Testing Zyzzyva

Ishan Pahwa[1], Joao Neto[2], Burcu Ozkan[3]

Technische Universiteit Delft

Introduction

- Distributed systems, such as the ones used in Blockchain, Cryptocurrencies, Peer-to-Peer networks (P2P) and cloud computing are incredibly important.
- Byzantine Fault tolerant algorithms attempt to give a guarantee of liveness and safety against the byzantine generals problem in distributed systems.
- Zyzzyva is one such Byzantine fault tolerant algorithm that attempts to give a liveness and safety guarantee. It introduces speculation, which cuts down on the number of stages for the protocol to the theoretical minimum [2].

The Liveness property is a guarantee that the distributed system does eventually reach a consensus.

The Safety property is a guarantee that the distributed system never reaches an incorrect result.

ByzzFuzz introduces a novel solution for the testing of BFT algorithms by introducing small-scope mutations as well as any-scope mutations [3].

Research Questions

1. Can ByzzFuzz implemented in ByzzBench find faults in the Safety and Liveness guarantees that Zyzzyva claims to provide?
2. What is the difference in performance between ByzzFuzz, a baseline testing method and possibly Twins?
3. How do small-scope and any-scope mutations in ByzzFuzz compare in the performance of bug detection?

Methodology

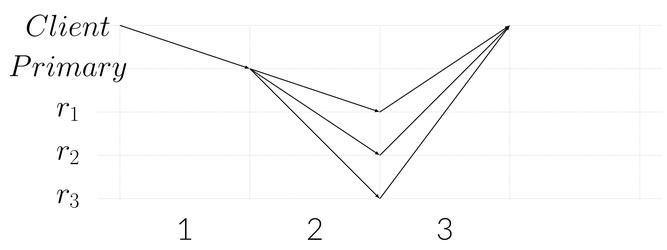

Figure 1. The fast-track in Zyzzyva

- We implemented Zyzzyva on ByzzBench
- We implemented a variation of Zyzzyva on ByzzBench with a safety issue
- We tested Zyzzyva using different parameters on ByzzBench

Experimentation and results

We found that ByzzFuzz and the baseline testing method were not able to find any liveness or safety violations in Zyzzyva, including the one introduced by Abraham et al.[1]. Twins wasn't able to find any faults with our implementation.

We introduced a faulty version of Zyzzyva that violates its safety and ran the three testing strategies on it. We find that ByzzFuzz and the baseline are able to find the violation but Twins isn't.

We also find that small-scope mutations are generally accepted more by the system and therefore can possibly violate safety or liveness in Zyzzyva.

Discussion

ByzzFuzz in general is able to catch the safety violation that we inject into Zyzzyva, possibly due to its round-aware injection that simulates Byzantine behaviour better than random injections.

We also discuss ByzzFuzz's limitations when strictly adhering to its criteria as a network wrapper. We discuss the limitations in terms of not being able to produce past values in Zyzzyva's history without access to its state.

Conclusion

We find that neither of our testing methods are able to find the previously known safety flaws in Zyzzyva. This could be due to relatively small number of scenarios run as well as the precise nature of the mutations applied.

We find that ByzzFuzz and our baseline are able to find the safety violation that we inject into Zyzzyva, with the former performing better, showing promise but this requires more testing. Twins does not manage to find a violation with the correct and faulty cases of Zyzzyva, possibly due to the way we compute rounds.

Small-scope mutations seem to cause less errors in our faulty implementation implying that they are more readily acceptable by the system and therefore have the potential to materialize violations with a higher frequency.

References

[1] Ittai Abraham, Guy Gueta, Dahlia Malkhi, Lorenzo Alvisi, Rama Kotla, and Jean-Philippe Martin.
Revisiting Fast Practical Byzantine Fault Tolerance, December 2017.

[2] Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong.
Zyzzyva: Speculative Byzantine Fault Tolerance.

[3] Levin N. Winter, Florena Buse, Daan De Graaf, Klaus Von Gleissenthall, and Burcu Kulahcioglu Ozkan.