SHF:Small: Data Debugging

Emery Berger, Alexandra Meliou

Intellectual Merit.

Data debugging, awesome.

Broader Impact. Data debugging will save the world. The investigators will make their benchmarks and tools publicly available, adding to the national research infrastructure.

The investigators' prior tools and systems are widely used in research and industry: the Hoard high-performance memory manager, the DieHard error-tolerant and DieHarder secure runtime systems now incorporated in Windows, *Alexandra's stuff*. Educational impact will include training graduate and undergraduate students in key language system technologies, contributing to the technology workforce, and outreach to under-represented groups via the inclusion of female students from nearby Mount Holyoke and Smith Colleges.

Key words: Programming Languages; Databases; Big Data; Errors

Data Management Plan

Data Collection

- *types of data, samples, physical collections, software, curriculum materials, and other materials to be produced in the course of the project*

  Data to be collected as part of this research include software source code, educational materials, performance results including benchmark execution times and other performance characteristics such as memory consumption, as well as logs of interactive proof assistants (e.g., Coq).

Data Storage

- *standards to be used for data and metadata format and content*

  All performance results will be stored in comma-separated files (.csv format). All other source code and output will be stored in ASCII text. Educational materials will comprise LaTeX files and PowerPoint presentations.

  Scripts written in publicly-available languages like Python or Perl will be used to drive all benchmarking. All scripts will be maintained together with other source code products of this research.

- *policies for access and sharing including provisions for appropriate protection of privacy, confidentiality, security, intellectual property, or other rights or requirements*

  The data generated by this project will comprise only publicly-available, non-confidential information. All source code will be released under an open source license, such as the MIT or GNU General Public Licenses.

Preservation, Documentation and Sharing of Data

- *policies and provisions for re-use, re-distribution, and the production of derivatives*

  All products from this project will be made publicly available both on a departmental website and via a public source code version control repository (e.g., `github.com`), including data, scripts, and source code. By releasing code under the GNU General Public License, all source code can be freely included in any open source project.

- *plans for archiving data, samples, and other research products, and for preservation of access to them*

  All artifacts will be suitably documented. Additional copies of all data and code will be archived on departmental backup systems (disks on-site, and tapes off-site) to further ensure their preservation. In addition, papers that present results based on our research will be made available via the PIs individual web pages.

University of Massachusetts
Budget Justification

Senior Personnel

- PI Emery Berger: 1 summer month/yr.
- Co-PI Alexandra Meliou: 1 summer month/yr.

A 3.5% increase is added per year.

Other Personnel

- 2 Graduate Students: Student salary is based on approximately $27.34 an hour, 20 hours a week, 38 weeks per academic year, plus 14 weeks in summer. A 3.5% increase is added per year.

Fringe Benefits

Benefited Positions / Fringe Rates:	
Fringe 7/1/12–6/30/13	25.98% + workers comp. 0.53%, + UI, UHI, MTX 1.29% = **27.80%**
Health & Welfare	$14.50 weekly = $754 annually
Sick Leave Bank	0.30%, not assessed on Faculty Salaries

- **Fringe** benefits applicable to direct salaries and wages are treated as direct costs. They are the rates identified in the Massachusetts Statewide Cost Allocation Plan approved by DHHS. This rate is comprised of Group Insurance and Retirement. The combined rate must be applied to all benefited personnel on any awards.
- **Health and Welfare** (H & W) for all benefited positions is $14.50 per week ($754) annual FTE (prorate on part-time positions). Health and Welfare must be assessed to all graduate student appointments. A 5% increase is added per year.
- **Summer Student Payroll:** *All students* (excluding Post Docs and Fellows) employed for the summer and not enrolled in classes are to be assessed 1.29% UI, UHI, MTX on the summer salary.
- **GEO Health Deferment Rate:** An appointment that is 20 hours per week both summer and academic year (1,040 hrs) would be assessed $5,439.20. A 5% increase is added per year.

Other Direct Costs

- **Domestic Travel:** Anticipate travel to the following conferences: PLDI, OOPSLA, ASPLOS, etc. Exact dates and locations to be determined. A 5% increase is added per year.

  Years 1–3: 2 conferences at approximately $1,666 for each of the PIs and graduate students, and 1 conference (West Coast) in year 3. Approximate costs include $450 RT airfare, $600 registration fee, $125/night hotel (2 nights), ground transportation $200, per diem $100, miscellaneous $66.

- **International Travel** Anticipate travel to the following conferences: PLDI, POPL, ICSE, FSE (overseas). Exact dates and locations to be determined. A 5% increase is added per year.

  Years 1–3: 1 conference at approximately $5,000 for one PI and one graduate student. Approximate costs include $2,500 RT airfare, $800 registration fee, $800–$1,000 foreign per diem rates (4 days), $350 ground transportation, $350 miscellaneous.

