Paper Review

CS6378 - Advanced Operating Systems

Aravind Menon Kadekuzhi
axk210275

Topic: Blockchain Consensus Protocols in the Wild

Christian Cachin, Marko Vukolic’

In this paper, the author provides a comprehensive overview of consensus protocols utilized in permissioned blockchains, examining their underlying principles, resilience, and trustworthiness. Blockchains are robust distributed ledger systems that offer reliable services to parties and nodes that may not have complete trust in each other. They support a variety of tasks, including smart contracts, and exist in different forms like permissionless, permissioned, and private blockchains. To ensure security solutions are reliable and effective, it is crucial to avoid ad-hoc approaches and "security by obscurity." Instead, security solutions must be based on well-defined security models and trust assumptions that are validated using mathematical reasoning and formal tools.

The author conducts a thorough analysis and comparison of consensus protocols in permissioned and permissionless blockchain platforms, while also exploring non-BFT consensus mechanisms.

To achieve consensus among distributed nodes, state-machine replication is a widely adopted approach that incorporates the service logic and a consensus protocol for disseminating requests among nodes. The eventual-synchrony network model is utilized to design resilient distributed systems, providing optimal resilience and avoiding assumptions about synchronized clocks and timely network behavior.

In blockchain systems, consensus protocols like the Nakamoto protocol employed in Bitcoin are used to establish consensus on a shared ledger through voting among nodes. Today, blockchain platforms can implement any consensus mechanism that matches their trust model. Consequently, existing consensus and replication mechanisms are receiving renewed attention for their potential application in blockchain systems.

The paper explores crash-tolerant consensus in distributed systems that are prone to node crashes. Atomic broadcast protocols like Paxos, Viewstamped Replication (VSR), Zab, and Raft are examined as they provide reliable message ordering. These protocols utilize a sequence of views with unique leaders and view change protocols to ensure agreement.

Byzantine consensus protocols, such as PBFT, are designed to tolerate malicious nodes. These protocols extend the Paxos/VSR family and can tolerate up to $f < n/3$ Byzantine nodes. BFT-SMaRt is acknowledged as the most advanced implementation of a Byzantine fault-tolerant consensus protocol.

Validation is a critical aspect of consensus which ensures that only "valid" transactions are given by the broadcast protocol. An external predicate, V(), formalizes this process. The majority of consensus protocols examined in the paper combine validation with ordering to prevent denial-of-service attacks.

Tangaroa, a BFT consensus protocol, is deemed neither live nor safe due to its inability to withstand malicious nodes. It violates liveness by permitting a malicious node to be elected as a leader and halt the protocol, and it violates safety because it fails to guarantee that all correct nodes deliver the same messages in the same order.

The author explores various consortium blockchain systems, including Hyperledger Fabric, Tendermint, Symbiont Assembly, R3 Corda, Iroha, Kadena, Chain, Quorum, and MultiChain. The implemented protocols assume independence among failures, selfish behavior, and subversion of nodes, and support a numerical level of confidence depending on the proportion of possibly flawed nodes.

Hyperledger Fabric, a distributed ledger platform, offers high levels of confidentiality, resilience, flexibility, and scalability. The consensus protocol used up to release v0.6-preview was a native implementation of PBFT. With Fabric V1, an Apache Kafka cluster can provide the ordering service responsible for conflict avoidance. Kafka has high throughput and low latency, and Fabric inherits its crash resilience from ZooKeeper. A second implementation using PBFT for the ordering service is currently under development. BFT-SMaRt is also being integrated into Fabric V1 as an ordering service.

Tendermint Core is a Byzantine Fault Tolerant (BFT) protocol, which is similar to PBFT. Clients use a gossip protocol to disseminate transactions to validators. The primary difference between Tendermint and PBFT is the continuous rotation of leaders. Despite having a livelock bug, the protocol has additional mechanisms that prevent it from occurring.

The paper highlights several blockchain platforms tailored to the financial industry. Symbiont Assembly is a proprietary distributed ledger platform that utilizes the open-source BFT-SMaRt toolkit to achieve a high throughput of 80,000 transactions per second.

R3 Corda differs from other platforms since it does not order transactions in a single blockchain. Instead, it defines states and transactions in a Hash-DAG. A notary service in Corda orders and timestamps transactions and notaries can be based on Raft or BFT-SMaRt, each with different resilience features.

Iroha is an open-source blockchain platform modeled after Fabric's original design. It relies on the Sumeragi consensus library, which draws heavily from BChain, and employs standard BFT consensus assumptions in the eventually-synchronous model.

Kadena's Juno, a smart contract platform, initially relied on the Byzantine Fault Tolerant Raft protocol for consensus. However, it was later deprecated in favor of ScalableBFT, a proprietary protocol inspired by Tangaroa. ScalableBFT's design and implementation are not publicly available, making it challenging to assess its resilience.

Chain Core is a platform designed for institutional consortiums to issue and transfer financial assets on permissioned blockchain networks. The Federated Consensus protocol, which is executed by nodes in the network, focuses on the financial services industry and supports multiple assets. It is a specialized version of a standard BFT-consensus protocol that utilizes a fixed "leader" as the block generator. The protocol is resilient against malicious signers but not a malicious block generator.

Quorum, primarily developed by JPMorgan Chase, is an enterprise-oriented version of Ethereum that uses the Ethereum Virtual Machine to execute smart contracts. It provides two consensus protocols: QuorumChain and Raft-based consensus. QuorumChain relies on a smart contract to validate blocks, while Raft-based consensus leverages the Raft protocol for replication. QuorumChain's resilience is limited, and it cannot ensure consensus in any practical sense. Raft-based consensus is appropriate for safeguarded environments without adversarial nodes.

MultiChain is a platform designed for permissioned blockchains in the financial industry, with a focus on compatibility with the Bitcoin protocol. It employs a mining-based consensus protocol that uses a round-robin scheme to select miners. A miner is chosen to create a block in each round, and other nodes validate and accept the block. MultiChain enables multiple assets and smart filters for transaction validation. Its resilience depends on the assumption that the majority of miners are honest and that a certain number of consecutive blocks must be mined to confirm a transaction.

The paper also delves into non-BFT consensus mechanisms employed by permissionless blockchains. Intel's Sawtooth Lake uses Proof of Elapsed Time (PoET), similar to Proof of Work but with significantly less energy consumption. It utilizes Intel's Software Guard Extensions (SGX) to create a trusted execution environment. However, PoET's resilience relies on the trustworthiness of Intel's SGX implementation.

Ripple and Stellar use Federated Byzantine Agreement (FBA) in their consensus protocols, a generalization of traditional Byzantine Agreement. In FBA, nodes form quorums by selecting trust relationships, and a transaction is considered valid once it is confirmed by a sufficient number of quorum slices. Ripple's consensus process is low latency and high throughput, but its resilience depends on the trustworthiness of a small set of nodes. Stellar's SCP protocol is secure under the FBA model, but it also relies on the trustworthiness of a limited number of nodes.

IOTA implements the Tangle, a Directed Acyclic Graph (DAG), as its consensus mechanism. Each transaction in the Tangle must approve two previous transactions, and only after being referenced by an adequate number of subsequent transactions, a transaction is considered confirmed. Unlike other blockchain platforms, the Tangle does not rely on miners or validators and does not charge transaction fees. However, the security of the Tangle depends on the assumption that an attacker cannot control a significant portion of the network's hash power.

In conclusion, the paper provides a comprehensive analysis of consensus protocols in permissioned blockchains and highlights the importance of rigorous evaluation, formal verification, and adherence to established security methodologies. It extensively reviews various consensus protocols, highlighting their strengths and weaknesses, and offers valuable insights for developers, investors, and users in the blockchain industry.