# 如何让公链快起来<下>

通过上一篇文章「区块链公链如何才能快起来 」，希望读者可以从文中解释的那些原因出发，明白这样一个结论：安全的共识机制本质上并不跟系统的吞吐性能以及容量相矛盾 。在区块链的系统里面，所谓的「不可能三角」——安全、去中心化、性能，这三者只能取其二的说法，并没有技术逻辑的论证，而只是对现有解决方案的观察。

作者：王嘉平

我的上一篇文章探究了单链公链系统在性能和容量上受限的本质原因。本来我以为文章过于「技术」，会遭到普通读者的嫌弃。完全出乎我意料的是，竟然有上万的读者通过各种渠道阅读了这篇文章，并且，我收到了不少朋友和读者的反馈。他们鼓励我继续写下去，把我自己在分布式系统研究和区块链投资领域的经验和心得分享出来。

我非常感谢这些朋友，还有每一位读者。但是我还是非常担心，我的文章有太多技术性语言和技术上的分析——比如今天这一篇。不过，我会努力用最简单的词句、最直白的语言，把区块链领域看似很深奥的理论和方案，用浅显易懂的方式介绍给大家。

通过上一篇文章「区块链公链如何才能快起来 」，希望读者可以从文中解释的那些原因出发，明白这样一个结论：安全的共识机制本质上并不跟系统的吞吐性能以及容量相矛盾 。在区块链的系统里面，所谓的「不可能三角」——安全、去中心化、性能，这三者只能取其二的说法，并没有技术逻辑的论证，而只是对现有解决方案的观察。

不过，对于现有的单链系统，确实有一个不可逾越的物理限制，那就是全节点平均带宽。以13Mbps的带宽为例(这是互联网带宽中位数, 4K电影在线播放带宽下限要求为15Mbps)，无论基于什么共识技术，即使完全丧失去中心化，并忽略传送协议，以及层层封包等额外引入的代价，以比特币网络现在的每秒处理 7 笔交易的吞吐量来算，其吞吐量理论上限为6825TPS（7TPS × 13Mbps / 8Mb × 600s）。否则就会出现部分全节点脱网，即长时间无法赶上全网的区块增长，而离最新头部块越来越远。

我的上一篇已经提到，性能瓶颈和容量瓶颈在现在的结构下无法突破，因为单个全节点的带宽和存储资源总是有限的，并且我们还不能要求部署全节点的机器都具备顶级的配置，因为只有保证了参与全网的壁垒较低，才能真正具备有良好的去中心化特性。从这个意义上来说，要获得大幅度的伸缩性，我们必须要能够有一个合理的设计，使的单个全节点仅仅负责整个网络吞吐量和容量的一部分，这样，才有可能在维持全节点的负荷比较小的前提下，使得全网的性能和容量有大幅提升。

分片(Sharding)技术就是针对这一思路提出的。分片最初是基于数据库系统提出的，泛指横向切分数据库服务器的负荷，用多个数据库服务器平行服务的方式，以提升整体的服务吞吐量和数据容量。实际上，分片并不单指一种特定的技术，而是一大类横向切分系统负荷，多实例并行，以提高整体性能和容量的技术。

不过，分片这个提法在区块链技术中过于宽泛。在这篇文章的最后部分，我会通过引入「异步共识组」这个概念，谈谈我认为什么会是区块链中理想的分片方案。

在讨论我认为的理想方案之前，我还是用我习惯的逻辑，先向读者解释为什么我们需要分片技术、理想的分片技术究竟应该解决什么问题这些最基本的知识；然后，我会根据自己看过的大量的分片方案，谈谈目前市面上已有的分片方案存在的五个误区或谎言。

**为什么需要分片：从全节点的负担谈起**

在分析具体技术方案之前，先来看看一个全节点都有哪些负荷:

网络带宽，这个部分同吞吐量(TPS)线性相关

**1. 新区块的广播**

全节点需要下载周期性出现的新区块，并进一步上载给其他需要的节点。对于现在7TPS的比特币区块链来说，平均下载带宽消耗是13.6Kbps，完全不是负担。但是如果TPS提高到7000TPS，则至少需要13.6Mbps的下载带宽才能承受——请注意，这个是关键数据，并且有序，如果区块数据得不到及时传输，区块链将无法持续构造，并且无法重建全网的账簿状态。

