# 如何看待swirlds的hashgraph可以成为下一代区块链技术？

作者：maxdeath  
链接：https://www.zhihu.com/question/51437877/answer/263550477  
来源：知乎  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

谢邀。

我一年前就看过这个，只能说，不要听风就是雨，看到所谓DAG，所谓hashgraph就以为他们搞了个大新闻，多学习文化知识。

稍微说点技术方面的东西：

一笔交易有O(N^2)的消息复杂度，叫unscalable，所有的传统BFT算法都是这一类，特点是每笔交易都要通过所有节点验证，验证结果需要被广播到网络，换句话说，一笔交易要先被广播到网络一次，然后每个节点都要再广播一次，这就是N^2的来源。

一笔交易有O(N)的消息复杂度，叫scalable，很多新的BFT算法属于这一类，严格来讲，完全不牺牲任何BFT的条件或者不做任何假设的话，这个是达不到的（其实完全不做任何假设的话，严格来讲异步系统容错都是达不到的）。但是，如果允许牺牲一些BFT的条件或者引入一些假设，这是达得到的。这里请注意，POW和POS都属于这一类，POW的scalability是另一个问题，而POW和POS能达到scalable的原因，就在于引入了激励机制和理性节点的假设。简单说，如果每笔交易都要广播到全网，那么至少消息复杂度是O(N)。

一笔交易的消息复杂度是O(1)，叫scale-out，这是目前研究的前沿。能实现的方法，有off-chain，分片（sharding），DAG，总之，一笔交易不广播到全网，或者说，有的交易有的节点并不知道，至少是这样，才有可能scale-out，这是目前的研究前沿。当然，通常情况下，对于一般应用是不可能scale-out的。

然后，这个swirlds首先，最多是scalable，而scalable的区块链现在比比皆是，也就是所谓的hashgraph比传统区块链结构没有任何优势，其次，不管本身对或不对，他们的证明本身有问题。

[编辑于 2017-11-22](/question/51437877/answer/263550477)

​赞同 36​​19 条评论

​分享

​收藏​感谢

[王文士](//www.zhihu.com/people/wwens)

[王文士](//www.zhihu.com/people/wwens)

4 人赞同了该回答

认为paper中论点有漏洞，主要由两个问题

一个是网络同步的时间复杂度消耗相对比特币并没有明显优势。

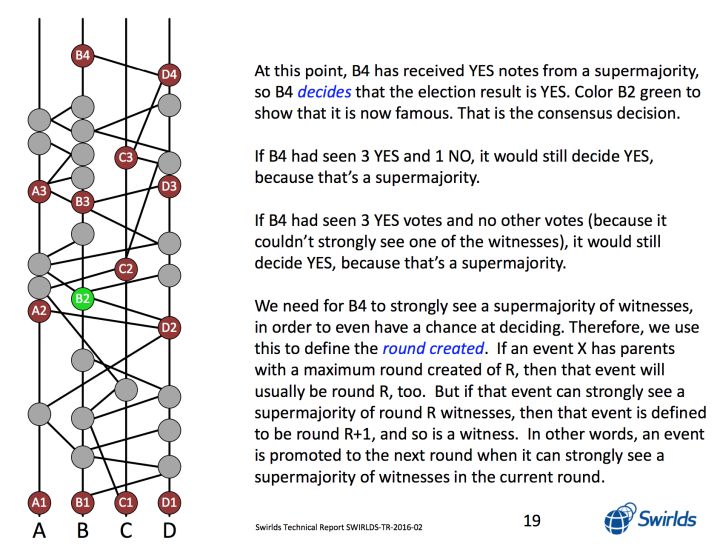
Gossip协议同步图谱最多就是次线性的复杂度，可以一次同步多个交易单元，这样共识需要的famous witness选举需要的全网大多数节点的网络同步时间复杂度是次线性时间复杂度，没有比比特币的O（N）带来质变。

二是算法总是假设诚实的节点会做强一致性的网络同步在开放的网络环境中不能成立。

Lemma 5.13. If hashgraphs A and B are consistent and both contain event x,  
then both will assign the same round created number to x.

觉得这里有问题，如果在A和B不一致的情况下如果出现图谱A和图谱B中决定的某节点的round number不一致，那么后面的算法逻辑推演就不能继续了，这样的话后面的选famous witness和共识的部分就不能保证是全网一致的。它们的网络节点不对外开放，总是假设consistent，或者说白皮书回避了这个问题，只讨论在假设成立的情况下算法的优势，放在公链环境下不能成立，so big problem。

举个栗子

<img src="https://pic4.zhimg.com/v2-09509c4a916c7ec361117336c4329437\_b.jpg" data-caption="" data-size="normal" data-rawwidth="1692" data-rawheight="1302" class="origin\_image zh-lightbox-thumb" width="1692" data-original="https://pic4.zhimg.com/v2-09509c4a916c7ec361117336c4329437\_r.jpg">

