北京大学
PEKING UNIVERSITY

《区块链》课程

孙惠平
sunhp@ss.pku.edu.cn

北京大学 软件与微电子学院
School of Software and Microelectronics, Peking University

北京大学
PEKING UNIVERSITY

PART 第七章

区块链共识

目 录

CONTENTS

第一节

共识算法介绍

01 共识算法概述

02 拜占庭将军问题

03 共识算法假设

04 区块链共识

1.1共识算法概述

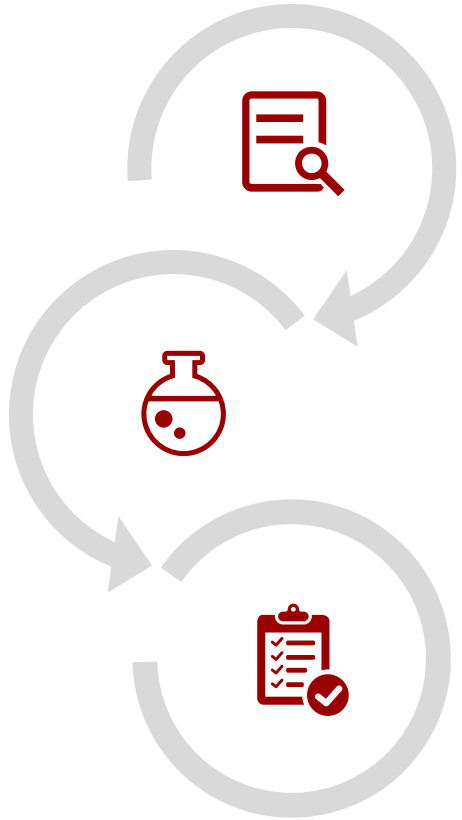
区块链的共识算法指的是区块链节点在一定时间内对区块链交易进行验证和确认的一种算法，是区块链的核心算法，影响区块链的可扩展性和安全性。

共识

即对于某些事情达成一致。
一般关注状态、决策、顺序等。

分布式系统语境下的共识

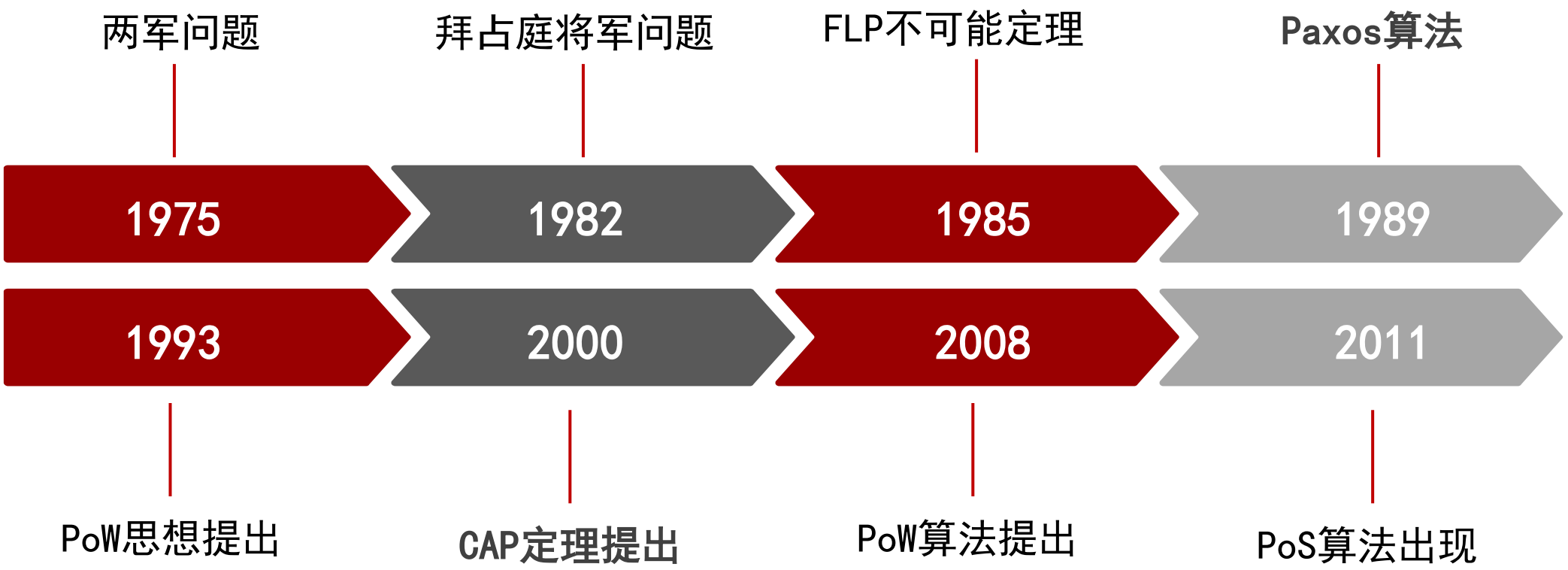
分布节点，即使面对不可信的其他节点和网络也可以确保就同一个数据或者状态达成一致的看法。



区块链

区块链是一种分布式系统，依赖共识算法来确保所有节点对交易历史的认可。

共识算法历史


两军问题
拜占庭将军问题
FLP不可能定理
Paxos算法
1975
1982
1985
1989
1993
2000
2008
2011
PoW思想提出
CAP定理提出
PoW算法提出
PoS算法出现

1.2拜占庭将军问题

莱斯利·兰伯特（Leslie Lamport）于1982年提出

拜占庭帝国有数十只军队，在一次军事行动中，每只军队驻扎距离较远，将军之间只能通过信使传递消息，不能面对面通信。

规定在军事行动中，拜占庭帝国的所有将军间必须达成统一的意见才能行动，确定是否攻占敌人堡垒。

但是在各个军队中的将军可能是敌军的间谍或者叛徒。它们可以影响、打乱军队的决策意见。从而扰乱将军之间对军事行动达成共识。

拜占庭将军问题就是指，在已知有恶意将军的情况下，其它忠诚的将军如何达成一致意见，完成军事行动。

1.2拜占庭将军问题

北京大学 PEKING UNIVERSITY

互联网系统中的拜占庭将军问题

节点对等，没有中央服务器。节点与节点之间想要达成一致的意见，只能通过相互之间的通信来完成。

由于恶意节点攻击、网络延时、节点宕机等，系统中可能存在异常节点，从而影响系统达成一致意见。

拜占庭容错算法： 3f + 1的经典结论

1.3共识算法的假设

容错假设

故障错误(Crash Fault, CFT)
拜占庭错误(Byzantine Fault, BFT)

身份假设

完全匿名身份
部分匿名身份
不匿名身份

同步假设

同步网络(Synchronous)
部分同步网络(Partial Synchronous)
异步网络(Asynchronous)

北京大学 PEKING UNIVERSITY

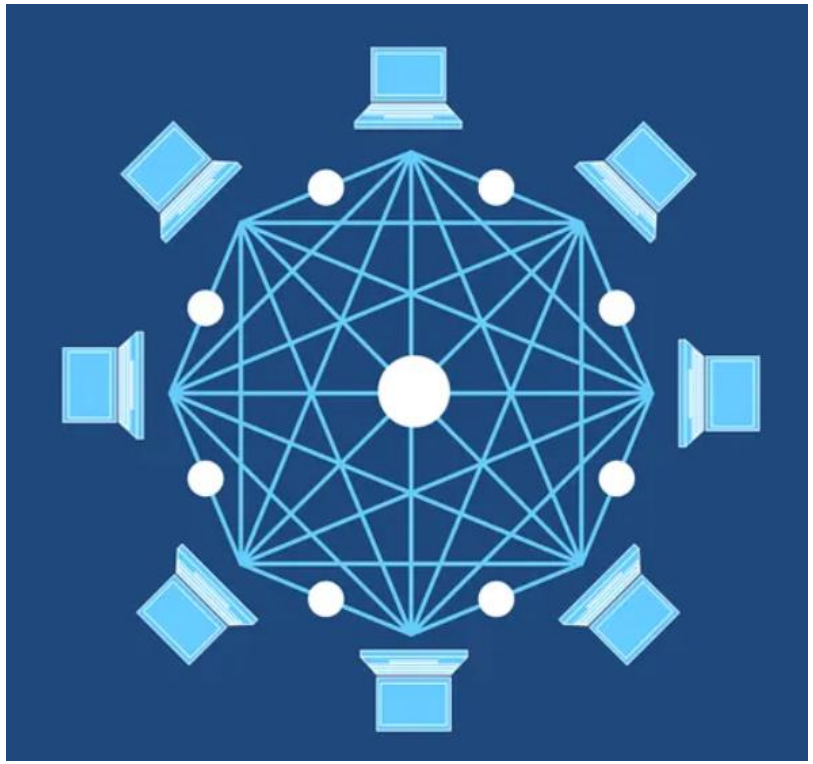
区块链共识的过程

01 主节点选举

02 区块生成

03 区块验证投票

04 区块上链



共识算法决定了共识流程的复杂度和节点交换消息的数量，也决定了达成分布式一致性的代价和时间。

第 二 节

PoW类共识算法

01 PoW基本概念

02 PoW面临问题

03 PoW最新发展

2.1PoW基本概念

- **Proof-of-Work简称PoW，意为工作量证明，也称为Nakamoto共识，是比特币、以太坊等数字货币采用的共识算法，也是当前使用最广泛的共识算法。**

Proof-of-Work

工作量证明最早被用来防止垃圾邮件，由Dwork和Naor提出。在邮件发送之前，发送方必须先完成某种数学难题的解密，该过程需要一定量的数学计算，只有该难题答案被成功验证后接收方才会接受到邮件。