**2. 未确认新交易的广播**

全节点需要持续发现，并下载尚未被确认的新交易，然后上载给其他节点。在全网不拥堵的情况下，这个数据量和新区块的数据量相当。相对的，这个数据不需要保序，即使下载不全也不会直接导致全节点本身工作异常。但是，如果这种情况大规模地发生，最终会导致部分交易始终没有被矿工看到，而长时间不被确认，最终超时。

内存容量，这个部分同用户量(address数量)以及智能合约数量线性相关

账簿的存储占据主要的内存消耗，其他部分的消耗不多，并且基本不会随着网络扩展而增加。

区块链的基本账簿, 至少需要维护每个用户(每个address)上的余额，每个用户至少需要16 + 32个字节。对于支持智能合约的区块链，每个用户至少需要额外32个字节记录每个智能合约的代币的余额。智能合约自身还可以自定义更多的字段，可以占用远超32字节的内容。

当然，并不是所有的用户都参与了所有的智能合约，所以我们并不以用户量和智能合约数量的乘积作为内存消耗的估量。现实的情况是，我们按照以太坊的现状来估计，约 4500 万个地址，至少占据了约3GB的内存。每个地址平均消耗了68个字节。这里并不是说以太坊有了4500 万个用户，通常而言，一个用户会拥有好多个地址。

**磁盘 I/O，这个部分同吞吐量(TPS)线性相关**

全节点收到区块并确认成链之后，区块会被写入磁盘归档起来。这个部分的吞吐量大体和下载新区块的带宽一致。

这里需要注意的是，按照以太坊的观点(账户模型)，账簿是指全网的某一时刻的全量状态，而区块(若干交易的集合)是对账簿的增量修改信息。在磁盘实际存储的是区块，而不是账簿。节点在启动的时候，需要重放所有已经记录的区块，以构造账簿的最新状态。截至目前，以太坊的全部归档区块数据量已经超过160GB。这里的一个优化是定期将账簿全量快照也记录下来(即checkpoint)，这样只需要从最近的checkpoint开始重放就可以了。

而UTXO模型是比特币的观点，这是一个非常有意思的结构，它的区块既是若干交易的集合，同时也是部分账簿的状态，即特定地址上的余额。整条链上余额不为0的UTXO都是账簿的最新状态。至今比特币区块链的全部归档区块数据量已经超过220GB。不过从磁盘I/O的角度来看，重建全网账簿还是需要扫描整条链找出全部余额不为0的UTXO，并没有什么优势。当然checkpoint技术对UTXO一样有效。

在这个领域，最近我看到了一些被热炒的改进，尤其是在已确认交易集合的聚合表示方面。我注意到斯坦福大学的一个团队最近提出，RSA累加器(RSA Accumulator)可以代替Merkle Tree，用来表示一个已确认交易的集合。这个想法的优异之处在于，可以直接用一个交易的Hash值来校验一个交易是否存在于已确认交易的集合，而Merkle Tree不仅需要一个交易的Hash，还需要从树根到该交易沿途路径上的所有兄弟节点上的Hash。不过，这个做法的代价是，这个表示本身要1.5K字节，Merkle Tree的根仅需要20或32个字节。同时，这并不意味着全节点可以不再保存历史交易，原因在于，首先集合的聚合表示是不具备枚举的能力的，也就是说，在交易未知的情况下，集合的聚合表示无法告诉你里面具体的交易是什么，所以你也无法判读这个集合本身是不是正确的，所以全节点也无法仅凭这个信息确认链头的正确性。全节点在自举的时候还是需要从创始区块(genesis block)开始下载，逐块验证全链的完整性和正确性。但是，当这个验证过程完成之后，为了节省空间全节点，是可以删除掉区块中的实际交易内容的，其代价是丧失枚举交易的能力，同时也丧失了帮助其他全节点自举的能力。

另外，通过验证一个交易是否在链头上，来确定账户余额的做法，仅对于UTXO交易模型有效。对于以太网这样每个交易仅为增量修改信息的交易模型，仅靠验证一个交易是否在链头上，是无法得知账户的余额的。

**CPU处理能力，这个部分同吞吐量(TPS)线性相关**

**1. 密码学相关计算**

对于每一个收到的新区块，无论最终时候成功成为链的一部分，全节点都需要验证其携带的每一个交易是否被正确签名。同时会验算每个区块的Hash，以确保其达到了当前工作量证明（PoW）难度的要求。对于拜占庭容错（BFT）算法，还需要验证各个委员会成员的签名是否正确。这里主要的工作量是验证签名。验证签名在一台普通计算机上(E5-1620@3.5GHz, C++并行实现)的速度可达 40K op/s，也就是4万TPS(每个交易需要验签一次)。

