Verifying Liveness Properties of Distributed Systems with Verdi

Saswat Padhi · Lun Liu

20 January 2017

Motivation

The last couple of decades have witnessed a tremendous growth in distributed systems, especially after the arrival of the Internet. In the distributed computing paradigm, several interconnected computing systems solve a single large scale problem by breaking it down to several simple tasks that handled by individual systems. But owing to the scale and complexity of these systems, guaranteeing desired properties about a distributed system as a whole is extremely difficult.

Verdi is a recent attempt to help programmers implement distributed systems which are correct by construction. It is implemented using the Coq proof assistant and the systems are extracted to OCaml for execution. Although Verdi supports verification of safety properties of systems, proving liveness properties is not supported. The current repository seems to have one been updated with one example of verifying liveness property of a simple lock server, but it has only been verified with idealized network semantics. It is not known whether the transformations defined by the existing transformers are sufficient to reason about liveness of the transformed system. We want to investigate the support for verifying liveness properties in Verdi, and extend it as required for a simple use case.

Goals

Our goal is to demonstrate verification of liveness properties of distributed systems using Verdi, by extending the framework as needed. When using Verdi to prove properties of systems, one needs to verify application-level guarantees under an idealized fault model first and then apply a verified transformer (e.g. Raft replication transformer that handles node failures) to obtain a system that tolerates more faults while still providing analogous guarantees. Thus, to support verifying liveness properties using Verdi, we have to verify that the transformers preserve liveness properties of systems.

As a case study, we will implement a simple distributed system using Verdi, and prove both its liveness and safety under perfect network semantics and under adversarial settings. For demonstrating verification of liveness properties, we will formalize a simple liveness property that we would verify under idealized network semantics and prove to be preserved by a certain transformer. In the case that the liveness property cannot be verified for the transformed system, we would investigate ways to augment the transformations to enable this.

Milestones

0. Understand the Verdi framework and its model (network semantics etc) by building a simple distributed system and verifying some safety property for it.

1. Formalize an application-level liveness property for this simple system and prove it under idealized network semantics, possibly using approaches similar to the simple example currently in the repository.

2. Verify that one of Verdi's existing system transformers preserves analogous liveness guarantees on the transformed system.

   (a) If the verification does not succeed, enrich the transformation as necessary.

   (b) Investigate if there is a generic modification that may be applied to all transformers to enable this.

3*. Extract an executable system with the transformer applied to the simple distributed system and compare its performance to an unverified implementation.

* = at the risk of exposure to unhealthy levels of caffeine.