

# James E. Mace

400 Mississippi Street  
San Francisco, CA 94107  
415-550-9366 (Home)  
jm5b@yahoo.com

---

## Profile

Highly experienced software engineer with an unusually broad background: scientific programming, compiler tool chain internals, networking, distributed systems, search engine implementation, enterprise software, web technologies, file systems, and Unix/Windows system programming. Well-regarded as a domain expert and mentoring resource in many of these subject areas. History of successful, clearly documented technical designs and effective, timely coding of complex products. Several projects have resulted in patent applications. Expert in multithreaded C/C++ development, debugging and profiling, with significant skills in java, python and Unix shell scripting.

Particularly interested in technical projects involving data structure design, large-scale information storage and retrieval, performance optimization, API/SDK design and implementation.

---

## Experience

### Riverbed Technology

*Member of Technical Staff (Team Lead)*, April 2006-present

- Led a team of five engineers to design and implement a high performance, scalable, fault tolerant, reference counted object store for small variable-sized fragments (128 bytes on average) produced by Rabin-Karp fingerprint segmentation of data streams. This object store was foundational technology for Riverbed's in-path deduplicating file system. Also designed and implemented data structures for representing and atomically updating deduplicated file data on disk. The store and file system were coded for Linux in C++ using a custom user-space cooperative threading library. The file system implemented both CIFS and NFS protocols.
- Led a team of five to design and implement a second-generation object store as the key component of an out-of-path file deduplication project based on NTFS reparse points. This version improved on the original by providing in-process file segmentation, more compact representation of deduplicated content, and an extremely high ingest rate, all while minimizing file system churn - ideal for use with file systems supporting block-level point-in-time snapshots. The store was coded as a standalone library for Linux in C++ using standard POSIX threading.
- Currently leading a team of three implementing a fault tolerant, transactional iSCSI block cache with asynchronous write-back to primary storage. This system is designed to operate efficiently in virtual machines with a small memory footprint and limited local I/O capacity. It's coded for Linux in C++ using a custom event dispatch framework.

### BEA Systems (through acquisition of Plumtree)

*Senior Staff Engineer, Business Interaction Division*, October 2005-March 2006

- Tasked with planning and designing the integration of BEA technologies, security infrastructure, additional web services standards, and new Java technologies (e.g. the Spring IoC Container, Hibernate/EJB3 persistence, and OSGi) into ongoing Plumtree development efforts.
- With a team of two direct reports, developed strategy for transitioning Plumtree's monolithic Java portal code base to a scalable service oriented architecture while preserving cross-platform compatibility and application server agnosticism.

## Plumtree Software (through acquisition of Ripfire)

### *Principal Engineer, November 2001 - October 2005*

- Led ongoing development of the Ripfire search engine through three major Plumtree product releases. Responsible for international text support, scalability enhancements and porting to additional operating systems (AIX and Linux variants).
- Designed a horizontally scalable and fault tolerant version of the Ripfire search engine intended to operate on commodity hardware. Led team of four engineers in initial implementation phases; scheduled to ship in second quarter of 2006.
- Provided support and mentoring for other Plumtree engineering groups in areas including Unix development, system administration, software build and packaging, performance analysis, and multithreaded cross-language debugging.
- Led technical investigations of service oriented architecture, enterprise service buses, asynchronous messaging, and related open source technologies. Produced white papers, prototypes and recommendations for appropriate service decompositions in the next generation of Plumtree's portal product.
- Defined and prototyped a unified security architecture for Plumtree's family of applications, capable of providing fine-grained object-level security.

## Ripfire

### *Senior Software Engineer, May 2000 - November 2001*

- Shared responsibility for maintaining and enhancing the C++ server component of Ripfire's search engine.
- Designed and implemented file structures for the compact representation of phrase-encoded documents, run-length-encoded index file structures, scalable disk-based lexicon storage, and algorithms for efficiently searching this compressed data.
- Reworked locking strategy within the search engine to eliminate performance bottlenecks. The new strategy provided optimal query performance regardless of concurrent background tasks.
- Ported the search engine from Linux to Windows NT, Solaris, HP-UX, and Alpha Tru64 UNIX.
- Performed 64-bit conversion of the code and was responsible for major scalability enhancements.

## NARUS, Inc.

### *Staff Engineer, July 1999 – May 2000*

- Maintained, documented and improved NARUS's Semantic Traffic Analysis (STA) Platform—a distributed system for near-real-time application-level IP traffic analysis.
- Engineering project lead for version 1.0 of the NARUS Software Development Kit (shipped in February, 2000). Continued in role for SDK version 2.0. This SDK was an NT/Solaris/Linux cross-platform development environment for the STA Platform. It includes a graphical programming language editor, network traffic simulator, code library and documentation.

## Silicon Graphics Computer Systems, Inc.

### *Member of Technical Staff, Compiler Group, October 1997 – July 1999*

- Working from Intel's technical specifications, designed and implemented an assembler for the IA-64 processor architecture. Used during compiler bring-up on IA-64 simulators, this assembler was then optimized for use in the code generation path.
- Maintained and improved SGI's runtime loader and related utilities.
- Added code generator and linker support for a new ELF object file section recording detailed information about function interfaces in C, C++ and fortran code. Developed utilities to extract and compare this information for automated object/library compatibility tests.

*Member of Technical Staff, Tools Co-Op*, October 1995–October 1997

- Designed, implemented, and maintained PatchWorks—a revised interface and workflow system for SGI's patch database.

**Higher Order Software, Somerville, MA**

*Programmer*, 1984-1987 (Summers)

- Shared responsibility for maintaining and extending USE.IT—a software package for the design, analysis and generation of provably correct code in a variety of target languages.
- Developed a dependency mapping mechanism and user interface for USE.IT code modules.

---

## **Education**

**University of California at San Francisco, San Francisco, CA**

Department of Biochemistry and Biophysics (Structural Biology Program)

*Ph. D. in Biochemistry*, August 1995

- Howard Hughes Medical Institute Predoctoral Fellow, 1989-1994

**Carnegie Mellon University, Pittsburgh, PA**

*B.S. with University Honors in Biological Sciences*, 1989

- Double major in Computational Biology and Biochemistry
- National Science Foundation Undergraduate Fellowship, 1988