**2. 交易验证相关计算**

对于每个交易，如果是支付交易，将需要至少查找一次账户的索引结构(通常为哈希表)，并做一次加法和大于的判断；为了更新账簿，还需要至少一次加法和减法。这样的一系列操作，在一台普通计算机上的速度可以轻松达到1M op/s。如果是智能合约，会需要在虚拟机中执行对应的智能合约指令，对于现有的金融本质的应用(包括类似Cryptokittes类或Fomo3D类的区块链游戏)，至少可以达到100K op/s。

**这意味着什么？**

针对上面的分析，我来做一个小结：对于一台普通的计算机，拥有13Mbps的互联网连接，E5-1620@3.5GHz 4Core的CPU，16G内存，512G SSD 硬盘 (250MB/s)。网络带宽导致的TPS理论上限约为7千TPS，硬盘文件I/O导致的TPS理论上限为5万TPS，CPU处理能力导致的的TPS理论上限约为5万TPS。另外，内存大小导致的地址理论上限约为 2.5 亿个地址。

这里可以看到，对于性能来说，网络带宽是最首要的瓶颈根源，其次是硬盘的I/O和CPU的处理能力。对于容量来说，内存是最主要的瓶颈根源。

**分片应该做到什么？**

从上面对瓶颈根源进行分析之后，我们可以得到这样的结论：特定的分片方案至少要切分系统的网络带宽、内存容量相关的工作量，才有可能大幅提高全网的伸缩性，才有可能打破所谓的「不可能三角」。必须再次强调，这个结论非常重要。因为只有真正了解这一点，才能进一步讨论分片方案是否真的可以打破「不可能三角」。

虽然在前面的分析中，带宽和内存是短板，但实际上，其他部分的约束也并不富余。理想的情况下，分片系统能够切分上述四个方面所有的负担。对于一个有n个分片的系统，其每一个分片仅需要承载1/n的全网负荷，将网络带宽、硬盘文件I/O、CPU处理和内存容量的消耗都降到原来的1/n。在分布式系统中，这个是高伸缩性能带来的性能和容量提升的上界。一个分片系统如果可以实现这个，即性能和容量可能随着分片个数n而线性提升，那么理论上，可以使得全网的性能和容量能够被无限地提升。

而现实情况是，分片系统为了实现应用逻辑的正确性，协调各个分片之间的运作以及实现整体的安全性，需要每个全节点引入额外的工作量(overhead)。这些额外的工作量是个常数O(1)，即和分片个数n无关，那么全网线性提升仍旧可以保证。

如果额外的工作量的阶(order)小于线性，例如O(log n)，那么就意味着随着n的增加，全网提升比额外工作量的增加要来得快，全网线性提升还是可以实现的。但是，如果额外的工作量的阶大于或等于线性，例如O(n log n) 或 O(n^2)，那么基本上全网提升到一定程度，就无法继续了。因为全节点额外引入的工作量成为了新的首要瓶颈。

**跨分片交易的原子性保证**

区块链中的每一个交易都是原子的，必须保证其涉及到的操作或者全部完成，或者一个都不开始，才能使得最终状态是正确的。

对于涉及到多个分片的交易，我们不得不协调发生在各个分片中的操作，保证它们都能被没有错误地完成，并且不受其他交易的干扰。这里看起来可以简单借用在多线程编程中常用的线程同步概念，用诸如临界区(critical section)等方式，锁住涉及到的状态，阻止其他交易触碰这些状态，在完成交易所有操作后释放。但是，这会导致部分分片的执行被阻塞，而导致实际性能大打折扣。并且随着全网规模扩大，分片数量增多，跨分片交易的比例必然会极剧上升，从而使得加锁变得非常频繁。

一个设计良好的分片系统，必须具备高效的跨分片交易处理算法，并且算法的开销应该和分片数量n无关。

**单个分片的安全性保证**

分片系统中，如果每个分片有独立工作的共识系统，也就是每个分片自己一条链，全网能最大获得的吞吐量的n倍提升。

共识系统的安全性依赖于出块权力的大多数(majority)是可信的，这个大多数在PoW系统中为50%，在BFT系统中为2/3，甚至更高。而当全网有n个独立工作的共识系统时，这时平均分到每一个分片的大多数变会降低到原来的 1/n。这样攻击这些分片中的共识系统的壁垒，对于PoW系统会降低到全网的 50/n % , 对于BFT系统则会降低到1/(3n)。从而导致，仅需要非常低的成本，就可以攻击某一个特定的分片，例如构造双花攻击(double spend)。而在一个分片系统中，只要有一个分片被攻击，其后果会藉由跨片交易迅速污染到其他的分片。

