Optimized BFT Replication from Authenticated Logging

Hanish Gogada[1], Christian Berger[2], Leander Jehl[1], Hans P. Reiser[3], and Hein Meling[1]

Abstract: Byzantine fault-tolerant (BFT) state machine replication (SMR) protocols experience increased attention, due to their envisioned utilization in blockchain systems. Several novel BFT SMR protocols that improve scalability assign specific roles to individual replicas, and at the same time, introduce a yet unsolved issue because the role assignment is highly sensitive to faults. In this presentation, we introduce SmartLog, a logging framework that collects and analyzes metrics for smart configuration decisions to improve performance despite the presence of faults. SmartLog presents local measurements in global data structures, to enable consistent decisions and hold replicas accountable if they do not perform according to their reported measurements.

1 Problem Statement and Motivation

Many BFT protocols improve scalability using optimization strategies effective under specific favorable conditions (e.g., [Go19, NMR21, St22]). In adverse scenarios, these optimizations become less effective or even defunct, necessitating a switch to a more resilient but less effective protocol. It is non-trivial to detect whether or not the operating conditions are favorable or adversarial. Often approaches rely on repeated trial-and-error to find a working system configuration and ignore the actual operating conditions, such as the latency between replicas and prior misbehavior, thus resulting in poor performance. Many systems that consider the operating conditions rely only on local measurements to select a system configuration (for example [MOR09]). This is problematic because local measurements across the replicas may be inconsistent, making it challenging to make global configuration decisions. In the Byzantine fault model, there is a lack of transparency and trust if a single replica, e.g., the leader, can make such decisions based only on its local measurements since it is impossible to verify that the decision was based on actual measurements.

2 SmartLog for Replicated State Machines

We advocate for a holistic, measurement-based approach to accurately identify the current operating conditions, promoting aggressive use of efficient protocols and reducing fallback to less efficient ones. We accomplish this through a shared append-only log of measurements.

SmartLog is an integrated shared log for recording various metrics and computing efficient system configurations from these metrics. SmartLog extends a generic RSM with sensors

[1] University of Stavanger, Stavanger, Norway, hanish.gogada@uis.no; leander.jehl@uis.no; hein.meling@uis.no
[2] Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany, berger@cs.fau.de
[3] Reykjavik University, Reykjavik, Iceland, hansr@ru.is

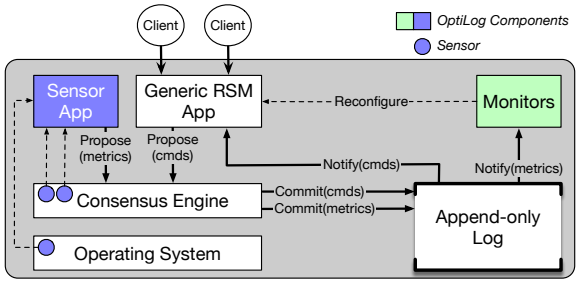

Fig. 1: SmartLog's component architecture

and monitors to capture and evaluate different metrics (see Figure 1). Individual replicas instrumented with sensors record measurements in the log, and corresponding monitors at replicas collate these measurements into data structures that are used to *compute and deploy efficient configurations*. Furthermore, SmartLog enables replicas to make consistent configuration decisions based on the same information. The log also provides transparency, allowing all replicas to verify decisions and recognize faulty behaviors. Moreover, some optimization techniques can be costly to (deterministically) evaluate for large configuration sizes. Thus, SmartLog allows heuristic optimization techniques to be used, where the resulting non-deterministic configurations are logged, such that the replicas can consistently determine a global ranking of configurations. Additionally, SmartLog supports collaborative optimization techniques where the search space is partitioned and distributed across replicas. To showcase the potential of our implementation, we apply it to Kauri [NMR21] to boost its performance in heterogenous networks by selecting a fast tree configuration which is decided based on logged replica latency measurements and indications about faulty replicas.

Bibliography

[Go19] Golan Gueta, Guy; Abraham, Ittai; Grossman, Shelly; Malkhi, Dahlia; Pinkas, Benny; Reiter, Michael; Seredinschi, Dragos-Adrian; Tamir, Orr; Tomescu, Alin: SBFT: A Scalable and Decentralized Trust Infrastructure. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 568–580, 2019.

[MOR09] Merideth, Michael G.; Oprea, Florian; Reiter, Michael K.: When and How to Change Quorums on Wide Area Networks. In: 2009 28th IEEE International Symposium on Reliable Distributed Systems. pp. 12–21, 2009.

[NMR21] Neiheiser, Ray; Matos, Miguel; Rodrigues, Luís: Kauri: Scalable BFT Consensus with Pipelined Tree-Based Dissemination and Aggregation. In: Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles. SOSP '21, Association for Computing Machinery, New York, NY, USA, pp. 35–48, 2021.

[St22] Stathakopoulou, Chrysoula; David, Tudor; Pavlovic, Matej; Vukolić, Marko: [Solution] Mir-BFT: Scalable and Robust BFT for Decentralized Networks. Journal of Systems Research, 2(1), 2022.