Partially synchronous programming abstractions for fault-tolerant distributed algorithms

Cezara Drăgoi[1], Thomas A. Henzinger[2], Josef Widder[3], and Damien Zufferey[4]

[1] INRIA/ENS/CNRS, France
[2] IST Austria
[3] TU Wien, Austria
[4] MPI Kaiserslautern, Germany

Fault-tolerant distributed algorithms play an important role in many critical/high-availability applications. These algorithms are notoriously difficult to implement correctly, due to asynchronous communication and the occurrence of faults, such as the network dropping messages or computers crashing.

One fundamental obstacle in having correct fault-tolerant distributed algorithms is the lack of abstractions when reasoning about their behaviors. In this talk we discuss the impact of partially synchronous programming abstractions in increasing the confidence we have in fault-tolerant systems. We will focus on partially synchronous models that view asynchronous faulty systems as synchronous ones with an adversarial environment that simulates asynchrony and faults by dropping messages. This view simplifies the proof arguments making systems amendable to automated verification. We apply partial synchrony to algorithms that solve agreement problems, such as consensus and state machine replication.

Technically, we take a programming language perspective and define a domain specific language which has a high-level partially synchronous semantics and compiles into efficient asynchronous code. We validate our technique by defining partially synchronous implementations of algorithms like Paxos, whose verification becomes now automated, and which compile into efficient asynchronous code, that preserves the properties verified under the partially synchronous semantics.