Yehuda Afek

Title: Software/Hardware Lock elision: Amalgamated Lock-Elision

Abstract:

Hardware lock-elision (HLE) introduces concurrency into legacy lock-based code by optimistically executing critical sections in fast-path as hardware transactions. Its main limitation is that in case of repeated aborts it reverts to a fallback-path that acquires the serial lock. This fallback-path lacks hardware-software concurrency, because all fast-path hardware transactions abort and wait for the completion of the fallback. Software lock elision has no such limitation but the overheads incurred are simply too high.

We propose Amalgamated lock-elision (ALE), a novel lock-elision algorithm that provides hardware-software concurrency and efficiency: the fallback-path executes concurrently with fast-path hardware transactions, while the common-fast-path reads incur no overheads and proceed without any instrumentation. The key idea in ALE is to use a sequence of fine-grained locks in the fallback-path to detect conflicts with the fast-path, and at the same time, reduce the costs of these locks by executing the fallback-path as a series segments, where each segment is a dynamic length short hardware transaction.

We implemented ALE into GCC and tested the new system on Intel Haswell 16-way chip that provides hardware transactions. Micro-benchmarks on linked-lists, hash-tables and red-black trees, as well as a real-world scenario, in which we convert KyotoCacheDB to use ALE in GCC, all show that ALE is significantly faster than HLE.

Joint work with: Alex Matveev, Oscar R. Moll, and Nir Shavit.