1997年，Back提出的Hashcash，该论文对工作量证明进行创新性设计和改进，首次实现借助哈希函数完成工作量证明。

2.1比特币中的PoW过程

北京大学 PEKING UNIVERSITY

01

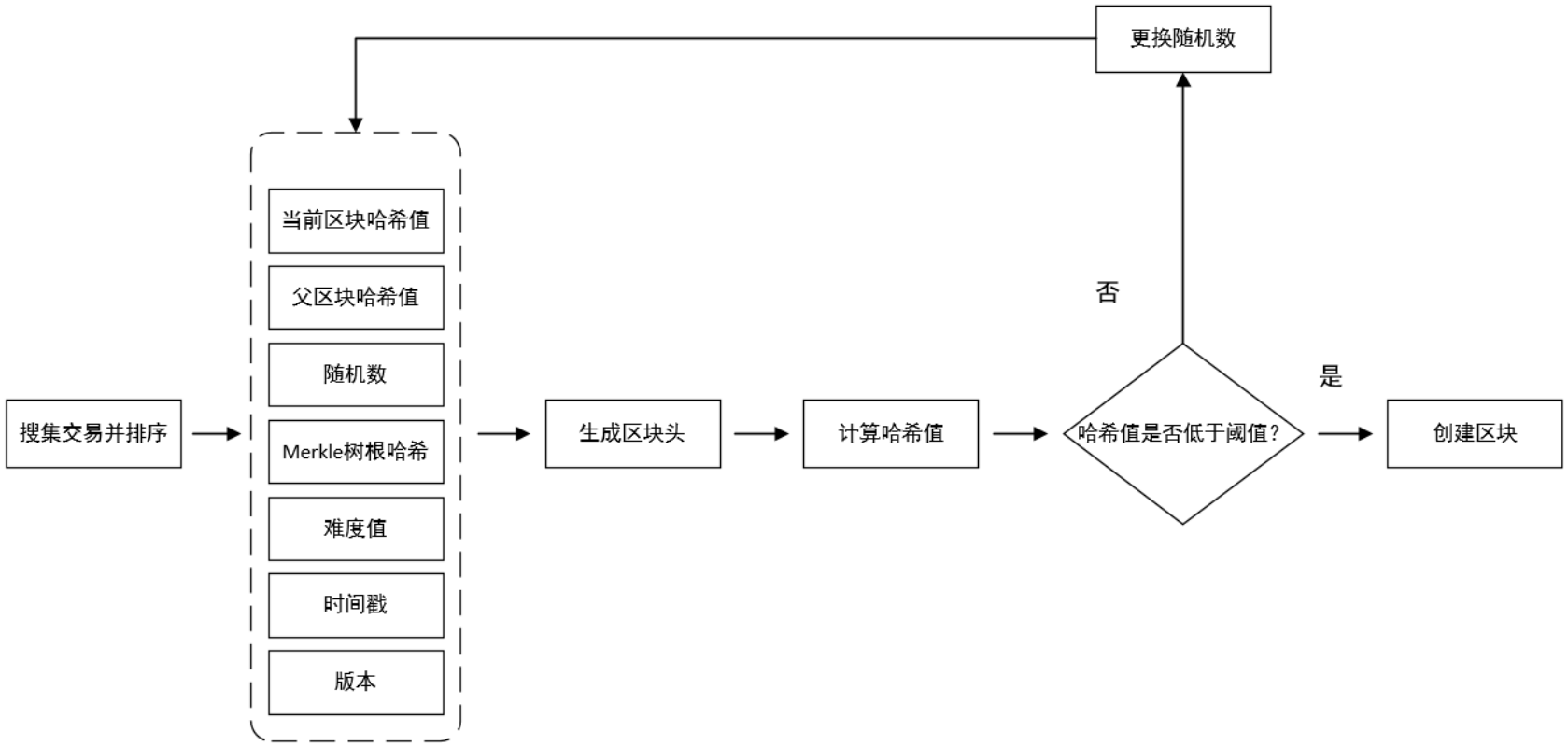
节点获取挖矿难度、交易信息以及历史数据。

02

节点执行循环过程寻找解。

03

新区块的传播与上链。



2.2PoW面临问题

北京大学 PEKING UNIVERSITY

01

性能与安全的权衡

比特币性能低下是一个比较突出的问题，目前比特币每秒的交易额是每秒7笔。

02

能源浪费严重

2019年11月的数据显示，比特币一笔交易平均消耗的能源是431KWh，相当于美国21个家庭一天的用电量。

03

日蚀攻击

如果一个算力较强的恶意节点控制了一个诚实节点与系统其他部分的通信路径，那么该诚实节点就不会为系统做出持续贡献。

04

自私挖矿

一个恶意节点基于某个区块进行挖矿，但不将挖到的区块传播出去，影响诚实节点挖出的区块的传播。

05

双花攻击

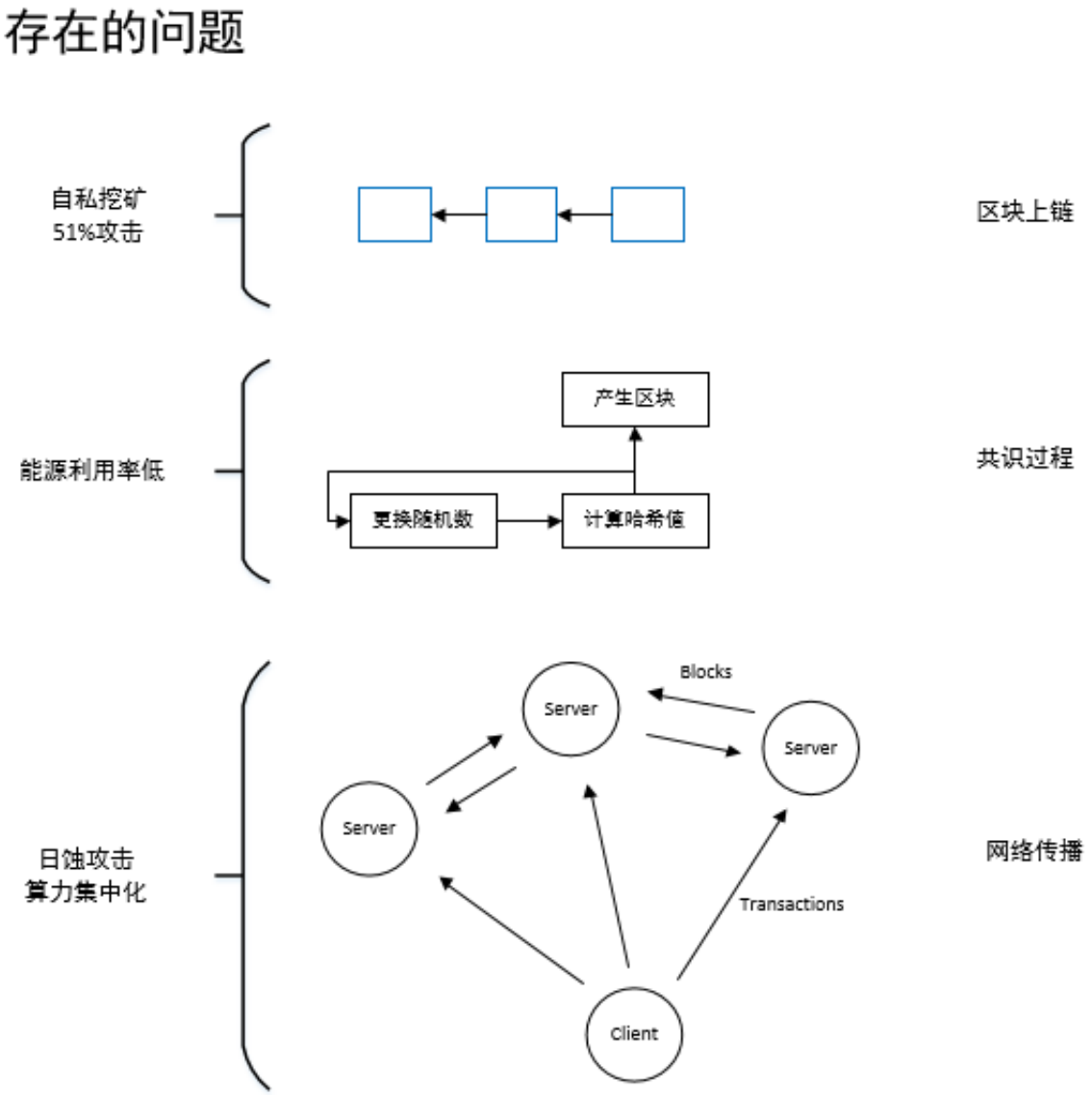
也称作51%攻击，因将同一个比特币进行多次支付交易而得名。双花攻击会使受攻击的区块链网络的安全性被破坏。

06

算力集中化

PoW算法的激励算法决定了矿工的收入与其算力成正比，一些节点会购买高昂的机器进行挖矿，使得其收入提高。

2.2PoW面临问题


存在的问题
自私挖矿
51%攻击
区块上链
能源利用率低
产生区块
更换随机数
计算哈希值
共识过程
日蚀攻击
算力集中化
Server
Server
Server
Blocks
Transactions
Client
网络传播

2.3PoW最新发展

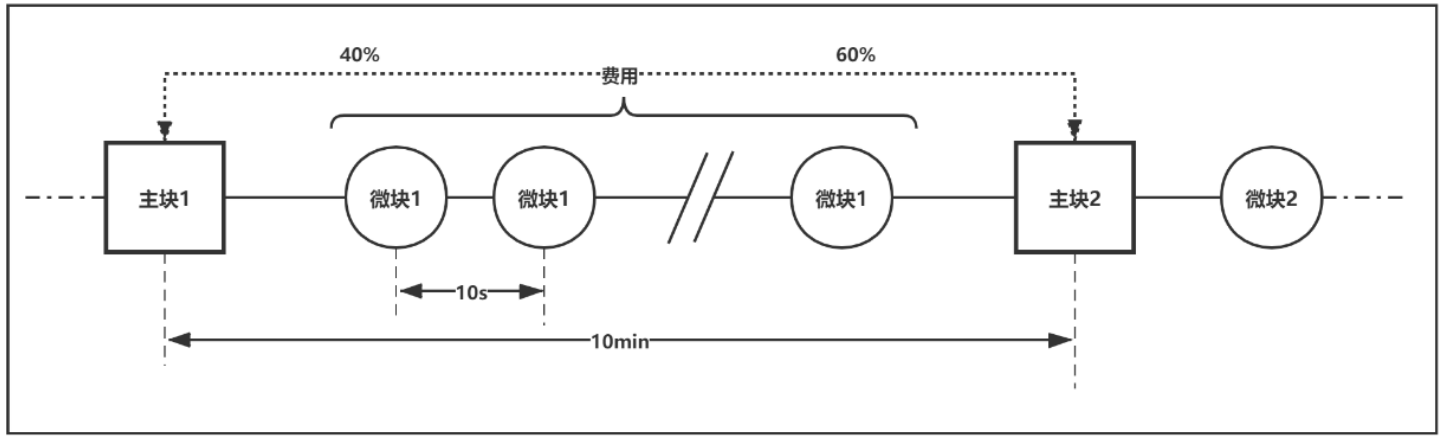
北京大学 PEKING UNIVERSITY

Bitcoin_NG

同比特币不同，在Bitcoin-NG中，区块被分成了两种，分别是核心块（key block）和微块（microblock）。矿工们对区块的操作也被分成了两个阶段，分别是选举领导者和打包交易。