在决定round number的时候，正常情况在D节点的图谱中有了D2 Event，那么因为第一个Strong See，所以该新启一层Round了，那么如果这个时候D2节点发生网络故障，长时间不能把图谱传给其他节点，其他节点就因为别的Event产生Strong See会选出新的Round，当出现不同节点的Round number冲突的时候，算法逻辑就推演不下去了。

Swirlds这个算法有问题也有优势，优势有以下：

1、算法有个假设，诚实的节点会做强一致性的网络同步，这个假设我们认为能成立，在这种情况下，做共识需要网络同步是次线性复杂度，比比特币要好点

2、节点做共识的famous witness选举的时候需要的时间应该也比挖矿暴力破解哈希值要低的多

3、新交易立刻加入网络，有利于削平波峰波谷，从而提高系统的整体吞吐能力

4、DAG网络交易量越大，认为诚实的节点能更快达成共识则交易确认时间越短。

这些是在完全去中心化的环境中相比比特币的优势，没有质变，不过还是带来了DAG应用到区块链上去中心化共识的落地，至少达到了比特币的安全和性能，iota，byteball的算法能做到很高的并发是因为它们采用中心化的方式，不做全网的一致性共识，到后期网络大量交易会带来性能的提升

问题有以下：

1、算法的假设诚实的节点会做强一致性的网络同步，在恶意节点中不成立的情况下，会带来达不成一致共识的情况，这种问题总会发生。恶意节点导致不一致的问题就是拜占庭容错问题，其实DAG在这个事情上没有看到相比的突破，又回到了POW、POS、DPOS上。一个可行的办法是结合DPOS选超级节点，把网络的共识分成两种状态，临时一致和最钟一致，在达不成不一致的时候由委托的超级节点来决定最终一致的状态然后同步全网，算法继续。这个故事也比较好讲，DPOS+BFT。

2、既然网络同步相比比特币的O（N）复杂度没有明显优势，思路就成了各个工程上的环节性能的优化，或者中心化思路优化更依赖超级节点。

无耻的更新线

又过了一边paper，发现对我上面说的问题二关于图谱同步和强一致性的理解有问题，其实在大多数节点做强一致性图谱同步的情况下可能会发生分叉的情况，但是并不会影响算法的正常工作。

先来两个定义，我们细看一下这个问题。

一致性：如果HG<A>和HG<B>是一致的，意味着如果都包含Event<x>那么将同时包含E<x>的Ancestor并且相同的Ancestor的Edge.

分叉：Event<x, y>被视为一个分叉，当E<x>、E<y>有同样的Creator，但是彼此不是self-ancestor关系

其实并不是说两个图谱HG<A>、HG<B>出现不一样的图谱结构就是不一致，只要满足上面的定义，父子引用关系不冲突就是一致的。诚实节点的图谱同步逻辑是：

- 当Alice向Carol发起一次Gossip时，Alice告诉Carol所有A知道&C不知道的Event

- Alice不应该给Carol发送Event<x>，除非Carol包含Event<x>的所有Ancestor(整个算法正确运行保证)

PROOF:

(1) 如果所有的节点都诚实那么，所有成员的图谱一致

而且只有在恶意节点故意向不同的节点发送有冲突的图谱同步才会导致分叉的情况出现，即便是在这种情况下，只要恶意节点不超过1/3，仍然不会影响算法核心的Round决策、Witness和Famous Witness的选举，算法仍然是成立的。paper中的证明如下：

假设:

(1) 全网大多数节点2/3为诚实节点

PROOF:

(2) 如果一对Event<x, y>是一个分叉，并且E<x>被HG<Alice>中的E<z> StrongSee，那么E<y>不会被任何HG<Bob>中与HG<Alice>相一致的Event StrongSee.

(3) 如果HG<Alice>和HG<Bob>是一致的并且都包含Event<x>, 那么会对E<x>确认相同的Round Number.

证明(1)、(2)、(3)表明了只要恶意分叉节点不超过全网的1/3，那么算法仍可以正常运作。

[编辑于 2018-05-25](/question/51437877/answer/376510442)

​赞同 4​​17 条评论

​分享

​收藏​感谢收起​

[gkzhong](//www.zhihu.com/people/gkzhong)

[gkzhong](//www.zhihu.com/people/gkzhong)

3 人赞同了该回答

属于胡扯。

DAG在计算机工程上的应用应该表达过程，用于表达支付这种结果状态强一致性要求的就是死路一条，必然花费非常高昂的代价去解决冲突问题（对于币而言就是双花的问题），在计算机技术里有一个非常大规模表达结果状态的应用就是版本管理git，看看冲突是怎么解决的。又回到中心化去了。

Hashgraph就更扯了，这根本不可能实现，这要能实现，全球部署Cassandra，写数据加上签名公钥。写2/3，读2/3，容错性比它高多了。但一样在拜占庭环境中无效。

[发布于 2018-02-23](/question/51437877/answer/326079294)