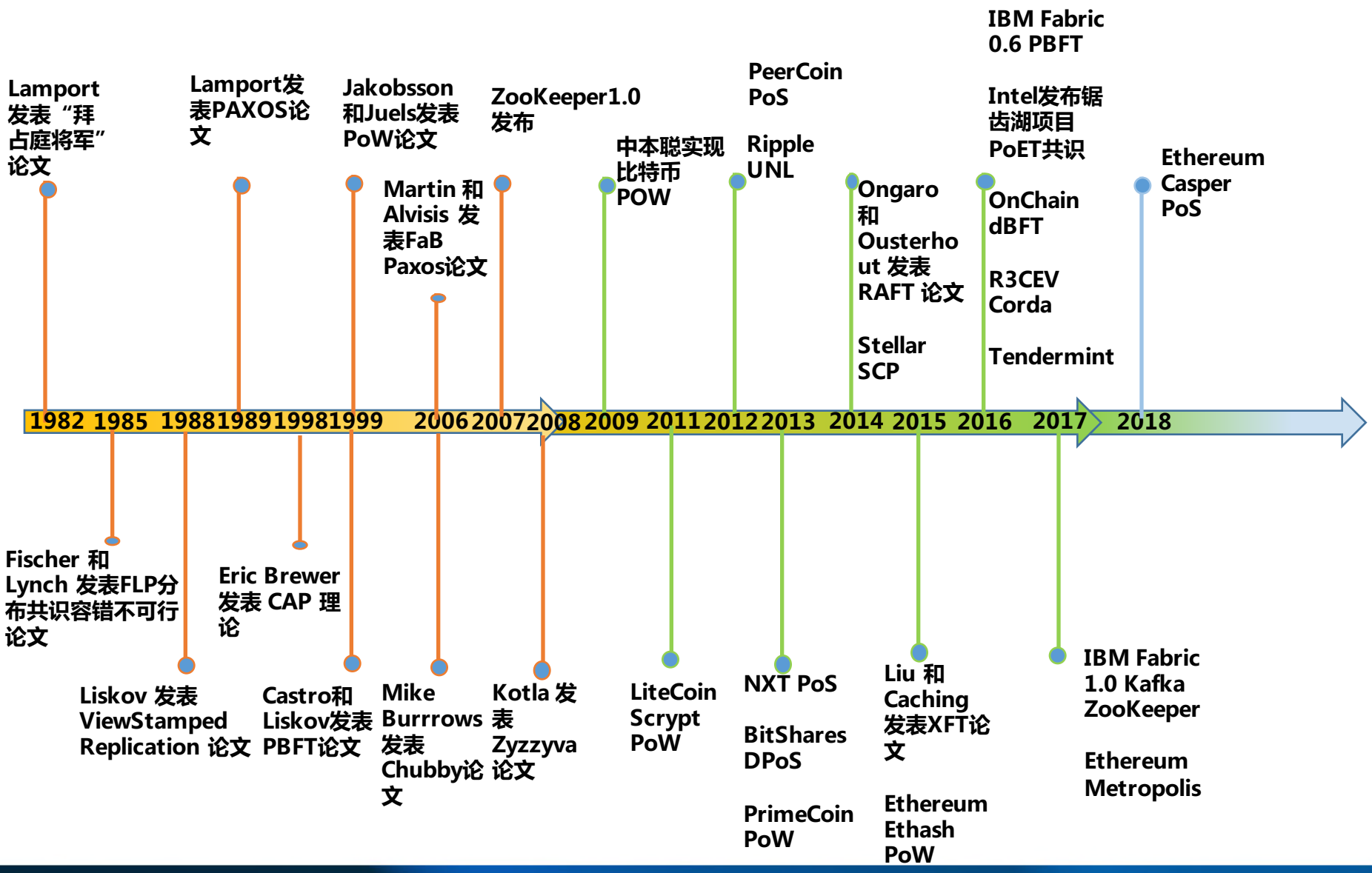
共识简史——回顾·现状·未来

邹均

目录

区块链共识简史


Lamport 发表“拜占庭将军”论文
Lamport发表PAXOS论文
Jakobsson 和Juels发表PoW论文
ZooKeeper1.0 发布
中本聪实现比特币 POW
PeerCoin PoS
Ripple UNL
IBM Fabric 0.6 PBFT
Intel发布锯齿湖项目 PoET共识
Ongaro 和 Ousterhout 发表 RAFT 论文
Stellar SCP
OnChain dBFT
R3CEV Corda
Tendermint
Ethereum Casper PoS
Martin 和 Alvisis 发表FaB Paxos论文
1982 1985 1988 1989 1998 1999 2006 2007 2008 2009 2011 2012 2013 2014 2015 2016 2017 2018
Fischer 和 Lynch 发表FLP分布共识容错不可行论文
Liskov 发表 ViewStamped Replication 论文
Eric Brewer 发表 CAP 理论
Castro和 Liskov发表 PBFT论文
Mike Burrrows 发表 Chubby论文
Kotla 发表 Zyzzyva 论文
LiteCoin Scrypt PoW
NXT PoS
BitShares DPoS
PrimeCoin PoW
Liu 和 Caching 发表XFT论文
Ethereum Ethash PoW
IBM Fabric 1.0 Kafka ZooKeeper
Ethereum Metropolis

