Errata Slip

Revised 3/29/14
In the paper "Warranties for Faster Strong Consistency" by Jed Liu, Tom Magrino, Owen Arden, Michael D. George, and Andrew C. Myers, Cornell University (Friday session, "New Programming Abstractions," pp. 503-517 of the Proceedings)

Page 514, Related Work
Delete sentence: Memoized results are not shared across clients.

Original Text:

The TxCache system [45] provides a simple abstraction for caching and reusing results of functions operating over persistent data from a single storage node in a distributed system. As with the Fabric implementation of computation warranties, functions may be marked for memoization. TxCache does not ensure that memoized calls have no side effects, so memoized calls may not behave like real calls. Memoized results are not shared across clients. Compared to Fabric, TxCache provides a weaker consistency guarantee, transactional consistency, requiring that all transactions operate over data that is consistent with a prior snapshot of the system.

Corrected Text:

The TxCache system [45] provides a simple abstraction for sharing cached results of functions operating over persistent data from a single storage node in a distributed system. As with the Fabric implementation of computation warranties, functions may be marked for memoization. TxCache does not ensure that memoized calls have no side effects, so memoized calls may not behave like real calls. Compared to Fabric, TxCache provides a weaker consistency guarantee, transactional consistency, requiring that all transactions operate over data that is consistent with a prior snapshot of the system.

Revised 4/1/14
In the paper "Software Dataplane Verification" by Mihai Dobrescu and Katerina Argyraki, École Polytechnique Fédérale de Lausanne (Wednesday session, "Software Verification and Testing," pp. 101-114 of the Proceedings)

Page 102, Introduction
Replace sentence:

Original Text
We share common ground with many verification tools, especially the ones that use compositional symbolic execution [4, 22], but those were designed for different goals (increase line coverage or find bugs), so they do not solve our problem.

Corrected Text
We share common ground with many verification tools, especially the ones that use compositional symbolic execution [4, 20, 22], but, to the best of our understanding, those tools alone cannot solve our problem (they were not designed with software dataplanes in mind).