Chris Newcombe

Aggressive Performance Improvements via Formal Methods

Leslie Lamport's TLA+ and PlusCal

1. Describe the hard parts of your system (concurrency, fault-tolerance, protocols) in expressive, precise pseudo-code; typically a few hundred lines.
2. Describe precisely what your system should do (correctness properties), in the same language.
3. Use tools to thoroughly confirm that your system implements your desired correctness properties.

Move from "plausible prose" to precise statements much earlier in the development process.

Just the act of writing precise-pseudo code and precise correctness-properties often uncovers ambiguity and gaps.

Intel QuickPath cache-coherence protocol

"*There was a time when some optimizations to cache coherence protocols were avoided out of fear of "race cases"; formal specifications and model-checkers like TLA+/TLC gave us the confidence to do more aggressive optimizations.*

This was my experience … when Robert & I architected the initial version of the Quickpath cache coherence protocol"

- Brannon Batson, former Intel Architect

Intel Nehalem

"*The protocol optimizations we had made for Nehalem worked so well that they ended up creating a bottleneck in one of the microarchitectures.* ***The architects and designers, knowing where not to go in the design due to all our TLA+/TLC work, proposed a solution with minimal state machine interactions that "just worked".***

Interestingly, the solution involved mechanisms they had outright feared early in the project. *Their confidence in the microarchitecture let them go into a solution space they dared not tread previously and find a near-optimal solution.* ***I fully credit their minimal worry to the confidence everyone had from the TLA+/TLC backing.***"

- Robert Beers, Intel Architect