拜占庭容错算法

官永辉

2021 年 3 月 26 日

简介

内容介绍

- 区块链的共识算法
- 区块链的链性能
- 拜占庭容错问题简介
- 原始的拜占庭将军问题
- 实用拜占庭容错问题
- Honey Badger拜占庭容错
- Dumbo拜占庭容错

区块链的共识算法

共识算法

1 POW：比特币，以太坊1.0

2 POS：以太坊2.0

3 DPOS：EOS，Cardano

4 BFT算法：联盟链；Filecoin的预期共识

区块链的链性能

解决方案

针对区块链的连性能不足问题，有如下解决方案

1. 高性能异步的BFT算法
2. Layer2扩容：以太坊的Rollup，包含ZK Rollup和Optimistic Rollup算法
3. Layer1 分片：Near, 以太坊2.0；包含网络分片，交易分片，状态分片

BFT算法的目的

PBFT算法因为性能问题只用于联盟链，没有用在主流公链；但随着异步的BFT算法和区块链分片技术的应用，BFT算法在区块链拥有更广泛的空间

拜占庭容错问题简介

发展历程

1. 1982年，图灵奖获得者莱斯利·兰伯特提出拜占庭将军问题(Byzantine Generals Problem)，是计算机分布式系统关于达成一致性的问题
2. 1999年MIT科学家提出了实用拜占庭容错协议，这是一种弱同步的拜占庭容错协议，用于传统的分布式系统和区块链项目

拜占庭容错问题简介

发展历程

3 2016年，为了满足区块链性能的需求，中美研究团队首次提出基于区块链的异步拜占庭容错算法Honey Badger BFT。

4 2020年中国科学院软件研究所联合美国新泽西理工团队，在国际上提出完全实用的异步共识算法“小飞象"拜占庭容错(DumboBFT)算法，在HoneyBadgerBFT的基础上大大提升了共识算法的性能。

原始的拜占庭将军问题

拜占庭将军问题介绍

拜占庭将军问题（Byzantine failures），是由莱斯利·兰伯特提出计算机系统的点对点通信中的基本问题。考虑的系统是在存在故障或者作恶节点的情况下，是否能达到一致性的问题。

拜占庭奖金问题达到的目标

拜占庭将军问题，是由莱斯利·兰伯特提出计算机系统的点对点通信中的基本问题。考虑的系统是在存在故障或者作恶节点的情况下，是否能达到一致性的问题。包含

1. 忠诚的将军遵循相同的命令
2. 如果主将是忠诚的，那么每个忠诚的副将遵循他的命令

拜占庭将军问题

两种拜占庭协议

1. 口头的拜占庭协议
2. 签名的拜占庭协议

口头的拜占庭协议条件

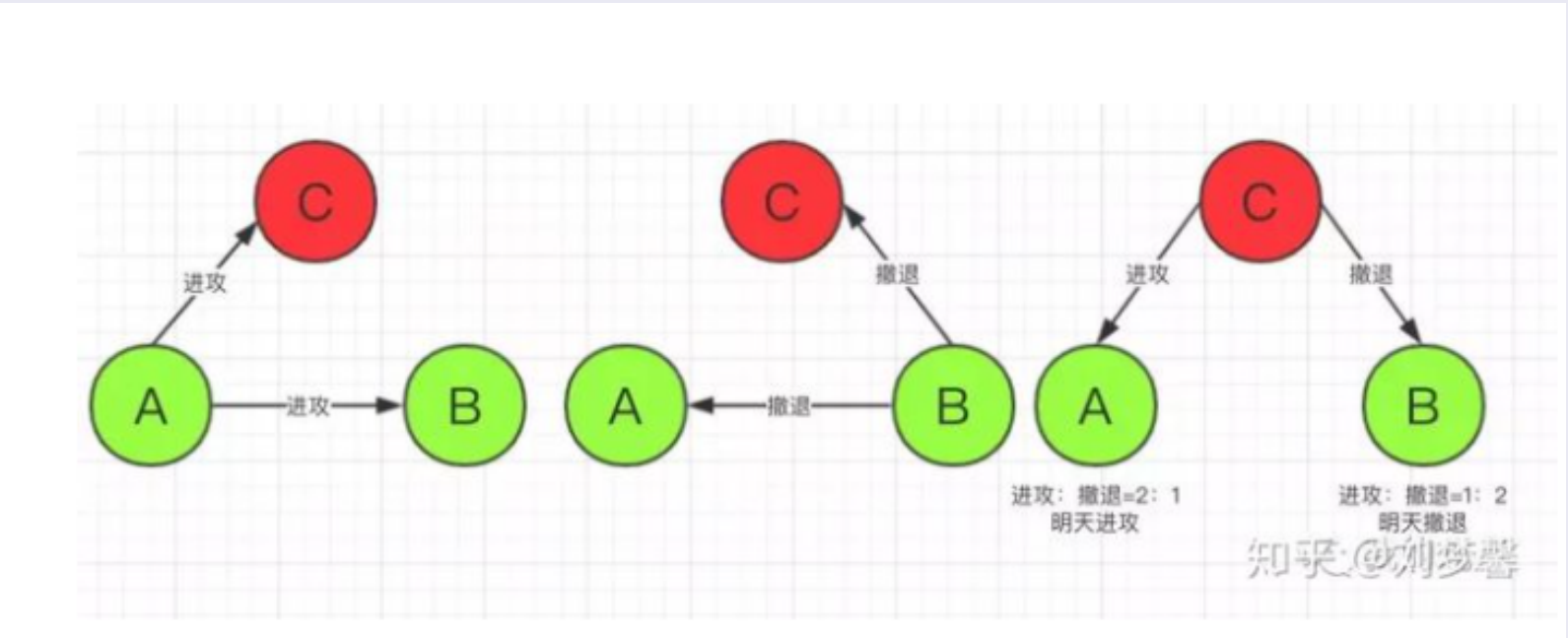
1. 每条消息直接正确传达
2. 接受者知道谁发送的消息
3. 不发送消息是可以被检测到的

