HQ Replication: A Hybrid Quorum Protocol for Byzantine Fault Tolerance

James Cowling, Daniel Myers, Barbara Liskov, Rodrigo Rodrigues and Liuba Shrira

Summarised: April 3, 2017

1 Main points

HQ Replication is about combining two known approaches for providing Byzantine-fault-tolerant state machine replication. One is query based approach, and the other is replica based approach. Query based approach is used when there is no contention, this is the case for the majority of time the system is running. This approach is very efficient because it only requires responses from the quorum of replicas to achieve state change. This approach scales very well as the number of replicas grows, since contention is more rare for this case a replica-based approach is used, with all-to-all communication to agree on the state under contention.

2 Protocol

2.1 Normal case

Client execcutes a command through client proxy. Write commands are executed in two phases, the first one is to get the quorum of $< OK >$ messages called grants, when the clients get the quorum it creates a certificate. With this certificate it can safely execute phase 2 of write operation, but it is still up to the replicas how they will process it, it might be that operation has already been completed by the time the original client sents phase-2-write request this happens usually during contention when the BFT (replica-based module) takes over.

2.2 Contention

Contention occurs when several clients are trying to write at the same timestamp. Clients notice contention after phase-1 write, they will receive conflicting grants for the same timestamp. At this point client will send a $< Resolve >$ to the replicas. There might be multiple $< Resolve >$ requests in the system, and normally they will all be resolved in order. Te replica that received a $< Resolve >$ is frozen and will delay all other client requests untill resolution. The replica will create a $< Start\ resolution >$ message to the server proxy running current "primary" or "leader" of the BFT module. When the "primary" receives a quorum of $< Start\ resolution >$ messages, it will procede with contention resolution.

3 Possible questions

Q: Is the BFT module active during operations without contention?
A: No BFT is only listening to $< \textit{Start contention resolution} >$ messages from replicas, and resolfs contention when quorum of these messages is achieved.
Q: How do clients know that there has been a contention?
A: Clients will receive conflicting phase-1 responses, called grants. Grants with the same timestamp but different operations mean that two clients are trying to execute at the same time, but also a faulty client might have sent different pahse-1 request to different servers. In any way client sends $< \textit{Resolve contention} >$ to the replicas.