分布式系统如何保证一致性？

- 分布式系统
  - 分布式是一种计算模式，指在一个网络中，各节点通过相互传送消息来通信和协调行动，以求达到一个共同目标
- 一致性的问题
  - 分布式系统由于各节点独立运行，相互通信也会出故障，因此要保证整个系统状态的一致性，需要有共识协议

- **拜占庭将军问题**
  - 所有忠诚的将军决定一致的计划；
  - 少数叛徒不能使忠诚将军采用错误的计划。
- **简化版拜占庭问题**
  - 一个指挥官要发一个命令给n-1个副官，希望：
    - 所有忠诚的副官都执行同一命令；
    - 如果指挥官是忠诚的，每个忠诚的副官都执行该命令。

- **共识问题（Consensus Problem）**
  - **特征**
    - 所有进程都有一个初始值（all processes have an initial value ）
  - **属性**
    - 一致协议（Agreement）
      - 所有的非缺陷进程都必须同意同一个值。
    - 正确性（Validity）
      - 如果所有的非缺陷的进程有相同的初始值，那么所有非缺陷的进程所同意的值必须是同个初始值。
    - 可结束性（Termination）
      - 每个非缺陷的进程必须最终确定一个值。

- **分布式系统节点类型**
  - 正常系统 – 忠诚的拜占庭将军
  - 任意故障系统 – 叛变的Byzantine将军
- **拜占庭缺陷（Byzantine Fault）**
  - 任何从不同观察者角度看表现出不同症状的缺陷
- **拜占庭故障（Byzantine Failure）**
  - 由于拜占庭缺陷导致的节点故障
- **故障的分类**

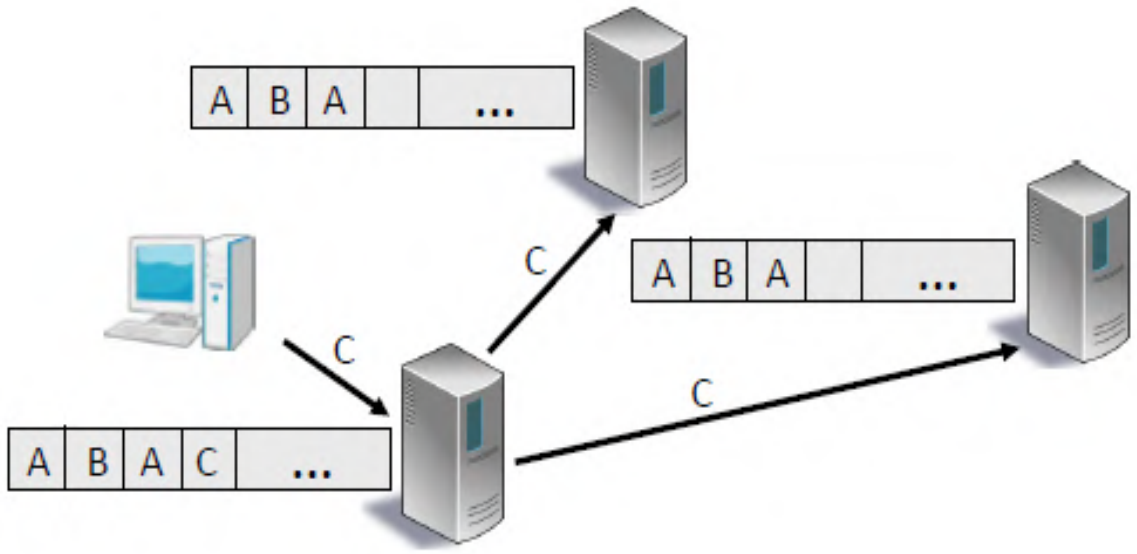
关于拜占庭将军问题的理论证明

- **Lamport （1982）**
  - 当叛徒多于将军总数的三分之一时，同时通信采用同步非防篡改方式，拜占庭将军问题无解。
  - 如果通信采用同步、认证和不可篡改方式，拜占庭将军问题可以在任意多叛徒情况下有解（如果将军总数少于叛徒数+2，问题就没有意义）
- **Fischer-Lynch-Paterson （1985）**
  - 在一个多进程异步系统中，只要有一个进程不可靠，那么就不存在一个协议，此协议能保证有限时间内使所有进程达成一致。