口头的拜占庭协议

结论1

如果存在f个叛徒将军，那么当将军总数n小于或等于3f 时，f代表叛徒将军个数，叛徒便无法被发现，整个系统的一致性也就无法达成。

以f=1为例子，主将是叛徒：



副官是叛徒,结果类似

口头的拜占庭将军问题

算法流程

1 OM(0)

   1 主将将他的值发给副将

   2 每个副将使用从主将接受到的值，没有接收到默认退

2 OM(m)

   1 主将将他的值发给副将

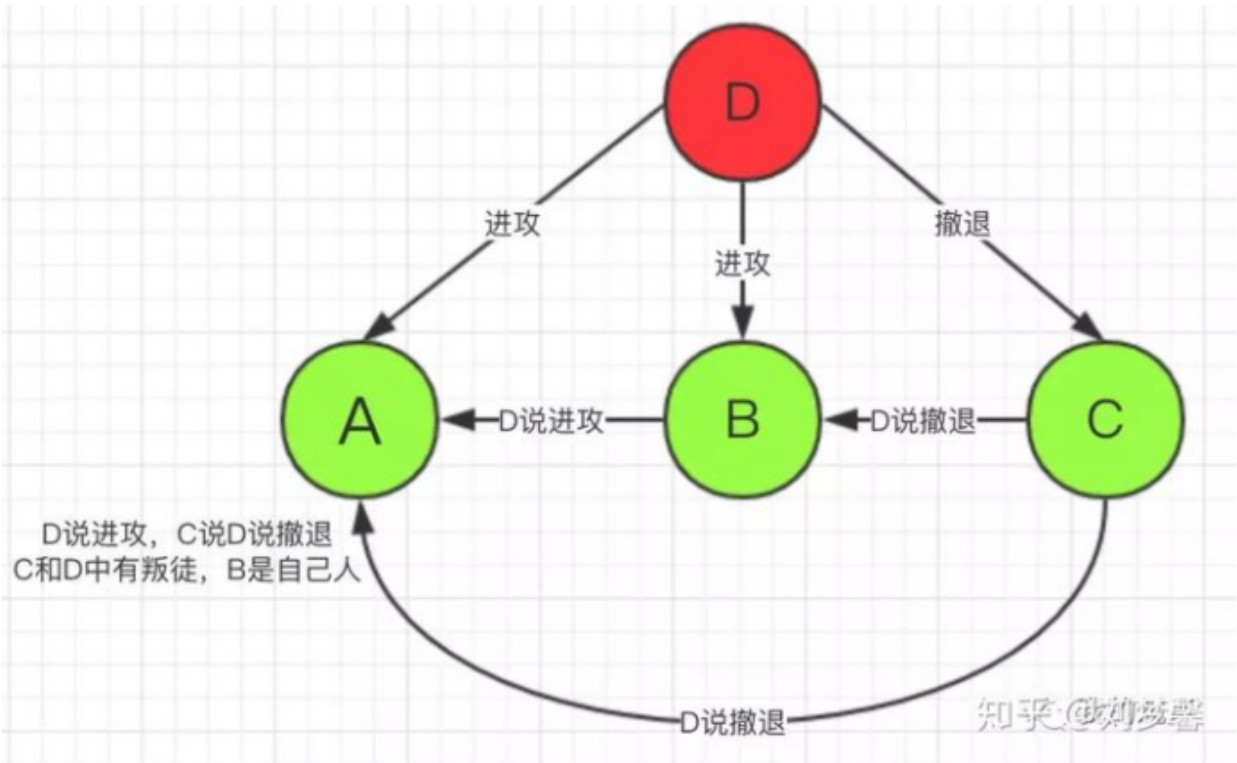
   2 每个副将i使用从主将接受到的值，没有接收到默认退；每个副将i作为主将运行OM(m-1)

