

# Review of Software and the Concurrency Revolution by Herb Sutter and James Larus

Riley Wood

March 29, 2016

## Key Ideas

The main goal of this paper is to describe the necessary trajectory for programming languages and compilers given the recent trend toward multiprocessor systems. Sutter and Larus highlight several key shortcomings in how concurrent programming is implemented today; they also summarize existing approaches to these problems as well as suggest what they believe to be the right solutions. They describe how the global nature of synchronization locks defies modularity and makes it hard to blackbox functionality, as a function that secretly uses a lock could cause deadlock to occur. Locks also have no easy way of being coupled with the data they protect - it is simply up to the programmer to remember how locks and data are correlated.

## Review

Surely there is some framework for correlating locks with the data they protect. I wonder how LabView would be categorized in terms of its concurrency. LabView is essentially parallelized by default. Operations are executed as soon as their inputs are available, and you can have many concurrent data paths all execute at once. A visual programming approach makes it easier to think about concurrency, since you can see all of the datapaths at once.

## Conclusions