Bitcoin-NG把时间分成了多个时代（epoch），每一个时代选举出一个领导（leader）。选举领导的过程同比特币的挖矿过程相同，每一个矿工在前一个领导生产的微块的基础上进行挖矿。

一旦某个矿工被选定成为领导，这个矿工就可以以预先设定好的速度持续生产微块（microblock），直到下一个矿工在自己生产的微块基础上挖矿成功并成为新的领导者。



2.3PoW最新发展

北京大学 PEKING UNIVERSITY

Fruitchain

Fruitchain主要解决比特币系统中存在的公平性问题。Fruitchain提出了一种新的数据结构“fruit”，可以理解为水果。交易保存在水果中，并且一个区块中会包含多个水果。

水果生成难度低于远低于生成区块的难度，可以理解为微区块的变换形式。在Fruitchain中，对于水果的挖矿以及区块的挖矿同时进行，并且都能够获得收益，因此矿工可以通过对水果进行挖矿获得收益而无需加入矿池，使得系统中的节点更加分散化、独立化，可以避免出现51%攻击的威胁。

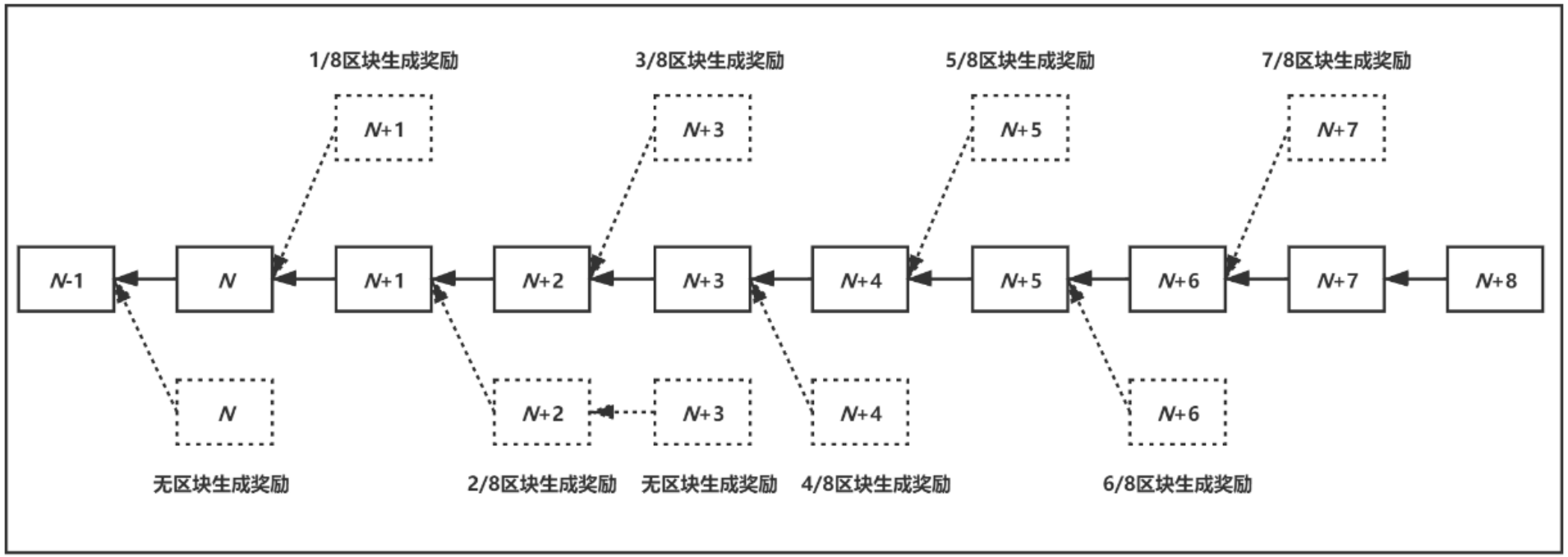
此外，fruitchain对于区块中包含的水果有新鲜度的要求，也就是区块只能包含最近时间内产生的水果，因此攻击者不能通过私自扣留水果的方式进行自私挖矿来获得优势。

2.3PoW最新发展

GHOST

GHOST对基础区块链（最长链为主链）的主要贡献是提供了一套新的选择主链的策略，并且维持了主链在对抗50%攻击时的安全阈值β/λ恒定保持为1，而不随协议λ值和区块大小b的提高（影响网络延迟）而降低。

基础区块链协议增加GHOST后，可以提高出块速率 λ（出块时间缩短），有效提高区块链协议吞吐量(TPS)；目前Ethereum已经采用了一个基础版的GHOST。



第 三 节

PoS类共识算法

01 PoS基本概念

02 PoS共识机制流程

03 PoS面临问题

04 PoS最新发展

3.1PoS基本概念

北京大学 PEKING UNIVERSITY

Proof-of-Stake

Proof-of-Stake（PoS）称为权益证明，是一种替代PoW的有效利用能源的共识协议。

所谓权益是指系统中节点具有的币龄或者币天数，即节点对特定数量货币的所有权。简单来说，该系统中的共识节点通过拥有的币龄数或者token数决定系统记账权。

从安全角度来看，PoS利用token所有权来缓解女巫攻击，与 PoW 矿工相比，PoS 矿工创建区块的机会与其股份价值成正比。

3.1PoS基本概念

北京大学 PEKING UNIVERSITY

Proof-of-Stake

PoS是作为PoW 的替代技术提出的，没有使用算力挖矿的方法，意在解决 PoW 共识速度慢且耗费资源的问题。

PoS 根据节点在系统中持有的权益而非算力的大小获得记账权，权益越大，获得记账权的概率越大。PoS在一定程度上解决了PoW算法能耗大的问题，缩短了区块的产生时间和确认时间，提高了系统的工作效率。

PoS不是一个，而是一类共识算法。

3.2PoS共识机制流程

北京大学 PEKING UNIVERSITY

01

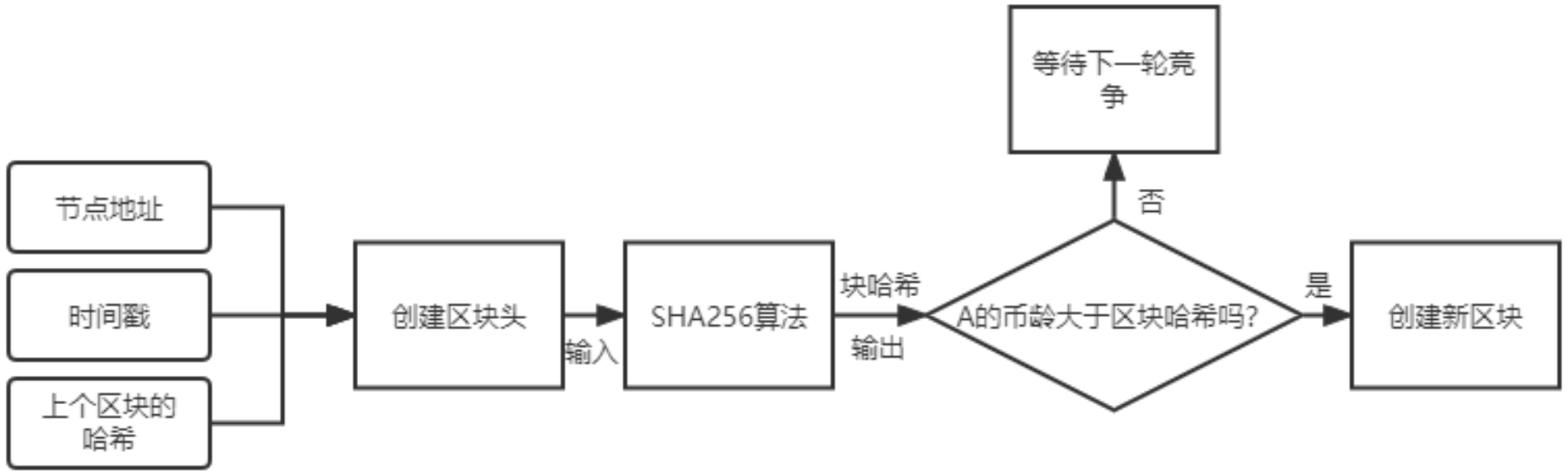
选举出块者：
用户将自己持有的数字货币进行抵押来获取币龄，并且根据PoS公式进行计算，一旦满足公式的条件即可成为出块者。

02

提出区块：
出块者收集系统中的交易，对交易进行验证并且将合法的交易打包进区块中，然后在系统内广播新区快。

03

验证并更新区块：
验证节点对新的区块进行验证，若验证成功，则将其添加到区块链的末端更新区块链，随即开始下一轮的共识。



3.3PoS面临问题

北京大学 PEKING UNIVERSITY

安全问题

PoS 系统中共识节点依赖的不再是算力而是权益，一定程度上降低了节点参与区块共识的代价。但每个区块共识轮次可以产生多个符合条件的共识节点，导致产生多个区块，增加了分叉的发生，降低了分叉进行双重支付的难度，影响了区块链状态的一致性。

PoS要选择恰当的权益标的以保证系统公平。此外，长程攻击也是PoS 系统中潜在的攻击行为，即具有权益优势的节点有可能从创世块开始就产生一条完全不同的区块链分叉。

3.3PoS面临问题

算力集中化

PoS面临与PoW类似的货币集中化的趋势。在PoS中，矿工可以合法地将他们的利润重新投资到永久性的资产中，这使得具有大量未使用代币的节点变得更富有并最终达到垄断地位。

当节点拥有超过50%的流通令牌，共识过程将由该参与者和系统主导并导致完整性得到破坏。以以太坊的Casper FFG为例，提出的PoS方案是建立在PoW基础上的。这使得已经拥有许多货币的节点而言带来了巨大好处，同时也带给系统更多不确定性。

3.4PoS最新发展

北京大学 PEKING UNIVERSITY

DPoS

DPoS（Delegated Proof of Stake）称作委托权益证明，是基于PoS的一种改进版的共识协议。

可以理解为节点通过选举选出共识委员会，由共识委员会负责系统记账权，该共识设计的主旨是控制共识委员会的大小从而达到将共识开销达到一个稳定的状态。

3.4PoS最新发展

北京大学 PEKING UNIVERSITY

Ouroboros

Ouroboros是Kiayias等人在2017年提出的，被数字货币Cardano使用。

Ouroboros将物理时间分为多个固定长度的周期，每个周期又被分为N个阶段，每个阶段选举一个出块者产生区块。在每个周期内，拥有足够权益的节点会通过MPC选择委员会，该委员会负责系统的记账权。

Ouroboros Praos由David等人在2017提出的，致力于解决Ouroboros存在的两个问题：

一是Ouroboros要求严格的同步网络，可能会遭受到去同步网络攻击；

二是Ouroboros的出块节点序列是一定的，因此恶意节点会向下一个周期的节点进行贿赂攻击。

Ouroboros Praos首先将同步网络改造成部分同步网络，允许消息传递延迟最大为δ。其次让节点通过可验证随机函数VRF产生可验证随机函数，如果其数值低于某个阈值，就可以成为出块者。出块者生成区块后将其产生的随机数和VRF对该随机数的证明一起发送到网络中，其他节点会对该消息进行证明。

3.4PoS最新发展

北京大学 PEKING UNIVERSITY

Snow White

Snow White是由Phil等人在2017年提出的共识协议。

该共识协议设计允许节点可以任意在线参与或者离线退出。Snow White采用MPC程序决定共识委员会，按委员会中节点权益占比随机选择出块者。

Snow White的一个亮点是设立了睡眠模型，该模型确保了节点可以在任意时刻离开或者加入网络，但依然可以参与共识。该模型更符合现实网络的情形。

第 四 节

经典共识算法

01 Paxos

02 Raft

03 PBFT

04 SBFT

05 Tendermint

Paxos概述

Paxos算法是由Leslie Lamport于1990年提出来的一种基于消息传递的分布式一致性算法，并且于2013年获得了图灵奖。Paxos算法在分布式领域中有着举足轻重的作用，但是该算法存在着两个明显的缺点：算法不易于理解、工程实现困难。

Paxos算法在Lamport于1998年发表的论文《The Part-Time Parliament》中首次公开。算法的名字来源于希腊一个名为Paxos的小岛，并在论文中描述了Paxos小岛上进行决议的流程，以此提出了Paxos算法。然而这样的描述对于读者来说实在难以理解，于是在2001年Lamport重新发表了更为简单明了的算法描述版本《Paxos Made Simple》。

在常见的分布式系统中，总会存在网络异常或是机器宕机等等情况。而Paxos算法首先解决的问题就是如何在一个可能发生异常的分布式系统中，可靠地在节点之间对某一个数据达成一致。

4.1Paxos

提议者(Proposer)

01

提议者的工作是负责提出提案(Proposal)。所提出的提案中包含的信息有提案编号n，以及提议的值Value。如果所提出的提案被超过半数的决策者接受，那么认为该提案被批准。

决策者(Accepter)

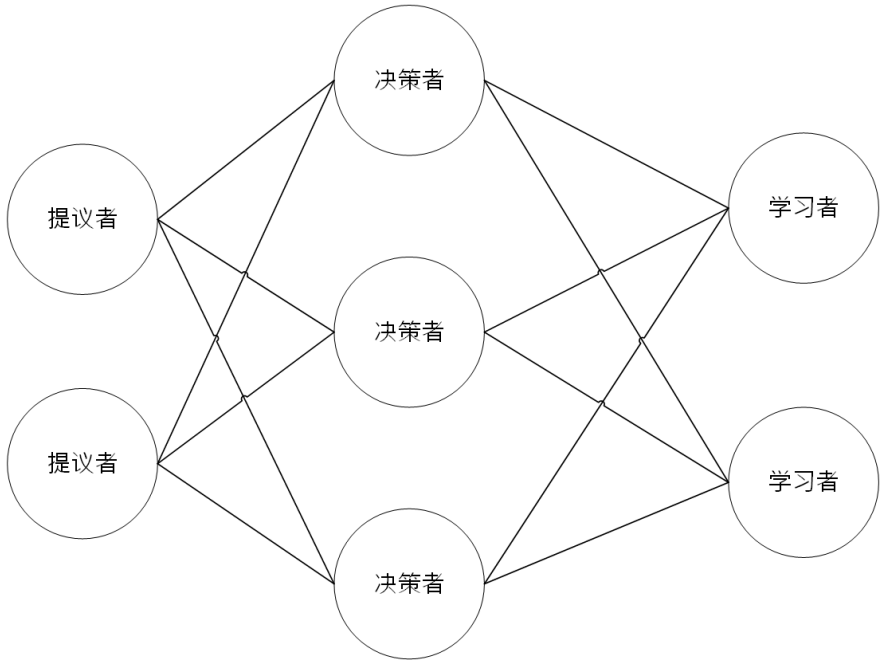
02

决策者的工作显然就是参与决策，对提案进行回应。决策者在收到提案之后可以选择接受提案或是不予以响应。

学习者(Learner)

03

只需要对共识的结果进行备份，不参与决策过程，会被告知共识的结果。



第一阶段：Prepare阶段。

提议者向决策者发出Prepare请求，请求信息中包含有提案编号n。

决策者接收到提议者的请求后，与自己保存的提案编号最大值max_n进行比较。

若n>max_n则更新max_n，并且将Promise承诺返回给提议者；否则决策者不做响应。

若提议者可以收到半数以上决策者的承诺，就可以进入下一阶段；否则重复第一阶段。

第二阶段：Accept阶段。

提议者收到足够多的承诺之后，向已经承诺的决策者发送一个Accept请求，Accept请求中包含编号n以及值Value。

如果决策者收到一个针对编号n的提案的Accept请求，只要该决策者对没有编号大于n的Prepare请求做出过响应（即此时的max_n与Accept请求中的n相等），决策者就会接受当前提议者的提案。

4.2Raft算法

北京大学 PEKING UNIVERSITY

Raft概述

与Paxos相同，Raft致力于解决分布式系统中存在网络异常或是节点宕机情况下的一致性问题。

与Paxos晦涩难懂的算法不同，由斯坦福大学的教授在2014年发表的分布式协议Raft更加注重协议的落地性以及可理解性。

4.2Raft算法

领导者(Leader)

01

负责接收客户端的请求，向跟随者同步请求日志，当大多数节点同步完成之后要求跟随者进行日志提交

跟随者(Follower)

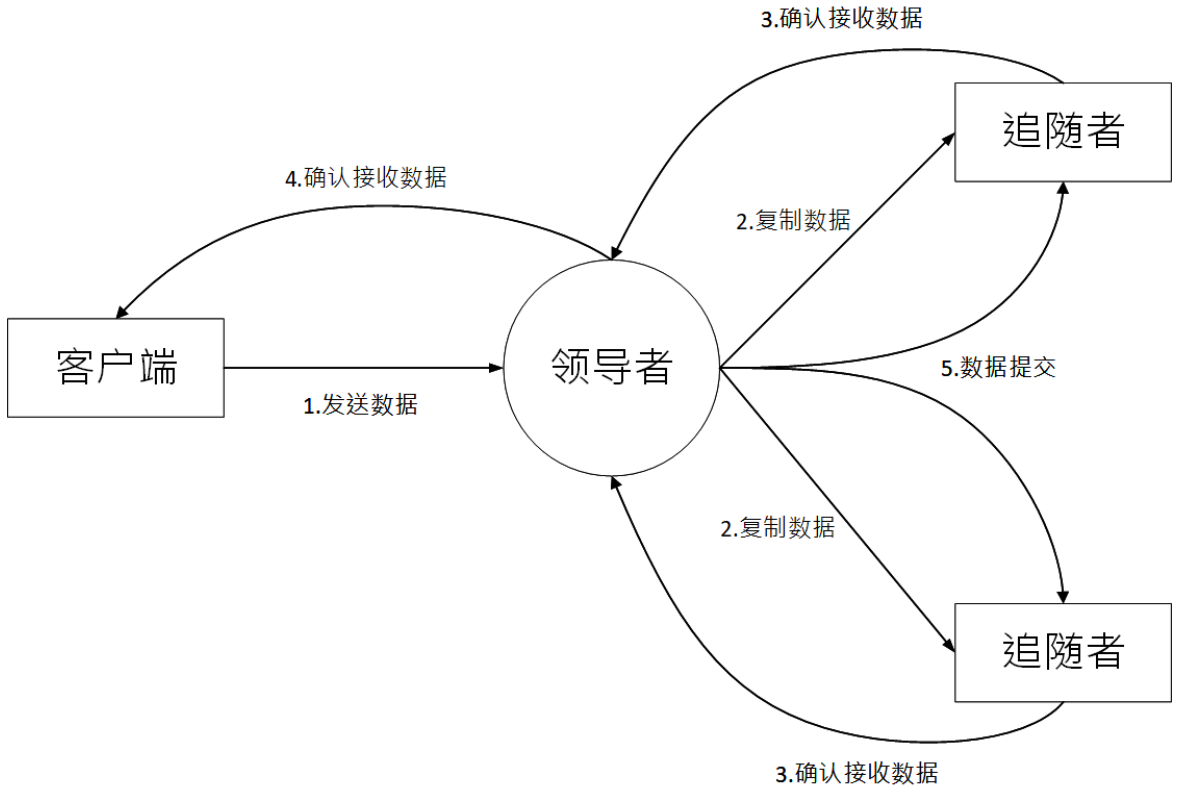
02

接收并且维护领导者同步的日志，在收到领导者的提交要求后进行提交。参与选举或是被选举。

候选者(Candidate)

03

在领导者选举时的临时状态，是跟随者以及领导者状态间的过度。



4.2Raft算法流程

第一阶段：领导者选举阶段。

一个使用Raft算法的网络中最小需要有3个节点才可以在选举过程中产生领导者。领导者通过心跳消息证明自己的存在，当节点没有收到心跳消息时认定领导者死亡，在一段随机长度的等待时间之后发起领导者选举。

系统在初始化时，所有节点都是跟随者状态。发起领导者选举的节点会成为候选者，此时也会有其他节点发现领导者死亡于是成为候选者。

接着节点间开始互相拉票，最先获得超过总节点数一半票数的节点成为领导者，并且广播自己的领导者状态。其余候选者节点回退到跟随者状态。新的领导者需要定时向跟随者发送心跳消息维持自己的“地位”。

