A Framework for Consensus-Agnostic State-Machine Replication based on Threshold Signatures

Alexander Heß
Franz J. Hauck
alexander.hess@uni-ulm.de
franz.hauck@uni-ulm.de
Institute of Distributed Systems, Ulm University
Germany

1 Motivation

State-machine replication is an established fault-tolerance technique that can be leveraged to build highly-available services [5]. Consensus protocols have emerged as the primary mechanism to enforce deterministic request ordering among the replicated servers [3]. Over the last two decades, a multitude of consensus protocols have been published that come with different characteristics but also with different communication and programming models. Most consensus protocol require specific client-side code for the interaction with the replicas. As a consequence, clients have to be aware of the internals of a given SMR system configuration.

We developed a communication wrapper, which provides a generic client-service interface and allows clients to be consensus agnostic. Hereby, we utilized BLS threshold signatures [2] to securely outsource the specific client responsibilities to the server-side. The threshold signature scheme allows to derive a group signature from a set of individual signatures created by different replicas. The group signature is used as a proof for a majority response for a given request invocation. With this approach, a client can simply accept the first response with a valid group signature and discard the remaining responses.

2 Approach

Our communication wrapper is called *Consensus-Agnostic Replication Toolkit (CART)* and is a framework-based approach that allows to wrap existing implementations of consensus protocols with minimal amounts of glue code. The high-level architecture of our approach is depicted in section 2. Hereby, the CART handlers on the client and server-side provide a generic interface, independent of the technology stack used by the wrapped consensus protocol. Dedicated replicas relay requests received by clients to the consensus protocol through the corresponding BFT client handler. After a decision has been reached by the consensus protocol, an application proxy intercepts the ordered request which is then submitted to the server-side application. The result of the request invocation is returned to the consensus protocol, and also forwarded to the server-side CART where it is signed by each replica. These signatures are then exchanged among the replicas, and afterwards t signatures are aggregated into a group signature, whereby t is equal to the number of matching responses required for a majority vote. The resulting group signature is returned to the client along with the result of the request invocation. Our approach is designed in such a way that clients are guaranteed to eventually receive a valid response, even if up to f replicas are faulty.

We developed a prototype implementation of the CART, which we used to conduct a first performance evaluation with the two SMR frameworks *BFT-SMaRt* [1] and *Themis* [4]. Since the mathematical construction of the BLS signature scheme has a significantly higher computational overhead compared to ECDSA signatures for example, the measurements conducted with our first prototype yielded poor performance numbers. At the same time, this mathematical construction provides certain isomorphisms that can be leveraged to reduce the number of computationally expensive crypto operations per request invocation. This allowed us to implement different optimization strategies which lead to significantly better performance. The talk will provide an overview of the CART approach, with a focus on certain implementation aspects including these optimization strategies.

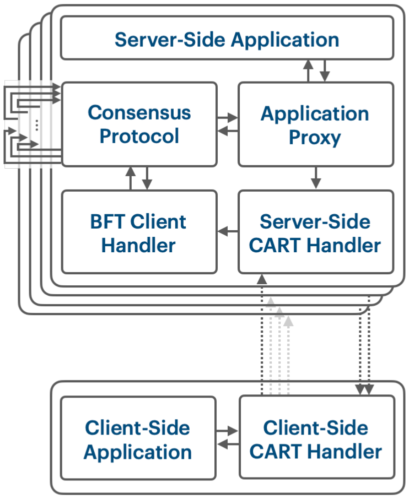

Figure 1. A high-level overview of the CART architecture.

References

[1] Alysson Neves Bessani, João Sousa, and Eduardo Adílio Pelinson Alchieri. 2014. State Machine Replication for the Masses with BFT-SMART. In *44th Annual IEEE/IFIP Int. Conf. on Dep. Syst. and Netw. (DSN)*. IEEE Computer Society, 355–362. https://doi.org/10.1109/DSN.2014.43

[2] Dan Boneh, Ben Lynn, and Hovav Shacham. 2004. Short Signatures from the Weil Pairing. *J. Cryptol.* 17, 4 (2004), 297–319. https://doi.org/10.1007/s00145-004-0314-9

[3] Miguel Castro and Barbara Liskov. 2002. Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Trans. Comput. Syst.* 20, 4 (2002), 398–461. https://doi.org/10.1145/571637.571640

[4] Signe Rüsch, Kai Bleeke, and Rüdiger Kapitza. 2019. Themis: An Efficient and Memory-Safe BFT Framework in Rust: Research Statement. In *Proceedings of the 3rd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers* (Davis, CA, USA) *(SERIAL '19)*. Association for Computing Machinery, New York, NY, USA, 9–10. https://doi.org/10.1145/3366611.3368144

[5] Fred B. Schneider. 1990. Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial. *ACM Comp. Surv.* 22, 4 (1990), 299–319. https://doi.org/10.1145/98163.98167