这就是为什么每一个分片系统都必须处理好安全问题，使得攻击特定分片的代价同攻击全网一样高。

**分片方案的一些误区**

**误区一：分片的颗粒度不够细**

分片的切分依据需要有足够的颗粒度，这样才能使得分片个数n取到足够大的值，并且不容易产生单点过热(hotspot)。例如按照交易参与方的地址分，或者按照交易本身的hash值来分。这些是可行的方案，因为地址和交易的数量本身足够的多。有些分片的方案按照业务来切分的，这是不可取的，颗粒度不够细，并且如果单个业务活跃度很高，需要更高的性能和容量，却无法获得分片系统所带来的提升。

在全网分片结构不发生改变的前提下(例如 给定分片数量n)，分片的切分依据应该是确定的，和账簿状态无关的，并且是可以被简单计算的。对于提交交易的轻量节点，可以根据交易内容唯一地确定这个交易应该被发送到对应的某个分片。早期看到动态的调整地址归属，试图将经常交互的地址划分在同一个分片，从来减少跨分片交易发生的概率。这种做法是不切实际的，因为本质上同分片数量相矛盾。就以太坊ERC20的历史支付交易来看，在4个分片的时候，跨分片交易比例为75%，在32个分片的时候，跨分片交易比例为97% (按支付发起者的地址 , 平均随机划片)。

**误区二：鼓吹智能合约的计算复杂性**

从前面的分析可以看到，CPU处理并不是短板。但是现实中有若干方案，强调智能合约非常复杂，不仅GPU不够用，甚至还需要GPU乃至集群来计算。

在分片设计中，每个分片依旧承载全网全部的吞吐和容量，只是将交易的验证以及状态更新分开在各个分片做，并且之后引入一个O(n^2)的通讯量，将部分更新过的状态传播到其他分片。而事实上，即使是现实世界中的业务，各种交易、清算其实际计算一点都不复杂，就是一些加减乘除，而更多的工作量是在安全验证以及通讯上面。

当然我们可以假想一种奇特的应用，需要巨大的计算量来完成逻辑层面的交易验证和状态更新。这会使得CPU处理最先成为瓶颈，并且使得GPU加速变得可能有意义。不过我会很怀疑这样的应用(或者应用的这个部分)是不是区块链的真实需求，是不是应该由区块链来承载。

**误区三：忽略容量瓶颈**

近期我看到一些分片方案，认识到了性能瓶颈的本质是带宽的约束，并从这一点切入划分工作量，让每个分片负责全网的一部分交易，包括其相关的计算。

不过非常遗憾，我看到的这些方案没有切分和用户容量相关的负荷，每个分片仍旧需要维护全网所有用户所有DApps的状态。并且，由于所有用户状态需要每个分片都维护，导致某用户在特定分片中的更新必须广播到其他的分片，这将额外引入一个O(n)的通讯量。由于用户容量相关的负荷没有被切分，导致系统无法承载更多的用户和DApp在链上发生交互，从而使得性能的提升无法被充分利用，约束上层应用的发展。

**误区四：忽略安全问题(被稀释的可信大多数)**

前面的分析已经提到，在分片之后可信大多数被稀释，攻击成本将极大下降。但是，到目前为止，我还没有看到有可靠的方案，能切实解决这个问题。

可能会有团队说，攻击成本即使降个几倍，也是很难攻击的，攻击要花很多资金，没人会真的去攻击。我不认同这种看法。

我的观点是，即使在不分片系统中，直接的暴力算力攻击已经屡有发生，更别说攻击成本被降低的分片系统了。如果我们不处理这个问题，只是祈祷没有人来攻击，那么在这样的系统设计之下，性能和安全就构成了直接的矛盾，越高的性能，即越多的分片，就会导致更低的攻击成本。这其中安全风险巨大。