   3 每个副将i,他从第j个副将收到的消息为v_j,那么他使用大多数值$(v_1, \ldots, v_{n-1})$

口头的拜占庭将军问题

结论2

如果存在f个叛徒将军，那么当将军总数n大于3f 时，经过OM(f)算法，叛徒可以被发现，整个系统的一致性可以达成。以f=1为例子，主将是叛徒



签名的拜占庭将军问题

签名的拜占庭协议条件

1. 忠诚将军的签名不能被伪造，并且其签名内容的任意变化都会被察觉
2. 每个人都能校验将军的签名

注记

对叛徒将军而言，这里没有假设，即叛徒将军的签名可以被伪造，叛徒间可以串通。

签名的拜占庭将军问题

算法流程

记V_i为第i个将军的命令集合

1. 主将签名并将值发送给每一个副将
2. 每个i
   1. 如果收到主将发来的v:0, 让$V_i = \{v\}$, 并且发送v:0:i给其它副将
   2. 如果收到$v:0:j_1:\cdots:j_k$, 并且v不在V_i里面,则把v加到V_i里面
      把$v:0:j_1:\cdots:j_k:i$发送给其他副将如果$k < m$
3. 每个i，当收不到消息时候，遵循choice(V_i)

签名的拜占庭将军问题

结论

对于签名的拜占庭将军问题，如果存在m个叛徒将军，那么经过SM(m)算法，叛徒可以被发现，整个系统的一致性可以达成。

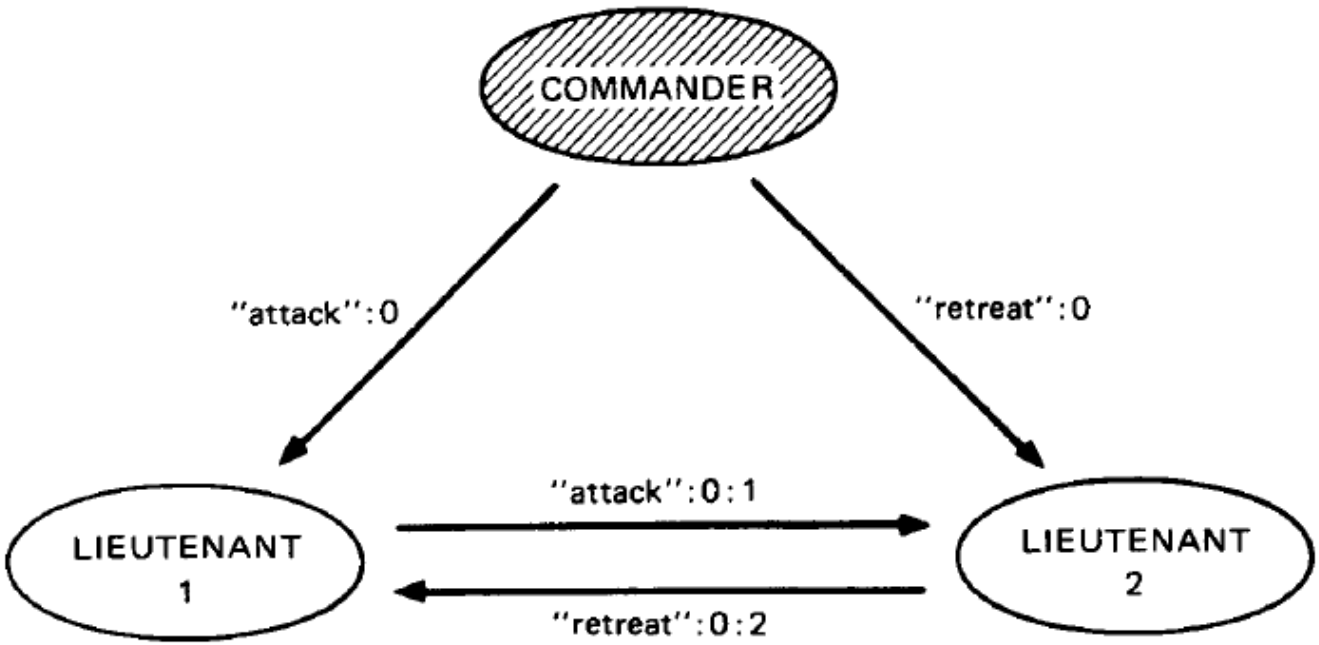


Fig. 5. Algorithm SM(1); the commander a traitor.

拜占庭将军问题

算法复杂度

SM(m)需要做m+1轮，复杂度指数依赖于m。1983年，有文章指出签名的拜占庭算法必须且只需要m+1轮，并实现了m+1轮多项式的算法。但后续文章实现了常数轮的多项式算法

拜占庭将军问题存在的问题

1. 要求网络同步同时
2. 通信复杂度太高