第二阶段：日志同步阶段。

Raft算法依靠领导者节点来确保系统数据的一致性，因此领导者节点的可用性是十分重要的。

当客户端向领导者发送请求后，领导者把请求作为日志条目写入它的日志中。并向跟随者节点发送这条日志条目，跟随者成功接收后向领导者响应。

领导者收到超过半数节点的接收响应后，向客户端确认请求已经被接收。客户端发出请求接收的Ack响应，表明此时的请求已经提交，领导者向跟随者告知请求已经提交。

PBFT概述

在经典的分布式算法中，RAFT以及Paxos都是非常高效的算法，然而他们都只考虑到了系统内节点宕机或是网络异常等情况，并未考虑到系统内可能存在的恶意节点，也就是拜占庭将军问题。

拜占庭容错问题（BFT, Byzantine Fault Tolerance）从上个世纪80年代开始被研究至今，PBFT（Practical Byzantine Fault Tolerance）是其中最著名的算法。

算法由Miguel Castro和Barbara Liskov在1999年提出，解决了原始拜占庭容错算法效率不高的问题，将算法复杂度由指数级别降低至多项式级别，使得拜占庭容错算法在实际系统应用中变得可行。

PBFT算法流程

PBFT是一种基于状态机副本复制的算法，要求网络中的所有节点采取一致的行动，共同维护一个状态。

PBFT算法包含了三类基本协议：一致性协议、检查点协议以及视图更换协议。在系统正常运行时，只有一致性协议和检查点协议被激活。

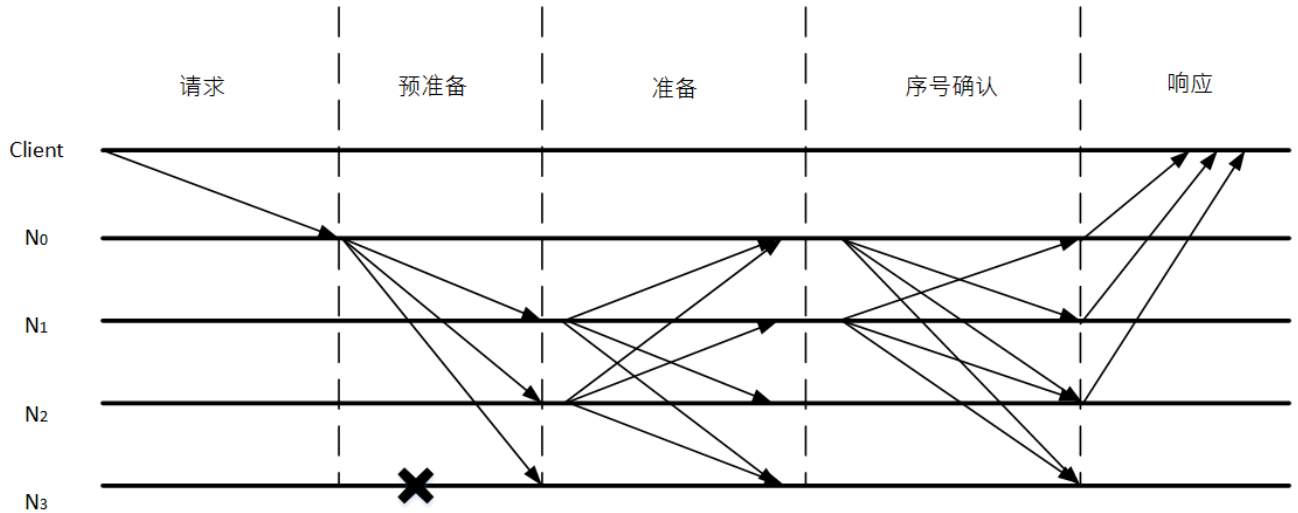
视图更换协议只有在主节点运行缓慢或是出错的情况下才会启动，以保证系统能够继续执行客户端的请求。

PBFT算法流程

PBFT系统中所有的节点被分为两类：主节点和从节点。

除了唯一的主节点之外其余节点都是从节点。主节点通过选举产生，所有节点都有选举权和被选举权，所有节点在选举过程中被选中的概率都是相等的。PBFT中将一个选举周期当作一个视图，即一个视图对应了某一个节点作为主节点时的情况。

一致性协议中包含了五个阶段：请求(Request)、预准备(Pre-prepare)、准备(Prepare)、序号确认(Commit)、响应(Reply)。

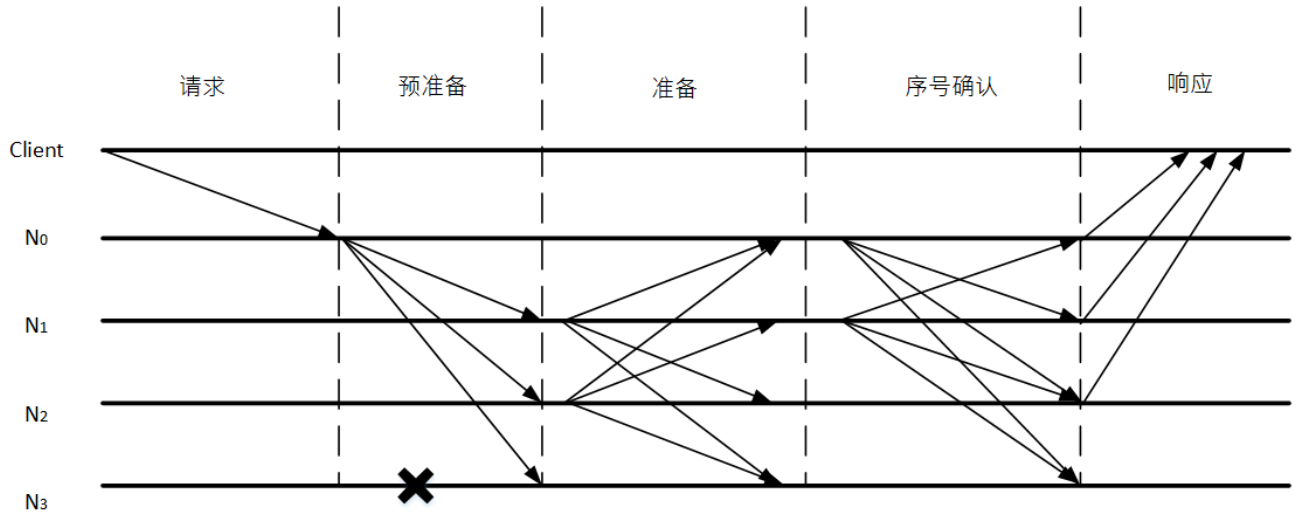


PBFT算法流程

PBFT系统中所有的节点被分为两类：主节点和从节点。

除了唯一的主节点之外其余节点都是从节点。主节点通过选举产生，所有节点都有选举权和被选举权，所有节点在选举过程中被选中的概率都是相等的。PBFT中将一个选举周期当作一个视图，即一个视图对应了某一个节点作为主节点时的情况。

一致性协议中包含了五个阶段：请求(Request)、预准备(Pre-prepare)、准备(Prepare)、序号确认(Commit)、响应(Reply)。



1、请求阶段

客户端发送请求到主节点。激活主节点服务。

2、预准备阶段

主节点给当前请求分配一个序号n，然后发送预准备消息给从节点。

预准备消息中包括视图编号v、当前的消息序号n、消息m的摘要d。

3、准备阶段

从节点在接收到预准备消息之后在消息上签名并转发给其他所有节点。其他从节点在接收到消息后会对消息进行校验和确认，例如：检查签名合法性、信息m与摘要d的一致性、当前视图编号是否与消息中的视图编号v一致。将校验通过的消息存入消息日志中，对消息签名并转发给包含主节点在内的所有节点，此时转发的消息称为准备(Prepare)消息。若校验未通过，则进入试图更换(View change)阶段。

4、序号确认阶段

每个节点收到其他所有节点的准备消息之后，先对消息进行签名验证和确认。把签名验证通过后的消息与自己的消息日志进行比较，如果超过2f个消息请求都一样，那么认为这是所有节点的共识请求。该节点对这个请求进行响应，给出运算结果，并且签名。

5、响应阶段

将序号确认阶段的处理结果发送个客户端，客户收到f+1个相同的运算结果后，就认为该结果正确。

4.3PBFT共识算法

PBFT的分析

PBFT算法最明显的优势在于改进了原始拜占庭容错算法复杂度过高的问题，不仅能够容纳网络中故障节点的存在，同时容纳了一定的恶意节点——在网络节点数量为3f+1时可以容纳f个恶意节点，保证了系统中的活性(Liveness)以及安全性(Safety)。

然而PBFT算法的通心复杂度为O(n^2)数量级，随着系统中的节点数量增加，算法性能会快速下降。PBFT算法仅仅适用于联盟链或是私有链。

4.4SBFT共识算法

SBFT概述

可扩展的拜占庭容错 (Scalable Byzantine Fault Tolerance，SBFT) 协议，是由Guy Golan Gueta等人在2018年首次提出来的。