- **Materials and Supplies:**

  One laptop at approximately $1,200 will be purchased in each year for a graduate student. The laptop is project-specific because of the need to experiment on state-of-the-art browsers and computing environments. Purchase of computer peripherals, upgrades, storage and parts for the laptop. A 5% increase is added per year.

- **Computer Costs:**

  Computer maintenance will be paid to the Computer Science Department's Computer Facility to provide maintenance and support of equipment, software, and communication networks, as well as, mass storage and file back-up services. The charges are based on a campus approved fee structure. There are also general computer science facility maintenance charges that are billed according to FTE faculty, staff, and students assigned to the project (for central/shared services) and according to the number of workstations and other general-purpose equipment assigned to the project (for maintenance of this equipment). A 5% increase is added per year.

- **Other:**

  The University charges a curriculum fee to all grad student appointments. FY14 – $8,618.40. Summer appointments are not charged curriculum fee. This fee is exempt from I.C. A 5% increase is added per year.

Indirect Costs

Indirect costs are 59% MTDC, 7/1/12–6/30/15.

SHF:Small: Data Debugging

Emery Berger, Alexandra Meliou

1 Broader Impact, Education, and Outreach

Broader Impact. FIX ME.

We will add to the national computing infrastructure by implementing and making available all of the component systems we build as part of this project. These tools will add to the national research and industry software infrastructure. Our prior tools are in wide use in industry and academia, and we expect these will be also.

Education. We plan to integrate the research developed here into both undergraduate and graduate classes. FIX ME.

Outreach. Improving participation in computer science from underrepresented groups requires outreach at multiple levels; the PIs have a strong track record of this work at both the undergraduate and graduate levels.

PI Berger has supervised undergraduate research and mentored women and minority undergraduates. PI Berger has recruited undergraduates from across the neighboring Five College consortium (which includes Amherst and Hampshire Colleges, and two female-only undergraduate institutions, Smith and Mt. Holyoke Colleges). FIX ME Alexandra stuff.

2 Results of Prior NSF Support

Emery D. Berger is a PI of three active NSF grants.

Professor Berger is a PI in a collaborative project with Professor Daniel Jiménez (Texas A&M University): *SHF: Large: Collaborative Research: Reliable Performance for Modern Systems* (NSF CCF-1012195, $550,000, August 2010-July 2013). This project aims to deliver reliable performance on modern computer systems. By introducing randomness into the way a computer runs programs, a reliably performant system will significantly reduce the probability that any small change will have a large impact on performance. It has thus far produced two publications by PI Berger, to appear at ASPLOS and in ACM TECS [5, 7], and one major software release: STABILIZER, a compiler and runtime system that enables statistically sound performance evaluation.

Professor Berger is sole PI of *Programming the Crowd* (NSF CCF-1144520, $300,002, August 2012-December 2013). This project introduces crowdprogramming, an approach that fully integrates human and digital computation. In crowdprogramming, humans are modeled as function calls in a standard programming language. This approach lets programmers focus on programming logic, while the crowdprogramming runtime system manages the critical tradeoffs between cost, time, and data quality. We have released a crowdprogramming system called AUTOMAN and reported on it at OOPSLA [1].

Professor Berger is a PI in a collaborative project with Professor Michael Hicks (University of Maryland) and Professor Kathryn McKinley (UT-Austin): *PASS: Perpetually Available Software Systems* (NSF CCF-0910883, $639,420, August 2009-July 2013). This project proposes a transformative paradigm shift to "perpetually available software systems" (PASS) that will make software more available and robust by directly addressing errors in deployed software. Thus far, this project has led to two publications by PI Berger at OOPSLA and SOSP [12, 11], and two major software releases: DTHREADS, a deterministic replacement for `pthreads`, and SHERIFF, a system that finds or eliminates false sharing.

PI Berger has been supported by three previous NSF grants:

The first of these grants is *CAREER: Cooperative System Support for Robust High Performance* (NSF CAREER CNS-0347339, $477,000, June 2004-2009). The purpose of this project is to attack the growing latency between main memory and disk with adaptive algorithms that leverage cooperation between the compiler, runtime systems, and operating system. It has thus far produced eight papers that have appeared at PLDI, OSDI (2), OOPSLA, ISMM, MSP, and USENIX [8, 10, 9, 16, 15, 6, 4, 17].

