以太坊的路线图应该改变吗?

一个 PoS 的幽灵，在以太坊的上空回荡。 —— 题记

自《A rollup-centric ethereum roadmap》（中文译本）一文发表以来，整个社区都对以太坊（尤其是以太坊 2.0）的路线图产生了疑问。

2020 年 11 月 18 日，在以太坊基金会的 Eth2.0 研究团队的第五次 AMA 活动中，Vitalik 明白地表示，路线图已经发生了变化：（1）暂时不再强调 Phase2 的重要性，Phase 1 致力于实现数据分片（shard data），供 rollup 方式使用；（2）信标链将具备执行功能，即 Eth1-Eth2 合并之后，信标链区块将直接包含交易；（3）Phase 0 实现后的三大工作：轻客户端支持、数据分片、合并，将并行推进，任一模块只要准备好了就退出。

本文的目的不是为原来的三阶段路线图辩护。相反，本文是想主张，三阶段路线图虚无缥缈，新路线图食之无味，没有一种与 Eth2.0 相关的路线图值得以太坊放弃当前的运作模式、转向以 PoS 为基础的系统。

在这里，我会先讲解初始的三阶段路线图的论证思路及其技术难点；然后分析新路线图的可扩展性。最后论证，新路线图的可扩展性优势，已经渺小到不足以使以太坊冒险转入 PoS。

Eth2.0 的三阶段路线图

在过去两年，广为流传的 Eth2.0 路线图规划了三个依次序实现的组件：

- Phase 0：以 PoS 为共识机制的信标链
- Phase 1：多条分片链
- Phase 2：为所有分片增加执行功能

从这一路线图可以清晰地看出，原来的以太坊 2.0 的目标是打造一个 “分片化执行（sharded execution）” 的系统，意思是：每个分片都有自己的状态，这些状态按各分片的状态转换规则来变更；变更后的状态由信标链来敲定；由此，以太坊 2.0 就成了一个多个分片可以并行处理交易的系统。这也意味着，以太坊 2.0 是一个 “共识” 和 “交易处理（验证）” 解耦的系统，被分配到各分片上的验证者负责验证交易和状态的正确性；但这些状态的敲定则依赖于信标链的 epoch 敲定机制，两个过程并不是完全同步的。

这种 “PoS 信标链 + 多分片” 的架构，似乎也非常好地利用了 PoS 算法本身的特点：为了解决 Nothing-at-stake 问题（PoS 出块不需要付出计算量，因此有资格出块的账户会尝试在不同的分叉上同时出块，使系统分崩离析），以太坊 2.0 所用的 Casper 算法要求用户先存入一部分押金才能获得出块资格，而如果验证者滥用了出块资格（比如同时支持两个分叉），则会被罚没押金；由此，像 Casper 这样的算法实际上在区块链上创造了两种可以相互沟通、但相互独立变更的状态：一种是普通用户的状态，另一种是验证者的出块权重状态；共识过程以出块权重状态为基础，达成共识也会更改出块权重状态；因此，共识过程先天独立于用户交易的验证，可以解耦；对任意的交易批次及结果状态而言，共识过程可以被抽象成一种 “终局性敲定机制”，逻辑上，多分片并行执行于是成为可能。

至于其可扩展性，以太坊分片技术的命名 “二次方分片（Quadratic sharding）” 透露了端倪：假设分片上的交易，其执行复杂性能够被化约到与区块头验证同样的难度，则分片化执行的架构，可以使整个系统的处理能力，随着参与节点处理能力的线性提高而呈平方级提高。通俗来说，如果参与网络的节点（平均而言）在一段时间内能验证 4 个区块头，这就意味着，在参与一个分片时，节点们可以在同等时间内验证 4 笔交易，此时系统总处理量是 4 条分片 × 4 笔交易/分片 = 16 笔交易；如果节点的处理能力变成了 8（2 倍），则处理量会变成 64 笔交易（4 倍）。

听起来很美好，但是，这个“平方级扩展” 的论证中包含了如下假设：

（1）存在一种技术，使得分片交易的验证，可以简化到与验证区块头同样的难度；

（2）不存在跨分片的交易，即各分片内的交易是完全不会相互依赖的。跨分片的交易需要占用多个分片的处理容量，也要占据信标链的处理容量，会使可扩展性大打折扣。

关于（1），这个假设是有可能得到满足的，无状态性（statelessness）就是这样的一种技术，它的思路是，在传播交易（或者传播区块时），附带交易所访问状态的证明（witness），使得交易的验证者无需持有交易执行之时的状态数据，就能验证交易的有效性。这一点极为关键，如果没有无状态性，参与分片验证的验证者就必须保存分片的状态，因为验证者会被不断分配到不同的分片链上，那就意味着他们必须保存所有分片的状态，在实践中也就意味着他们要不断下载所有分片的区块并处理交易，从而使整个系统坍缩为一个大区块系统（例：投入能处理 16 笔交易的资源，处理 16 笔交易）。遗憾的是，至今，以太坊 1.0 也没有研究出足够轻量的无状态方法。