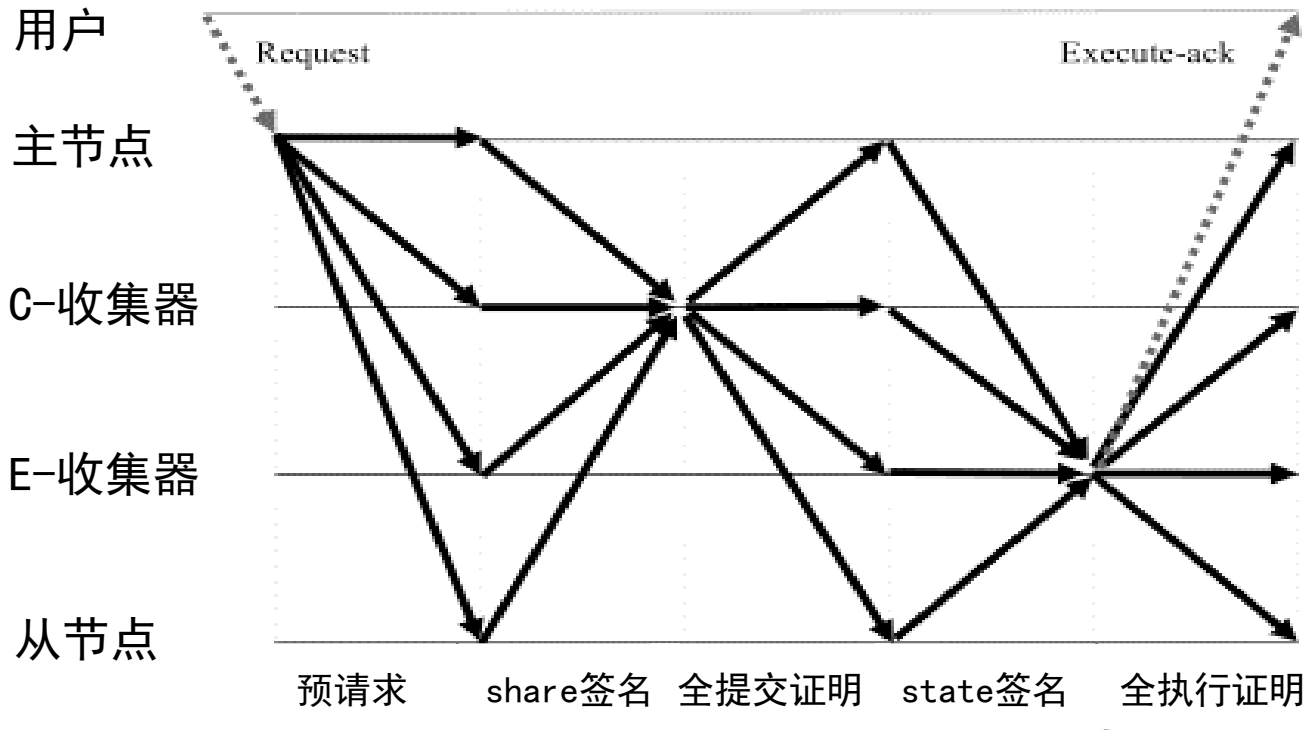
该协议的核心思想是使用收集器（collectors）来减少以实用拜占庭容错协议为代表的共识算法中的all-to-all通信。即，在SBFT中节点不用广播自己的消息给其他节点，每个节点都发送消息给收集器而收集器再向每个节点广播。

节点发送消息时，以阈值签名（ threshold signatures ）加密，收集器的存在可以将时间复杂度大小从线性级减少到常量级。

4.4SBFT共识算法

SBFT算法流程

可扩展的拜占庭容错协议主要分为Pre-prepare、Sign-share、Full-commit-Proof、SignState和Full-excute-proof共5个阶段。



4.4SBFT共识算法

北京大学 PEKING UNIVERSITY

1、Pre-prepare阶段

主节点将客户的请求打包成块，广播给去其他节点；

2、Sign-share阶段

节点检查view,sequence等，最主要的是检查交易r和交易的摘要SHA256(r||s||v) = h是否一致，通过后将摘要等信息用自己的share签名成σ(h)部分提交给C-collector节点；

3、Full-commit-Proof阶段

收到一定数量的各节点的σ(h)部分后，得到完整的签名σ(h)，将σ(h)发给所有节点；

4、SignState阶段

b节点收到σ(h)后，验证有效性，之后执行request的操作（比如将块写入区块链），然后，向E-collectors发送阈值签名状态消息；

5、Full-excute-proof阶段

每个E-collector都会收集签名share，并为决策块创建一个简洁的完全签名，它将证书发送回节点，表示其操作已执行。

4.4SBFT共识算法

SBFT的分析

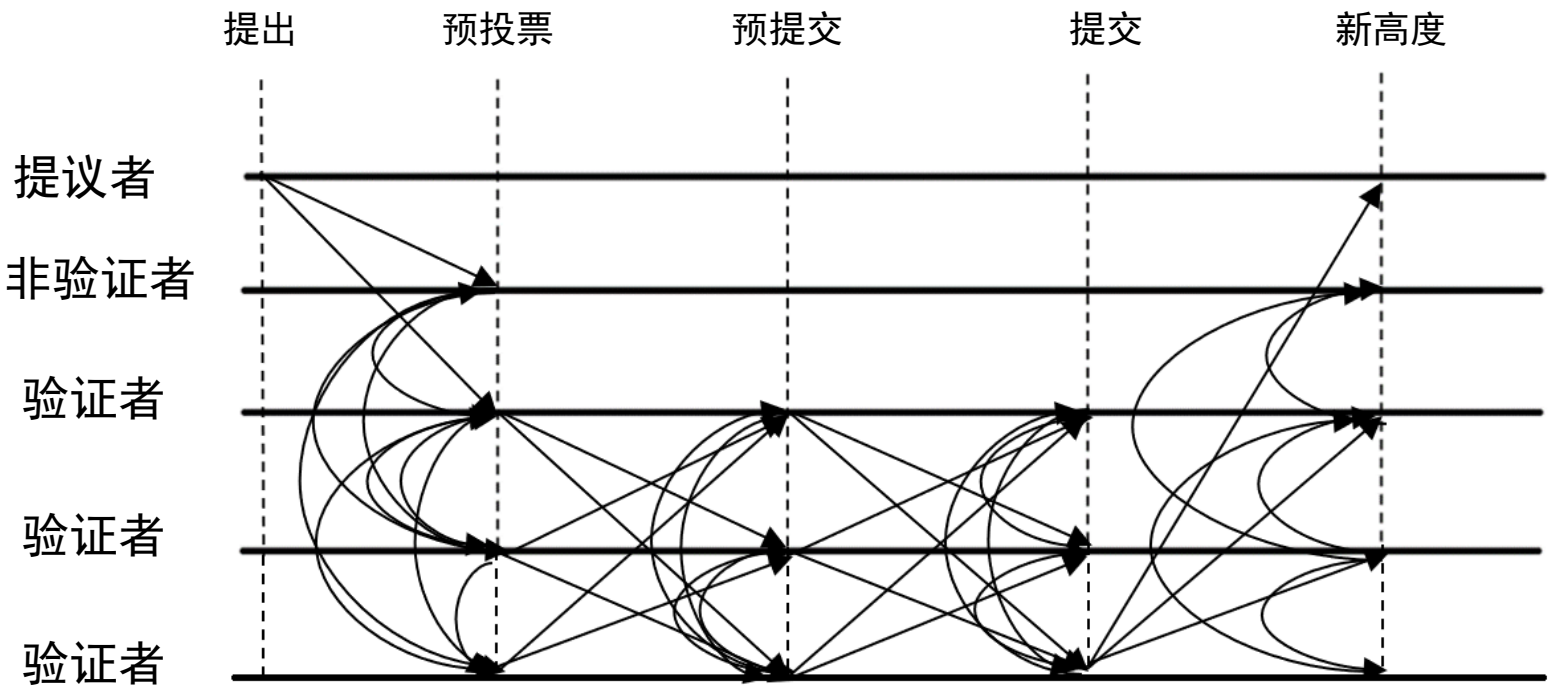
和实用拜占庭容错协议相比，由于收集器的存在，可扩展的拜占庭容错协议交易吞吐量更高，处理效率更快。

但是所有节点都只和收集器通信有点趋于中心化的趋势。收集器对系统影响很大。

4.5Tendermint算法

Tendermint概述

Tendermint 由Jae Kwon2014年提出来的改进PBFT的一种拜占庭容错算法，它通过加权轮询的方式产生验证者集合。共识过程一共包括NewHeigh、Propose、Prevote、 Precommit 、Commit共5个阶段。



4.5Tendermint算法

北京大学 PEKING UNIVERSITY

1、Propose

Proposer节点会通过gossip协议将提议发送给所有节点（Tendermint中包括Validator节点和Non-Validator节点）；

2、Prevote

Non-validator节点收到Proposal消息后只转发消息，Validator节点收到Proposal后会进行第一轮投票。如果当前结点还lock在上一块则给会给上一块投票，得票率超过2/3则进入下一阶段；

3、Precommit

对提议进行第二次投票。若通过则进入Commit阶段，超时则会出现进入Propose阶段。值得注意的是一块数据要进入Commit阶段可能要经过多轮共识阶段；

4、Commit

设置本次提议Commit时间，将区块写入链。节点进入下一个NewHight阶段。

有可能会有部分节点由于没有收到足够的Precommit投票导致无法Commit，这个时候可以通过同步来使各个节点的状态尽量保持一致，即节点已经Commit时，就可以广播一条消息携带VSet给其他节点，其他节点验证对于块的Commit是否有效；

5、NewHight

区块链高度+1，表示上一轮Commit结束，下一轮共识开始。同时该阶段会选定一个Proposer。

4.5Tendermint算法

Tendermint的分析

相较于PBFT，Tendermint 加入了PoLC和lock等概念，使得它对网络延时，宕机等造成影响可以进一步减小（一批交易，一次共识不通过，可以多轮共识）。

另外Tendermint通信采用的是gossip协议，因而它的可扩展性更强，可以允许节点的增多，减少。信息传播收敛性更强。实验称Tendermint每秒可以处理10,000笔交易。另外gossip通信对节点宕机，断电等的容错性也更强。

由于在BPFT之上增加了其他概念，会增加了其他的额外的计算通信开支。另外gossip通信协议的引入也存在消息冗余、带宽负担增加、消息延迟（提议要经过多次传播才能覆盖全网）等问题。当共识节点增多时带宽负担增加，Tendermint效率应该会明显下降。

第 五 节

其他共识算法

01 基于TEE的共识算法

02 分片共识算法

03 DAG共识算法

04 其余算法

5.1基于TEE的共识算法

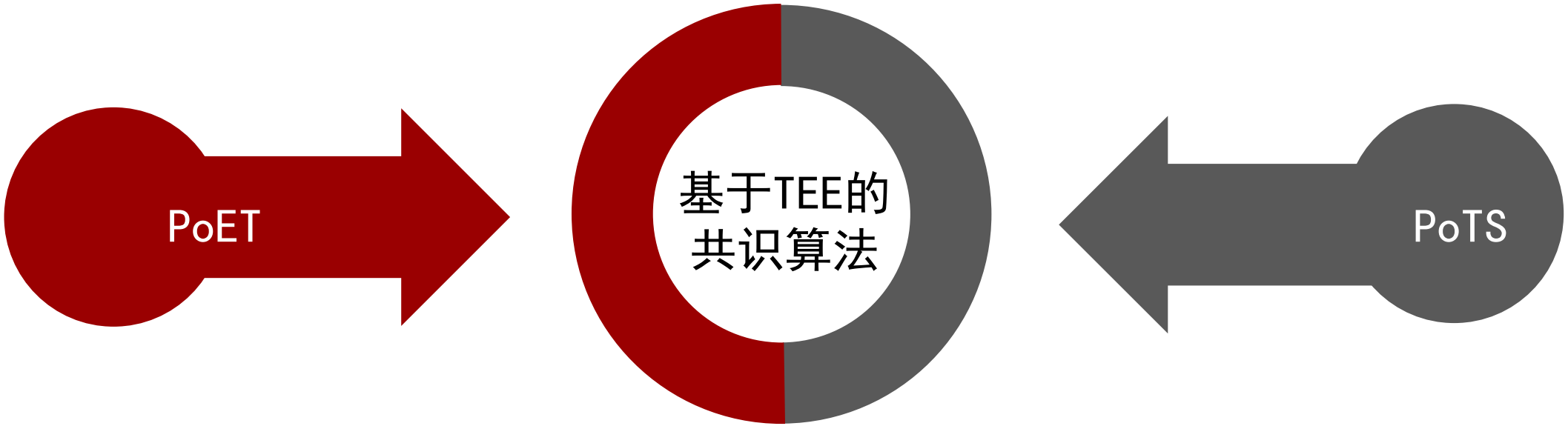
基于TEE的共识算法

与可信计算技术结合使用。可信计算环境（trusted execution environment，TEE），是应用程序在内存中运行时开辟的隔离区域（也叫做enclave），这类技术扩展了处理器的指令集向外提供数据完整性和机密性。

