

Accountable Distributed Computing

Petr Kuznetsov ✉

LTCI, Télécom Paris, Institut Polytechnique de Paris, France

Abstract

There are two major ways to deal with failures in distributed computing: *fault-tolerance* and *accountability*. Fault-tolerance intends to anticipate failures by investing into replication and synchronization, so that the system's correctness is not affected by faulty components. In contrast, accountability enables detecting failures *a posteriori* and raising undeniable evidences against faulty components.

In this talk, we discuss how accountability can be achieved, both in generic and application-specific ways. We begin with an overview of fault detection mechanisms used in benign, *crash-prone* system, with a focus on the *weakest failure detector* question. We then consider the fault detection problem in systems with general, *Byzantine* failures and explore which classes of misbehavior can be detected and which – cannot. We then study the mechanism of *application-specific* accountability that, intuitively, only accounts for instances of misbehavior that affect particular correctness criteria. Finally, we discuss how fault detection can be combined with *reconfiguration*, opening an avenue of “self-healing” systems that seamlessly replace faulty system components with correct ones.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Fault-tolerance, fault detection, accountability, application-specific

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2021.2

Category Invited Talk



© Petr Kuznetsov;

licensed under Creative Commons License CC-BY 4.0

25th International Conference on Principles of Distributed Systems (OPODIS 2021).

Editors: Quentin Bramas, Vincent Gramoli, and Alessia Milani; Article No. 2; pp. 2:1–2:1

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany