Synchrony Assumptions for Blockchain Systems

Goals: Determine the trade-offs between consistency and performance in permissionless and permissioned blockchains.

Tools: Logic, algorithmic reasoning, programming

Prerequisites: basic knowledge of distributed algorithms (with a focus on state-machine replication, Byzantine Fault-Tolerance, storage systems), basic concurrent programming skills, curiosity and persistence

Summary

The prominent *blockchain* technology aims at implementing a public "ledger": a decentralized consistent history of transactions proposed by an *open* set of participating processes, with no static membership. This problem can be seen as an instance of fault-tolerant *state-machine replication* [14], prominent examples of which are the *crash-tolerant* Paxos protocol by Lamport [11] and the BFT (*Byzantine* fault-tolerant) system by Castro and Liskov [3]. These systems use instances of *consensus* protocols in order to ensure that users get consistent views of the system evolution.

Principal downside of classical consensus protocols are lack of scalability and the need for a fixed or properly reconfigurable set of participants out of which only a bounded fraction (up to one third) can be faulty. This can be hard to ensure in an open system, where an arbitrary fraction of participants can be controlled by the adversary [5]. Prominent blockchain protocols [13, 15] achieve (nondeterministic) consistency by assuming that (1) the system is synchronous, (2) participants can use asymmetric cryptography, and (3) the adversary can control at most a minority (in practice, a minor fraction) of computing power.

Intuitively, these assumptions are used to overcome the folklore CAP theorem [2, 8] stating that no system can combine Consistency, Availability, and Partition-Tolerance. In particular, these protocols avoid partitioning by enforcing the *proof of work* (PoW) mechanism requiring that a participant must solve a time-consuming cryptographic puzzle before updating the ledger. The resulting protocols are notoriously slow and energy-demanding. More recent blockchain prototypes propose to obviate the energy demands via using *proof-of-stake* [1, 10], *proof-of-space* [6], or *proof of space-time* [12]. However, the proposals still resort to synchronous networks and/or impose restrictions on the fraction of honest players to ensure proper security levels. An immediate question is whether these costs and assumptions are unavoidable.

In this project, we intend to characterize the model assumptions that enable strong ledger consistency in an open system. This will involve determining precise bounds on the amount of synchrony [4, 7] and energy/space/time consumption for implementing a generic distributed transaction ledger. This might lead to improving the conventional "proof" mechanisms, used, e.g., in Tezos [9] and Cardano [10] platforms.

Theoretical in its nature, the project is motivated by viable practical concerns. Besides provable complexity and computability bounds, it intends to develop system prototypes that are not only formally proved correct but also studied experimentally.

Contact

Prof. Petr Kuznetsov
`http://www.infres.enst.fr/~kuznetso/`
`petr.kuznetsov@telecom-paristech.fr`
INFRES, Télécom ParisTech
Office C213-2, 46 Rue Barrault

References

[1] I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies without proof of work. In *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, pages 142–157, 2016.

[2] E. A. Brewer. Towards robust distributed systems (abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '00, pages 7–, 2000.

[3] M. Castro and B. Liskov. Practical byzantine fault tolerance. In *OSDI: Symposium on Operating Systems Design and Implementation*. USENIX Association, Co-sponsored by IEEE TCOS and ACM SIGOPS, Feb. 1999.

[4] T. D. Chandra, V. Hadzilacos, and S. Toueg. The weakest failure detector for solving consensus. *J. ACM*, 43(4):685–722, July 1996.

[5] J. R. Douceur. The sybil attack. In *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, MA, USA, March 7-8, 2002, Revised Papers*, pages 251–260, 2002.

[6] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak. Proofs of space. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 585–605, 2015.

[7] F. C. Freiling, R. Guerraoui, and P. Kuznetsov. The failure detector abstraction. *ACM Comput. Surv.*, 43(2):9:1–9:40, 2011.

[8] S. Gilbert and N. Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, June 2002.

[9] L. Goodman. Tezos: A self-amending crypto-ledger: Position paper, August 2014. `https://tezos.com/static/papers/position_paper.pdf`.

[10] A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 357–388, 2017.

[11] L. Lamport. The Part-Time parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998.

[12] T. Moran and I. Orlov. Proofs of space-time and rational proofs of storage. *IACR Cryptology ePrint Archive*, 2016:35, 2016.

[13] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, May 2009. `https://bitcoin.org/bitcoin.pdf`.

[14] F. B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, 22(4):299–319, Dec. 1990.

[15] G. Wood. Ethereum: A secure decentralized generalized transaction ledger. White paper, 2015.