

William McCloskey
bill.mccloskey@gmail.com
(617) 501-8417

622 Midrock Corners
Mountain View, CA, 94043

Summary

Experienced engineer looking for interesting new challenges in systems programming, distributed systems, concurrency, compilers, memory management, and/or debugging.

Education

Computer Science Ph.D., U.C. Berkeley, 2010.

Advisor: Eric Brewer.

Specialization: programming languages and program analysis.

Thesis: Practical Shape Analysis.

Computer Science B.S., Cornell University, 2002.

GPA: 3.95. Graduated magna cum laude.

Skills

C++, JavaScript, Python, C, Objective Caml, Java (in order of recency); Linux, Windows, MacOS, though I'm most familiar with Linux.

Experience

- Mozilla Corporation, senior staff engineer (currently)
2010 to present

2016-2017: Currently leading Mozilla's [Quantum DOM project](#) [1], which spreads web rendering workload across multiple threads and prioritizes important tasks within threads.

2015-2016: Developed [WebExtensions](#) [2] (an adaptation of Chrome's extension model) for Firefox. Wrote the initial code, advocated for it to management, selected team members, and started standardization process.

2015-2016: Created [Searchfox](#) [3], a code search and blame tool for Mozilla developers.

2013-2016: Revived Mozilla's [Electrolysis](#) [4] project to browse the web using separate content processes. Served as technical lead throughout the project, starting from two engineers until eventually most engineers in the company were involved in some way.

2010-2012: Implemented [incremental garbage collection](#) [5] in the SpiderMonkey JavaScript engine. Also provided technical leadership to other GC projects, especially generational GC. Additionally worked on JIT optimizations and heuristics.

- U.C. Berkeley Research Assistant (advisor: Eric Brewer)
Fall 2002 to Spring 2010

Wrote a [shape analysis](#) [6] to find memory errors in C systems code. The analysis supported arrays and linked data structures, hash tables, procedure calls, function pointers, casting, pointer

arithmetic, and reference counting. Before that, worked on concurrency ([Autolocker](#)[7]), and the automatic transformation and refactoring of C programs ([ASTECC](#) [8]). The unifying goal was to move C systems code to a [safer, stronger foundation](#) [9].

- Microsoft Research, Summer Intern (advisor: Sumit Gulwani)
Spring 2007

Implemented a program analysis for analyzing programs that manipulate arrays. Unlike a standard numerical abstract interpretation, our analysis could infer quantified invariants, such as "forall $0 \leq i < n$, $A[i] = 0$ ". The key insight was that reasoning about quantifiers requires both over-approximation and under-approximation. We used [our analysis](#) to prove properties of some sorting, searching, and partitioning algorithms [10].

- IBM Research, Summer Intern (advisor: David Bacon)
Summer 2006

Rewrote parts of the Metronome real-time garbage collector for Java to support mutator/GC concurrency in all principal phases (mark, sweep, and compaction). Our compaction algorithm permitted objects to be moved while the mutator ran and did not require the mutator to perform any atomic operations in the common case. This new collector, called [Staccato](#) [11], improved throughput and worst-case latency over Metronome while scaling up to larger multi-CPU systems. It also increased flexibility in GC scheduling so that important threads were almost never impacted by the collector at all.

References

Bugzilla profile. https://bugzilla.mozilla.org/user_profile?login=wmccloskey%40mozilla.com
Github profile. <https://github.com/bill-mccloskey>

1. Mozilla's Quantum project.
<https://billmccloskey.wordpress.com/2016/10/27/mozillas-quantum-project/>
2. WebExtensions. <https://developer.mozilla.org/en-US/Add-ons/WebExtensions>
3. Searchfox. <https://billmccloskey.wordpress.com/2016/06/07/searchfox/>
4. Multiprocess Firefox. <https://billmccloskey.wordpress.com/2013/12/05/multiprocess-firefox/>
5. Incremental GC in Firefox 16!
<https://blog.mozilla.org/javascript/2012/08/28/incremental-gc-in-firefox-16>
6. [*Statically inferring complex heap, array, and numeric invariants.*](#)
Bill McCloskey, Thomas Reps, Mooly Sagiv. SAS 2010.
7. [*Autolocker: Synchronization Inference for Atomic Sections.*](#)
Bill McCloskey, Feng Zhou, David Gay, and Eric Brewer. POPL 2006.
8. [*ASTECC: A New Approach to Refactoring C.*](#)
Bill McCloskey and Eric Brewer. FSE 2005.
9. [*Thirty Years Is Long Enough: Getting Beyond C.*](#)
Eric Brewer, Jeremy Condit, Bill McCloskey, and Feng Zhou. HotOS 2005.
10. [*Lifting Abstract Interpreters to Quantified Logical Domains.*](#)
Sumit Gulwani, Bill McCloskey, and Ashish Tiwari. POPL 2008.
11. [*Tax-and-Spend: Democratic Scheduling for Real-Time Garbage Collection.*](#)
Joshua Auerbach et al. EMSOFT 2008.