随着TEE技术的出现，拜占庭容错（BFT：Byzantine Fault Tolerance）问题也变得不那么重要了，因为只要参与了共识的节点都采用了TEE，那么节点的共识代码就具备了防篡改的效果。目前，Intel SGX以及Arm TrustZone是主要的两种TEE解决方案。

目前的研究将可信计算应用到共识算法设计中，以此提高共识的安全性和系统的稳定性。在实际的应用中，目前TEE的安全区域还不能将整个共识算法保存在其中，但是仍然可以将某一部分可能被用来作弊的部分代码放进去。

5.1基于TEE的共识算法



5.1基于TEE的共识算法

PoET

Proof of Elapsed Time（PoET）由Intel在2016年提出并用于Hyperledger Sawtooth系列项目中。该共识协议要求每个节点在发布区块之前任意选择一个时间进行回退，选择的时间符合指数分布，回退算法会安装在enclave中。以下是该协议的步骤：

1、节点i首先从可信的安全服务源获得PoET代码，同时将代码安装在本地带有SGX的机器上，其中关于回退算法的代码安装在enclave中。PoET对节点i生成一个密钥对<Pki，Ski>，节点利用该密钥对生成远程认证报告发送到网络中，其他节点将会通过Intel认证服务（IAS）对节点i的认证报告进行验证。

2、节点i通过验证后开始进行挖矿。在每一轮区块周期内，节点i根据回退算法生成的随机数等待一段时间，等待结束后将新区块以及用私钥签名的证书广播到网络中。其他节点收到消息后用节点i的公钥进行验证，如果验证通过后新的区块会被添加到系统账本中。

5.1基于TEE的共识算法

PoTS

Proof of TEE-Stake（PoTS）是一个采用TEE的共识协议。

该协议中的节点在前期的注册和准备工作与PoET类似，PoTS的后期则采取不同的步骤：共识首先根据节点权益贡献大小随机选举组成共识委员会，共识委员会中的节点都有资格产生下一个区块。

节点使用私钥对生成的区块进行签名同时将区块传播到网络中，其他节点会对收到的消息进行验证并使用最长链规则决定是否接受该区块。

5.2分片共识算法

北京大学 PEKING UNIVERSITY

分片共识机制概述

分片共识机制是为了解决区块链处理交易的可扩展性，从而利用多个并行的委员会来处理网络中不同分片的交易的混合共识机制。

分片的方式有：通信分片、计算分片以及存储分片。

5.2分片共识算法

北京大学 PEKING UNIVERSITY

通信分片

将系统中的网络分为不同的片区，每一个片区有一个委员会负责，每一个委员会内部成员在大多数情况下都只需要进行片区内部通信

计算分片

每一个分片的委员会负责处理其对应的交易。计算分片的方式可以随着网络中交易量的增加而增加委员会，从而实现交易吞吐量的可扩展性。

存储分片

将验证后的交易交给对应分片的委员会进行存储，每一个分片委员会只负责处理本分片对应的交易并且将将以存储到本分片所属的区块链中。

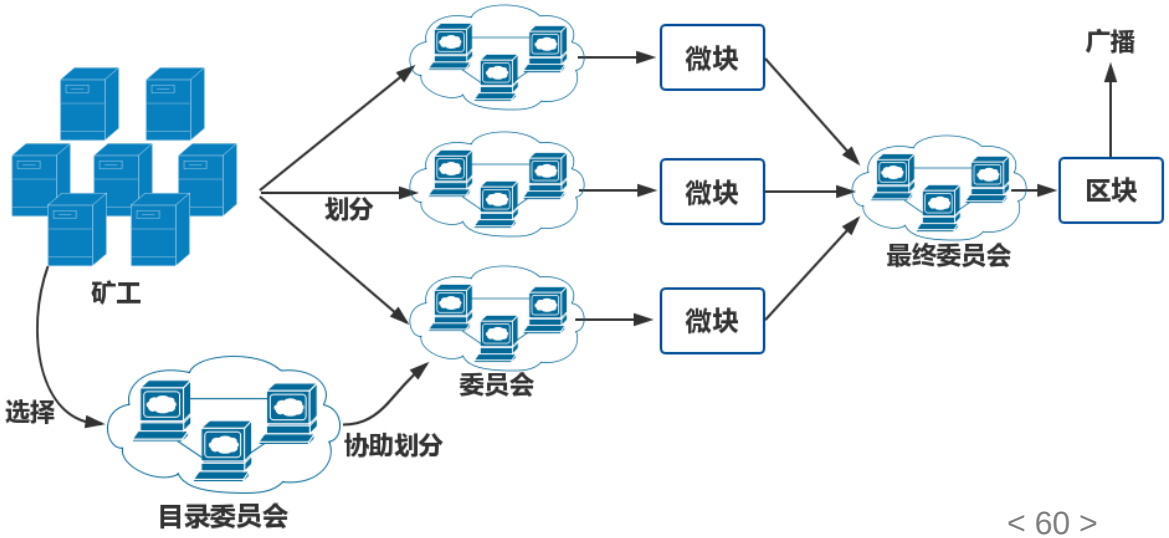
5.2分片共识算法

Elastico

Elastico由Luu等人提出，其核心思想是将网络中的节点分成多个规模较小的委员会，每个委员会处理彼此独立的交易集合，这些交易集合被称为shard。

Elastico将时间分成多个周期，每个周期包括五个步骤：一是创建身份并成立共识委员会会，二是对委员会进行配置，三是委员会使用拜占庭协议打成出块共识，四是委员会将最终的结果进行传播，五是为下一个周期的委员会筹备进行随机数的生成。

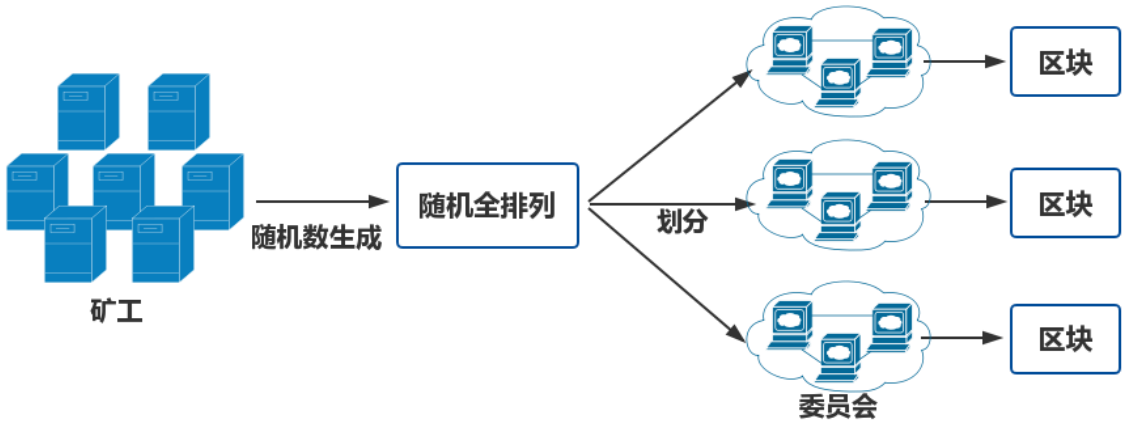
每个委员会内部打成共识形成，完成了对性能的线性扩展。



5.2分片共识算法

Omniledger

Ominiledger由Kokoris-Kogias等人提出，旨在解决分片区块链目前存在的一些问题。Omniledger采用UTXO模型，每个分片内的节点只需要处理和存储该分片对应的UTXO数据。Omniledger存在两种区块链，即身份区块链和交易区块链，身份区块链用于记录每个时期参与的节点和其对应的分片信息，交易区块链用于记录每个分片产生和维护的交易。

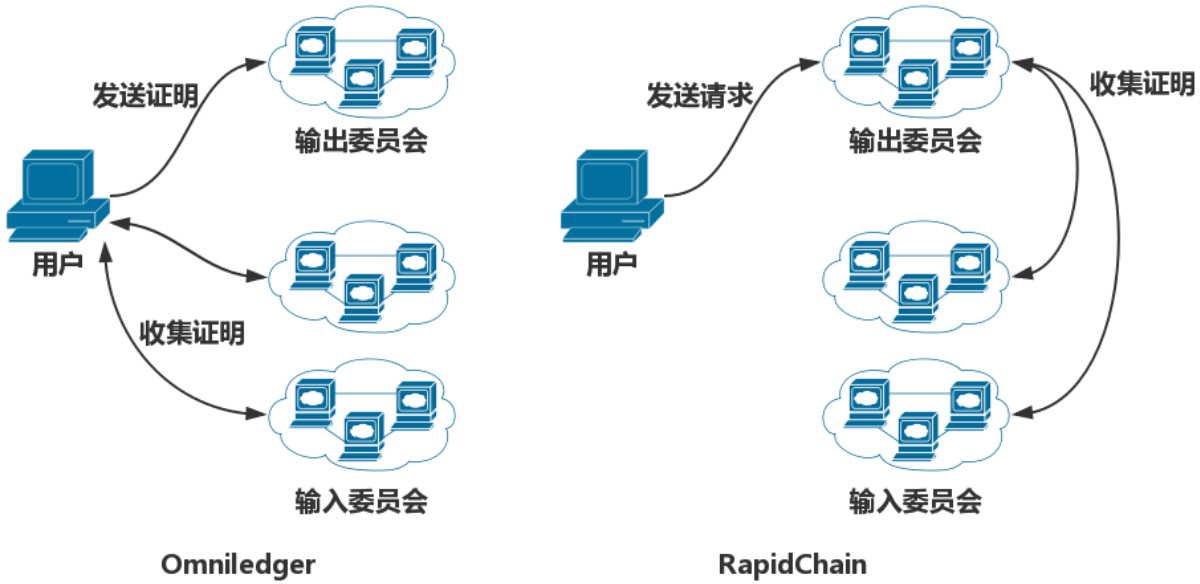


5.2分片共识算法

RapidChain

RapidChain由Zamani等人提出，该分片协议采用状态分片技术完成对区块链性能的线性扩展。

该共识协议分为三个阶段：引导启动阶段、共识阶段以及重构阶段。引导启动阶段只在RapidChain系统初始化时刻运行一次，创建参考委员会对节点进行随机分配，实现分片委员会。共识阶段分片委员会负责处理交易和达成共识。在重构阶段，委员会会验证新加入节点并将新节点随机分配到各委员会中。