**误区五：双层设计**

有不少分片方案，在各个独立的分片之外，引入了一个根链(有些叫主链或母链)，由其完成各个分片之间的沟通和协调，或者由其来保障各个分片的安全。

对于这类方案，需要比较细致的个例分析，不排除有可行的方案出现。但是总体上，当跨分片交易的比例很高的时候，根链有极大的可能成为系统的瓶颈。而利用根链作为安全保障的系统，其实本质上很可能就是一个单链系统。

在我看来，优异的分片设计方案，不应有分层的结构，而是各个分片应该是同质的，在功能上完全一致，地位上也完全平等。

**异步共识组**

可以说，以伸缩性为目标的公链研发项目，其方案中如果没有分片的设计，都是不切实际的，在未来的实际应用中必将面对性能瓶颈和容量瓶颈，至少会遭遇这两个瓶颈中的一个。

由于分片这个术语本身太过于宽泛，我想在下面引入异步共识组(Asynchronized Consensus Zones)这个概念来，大家讨论一下理想的分片设计究竟应该是什么样的。

我先来解释一下，什么是异步共识组。

异步共识组由多个同质的、功能上完全一致、地位上也完全平等，并逻辑上尽量隔离的独立共识系统的实例所构成，他们并行工作，分摊全网的吞吐压力，分摊全网状态的维护工作。

0. 具备独立的相对稳定的节点集合，逻辑上不要求一个节点参与到多个共识组

1. 具备独立的账簿，承载全网的一部分用户(组内用户)。各个共识组的组内用户没有交集

2. 具备独立的Chain of Blocks，仅记录已经确认的和组内用户相关的交易

3. 具备独立的非阻塞的出块过程，各个组之间没有任何同步的需要 (如需要互斥锁定特定资源)

4. 具备独立的未确认交易集合，仅有和组内用户相关的未确认交易会被暂存

5. 具备独立的出块候选或竞争机制，矿工仅限于组内竞争，和其他组的矿工无直接竞争关系

6. 具备独立的Gossip网络，完成区块和未确认交易的广播，不波及其他共识组的节点

异步共识组中每一个共识组采用具体的共识算法可以是PoW/PoS，也可以是BFT类算法，或这些算法的改进。异步共识组的设计是如何让他们完全独立地并行工作，由于各个共识组在IT资源的消耗上是互相独立的，那么全网的吞吐性能和账簿容量将随着全网共识组的数量增加而线性增加，而参与其中的全节点的工作压力依旧是单个共识组的负荷。

异步共识组是共识算法中立的，并不限定具体的在各个分片中采用的共识算法。有很多团队在研究和改进共识算法，无论共识算法被改进到什么程度，任何已知的共识算法或者是今后的改进版本，都可以被应用在异步共识组这个技术架构之中，从而得几个数量级的性能和容量的提升。

假设我们用比特币的PoW算法，用同样的配置参数(1MB的块，10 分钟的出块间隔)作为共识组的共识算法，对于一个具有1024个共识组的网络，全网将获得7000多TPS的吞吐性能。分片的全节点按照一台普通的计算机来算，在13Mbps的互联网连接、E5-1620@3.5GHz的CPU、16G内存、SSD 硬盘 (250MB/s)的情况下，全网将具备等效的13Gbps带宽、16TB内存、4096个Core的计算能力，以及250GB/s的I/O能力，用来承载所有用户的账簿和DApp的状态，以及其中涉及的计算。而对于每个共识组里面的每个节点，它们所负担的通讯、计算和存储的压力，仅同运行一个比特币系统的全节点的负荷相当。无论全网扩张到多大的程度，网络中的单个参与者始终不会有明显増大的负担。

**异步共识组的挑战有哪些？**

异步共识组是一个理想方式的模型，不仅概念简单，而且可以完美突破全网的性能瓶颈和容量瓶颈，实现高伸缩性的区块链系统，并且对去中心化程度没有任何的牺牲。但是，这个模型要成为切实可行的方案，正如上文提到的，需要让这样的一个系统正确工作并保障安全性。这里需要两个方面的设计巧思：

第一， 如何正确高效地完成跨共识组的交易；

第二， 如何保障每个共识组的安全性。

只有，解决了这两个挑战，才能使得异步共识组系统能够正确安全地工作，释放其强大的性能和容量，打破所谓的「不可能三角」，服务于大体量的区块链应用。

具体这些挑战如何解决，并且如何最小化所引入的额外开销？我会留点时间，在未来的文章中和大家继续探讨 :)。欢迎大家通过我的微信公众号「王嘉平」和知乎专栏「去中心化数字世界随想」，就这个话题展开更多讨论。