions. Finally, we sketch new types of research that would be enabled by a DataForge.

2. DATA ISSUES

Gathering Data is Costly. Gathering data requires a significant investment. Some of these investments are technical, such as the cost of building an infrastructure for measurement. Others are more social in nature. For example, when collecting data from companies, researchers must pass through several layers of indirection for an approving signature. Furthermore, corporate lawyers spend several months drafting tedious legal agreements for the collaboration. Similarly, when conducting user studies for research, students are often required to obtain approval from an institutional review board, which requires considerable paperwork and has high latency.

Gathering Data is Time Consuming. It is accepted wisdom that time is our most precious resource. Yet, large scale experiments can take days, weeks, or months to carry out but often we fail to amortize these costs and leverage these investments. If we are to be more effective and efficient as a research community, then we must find ways to use and build upon the time investments of our peers. Data collection in experimental computer science and engineering has always been time consuming. However, as we pursue research agendas which increasingly connect the physical and virtual worlds, or embark on projects which require data as an *input*, like applications of statistical learning theory to systems problems, the data collection, management, and sharing challenges can only grow.

We have experienced first-hand the time consuming and overhead-laden nature of field data collection. To clarify what we mean by "time consuming and overhead-laden," consider a typical data collection day in the life of a sensor networks researcher in which only two useful hours of field data are collected. The experimental site had to be reserved well in advance of the field day since the site was owned by a third party. Attorney representing both the organization on whose grounds we planned to carry out experiments as well as university lawyers were involved to ensure that each party's rights were protected and liabilities limited. And, of course, experiments had to be designed, simulations carried out, and scripts written to control the experiment. Still, much of our effort was concentrated on the actual day of data collection. Figure 1 shows the schedule for a typical day of field data collection involving a handful of graduate students.

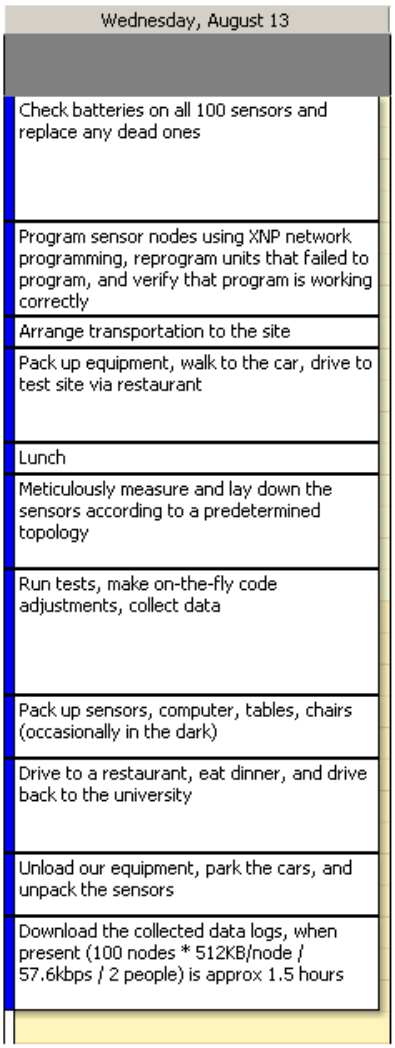

Figure 1: Typical schedule spanning from 9:00 A.M. to 11:30 P.M. for collecting a less than two hours of sensor data.

A review of the data collection day schedule reveals a ten hour overhead for a medium scale experiment. As a result, data collection occurred infrequently. Even though we knew that others

were carrying out the exact same experiments with the exact same hardware elsewhere, we were not able to make use of others' experimental setups, datasets, or data analysis because we simply did not know at the time that such data existed. In the most extreme cases, replicating an experiment may be impossible in cases of exception events like volcanic eruptions or earthquakes.

Files Are Not Enough. Data may be in obsolete formats or require special programs to read. Even if the data set is publicly available, this does not guarantee it is useful. It is important to enforce data-descriptors and adaptable data formats to allow future researchers to build from previous work. Additionally, a data evolution log might be instrumental in helping researchers understand exactly how the original data set was used for problem solving.

Collaboration Models. Academia uses tested mechanisms for sharing data with other academic institutions as well as industry. When two universities are collaborating on a project, they adopt a "gentlemans agreement" whereby credit is duly given to all individuals who contribute to a paper. Furthermore, each researcher clearly defines his project boundaries to avoid redundant work. However there is inherent competition among researchers working in the same problem domain, often resulting in data hoarding. Making the data available to peer researchers can lead to the unfortunate consequence of leapfrogging analysis results. A more concrete agreement regarding scope of data usage would reduce such risks and benefit this collaboration model.

Collaborations with industry have a slightly different flavor. When obtaining data from industry, a common model is to sign a non-disclosure agreement. Research groups are given access to the raw data, yet there are constraints on what can be publicly revealed about the data set. In return, industry benefits from the analysis of their data, often allowing them to enhance the performance/fault-tolerance of their systems. There have also been precedents for academia sharing data with industry. Traces from Baker et al.'s paper were used by industry to evaluate proprietary file systems. Such cross pollination of data and resources benefits the entire research community.

3. WHY CURATE SYSTEMS DATA?

Improve Experimental Rigor. Small et al. reviewed systems papers and asked the question whether the claimed results could be repeated given the data in the publication [6]. They found that most papers included insufficient data and insufficient statistical detail to interpret the results. A SourceForge for data facilitates sharing information too large to fit within a standard paper submission.

Common Comparisons. Mogul outlines requirements for making claims based on performance data and evaluates a wide range of systems papers against these requirements [5]. He finds that many papers do not meet the requirements; for example, relatively few use common benchmarks, even in fields where these benchmarks are available. If the data gathered from these systems were widely available, comparative studies could be done even without benchmarks standard to the field. This would in turn allow for research that pushes the envelope beyond the scope of accepted benchmarks without sacrificing the ability to compare to other systems.

Increase Return on Research Investment. We have argued that gathering data is costly. We must maximize the benefits of this investment and make data readily available for reuse. A SourceForge for data makes this simpler by removing some of the burden from the original researchers.

Data Transience. In the academic setting, data gathered for a PhD thesis often "graduates with the student." As an account is reclaimed after graduation, research data kept on the account may be lost. Even if the data is physically present at an institution, it may be "hidden" on some obscure server or require tracking down the appropriate people to restore it from backup. This process may be time consuming and researchers might lose interest in the data by the time they actually have access to it. Similar issues apply in industry when people change jobs or when companies change hands. The person-hours invested in the data collection are lost.

4. WHERE IS THE *SYSTEMS* RESEARCH?

Little expertise in building systems transfers to negotiating an NDA over a data set. Likewise, questions such as whether we should have a mechanism for formally publishing a data set as we publish papers are primarily social, not technical, in nature. Put another way, the data issues we raise appear to be purely questions of computer science

policy rather than computer systems research.

This appearance is mistaken. First, the space of policy decisions considering data issues is directly affected by the technical feasibility of building a research data utility, which in turn raises problems in systems research. A utility that costs one million dollars per year to run will give us a different set of options than a utility that costs one hundred. Second, a data utility can have impact even on a small scale and yield research problems of independent interest. Finally, researching what we would *do* with such a utility tells us something about how much we might *desire* it as a policy objective. We now elaborate on these points.

4.1 Building a Sourceforge for Data

Representing Experiments Repeatably. Once the initial data collection has taken place, the "raw" measurements are analyzed to produce conclusions. Two different teams can come to dramatically different conclusions depending on the assumptions they hold regarding the "raw" data and the analysis they perform. For example, one team might look at the mean value of a data set, while another might look at the median. We need a way to document analyses so that they can be repeated and critiqued later.

Reducing Cost of Data Curation. There is currently no single repository that can be queried for data sets. We rely on the knowledge of peer researchers to point us to the right person to obtain data. A SourceForge for data provides a single interface to numerous data sets, eliminating the unnecessary downtime of waiting for responses. There are various design and maintenance considerations for building such a repository, some of which are enumerated below.

Centralized vs Distributed Management. One of the biggest challenges of building a DataForge is determining management logistics. A centralized repository would be simpler and more cost effective to monitor. Designating a single organization to maintain the system introduces issues related to economics as well as trust. A decentralized repository, on the other hand, would be more fault tolerant (eliminating the single point of failure) but would require sophisticated consistency mechanisms to assure data integrity. A related question is whether *federated* management is feasible; a federated scheme would allow member sites to choose which features to open or not to the outside.

