The Fixed Logical Execution Time Assumption

Thomas A. Henzinger, *University of California, Berkeley*

Abstract:

A central challenge in real-time programming is the definition of a programming model at a level of abstraction that supports both implementability and verifiability. If a programming model is too close to an abstract specification, then it is difficult to generate efficient code. On the other hand, if a programming model is too close to the execution platform, then the gap between specificationand program is difficult to bridge. Many traditional real-time programming models are based on priorities. These models are arguably not sufficiently abstract, and the resulting code is often unpredictable with respect to both timing (jitter) and function (data races). Some newer programming models are based on the synchrony assumption, which postulates that computation is infinitely faster than the physical environment. These programming models are often too abstract and difficult to compile onto resource-constrained and distributed platforms. We present a novel real-time programming model, based on the FLET (fixed logical execution time) assumption, which is less abstract than synchronous models but more abstract than priority-based models. We demonstrate that FLET-based programming leads to predictable, composable, portable, and efficient real-time code.