Tough on the Outside, Reliable on the Inside: Utilizing System Composition for Improved Resilience

Laura Lawniczak and Tobias Distler
Friedrich-Alexander-Universität Erlangen-Nürnberg

Abstract

The notoriously high complexity of Byzantine fault-tolerant (BFT) protocols leads to a high risk of bugs and vulnerabilities and makes diversification costly. We aim to increase the resilience of BFT systems by introducing a new hybrid fault model focussing on the composition of the system. SHELLFT structures a BFT system into two main parts: 1. A BFT shell that handles client interaction and is connected to the outside world, and 2. a crash fault-tolerant core providing the protocol logic that is only connected to other replicas. With this approach, we aim to significantly reduce the code base of the BFT parts, making it easier to find vulnerabilities and use diversification. Exploiting the intrinsic modularization of the design paradigm micro replication, we devise a SHELLFT variant of the Mirador protocol.

1 Problem Statement

The main goal of Byzantine fault tolerance is to ensure integrity and availability of a system in the presence of arbitrary behavior of parts of the system (potentially caused by attacks). This can be achieved by using state-machine replication in combination with BFT protocols to reach consensus on a total order of commands on all (correct) replicas. Without further specialized hardware, this usually requires $3f + 1$ replicas to tolerate f faulty nodes and is commonly complemented by diversification (e.g., by using different hardware and/or n-version programming) to reduce the chance that a successful attack does affect more than f replicas.

However, BFT systems are notoriously complex, resulting in them being difficult to formally verify and often comprising extensive code. Unfortunately, this increases the danger of bugs or vulnerabilities, counteracting the purpose of BFT.

Amongst the large amount of work tackling this complexity, one area focuses on adjusting the fault model to a more realistic (and usually mitigated) scenario, often referred to as *hybrid fault model*. However, existing hybrid fault models often make assumptions regarding the *whole* system environment and even short disturbances can break these. For example, both fault models XFT [2] and VFT [3] assume a synchronous network for at least part of the nodes and unfortunate combinations of asynchronicity and Byzantine faults can endanger the consistency of the system. With SHELLFT, we therefore propose a different kind of hybrid fault model: Focusing on the composition of the system itself instead of general assumptions about the environment.

2 SHELLFT: Basic Approach

On closer examination of existing BFT systems, we made the following observations: Certain functionality, especially the client interface for handling requests and responses, needs to be directly connected to the outside world and is hence more exposed against potential attacks. For this reason, we expect this part of the system to be more likely to be taken over by an attacker and consider it potentially Byzantine faulty. Other functionality, however, for example the agreement protocol itself, only requires communication within the replica group and needs no direct connection to the outside world. This makes it easier to safeguard against attacks, hence we can assume no Byzantine behavior but only failures by crashing.

We therefore propose the following system model consisting of two main parts: An outside BFT shell handling client interaction and a crash fault-tolerant (CFT) core consisting of the agreement protocol and other internal functionality. Of course, this separation only works when the two parts are also split physically, so do not reside on the same hosts.

As monolithic replicas often use data structures across multiple protocol phases, it is challenging to separate functionality as cleanly as required for our approach. Hence, we leverage micro replication [1], a recently presented design approach that splits a protocol into specialized and individual micro replicas, each responsible for only a single protocol phase. This clean separation allows us to easily adapt micro-replicated protocols to the SHELLFT fault model, which we validate by designing a SHELLFT variant of Mirador [1].

As shown in Figure 1, we organize the micro replicas in the Mirador protocol into three layers:

- **Outer Shell:** All replicas that communicate with clients (i.e., the outside world) need to be considered to be more exposed to attacks and are in the BFT shell of the system.
- **Filter:** Replicas that have no contact to the outside world can be considered to fail only by crashing, but need to be

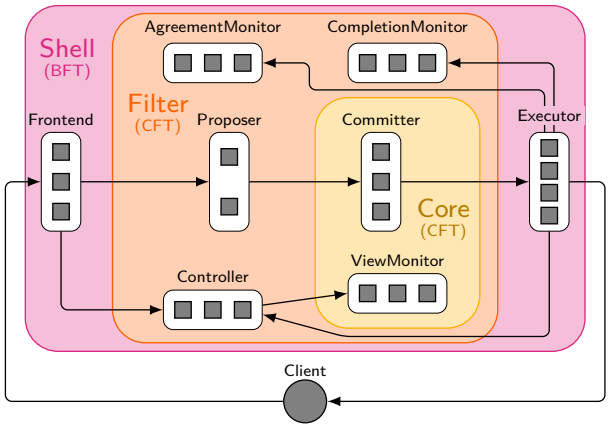

Figure 1: SHELLFT fault model applied to the micro-replicated protocol Mirador. For simplicity, some message flows irrelevant to the fault model are omitted.

split (logically) into two categories. In the filter category, a micro replica will not introduce any Byzantine faults but may be subjected to them via information from shell replicas. Hence, micro-replicas groups in the filter need to be seen as a blend between Byzantine and crash fault-tolerant: They must be able to tolerate Byzantine input but cannot produce any Byzantine output.

- **Inner Core:** Micro-replica groups in the core have neither contact to the outside world nor receive potentially Byzantine input. Hence, these replicas can use both a crash fault-tolerant configuration and implementation.

With this approach, we aim to **reduce the complexity of the BFT part** of the system, making it **more resilient** and **easier to diversify**. Additionally, by using **less costly CFT functionality** in the core and filter, we assume the system as a whole to be **more resource efficient in certain scenarios**.

References

[1] Tobias Distler, Michael Eischer, and Laura Lawniczak. Micro replication. In *Proceedings of the 53rd International Conference on Dependable Systems and Networks (DSN '23)*, pages 123–137, 2023.

[2] Shengyun Liu, Paolo Viotti, Christian Cachin, Vivien Quema, and Marko Vukolic. XFT: Practical fault tolerance beyond crashes. In *Proceedings of the 12th Symposium on Operating Systems Design and Implementation (OSDI '16)*, pages 485–500, 2016.

[3] Daniel Porto, João Leitão, Cheng Li, Allen Clement, Aniket Kate, Flavio Junqueira, and Rodrigo Rodrigues. Visigoth fault tolerance. In *Proceedings of the 10th European Conference on Computer Systems (EuroSys '15)*, pages 1–14, 2015.