PI Berger was the sole PI of *Probabilistically Correct Execution: Hardening Applications Against Error and Attack* (NSF CNS-0615211, $300,000, September 2006-2009, no cost extension). Probabilistically correct execution is an approach that allows programs to execute correctly with high probability in the face of errors or attack, and functions by both *randomizing* the runtime system

and *replicating* differently-randomized executables. This combination transforms unreliable software components into a quantifiably more reliable system. This grant has thus far produced four papers, two at PLDI, one at ASPLOS, and one at OOPSLA [3, 14, 13, 2]. One result of this work is DieHard, a freely-available system that increases the security and resilience to memory errors of C/C++-based applications; DieHard has been downloaded over 20,000 times since its release, and directly inspired the Fault-Tolerant Heap included in Windows since version 7. Another product of this research, the DieHarder secure heap, an inspiration for the security hardening features incorporated into Windows 8.

PI Berger was also a co-PI of *MRI: Cluster Acquisition for Computational Research into Large Scale Data Rich Problems* (NSF CNS-0619337, $350,000, 9/1/2006–9/1/2008).

FIX ME add Alexandra's stuff.

3 Summary

Data debugging is clearly the bomb. Fund us.

References

[1] D. W. Barowy, C. Curtsinger, E. D. Berger, and A. McGregor. AUTOMAN: a platform for integrating human-based and digital computation. In *Proceedings of the ACM international conference on Object oriented programming systems languages and applications (OOPSLA 2012)*, pages 639–654, New York, NY, USA, 2012. ACM.

[2] E. D. Berger, T. Yang, T. Liu, and G. Novark. Grace: safe multithreaded programming for C/C++. In S. Arora and G. T. Leavens, editors, *Proceedings of the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2009)*, pages 81–96, Oct. 2009.

[3] E. D. Berger and B. G. Zorn. DieHard: Probabilistic memory safety for unsafe languages. In *Proceedings of the 2006 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2006)*, pages 158–168, New York, NY, USA, 2006. ACM Press.

[4] B. Burns, K. Grimaldi, A. Kostadinov, E. D. Berger, and M. D. Corner. Flux: A language for programming high-performance servers. In *Proceedings of the 2006 USENIX Annual Technical Conference*, pages 129–142, June 2006.

[5] F. J. Cazorla, E. Quinones, T. Vardanega, L. Cucu, B. Triquet, G. Bernat, E. Berger, J. Abella, F. Wartel, M. Houston, L. Santinelli, L. Kosmidis, C. Lo, and D. Maxim. PROARTIS: Probabilistically analysable real-time systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 2013.

[6] J. Cipar, M. D. Corner, and E. D. Berger. Transparent contribution of memory. In *Proceedings of the 2006 USENIX Annual Technical Conference*, pages 109–114, June 2006.

[7] C. Curtsinger and E. D. Berger. STABILIZER: Statistically sound performance evaluation. In *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '13, New York, NY, USA, 2013. ACM.

[8] Y. Feng and E. D. Berger. A locality-improving dynamic memory allocator. In *Proceedings of the ACM SIGPLAN 2005 Workshop on Memory System Performance (MSP)*, Chicago, IL, June 2005.

[9] M. Hertz and E. D. Berger. Quantifying the performance of garbage collection vs. explicit memory management. In *Proceedings of the 20th annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA 2005)*, San Diego, CA, Oct. 2005.

[10] M. Hertz, Y. Feng, and E. D. Berger. Garbage collection without paging. In *Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2005)*, pages 143–153. ACM, June 2005.

[11] T. Liu and E. D. Berger. SHERIFF: precise detection and automatic mitigation of false sharing. In *Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications*, OOPSLA '11, pages 3–18, New York, NY, USA, 2011. ACM.

[12] T. Liu, C. Curtsinger, and E. D. Berger. DTHREADS: efficient deterministic multithreading. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 327–336, New York, NY, USA, 2011. ACM.

[13] V. B. Lvin, G. Novark, E. D. Berger, and B. G. Zorn. Archipelago: trading address space for reliability and security. In *ASPLOS XIII: Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, pages 115–124, New York, NY, USA, Mar. 2008. ACM.

[14] G. Novark, E. D. Berger, and B. G. Zorn. Exterminator: automatically correcting memory errors with high probability. In *Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2007)*, pages 1–11, New York, NY, USA, 2007. ACM Press.

[15] T. Yang, E. D. Berger, S. F. Kaplan, and J. E. B. Moss. CRAMM: virtual memory support for garbage-collected applications. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)*, pages 103–116, Berkeley, CA, USA, 2006. USENIX Association.

[16] T. Yang, M. Hertz, E. D. Berger, S. F. Kaplan, and J. E. B. Moss. Automatic heap sizing: Taking real memory into account. In *Proceedings of the 2004 ACM SIGPLAN International Symposium on Memory Management (ISMM)*, pages 61–72. ACM Press, Nov. 2004.

[17] T. Yang, T. Liu, E. D. Berger, S. F. Kaplan, and J. E. B. Moss. Redline: First class support for interactivity in commodity operating systems. In R. Draves and R. van Renesse, editors, *Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2008)*, pages 73–86. USENIX Association, Dec. 2008.