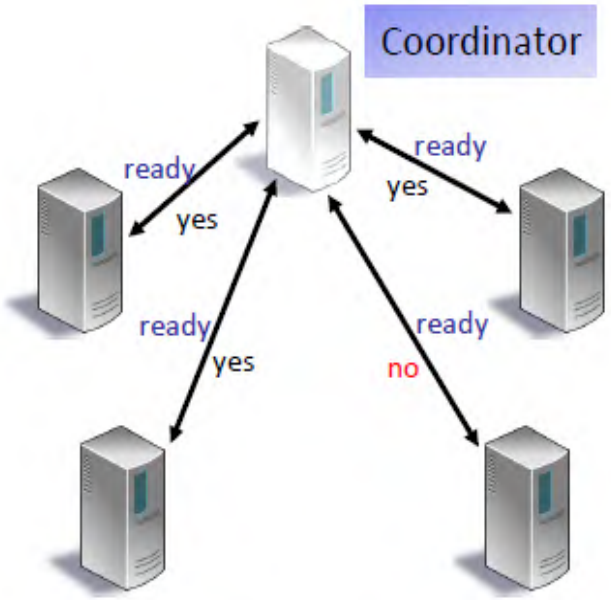
- **分布式系统假设**
  - 故障模型
    - 非拜占庭故障
    - 拜占庭故障
  - 通信类型
    - 同步
    - 异步
  - 通信网络连接
    - 节点间直连数
  - 信息发送者身份
    - 实名
    - 匿名
  - 通信通道稳定性
  - 消息认证性
    - 认证消息
    - 非认证消息

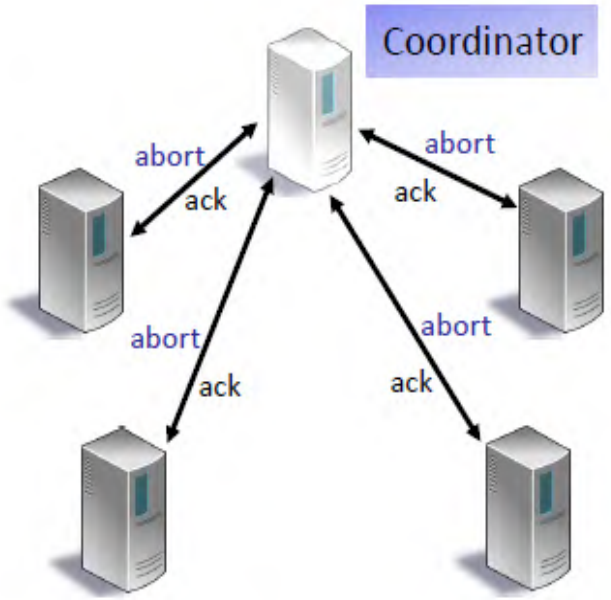
- **强一致性共识算法**
  - 宕机故障（Crash Failure）
    - 主副本, 2PC, 3PC提交
  - 宕机-恢复故障 （Crash Recovery Failures）
    - Paxos, Chubby，ZooKeeper，etcd
    - RAFT，ViewStamped Replication
  - 拜占庭故障 （Byzantine Failures）
    - PBFT（实用拜占庭容错）， Zyzzyva，Fab Paxos

