An abstract framework for faulty distributed systems

Denisa Diaconescu
University of Bucharest
Runtime Verification Inc.

In this talk we present the theory of *VLSMs* – validating labelled state transition and message production systems – a theoretical tool for specifying and formally analysing faulty distributed systems [3]. The theory of VLSMs is in alignment with the *correct-by-construction* ideology: define an abstract class of consensus protocols (satisfying some generic abstract properties), prove general safety and liveness results for the protocols belonging to that class, and then construct correct-by-construction protocols by concretely instantiating the abstract components, or, alternatively, prove that concrete protocols satisfy those requirements.

The central problem we investigate in this talk is that of *detecting equivocation*. Byzantine faulty components in a distributed system might behave in an arbitrary way in order to disrupt the operations of the system [2]. An important subset of byzantine behaviour is that of equivocation behaviour. In the consensus literature, equivocation refers to claiming different beliefs about the state of the protocol to different parts of the system in order to steer the protocol-following components into making inconsistent decisions; messages received from equivocating components seem to be valid messages [1]. For example, if a system is trying to come to consensus about the value of a bit, an equivocating component may claim to think the bit is 0 to one part of the system, and 1 to another part. Equivocation behaviour of a component cannot be produced by a single execution of the protocol, but could be produced by more than one protocol execution, i.e., an equivocating component behaves as-if running multiple copies of the protocol. It turns out that it is easier to detect equivocating components than byzantine components which can behave completely erratic.

In consensus protocols, it is common for components to *validate* received messages in order to ensure that they are not malformed. We formalise this idea into a general formal notion of *validators*. We are then able to show that, in the context of a distributed system without synchronisation assumptions, the effect that byzantine components can have on honest validators is no different than the effect equivocating validators can have on non-equivocating validators. Replacing byzantine components with equivocating validators forms the foundation for an alternative to byzantine fault tolerance analysis. This work opens the way for protocol designers to reason precisely about different types of faults and to thereby create more robust consensus protocols than are possible when budgeting for byzantine faults.

Our definitions and results have been formalised and machine-checked in the Coq proof assistant.[1] The formalisation is compatible with Coq 8.15 and uses the Coq-std++ library version 1.6.03.

[1] `https://github.com/runtimeverification/vlsm/releases/tag/v1.1`

Submitted to:
FROM 2022

References

[1] Allen Clement, Flavio Junqueira, Aniket Kate & Rodrigo Rodrigues (2012): *On the (limited) power of non-equivocation*. In: *Symposium on Principles of Distributed Computing*, pp. 301–308.

[2] Leslie Lamport, Robert Shostak & Marshall Pease (1982): *The Byzantine Generals Problem*. *ACM Transactions on Programming Languages and Systems* 4(3), pp. 382–401.

[3] V. Zamfir, M. Calancea, D. Diaconescu, B. Moore, K. Palmskog, T. Şerbănuţa & M. Stay (2022): *VLSM: Validating Labelled State Transition and Message Production Systems*. *doi:10.48550/arXiv.2202.12662*.