Database Schema: There are numerous repositories created by research groups to hold different types of data ranging from failure data to http and Apache logs to sensor data. Little effort is spent on making these repositories easy to access. Systems researchers would benefit from a unifying schema that accommodates all these data types. We might use XML-like languages to write headers describing the data set and index the actual data in tables pointed to by these entries.

Access control for contributing and retrieving data: It is important to verify the authenticity of data (and the contributing entity) to avoid plaguing our repository with fake data. We also need mechanisms to verify that people using the data give due credit to the data contributors. This task is challenging as the purpose of the repository is to provide data access to any and all organizations while reducing the likeliness of misuse.

Anonymization Infrastructure. People often have stringent privacy requirements for sharing data. We can meet these requirements by providing an infrastructure to anonymize sensitive data at the point of collection, a model that has already been adopted by some systems researchers. For example, Gummadi et al. ensured that the IP address was removed from their P2P traces before the traced packet hit stable storage. Another concern is that no details regarding individuals should be reproducible from cross-correlating various data sets.

4.2 Payoffs From a DataForge

Executable Systems Papers. We can write new "executable papers" that make use of the DataForge to present re-analyses of system measurements as the reader desires. What if you could re-graph figures on the fly with different scales, or zoom in on a strange point in a graph? What if you could compare two data sets which were originally placed in different figures of the paper?

Remote Data Access. A related payoff is the ability to access data remotely. Some data sets are so large that it is infeasible to download them. In this case, researchers could log in to the DataForge and obtain a shell with access to R, matlab, and other useful utilities.

Gap Analysis. It is often beneficial to revisit previous work with a more modern perspective and try to extract principles applicable to ad hoc tech-

nology trends. Currently, there is no clear method validate or build upon previous experiments due to the lack of source code and/or data. A DataForge would enable researchers to redo experiments and discover and analyze additional features of a system.

META-ANALYSIS. Similarly, a DataForge also enables meta-analysis, in which several different papers on closely related topics are compared and conclusions about the research area are formulated. Typically this kind of meta-analysis is rarely done in systems research, because the technology changes rapidly and the cost of recreating data sets is high. By changing the cost to recover data, we can lower the barrier to this kind of research.

AUDIT TRAIL. Depending on how it is designed, a DataForge could offer the ability to create an audit trail for data or enforce a privacy policy.

5. DISCUSSION

We have argued that designing and building a DataForge will both require and enable new systems research. Of course, a DataForge will also involve tricky political, social, and legal issues. For example, who should own a data set? Jacobs and Humphrey argue no one should, that data should be free [3]. Our experience tells us that there is a long way to go before that point. Another issue is whether placing a data set onto DataForge should count as a formal "publication." We believe that the economic and technical issues involved in building a DataForge will influence these discussions. Furthermore, a DataForge will be useful under a wide class of answers to these questions.

Another unsolved issue is that research can be heavily skewed by the available of data in that area. Providing a DataForge does not eliminate this problem; perhaps the problem may even be magnified. Among other social issues, technical and legal minds must work in sync to implement strong policies to share data more efficiently and safely. Designing a first draft of a DataForge will help both computer scientists and lawyers understand points of contention and help us derive a more robust and mutually agreeable system. In the end, we expect a DataForge would benefit several generations of Computer Science researchers.

6. REFERENCES

[1] J. Gray and A. Szalay. Where the rubber meets the sky:bridging the gap between databases and science, 2004. Microsoft Research TR-2004-110.

[2] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *19th ACM Symposium on Operating Systems Principles (SOSP-19)*, October 2003.

[3] J. Jacobs and C. Humphrey. Preserving research data. *Communications of the ACM*, 47(9):27–29, sep 2004.

[4] P. Maniatis, M. Roussopoulos, T. Giuli, D. S. H. Rosenthal, and M. Baker. The lockss peer-to-peer digital preservation system. *ACM Transactions on Computer Systems (TOCS)*, 2004.

[5] J. C. Mogul. Brittle metrics in operating systems research. In *Workshop on Hot Topics in Operating Systems*, pages 90–95, 1999.

[6] C. Small, N. Ghosh, H. Saleeb, M. Seltzer, and K. Smith. Does systems research measure up?, November 1997. Harvard University Computer Science Technical Report TR-16-97.