关于（2），那就没有什么好说的了。如果不能实现跨分片交易，分片化执行的系统就没什么意义，因为各分片各自为政。必须使得 ETH 有办法存在于各个分片上，这个系统才能仍然以 ETH 为主体。而直到今天为止，还没有出现一种跨分片交易方案，能够不增加信标链的处理量。道理也很简单，对于任意 A 分片来说，因为并行处理，任意 B 分片上正在发生什么交易，需不需要改写本分片的状态，是不可知的，因此必须存在一个通信层，可信地证明 B 分片上发生了一笔试图改写 A 分片状态的交易。而一旦需要让信标链具备处理交易的功能，平方级扩展的效果就会被打破。（顺带说一句，满足了这一可信通信层的需要的链，就变成了事实上的 Layer-1，而其它分片则变成了事实上的 Layer-2，像极了 “Layer-1 + Layer-2”。）

除了存疑的可扩展性，分片化执行还带来了许多经济上的有趣问题。例如，如果跨分片交易的处理时间超过一笔分片内交易的处理时间（这是必然的），这就意味着，不同分片上的 ETH 价值也不会相同。就好像美国国内的 1 美元，与美国国外的 1 美元，实际上并不是同一种东西。不论有多少个分片，都至少会有两种 ETH 价格，一种，是那个金融应用最繁茂的分片（也就是 Eth1 分片）上的 ETH 的价格；另一种是其它分片上的 ETH 的价格；后者必须支付一定的手续费并付出一定的时间，才能换成前者，因此对前者必定有一些折价。同理，即使每个分片上都有 uniswap，不同分片上市场的交易滑点也必定不相同，最终大家都会汇集到一个分片上，因为大家都在一起的时候，流动性最充沛，资金效率最高。某种程度上，可以认为跨分片交易的需要是很少的 —— 但这也意味着，其它分片上闲置的交易处理容量，也根本没有意义。

分片化执行系统的技术难点，此处不再赘述，感兴趣者可以自己想想分片化执行系统怎么支付手续费的问题。但我在这里想说的是，分片化执行系统的设计理念违背了大家的实际需要，也违背了事物的发展规律。全局状态（可组合性），并不是一个问题，而正是大家需要的东西；正是因为以太坊使得所有金融应用都能瞬间组合，创造了一个价值可以零摩擦流通的空间，以太坊才有了变革世界的潜力；在协议层为价值流通创造摩擦，是自废武功。有了一个良好的基础层时候，应该想办法维护这个基础层，剩下的事情让用户自己选择，让生态自己演化 —— 不要以为设计能设计出一个生态，过度设计只是给所有人强加成本。

分片化执行（Phase 2）的搁置，侧面印证了其中的难度 —— 在可预见的未来，这条道路无法产生令我们满意的成果。尽管如此，我并不认为 Eth2.0 的研究员们已经完全放弃了三阶段路线图，Vitalik 也还强调，变更后的路线图，跟 Phase 2 也是完全兼容的，只是 Phase 2 不再具有优先级。

但是实际上，放弃分片化执行，才是以太坊应该选择的道路。

可执行信标链路线图

在以太坊 2.0 的新路线图中，最令人瞩目的一点是：信标链区块将包含合并后的 Eth1 分片的交易，也即信标链具备了执行功能。其它分片仅具有保存数据的功能。

实际上，新路线图中 “数据分片” 的定位是 “供 rollup 使用的数据可得性（data availability）层”。

没了执行化分片，平方级扩展就无从谈起了。那么，这种 “PoS Layer-1 + rollup + rollup 数据不占据主链区块空间” 架构的可扩展性如何呢？

要解答这个问题，我们先来看看 rollup 方案与主链的交互模式。

首先，你可以把一个 rollup 系统理解为一个无状态的合约，这个合约的内部状态（哪个用户有多少钱），对外是不可见的；但是，该合约内发生的所有交易，其数据会定期公开出来，发布到主链上，使得任一第三方，得到这些数据后，都可以重建出该合约的内部状态。

使用有效性证明的 rollup （例如 zkRollup）的特点是：该合约每次公开交易数据时，都附带一个这些交易已被正确执行、因此新的状态根应是 XXX 的 “计算完整性证明”；如果该证明能通过合约的验证，则该合约更新状态根；如果该证明不能通过验证，则该合约拒绝更新。

使用错误性证明的 rollup（例如 Optimistic Rollup）的方案则相反：任一人每次为合约公开交易数据时，都必须存入一笔押金，并断言合约的新状态根是 YYY；此后一段时间内，任意其他人都能存入押金、发出错误性证明来挑战该断言；错误性证明即证明该批交易有瑕疵，或者交易处理后的新状态根不是 YYY；如果挑战成功，则发布错误断言的人会损失押金；如果一段时间内无人挑战，则合约更新状态根为 YYY。

这两种方案，都必须在链上发布数据，因此会占用链上空间；而且，链上空间的大小，决定了 rollup 系统在单位时间内的处理量（即 TPS）。想得更深一些，如果这些交易数据，能够发布在一个数据量的约束更小的地方，或者说，不去占用 Layer-1 区块的空间，则其处理量，能产生倍加的效果。如果这样的西方有很多，那还可以产生倍乘的效果。