5.2分片共识算法

北京大学 PEKING UNIVERSITY

Chainspace

Chainspace由Al-Bassam等人提出，Chainspace实际上是一个基于分布式账本的去中心化的基础设施，在分片的基础上创建了智能合约应用平台，该平台允许任何节点加入其中发布智能合约，实现的交易吞吐量为350tps。

5.3 DAG共识算法

北京大学 PEKING UNIVERSITY

DAG共识算法概述

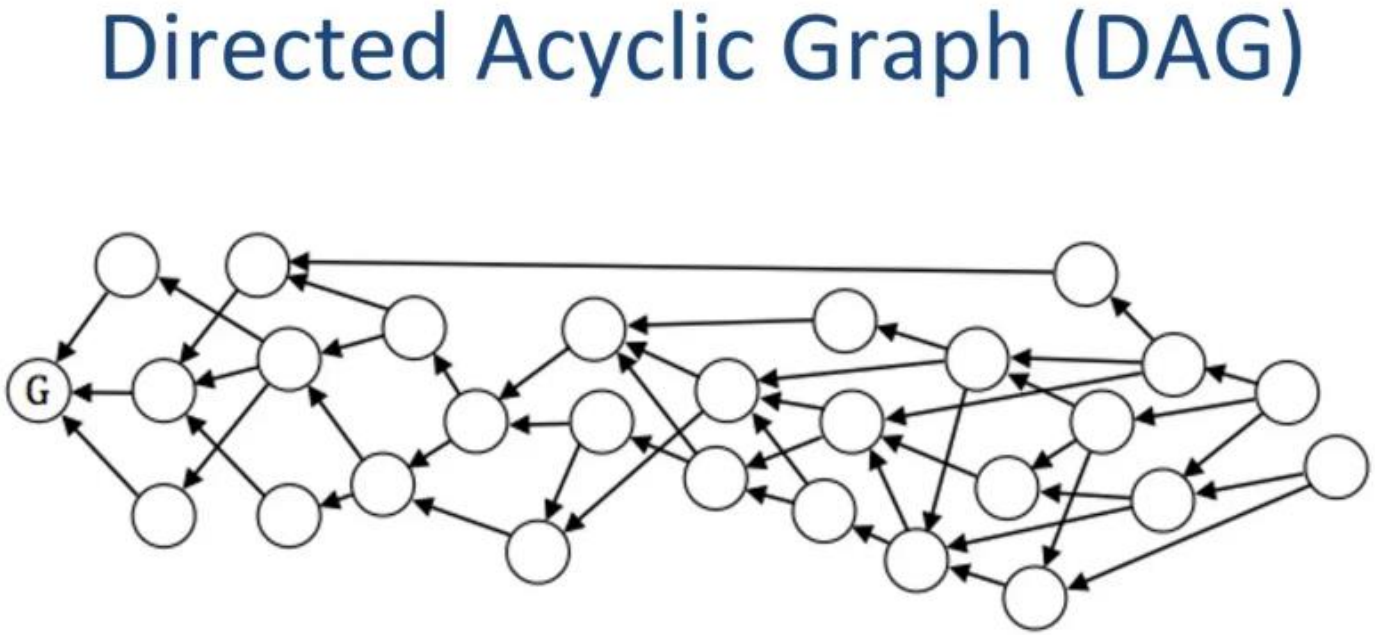
近几年基于有向无环图的区块链共识成为提高区块链性能的一个方向。有向无环图是计算机专业中的一种数据结构，图中的每一个节点可以看做是一个区块或者一笔交易，根据这样的分类分成blockDAG以及txDAG。

DAG区块与传统区块链区块的最大却别在于每一个区块链区块只能指向唯一的父区块，而DAG区块可以指向不同的多个前代区块。

5.3 DAG共识算法

DAG共识算法概述

简而言之，DAG可以堪称一种允许分叉的区块链，而允许分叉就意味着系统的出块速度可以超过广播速度。以比特币为例，为了尽可能避免分叉的产生，所以设置了平均出快速度为10分钟，因此降低了系统的交易吞吐的性能。因此采用DAG结构的区块链能够实现交易处理的提升。下面介绍几种比较经典的基于DAG的共识协议。



5.3 DAG共识算法

北京大学 PEKING UNIVERSITY

SPECTRE

该方案由Sompolinsky等在2016年提出，SPECTER是第一批有据可查的blockDAG提案。

SPECTER允许任何想要的节点挖掘一个新块以查找度数为零的所有块，并将新的块标题散列到在开始对新区块进行PoW挖掘之前。节点将新开采的块广播到网络，之后每个节点都会启动一个递归投票程序来确定当前DAG中任何两个块的顺序。该递归过程最终导致成对排序 blockDAG。这种成对排序方案本质上允许SPECTER在两个冲突的区块之间做出决定。

5.3 DAG共识算法

北京大学 PEKING UNIVERSITY

PHANTOM

PHANTOM是SPECTER的blockDAG提案。

SPECTRE的一个显着缺点是成对排序的块可能不会扩展为完全线性的顺序。这导致SPECTRE的活性度较弱（即仅自然支持按时间顺序排列的交易）和增加平衡风险冲突中的攻击可能无法解决。

相比之下，PHANTOM实现了交易和交易的完全线性排序通过以下算法在DAG中进行块处理。每个节点在blockDAG中搜索块的最大k簇。k表示集群中的节点度。k集群被认为是诚实的并且其中的所有块都是线性排序的。交易然后按新顺序验证群集覆盖的范围。值得注意的是，最大的0簇情况等同于中本聪共识的最长链规则。

PHANTOM可以与SPECTER一起使用来获得更灵活的共识表现，这两个顺序方案是相互补充的。

5.4其余算法

北京大学 PEKING UNIVERSITY

PoA

PoA（proof of authority）即授权证明，是以太坊联合创始人Gavin Wood提出的一种代替PoW以及PoS的共识算法，是基于PoS进行改进的增强版本。

PoA运用节点身份的价值，将节点的信誉作为系统记账权的凭证，这种节点也称为validator。Validator负责生成区块以及记录交易并获得区块奖励以及交易费用。如果validator进行作恶，该节点会被用户或者其他validator剔除掉。

PoA共识显然存在一个中心化的问题，Validator就是系统网络中的一个明确中心。普通节点对于validator几乎没有监督能力。如果validator自己对网络发起攻击，就非常容易成功。为了保证PoA的去中心化特性，PoA系统中也可以同时有多个validator，并且所有的validator通过多重签名机制来签署验证的交易，只要保证了多个validator来自不同利益群体，就可以有效抵御来自validator发起的攻击。

5.4其余算法

PoR

PoR（proof of retrievability）即可检索证明，是由Juels等人在2007年提出的。

该共识的核心特性是允许文件所有者通过挑战应答协议检查文件或者文件片段是否被安全存储，在某个远程节点上针对文件的可检索性可以证明该节点确实消耗了存储资源保存文件。使用存储资源用来替换计算资源是一个十分有效的方式，因为存储资源可以被回收，同时也可以用来保存一些有意义的数据集实现资源的双重利用。

5.4其余算法

PoSV

PoSV（proof of stake velocity）即权益-速度证明，由Larry Ren在2014年提出的Reddcoin中实现，该共识协议致力于解决到PoS的缺点：币龄与时间成线性增长关系导致的持币人屯币的现象。

该协议分为两个阶段，第一个阶段采用PoW协议实现代币的分配和统计，第二阶段采用PoSV协议实现共识，将币龄与时间的线性关系修改成指数衰减函数，随着时间的增加，节点持有的旧币的币龄增长率会不断下降，持有的新币的币龄增长率比旧币币龄增长率快，使得屯币的利益降低，一定程度上减缓了持币者恶意屯币的现象。

5.4其余算法

PoB

PoB（proof of burn）即燃烧证明，是一种致力于解决PoW高耗能问题的共识协议，该共识协议被称为无资源浪费的PoW协议。

该共识在系统初期使用PoW产生初始的代币，随着时间的发展和系统规模的扩大，再使用PoS争夺系统记账权。

该共识的特色是节点将持有的货币发送到特定的地址中，该地址的特点是无法找回的，该过程被称为燃烧。燃烧的货币越多，节点越大概率挖出下一个区块获得记账权。

5.4其余算法

PoI

PoI（proof of importance）即重要性证明，该共识由NEM提出并使用。

该共识将节点的重要程度作为争夺系统记账权的凭证，而不是节点的计算资源或者所持有的的权益数量，重视节点对整个网络的整体贡献和支持。NEM根据钱包的交互次数以及过去30天内交易数量和规模作为重要程度的指标。

本章小结

共识算法是区块链和分布式系统的核心，本章介绍了共识算法的基本概念、发展历史和基本假设，详细介绍了区块链中最常见的PoW和PoS两类共识算法和几种改进算法，及其常用于联盟链的PBFT和Raft等几种经典共识算法，最后简要介绍了一些适用于TEE、分片、DAG等环境的共识算法。

当前共识算法的研究既有挑战也有机遇，是否能发现新的代价较低的竞争性资源以降低共识算法的开销？结合不同的竞争性资源是否可以提高共识算法的效率？集成不同共识算法扬长避短是否可行？是否有可以和资源受限的智能终端和物联网设备适配的共识算法？共识算法需要综合考虑安全、性能、效率、激励、公平等多种因素，还需要衡量所适用应用场景和环境。

北京大学
PEKING UNIVERSITY

感 谢 观 看