Blockchain in the lens of BFT

Dahlia Malkhi
VMware Research

ATC 2018
“Centralized services are a security hole”

Nick Szabo, 2001
cryptographer and legal scholar
inventor of “smart contracts”
Agenda

What ?
Why ?
How ?
Blockchain Technology: A 3-Layer View

- Distributed Ledgers Technology (DLT)
- Contracts / Transactions
- Distributed Apps
Unspent Transaction Outputs (UTXOs)

new minted coins C, D
transfers:
A ← 0.0 of coin A
B ← 1.0 of coin A
...

new minted coins E, F
transfers:
A ← 1.0 of coin A
X ← .5 of coin C
C ← .5 of coin C
...
/* multi-sig */
set of signatories;
transfer(to, value):
  insert address to signatories;
  if (|signatories| > 2/3 owners) then
    to.call.value(value);

/* I want to transfer X to Y */
invoke transfer(Y, X);
Use-cases?

“ I need blockchain because my CxO says I do”
Use-cases?

``Centralized Services are a Security Hole”, Nick Szabo, 2001

- crypto-currencies
  - “utility tokens”
  - multi-sig

- reliable, ordered broadcast channel
  - identity management
  - audit
  - provenance tracking

- a hash-chain
  - timestamp
  - immutable

- platform for privacy preserving information sharing and processing
/* I want to sell my car */

sell(payment):
    if (payment > $1K from X) then
        assign my title to X

/* I want to buy the car */

invoke sell(my signed payment)
Blockchain core: BFT consensus (aka Distributed Ledger)
“Bitcoin is the first practical solution to a longstanding problem in computer science called the Byzantine Generals Problem”

Marc Andreesen, NYTimes, 2014
inventor of Mosaic, VC, thought leader
Nakamoto Consensus [Santoshi Nakamoto 2008]  
a triumph of math, algorithms and crypto

- **Hash-chain**  
  [Haber and Stornetta 1991, ”How to Timestamp a Digital Document” ]

- **Proof-of-Work (PoW)**  
  [Dwork and Naor 1992, “Pricing via Processing or Combatting Junk Mail”]
Nakamoto Consensus: Longest-Fork-Wins (LFW)
Nakamoto Consensus: Longest-Fork-Wins (LFW)
are we decentralized yet?

- Energy cost / waste
- High latency to “finality”
- Limited throughput
- Forking attacks
- Concentration of power
  - http://arewedecentralizedyet.com:
    - # entities controlling > 50%
      
      BITCOIN: 3
      Ethereum: 3
      Ripple: 1
      Stellar: 1
      ...
Revisiting BFT

- **Hybrid Blockchain**
  - Combination
  - Example: Ethereum Casper

- **Consortium Blockchain**
  - Known group of participants
  - Example: Banking

- **Public Blockchain**
  - Permissionless: Anyone can join
  - Example: Bitcoin
HOORAY

I'M RELEVANT
BFT in the lens of Blockchains and
Blockchains in the lens of BFT
BFT Consensus

- $n=3f+1$
- authenticated communication channels
- agreement, eventual termination, validity
- partial synchrony
  - eventually known bound $\Delta$
  - safety maintained against asynchrony
  - liveness during synchronous periods
DLS [Dwork Lynch Stockmeyer, 1988]

Landmark in asynchronous BFT agreement solutions
No Liveness

“The Saddest Moment” [Mickens 2013]
BFT in the Lens of Blockchains

- Tendermint [Buchman, 2016, “BFT in the Age of Blockchains”]
- Casper [Buterin and Griffith 2017, “Casper the Friendly Finality Gadget”]
- Hot-Stuff [AGM 2018, “Hot-Stuff the Linear One-Message BFT Devil”]
BFT in the lens of Blockchains

voting protocol possible decision

height 53

height 54
DLS in the lens of Blockchains

1-step decision by 2f+1 votes (QC)

at most 
f good + f bad 
votes
Blockchains in the lens of BFT

implicit voting by two-thirds thru PoW

height 53
height 54
DLS in the lens of Blockchains
BFT in the lens of Blockchains