这就是 “数据分片” 及 “以 rollup 为中心的路线图” 的理念：让 rollup 方案把交易数据都放到分片区块中，分片有多少个，处理量就能提升多少倍；当前的以太坊区块数据量大概是 20～30 KB，这个数据量显然是安全的，则，如果我们有 64 条分片，我们每 15 秒就能提供 64*30 = 1920 KB = 1.9 MB 的数据量。而且，使用端我提供了这么大的数据吞吐量，但它不会成为全节点的负担，因为这些数据你想下载就下载，不想下载就可以不下载（也就是 “分片” 的含义），大家你下载一点，我下载一点，节点的负担还是很轻的 —— 反正，验证这些 rollup 合约的状态，并不要求我拥有该 rollup 的所有历史交易数据。以太坊的状态仍然是安全的。

听起来很合理，但还是那句话，太乐观了，太多假设了：

（1）这种 “想下载就下载，不想下载就不下载” 的方法，在 zk rollup 上根本行不通：当 zk rollup 要更新状态根时，zk rollup 合约更新操作的验证者（也即 Layer-1 的全节点）在接受证明时也必须获得与该证明对应的交易数据，否则就无法通过验证。（不需要提供交易数据，仅验证证明就推进合约状态根的方案也有，叫 Validium，那不是 rollup）。也就是说，如果仅考虑 zk rollup，那么 “数据分片” 的方法，从带宽上来说，与大区块没有任何分别。不管这些数据一开始被发到了谁手上、存到了哪里，全节点都要下载它们。

（2）对 optimistic rollup 来说，如果你愿意采取更乐观一点的假设，当然可以，你可以平时完全不下载交易数据，仅保留获得终局性的最新状态根，仅在发生争议时，再下载相关的交易数据，从全节点的角度看，并没有因此丧失对合约状态的验证能力；但是从用户的角度看，事情就完全不同了：你开始不确定自己到底是不是随时能够重构自己的状态，来完成取款。也就是说，用户将不能确定自己用的到底是 optimistic rollup，还是 plasma。本来，optimistic rollup 的方案就是保证了所有全节点都有历史交易的备份，所以用户可以容易地重建自己的状态，并提交状态证明（或断言）完成取款；但如果这一点保证失去了，你就不确定自己能不能重建状态了。optimistic rollup 的安全性也会受到影响：它的安全假设是，获得了交易数据的人之中至少有 1 个是遵守协议的；在数据分片模式下，你并不知道，有多少人会去请求这部分交易数据。

总而言之，“数据分片”模式搭配 zk rollup 时，在带宽的意义上，无法提供更大的可扩展性，而与扩大区块空间的效果相同；在搭配 optimistic rollup 时，相对于大区块，其可扩展性优势与挑战发生的频率成反比；更严重的是，它使 optimistic rollup 有退化为 plasma 的风险（从定义上来说，也不存在 optimistic rollup 了，应该用另一个名字来指称这种介于 optimistic rollup 和 plasma 之间的东西）。

结论

Rollup 方案其实是从 Layer-2 发展过程中吸取了血淋淋的教训而飞出来的凤凰。它最大的特点在于，给用户的资金安全提供了充分的保护。因为任意得到了交易数据的人都可以重建状态，而区块链保证了这些交易数据的永续数据可得性，rollup 方案得以提供 layer-2 方案中首屈一指的用户保护。只有这样的方案，用户才敢真的去使用。舍弃了这种好处，按最大化性能的乐观假设来设计系统，只能设计出用户不敢去用的东西。

只要你意识到，rollup 本质上是一种合约的设计模式，“PoS + 数据分片 + rollup 可以提供更大吞吐量” 的迷思便可一眼洞穿 —— rollup 不管在哪个共识中，都可以提供同样的可扩展性，数据分片能提供更多，只是因为引入了别的安全假设，使 rollup 牺牲安全性来换取吞吐量而已 —— 问题在于，这样的合约，比 rollup 的安全性更弱、可扩展性更强的合约，不是没出现过，不是在 pow 链上就设计不出来，而是设计出来了也没人用而已。

自 2017 年以来，以太坊社区就为着实际的需要艰难地探索安全的可扩展性方案。许多人可能都相信过，“PoS + 分片” 能提供强大的可扩展性，但那是 “分片化执行系统”，有自身的一堆问题。眼前的 “可执行信标链路线”，也不过是牺牲合约本身的属性来换吞吐量而已。时至今日，已经找不到证据，证明为了可扩展性，以太坊应该拥抱 PoS。

归根结底，只有契合用户需要的性能提升，才是真正有意义的性能提升。如果不从用户的实际需要出发，相反，从技术美感或者最大化性能的假设出发，只能设计出空中楼阁。如果可以，那就让用户自己来做决定，在协议层操心太多，往往徒增摩擦。

以太坊的路线图应该改变吗？当然，应该放弃这些不切实际的幻想，回头问问用户需要什么。