- 状态机复制（State Machine Replication)
  - 采用把系统状态复制到各冗余副本节点以及协调客户端对服务器节点访问的通用容错方法



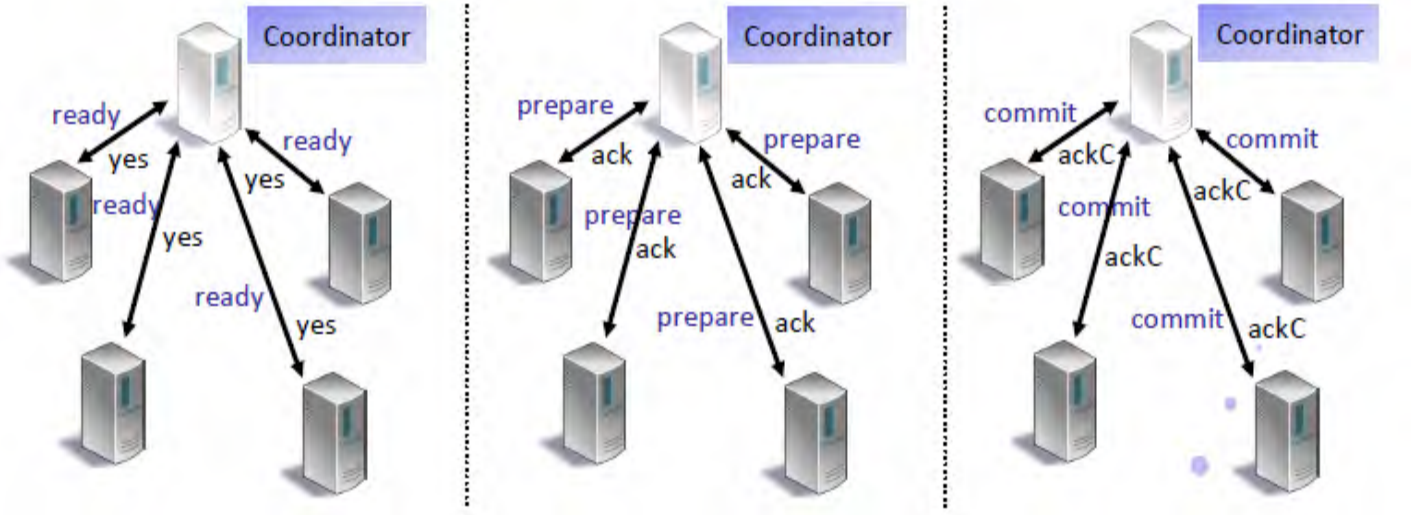
- 2PC 两阶段提交





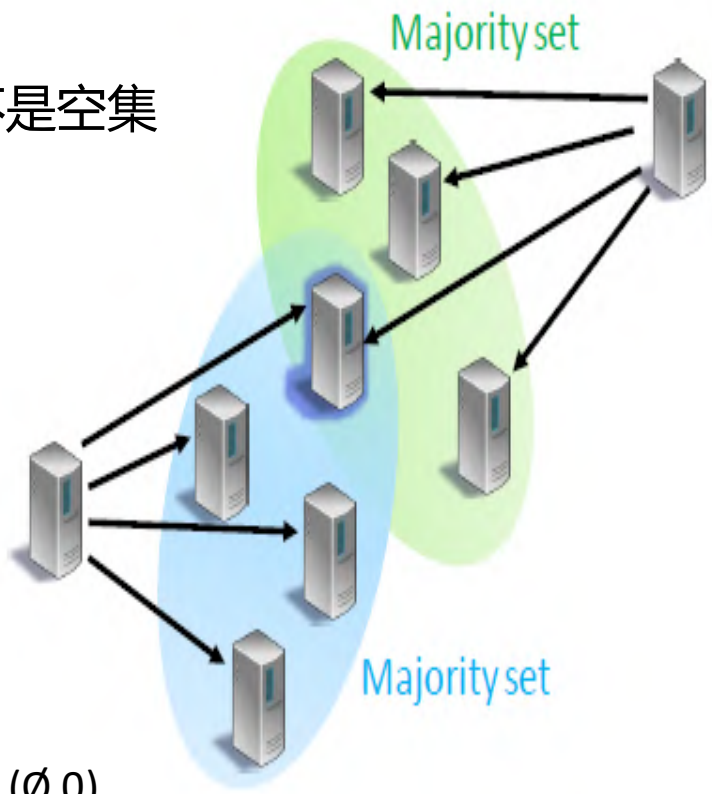
- 2PC 缺点
  - 当Coordinator和其它一个节点发生故障，其它节点无法知道是应该提交还是回滚

- 3PC提交
  - 把第二阶段再划分，从协议上使得第一阶段状态完全确定;
  - 其它节点第二阶段只能确认收到消息，不能提出回滚要求;
  - 第三阶最后交易提交。



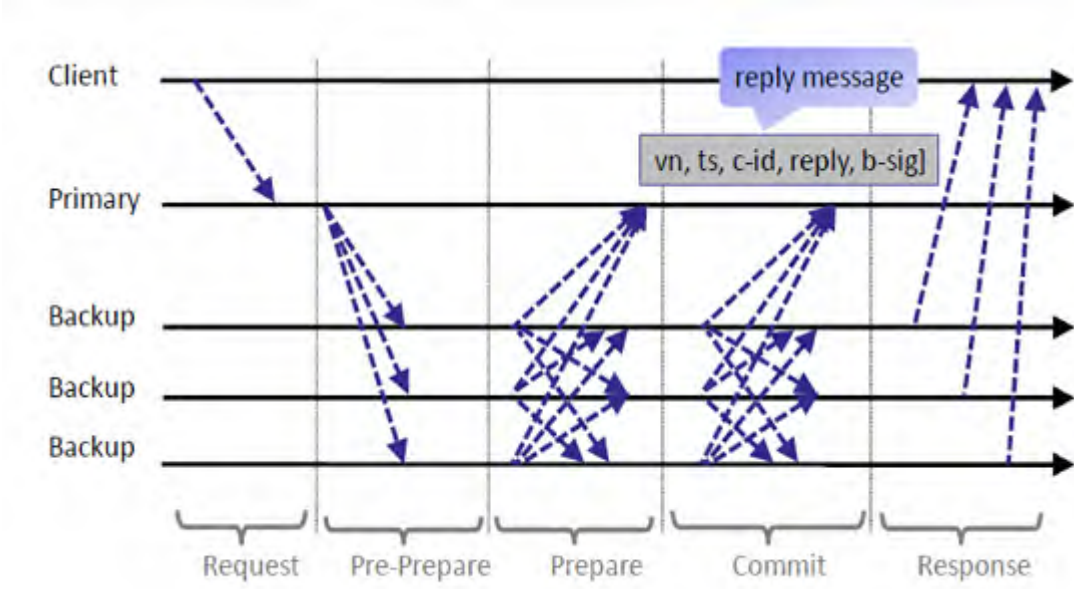
- 3PC的局限
  - 过于依赖协调节点，如果部分节点认为协调节点故障，将比较复杂
  - 如果一个Coordinator故障节点恢复上线, 如何处理？