- Voting protocol: possible decision
- Liveness gadget

Height 53

Height 54
Liveness Gadgets

- when can you guarantee progress?
  - proposer extends a safe branch
  - no correct replica locked on a different branch
  - synchronous communication with proposer
DLS in the lens of Blockchains

height 53

height 54

f+1 votes

evidence that f+1 vote against other branch

f+1 votes
practical BFT protocols

DLS 1988
O(n) rounds
wait $\Delta$ for latest
$O(n^4)$ comm

PBFT 1999
leader collects “proof” from $2f+1$
O(1) rounds
$O(n^4)$ t-missions

Zyzzyva 2007
fast track
$O(n)$ t-missions on common day

Tendermint 2016, Casper 2018
leader waits $\Delta$ for latest
O(1) rounds
$O(n^3)$ t-missions

Byzcoin 2016, SBFT 2018
signature combining / $O(n)$ t-missions
Current conundrum

one block per period

synchronous:
always slow

asynchronous:
sometimes very slow/expensive

8,000,000,000 msgs
Are we decentralized yet?

- **1976** Byzantine Generals
- **1988** DLS quartic
- **1999** PBFT cubic
- **2007** Zyzzyva optimistic fast track (flawed)
- **2018** SBFT (optimistic fast track and signature-combining)
- **2018** Hot-Stuff (quadratic)
Abstract

In this note, we observe a safety violation in Zyzzyva [7, 9, 8] and a liveness violation in FaB [14, 15]. To demonstrate these issues, we require relatively simple scenarios, involving only four replicas, and one or two view changes. In all of them, the problem is manifested already in the first log slot.
message pattern, but votes for a nil block.

Tendermint as originally described by Buchman [13] suffers from a livelock bug, pertaining to locking and unlocking votes by validators in the protocol. However, the protocol contains additional mechanisms not described in the cited report that prevent the livelock from occurring [14]. While it appears to be sound, the Tendermint protocol and its implementation are still subject to a thorough, peer-reviewed correctness analysis.
A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform

João Sousa
LaSIGE, Faculdade de Cincias, Universidade de Lisboa

Alysson Bessani

Marko Vukolić
IBM Research Zurich

Abstract

Hyperledger Fabric (HLF) is a flexible permissioned blockchain platform designed for business applications beyond the basic digital coin addressed by Bitcoin and other existing networks. A key property of HLF is its extensibility, and in particular the support for multiple ordering services for building the blockchain. Nonetheless, the version 1.0 was launched in early 2017 without an implementation of a Byzantine fault-tolerant (BFT) ordering service. To overcome this limitation, we designed, chaincode) and pluggable services [26]. The support for pluggable components, gives the HLF an unprecedented level of extensibility, and in particular the support for multiple ordering services for writing transactions on the blockchain. Despite of that, the version 1.0 (launched in early 2017) comes without any Byzantine fault-tolerant (BFT) ordering service, supporting only crash tolerance
Quorum

Quorum is an Ethereum-based distributed ledger protocol with transaction/contract privacy and new consensus mechanisms.

Quorum is a fork of go-etherum and is updated in line with go-etherum releases.

Key enhancements over go-etherum:

- **Privacy** - Quorum supports private transactions and private contracts through public/private state separation and utilising Constellation, a peer-to-peer encrypted message exchange for directed transfer of private data to network participants
- **Alternative Consensus Mechanisms** - with no need for POW/POS in a permissioned network, Quorum instead offers multiple consensus mechanisms that are more appropriate for consortium chains:
  - **Raft-based Consensus** - a consensus model for faster blocktimes, transaction finality, and on-demand block creation
  - **Istanbul BFT** - a PBFT-inspired consensus algorithm with transaction finality, by AMIS.
- **Peer Permissioning** - node/peer permissioning using smart contracts, ensuring only known parties can join the network
- **Higher Performance** - Quorum offers significantly higher performance than public geth

Note: The QuorumChain consensus algorithm is not yet supported by this release.

Architecture
SBFT
200 WAN nodes, 2 months of Ethereum contracts

Tput: 170/sec
30x ETH

Latency: 600 ms
24x ETH
1000x XBT
VMware Blockchain

VMware platforms will be

*The best place to run blockchain software*
Thank You