Debugging and improving the C/C++11 memory model

Viktor Vafeiadis, MPI-SWS

Abstract:

The 2011 C and C++ standards introduced a weak memory model, whose aim was to enable many compiler optimisations as possible, while providing some useful guarantees to the programmers. Sadly, however, as I will demonstrate, the formal model falls short of its intensions: it not only does not provide its expected DRF guarantee, but it also does not permit many common source-to-source program transformations (such as expression linearisation and “roach motel” reorderings) that modern compilers perform and that are deemed to be correct.

I will then focus on the release-acquire fragment of C/C++11, which is much better behaved, but still has some drawbacks. I will introduce a mildly stronger semantics for this fragment that forbids dubious behaviors that are not observed in any implementation, and admits an equivalent intuitive operational semantics. This strengthening has no additional implementation cost: it allows the same local optimizations as C11 release and acquire accesses, and has exactly the same compilation schemes to the x86-TSO and Power architectures. In fact, the compilation to Power is complete with respect to a recent axiomatic model of Power; that is, the compiled program exhibits exactly the same behaviors as the source one.

Bio:

Viktor Vafeiadis is a tenure-track researcher at the Max Planck Institute for Software Systems (MPI-SWS) working on software analysis and verification. He received his PhD in 2008 from the University of Cambridge, for which he was also awarded the ACM SIGPLAN annual distinguished dissertation award. Prior to joining MPI-SWS in 2010, Viktor was a postdoc at Microsoft Research and at the University of Cambridge.