- Paxos
  - 多数集
    - 任意两个多数集之间的交集不是空集
  - 三种角色
    - Proposer
      - 提议一个被希望接受的值
    - Accepter
      - 接受提议或拒绝提议
    - Learner
      - 不参与决策
      - 但必须知道最后结果
  - 两阶段
    - Prepare
      - *Prepare*(*x*, *n*) / *acc*(*y*, *m*) | *acc* (Ø,0)
    - Propose
      - *Propose*(*y*, *n*) / *ack*(*y*, *n*)



- Paxos 优点
  - Paxos是一个可以容忍 f<n/2 Crash-Recovery 故障的异步强一致性协议
    - Paxos能保证协议性（Agreement）和正确性（Validity）
- 局限
  - 不能保证可终止性（Termination）
    - Paxos 只能保证如果一个值被选择，其它节点只能选择同意一个值
    - 但它不能保证一个值能被选择到。

- PBFT 需要5轮通信步骤
  - 第一轮
    - 客户端发命令 op 给主节点
  - 第二轮
    - Pre-prepare
  - 第三轮
    - Prepare
  - 第四轮
    - Commit
  - 第五轮
    - 客户端接收服务端的回复
      - 如果 *f*+1 回复一样，这个结果被接受。
      - 因为只有 f 个拜占庭服务器，至少有一个正确的服务器支持这个结果。



- PBFT 特性
  - PBFT在一个异步系统中能达成共识，容忍f<n/3拜占庭故障
  - PBFT的异步通信是一个有固定延迟限制的异步通信系统（接近同步通信）
  - 采用认证的消息
    - 可以验证服务器是否发过一个消息
  - 在最坏情况下，PBFT需要用f轮才能达到共识。

目录

1. 回顾
2. 现状
3. 未来
4. 小结

比特币PoW – 最终一致性共识

- **挖矿过程**
  - 生成铸币交易，并与其他所有准备打包进区块的交易组成交易列表，通过Merkle树算法生成Merkle 根哈希；
  - 把Merkle根哈希及其他相关字段组装成区块头，将区块头的80字节数据作为工作量证明的输入；
  - 不停的变更区块头中的随机数即nonce的数值，并对每次变更后的区块头做双重SHA256运算（即SHA256(SHA256(Block_Header))）
  - 将结果值与当前网络的难度目标值做对比，如果小于目标值，工作量证明完成
- **难度调整**
  - 由于矿工数量动态变化，比特币系统动态调整挖矿难度，使得大约每10分钟产生一个区块
- **共识区块链**
  - 最长区块链为共识区块链，代表具有最大工作量的区块链
- **概率性拜占庭（Probabilistic BA）**
  - **协议（Agreement）**
    - 在不诚实节点总算力小于50%的情况下，同时每轮同步区块生成的几率很少的情况下，诚实的节点具有相同的区块的概率很高。
  - **正确性（Validity）**
    - 大多数的区块必须由诚实节点提供
    - （严格的说，当不诚实算力非常小的时候，才能使大多数区块由诚实节点提供）
  - **可终止性（Termination）**
    - 约每10分钟完成一次共识

