Michael L. Scott

Title: Linearizability of Persistent Memory Objects

Abstract: The prospect of ubiquitous nonvolatile main memory suggests the possibility of maintaining long-lived data unmediated by the file system. Since main memory is customarily seen through the filter of current register and cache contents, however, data must be carefully managed to ensure consistency in the wake of a crash. This talk will introduce formalism and design patterns for the implementation of correct persistent data structures. In keeping with "real world" systems, we assume a failure model in which all transient state (of all threads) is lost on a crash. We introduce the notion of *durable linearizability* to govern the safety of concurrent objects under this failure model and a corresponding relaxed, buffered variant which ensures that the persistent state in the event of a crash
is consistent but not necessarily up to date. At the implementation level, we present *explicit epoch persistency*, a formal model that builds upon and generalizes prior work. Our model captures both hardware buffering and fully relaxed consistency, and subsumes both existing and proposed instruction set architectures. Using the persistency model, we present an automated transform to convert any linearizable, nonblocking concurrent object into one that is also durably linearizable. We also present a design pattern, analogous to linearization points, for the construction of other, more optimized objects. Finally, we discuss generic optimizations that may improve performance while preserving both safety and liveness.

Biography: Michael L. Scott is the Arthur Gould Yates Professor of Engineering and past Chair of the Department of Computer Science at the University of Rochester. During the 2014 – 2015 academic year he was a Visiting Scientist at Google in Madison, WI. He received his Ph.D. from the University of Wisconsin–Madison in 1985. He is best known for work in synchronization algorithms and concurrent data structures, in recognition of which he shared the 2006 SIGACT/SIGOPS Edsger W. Dijkstra Prize. His textbook on programming language design and implementation (*Programming Language Pragmatics*, fourth edition, Morgan Kaufmann, Dec. 2015) and his monograph on *Shared Memory Synchronization* (Morgan & Claypool, 2013) are standard references in the field. Scott was named a Fellow of the ACM in 2006 and of the IEEE in 2010. In 2001 he received the University of Rochester's Robert and Pamela Goergen Award for Distinguished Achievement and Artistry in Undergraduate Teaching.