

Tools and Techniques for Failure Data Collection and Analysis

Archana Ganapathi

Computer Science Division, University of California at Berkeley
archanag@cs.berkeley.edu

Abstract

Failure data analysis is an essential component of computer systems research. Unfortunately, academic researchers are often hindered from making progress on their research agenda due to the lack of publicly-available real-world failure data. To address the gap between data abundance in industry and data scarcity in academia, we propose several tools and techniques to address privacy concerns and facilitate data sharing. We suggest using publicly-available tools such as "Berkeley Open Infrastructure for Network Computing" for data collection, "Computer Failure Data Repository" for allowing data reuse, and various Machine Learning algorithms to identify, anonymize and extract useful information from the failure data.

1. Introduction

Data analysis is a critical component that enables academic researchers to identify and pursue research agendas with real impact in industry. However, due to industry's stringent privacy concerns and legal formalities, academia often suffers from data starvation. On the other hand, despite the plethora of data available in industry, not very many companies invest time and money in data analysis techniques other than for forensic debugging purposes. As a result, there is a huge gap in our understanding of failures in systems.

Among many applications for data analysis (especially pertaining to failures) it would be useful to develop an oracle of system behavior that has knowledge about safe software configurations and unsafe configurations that consistently cause failures. Such knowledge has to be backed by real data and must also be continuously updated to account for changes in system environment. Maintaining such information will also help us gauge behavioral changes over time and predict future system behavior. Overall failure data analysis helps us move towards building

more reliable and failure-resilient systems and globally reduce computer user frustration.

Below, we address typical concerns with sharing data and identify tools and techniques that can be used to amortize the cost of data collection.

2. Obstacles of Failure Data Collection

There are several reasons why individuals as well as organizations hesitate to share data; these concerns are even more stringent with respect to failure data. Individuals worry about revealing illicit (perhaps not illegal) activities that may be reverse-engineered from the collected data. As failure data can potentially reveal product dependability statistics or loopholes that can be misused, organizations fear that competitors may access and/or misuse data and analysis results.

Often, while industrial researchers are willing to share data with academic researchers, legal departments intervene and impose long delays to draft legal agreements. If and when these agreements are ready, they are often so limiting that it would be impossible to publish any meaningful research results based on the shared data.

3. Tools and Techniques

There are two fundamental axioms that can facilitate data sharing between industry and academic institutions.

1. Use a neutral/trusted third party for data collection and storage
2. Amortize the cost of data collection by simplifying data reuse.

To satisfy the above two axioms, we suggest a few tools and techniques. For data collection, we suggest using Berkeley Open Infrastructure for Network Computing (BOINC). For the purposes of storing data and allowing data reuse, we suggest using the Computer Failure Data Repository (CFDR). Lastly, we propose creating a toolbox to assist data analysis. Each of these suggestions is elaborated below.

3.1. Data Collection using BOINC

BOINC is a non-commercial framework that allows researchers to use spare computing cycles from volunteers to perform distributed computing [1]. We can take advantage of this open-source framework to deploy data collection software (a rather non-traditional use for grid computing) [2]. BOINC provides a platform to send and receive data from volunteers using the HTTP protocol and aggregate it in in-house BOINC servers. It has built-in anonymization mechanisms that decouple data from its source.

As BOINC has positive representation and a large volunteer customer base, it is fairly easy to gather large quantities of data from this user population. However, there is an inherent self-selection bias in using this mechanism. While many people enthusiastically contribute to research causes facilitated by BOINC, this user community is not representative of the entire PC-user population.

We can use the BOINC infrastructure to collect data from various organizations as well. We can write custom applications to cater to the software environment within each organization.

3.2. Data Storage using CFDR

Often it takes several months to years to collect a significant quantity of data. It is important to amortize the time and effort spent on this process, and make the data available to other researchers. Carnegie Mellon University is leading an initiative to make failure data publicly available [3]. USENIX has agreed to host and manage the repository.

It is important for academic and industrial researchers to contribute data to CFDR and use such data to validate research results. Such a repository enables us to derive realistic failure and performance benchmarks to compare system reliability.

3.3. Data Analysis Toolbox

Interpreting failure data is a tedious task that requires good intuition and a useful set of tools. There are several tasks that are generic to any data analysis and it seems inefficient for each researcher to redevelop tools to perform these tasks.

It is critical to provide anonymization tools for any failure data analysis. The tools should accommodate company/user specific privacy policies while producing anonymized data that is still usable. Among useful features are the ability to hash values of certain fields and mask parts of revealing data such that the relative values and proportions of data are preserved.

Almost all data analysis requires filtering and preprocessing to remove duplicates, identify outliers and find temporal correlations. These tasks are fairly straightforward and can be bundled for reuse.

Statistical Machine Learning (SML) algorithms are becoming more commonplace in systems data analysis. There are several tools, such as R and Yale [4,5], that can be used for applying SML algorithms on data. The most applicable data analysis algorithms include dimensionality reduction and correlation analysis. It would be useful to have standard interfaces to these data analysis packages that can be used by computer data analysts.

Lastly, it is fairly important to incorporate good data visualization packages that help interpret results from SML algorithms. Such packages would help bridge the gap between theoretical data analysis and practical interpretations of the results.

4. Conclusions and Open Questions

We have identified a set of mechanisms that can readily be used for data collection and analysis. We must cooperate to integrate these tools and make them easily accessible to researchers. However, there are still some unaddressed issues in the data aggregation and sharing process.

Several open questions arise with respect to the semantics of data sharing. While collecting useful data is tedious, what information is necessary and sufficient to interpret data trends? Also, is it possible to draft End User License Agreements for data collection/usage that are legally sophisticated yet not too confining? Addressing these questions will help bridge the gap between academic and industrial research and allow us to work together to tackle some of the most challenging systems problems.

5. References

- [1] D. Anderson, "Public Computing: Reconnecting People to Science," *The Conference on Shared Knowledge and the Web*, Madrid, Spain, November 2003.
- [2] A. Ganapathi, "Why Does Windows Crash?" UC Berkeley Tech. Report UCB//CSD-05-1393, May 2005.
- [3] B. Schroeder, G.A. Gibson, "The Computer Failure Data Repository (CFDR)", 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06), Seattle, WA, November 2006.
- [4] R: <http://www.r-project.org/>
- [5] Yale: <http://rapid-i.com/>