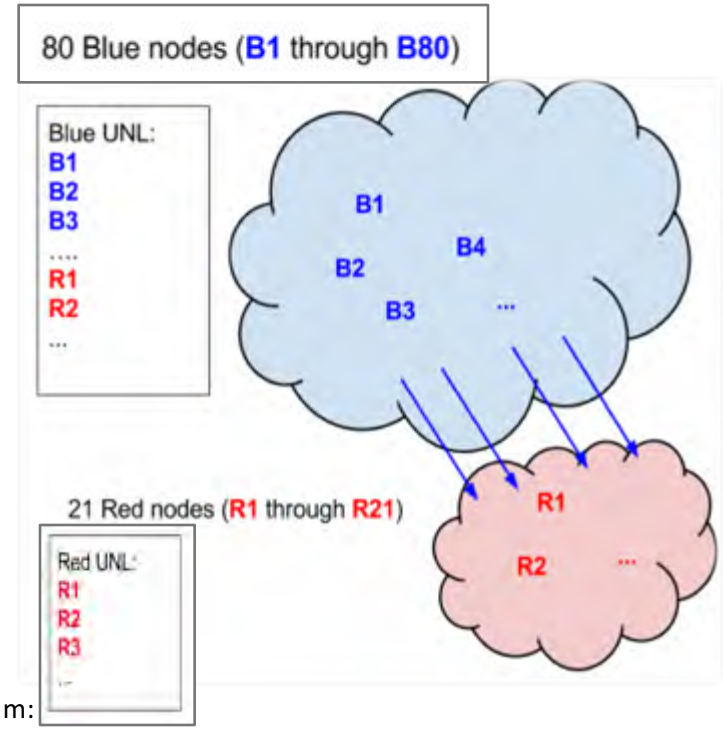
其它PoW共识算法

- LiteCoin PoW
  - 采用scrypt算法，不但需要计算能力，还需要内存，可以防止暴力破解，对ASIC挖矿也有一定的抵抗力
- PrimeCoin PoW
  - 工作量证明不像比特币那样做无用功，而是寻找质数，具有科学价值，可称为“有用工作量证明”（Proof of Useful Work”）
- Ethereum Ethash PoW
  - 比特币PoW基础上引入GHOST（Greedy Heavest-Observed Sub-Tree), 在共识区块链计入叔区块
  - 挖矿不但需要计算能力，还需要内存，对ASIC挖矿和算力中心化有一定抵抗力
- Intel SawtoothLake PoET
  - 采用新的CPU安全指令，通过可信任运行环境（TEE）如 Intel® Software Guard Extensions (SGX)，根据验证者等待时间来确定Leader，实现公平、随机选取共识Leader
    - 每个验证者向一个enclave (信任函数）请求一个等待时间，具有对一个区块最短等待时间的验证者会被选成Leader.
  - 实现“一个CPU一票”

- **权益证明Proof-of-Stake (PoS)共识算法**
  - **PeerCoin POS**
    - 权益证明机制结合了随机化与币龄的概念，未使用至少三十天的币可以参与竞争下一区块，越久和越大的币集有更大的可能去签名下一区块。
  - **NXT、Blackcoin POS**
    - 采用随机方法预测下一合法区块，使用公式查找与权益大小结合的最小哈希值， 由于权益公开，每个节点都可以以合理地准确度预计哪个帐户有权建立区块。
  - **BitShares DPOS**
    - 引入了见证人这个概念，见证人可以生成区块，每一个持有比特股的人都可以投票选举见证人。
  - **Tendermint**
    - 验证者将押金锁定，投票权力和押金相等。验证者如果作弊，押金会被销毁。
    - 投票步骤：Propose, Prevote, Precommit， Commit, NewHeight
    - Precommit和Commit需超过2/3投票
  - **Ethereum PoS (Casper)**
    - 权益持有者通过用押金方式来赌下一个区块，赌中有奖，不中会扣掉一部分押金

Ripple用子网共识提升共识效率，但能保证一致性吗？

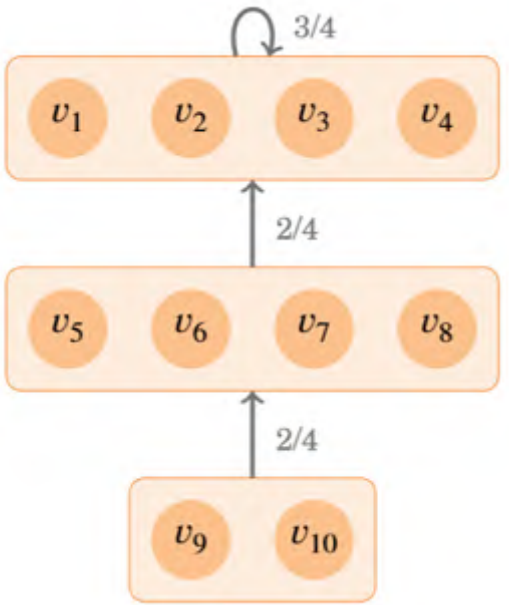
- Ripple共识
  - 假设拜占庭节点数少于所有节点数的20%（f <= (n-1)/5）
  - 共识基于有信任关系的独一节点群UNL（Unique Node List)
  - UNL 的任意一个节点串通作恶概率小于20%
  - 任意两个UNL间重合的节点数至少占最大UNL节点数20%
  - 验证的交易需获得80%以上UNL成员投票
- 极端情况下Ripple能保证不分叉吗？
  - UNL1: 80 蓝 + 21红
    - 80个节点，都投Yes票
    - 3个红节点投Yes票
    - 共识结果是Yes > 80
  - UNL2：21红
    - 21个节点，其中18个投No
    - 共识结果是No >80%
  - UNL 1和 UNL2 共识结果分叉

