Distributed runtime verification*

where combinatorics, fault-tolerance and formal methods meet

Sergio Rajsbaum

Instituto de Matemáticas, Universidad Nacional Autónoma de México, D.F. 04510, Mexico

Abstract of Keynote Talk

Runtime verification RV techniques are concerned with monitoring software and hardware system executions. They are complementary, and sometimes more versatile than conventional testing, and more practical than exhaustive formal verification, such as model checking and theorem proving, as well as incomplete solutions such as testing and debugging. There is an international conference, *RV* dedicated to these techniques.

Distributed runtime verification This talk gives an overview of distributed runtime verification (DRV). Building a decentralized runtime monitor for a distributed system is an especially difficult task since it involves designing a distributed algorithm that coordinates the monitors in order for them to reason consistently about the temporal behavior of the system. DRV techniques are less developed; they involve designing a distributed algorithm that monitors another distributed algorithm.

In an asynchronous system where processes may crash, it is impossible for the monitors to agree on the order of events in the system, due to the impossibility of solving consensus. Thus, it is unavoidable that monitors emit different opinions about the validity of the computation, that nevertheless, should be consistent with each other. Lower and upper bounds on the number of opinions that may have to be emitted, can be derived, as a function of the specification φ that is being monitored.

At the crossroads where distributed algorithms and formal methods meet An overview of the different types of techniques used in DRV is presented, which range from formal methods techniques related to LTL and multi-valued logics, on the one hand, to algorithmic techniques related to computing snapshots in an efficient manner to reason about temporal properties of a distributed system, on the other hand, and passing through combinatorial and topological techniques. RV is an exemplary area for interdisciplinary research opportunities, given that logic and algorithmic techniques converge, and few papers explore the difficulties introduced when failures and asynchrony can occur in the system.

Acknowledgements The results presented involve joint work with Borzoo Bonakdarpour, Pierre Fraigniaud, Matthieu Roy, David Rosenblueth and Corentin Travers. Some of them have been published in DISC'11, OPODIS'14, RV'14, and *Distributed Computing* (2013).

* Supported by a UNAM-PAPIIT Grant