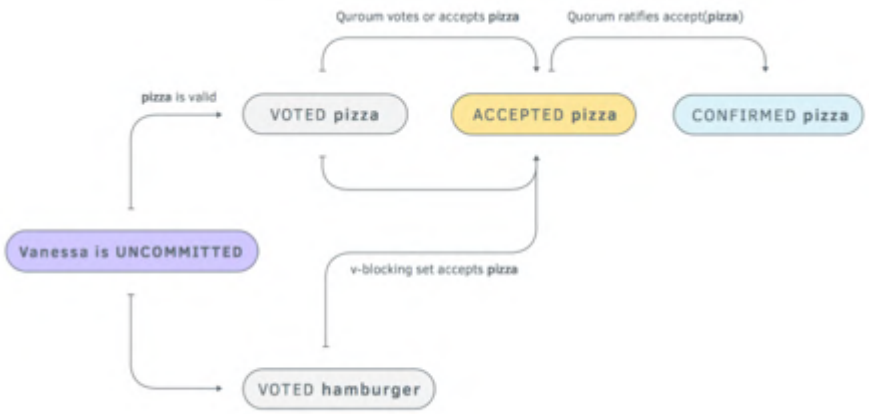


*source: Ripple Forum: https://forum.ripple.com/viewtopic.php?f=2&t=7801

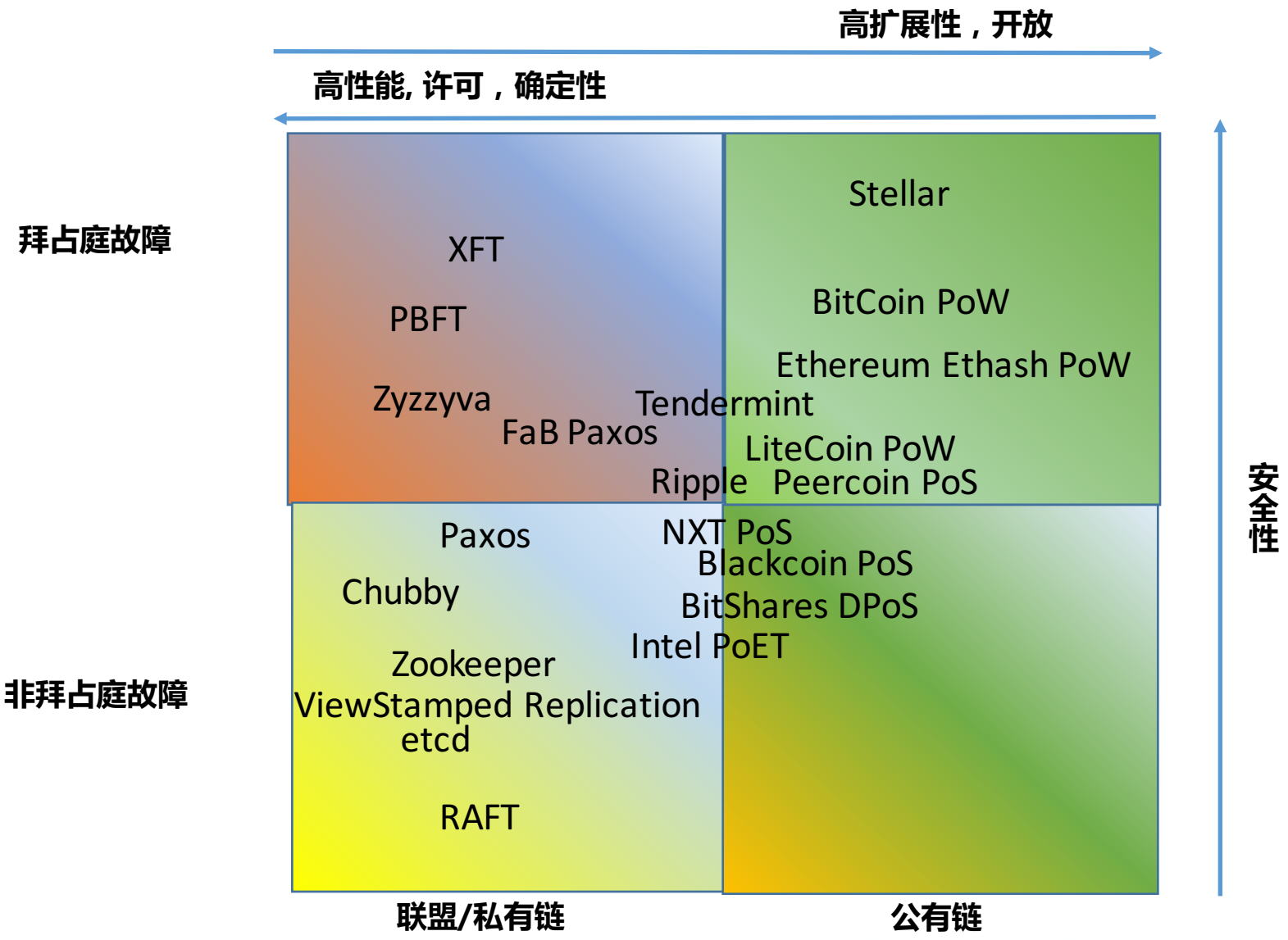
Stellar联邦式BFT共识算法

- SCP共识
  - 特点 – 分散控制、低延迟、灵活信任、渐近安全
    - 开放成员，每个节点可以决定信任对象 （ quorum slices)
  - 共识条件
    - Quorum
      - 达成一致的多数集
    - Quorum Slice
      - Quorum中的子集，能说服一个节点达成一致
    - Quorum Intersection 条件
      - 任意一个节点的Quorum Slices函数必须满足任意两个Quorum之间必须有一个共同的节点的条件
    - 删除破坏节点后还能达到Quorum Intersection
  - SCP协议 – 提名协议（ Nomination Protocol ）
    - 生成候选值
  - SCP协议 – 投票协议 （ Ballot Protocol)
    - Prepare （ 准备 ）
    - Confirm （ 确认 ）
    - Externalize （ 外部化 ）





*source：Stellar White Paper and Technical Summary


高扩展性，开放
高性能，许可，确定性
拜占庭故障
非拜占庭故障
安全性
联盟/私有链
公有链
XFT
PBFT
Zyzzyva
FaB Paxos
Tendermint
Ripple
Stellar
BitCoin PoW
Ethereum Ethash PoW
LiteCoin PoW
Peercoin PoS
Paxos
Chubby
Zookeeper
ViewStamped Replication
etcd
RAFT
NXT PoS
Blackcoin PoS
BitShares DPoS
Intel PoET

- **私有链多数采用状态机复制技术（SMR）**
  - Paxos-based
    - Chubby, Zookeeper, etcd, RAFT
  - BFT-based
    - PBFT
    - Zyzzyva
- **SMR技术扩展性不好**
  - 成员固定
  - 一般记账节点10-20个
  - 多轮消息往返
    - PBFT：五阶段， 节点平方级的消息量
  - 节点增加，性能降低

公有链上共识算法存在问题

- **比特币– POW机制**
  - 能源消耗大，工作量没有实际价值
  - 交易速度慢
    - 1秒钟最多7笔交易
  - 缺少最终确定性（Finality）
  - 算力集中
    - 矿池
    - ASIC矿机
- **POS 机制**
  - 破坏者攻击成本小，安全性成疑
  - 公平性有问题
- **有用工作量机制 – Proof-of-useful Work**
  - Primecoin
  - 如何满足容易验证，难度可调，几率和贡献工作量成正比（Progress-free）

目录

发展趋势 – 融合

- 未来区块链平台支持可插拔共识模块
  - Hyperledger Fabric, Ethereum， Hyperledger Sawtooth Lake
  - 根据不同场景，选择合适的共识算法
    - Fabric1.0， 提升交易吞吐率，采用Kafka
- 未来更多的融合，取长补短
  - 多链，侧链，闪电网络
  - 公有链/联盟链融合
  - 去信任 + 信任
    - Ripple, Stellar
- 行业区块链
  - R3CEV Corda -专注于金融行业，（不是区块链的区块链DLT）
    - 数据只在相关方共享，不交给无关第三方
    - 交易正确性共识
      - 通过合约运行和签名校验来验证
    - 交易唯一性共识
      - 需要事先设置的独立公证

目录

分类方式

- 按共识机制分类
  - PoW, PoS, PoET, BFT, Paxos
- 按区块链部署模式
  - 公有链共识算法
  - 联盟链/私有链共识算法
- 按容错类型分类
  - 宕机容错共识算法
  - 拜占庭容错共识算法
- 按一致性类型分类
  - 强一致共识算法
  - 最终一致性共识算法
- 按副本复制方式分类
  - Primary-backup 主从备份
  - State Machine Replication 状态机复制

区块链共识算法的选择考虑

- 公有链
  - 最终一致性（Eventually Consistency）
  - 故障类型: 拜占庭故障
  - 共识成本
    - POW vs POS
  - 共识效率
  - 安全性
- 联盟链 / 私有链
  - 强一致性（Strong Consistency）
  - 故障类型
    - 非拜占庭故障
    - 拜占庭故障
  - 共识效率

- **共识算法属性**
  - 一致性（Agreement）
  - 正确性（Validity）
  - 活性（Liveness）
  - 性能（Performance）
  - 扩展性（Scalability）
  - 公平性（Fairness）
  - 安全性（Security）
- **共识算法的理论限制**
  - Fischer-Lynch-Paterson定律
    - 在一个多进程异步系统中，只要有一个进程不可靠，那么就不存在一个协议，此协议能保证有限时间内使所有进程达成一致。
- **实用共识算法**
  - 实际情况下假设不同的条件
    - 同步/异步
    - 身份认证/匿名
    - 宕机-恢复故障 / 拜占庭故障
    - 故障节点占比

小结

- **区块链是信任机器**
  - 信任由共识产生
- **共识算法有强一致性和最终一致性共识算法**
  - 强一致性
    - Paxos，PBFT, RAFT 等
  - 最终一致性
    - PoW， PoS，DPoS
- **区块链共识算法因公有链、联盟链、私有链不同而有所不同**
  - 公有链：最终一致性共识算法
  - 联盟链，私有链
    - 强一致性共识算法

附：区块链技术指南

《区块链技术指南》新书介绍本书和其他描述应用场景的书不同，重点介绍各类区块链技术的特征和开发方法，包括比特币、以太坊、超级账本、共识算法、闪电网络、比特币开发技术、以太坊智能合约开发等等。对区块链技术感兴趣的朋友，请参考邹均博士、张海宁先生、唐屹博士、李磊博士和陈晖先生等作者合著的新书：《区块链技术指南》，机械工业出版社

扫描二维码购买

京东购买链接：
http://item.jd.com/12007317.html

谢谢！