

**MOAC 黄皮书**

实施细节

**说明:该黄皮书内容由群友翻译,内容存在瑕疵,如有异议,请以官方发布为准**

**1.基本设置:**

MOAC使用分层共识堆栈，通过分层结构和分片处理来扩展事务处理，在同一个区块链内同步完成。

对于底层，我们利用POW作为主要共识协议，因为POW是验证最为广泛的共识协议,最能够解决大规模的网络设置问题。目前MOAC使用类似于以太坊的POW，但是我们将来会让POW协议可拔插，如果需要，我们可以很容易地用另一种高效的协议换掉POW。

顶层补偿POW的缺点，在POW层中只处理关键事务和控制流程事务。顶层采用可配置的共识协议和分片技术，提供更快更高的吞吐量解决方案。

下层节点叫V-node，每一个v-node有一个智能合约服务器(SCS)节点，Smart Contract Server 简称(SCS)，这个智能合约服务器（SCS）身份可由相应的v-node完全验证。

每个SCS完全由相应的v-node验证身份，为了处理上层合约，每一个SCS节点都得进行绑定。

注意，SCS处理智能合约调用，所有顶层事务都采用智能合约调用的形式，并非所有SCS同时处理单个事务，而是选定的SCS的一部分将处理特定的事务。

SCS的选择是通过启动智能合约调用或刷新调用。启动/刷新调用实际上被传递到底层并达成共识。启动/刷新调用包括的选择标准包括处理节点的百分比。

然后每个v-node将使用预定义的算法在其SCS上调用该调用。 SCS决定是否选择自己来处理此Smart合约。 注意这是一个确定性的过程，SCS的参与可以由任何人验证。

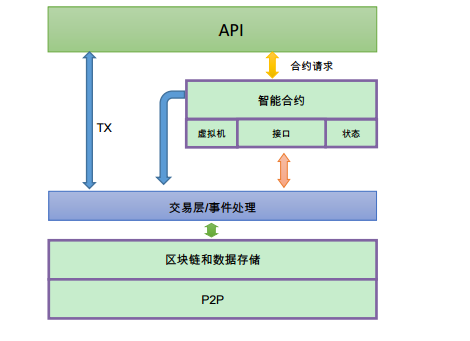
一些SCS被选为某些智能合约的组合后，它们就会相互沟通，形成一个小的共识组。 该组就会处理那个智能合约的调用。 此外，它们之间如何达成这个共识可由启动调用指定。更有效的是这些SCS节点形成了一个子链，并且基于预定协议或用户定的协议来执行共识。 请注意，共识协议与实际的智能合约代码不同。

智能合约状态保存在每个SCS中， 然而，这不是实际的全部区块链，为了实现全部区块链的好处，状态需要定期或按需地刷新到底层的区块链。

当在共识模式下刷新时，SCS节点将在底层v-node上启动数据存储请求。这是从共识分片成员合理认证发起的TX，TX将被广播到所有v-node， 其有效性可以由任何节点进行验证，同一个TX中，它还为每个分片参与者指定了奖励费用。如果TX被证明是无效的，则可以实施额外的过程以没收SCS的保证金。

MOAC，大多数合同将在顶层处理，而在v-node层只处理少量的控制流事务和资产交易。 这是可行的，因为顶层提供快速，灵活和低成本的服务，而v节点层提供缓慢，依赖性和昂贵的服务

**1.1分层结构**

****

1. P2P 网络层： 这个层定义了基础的 p2p 协议。

2. 区块链层：该层处理与区块链操作相关的所有操作，如共识，数据访问等。

3. 交易（TX）层： 该层处理 TX 请求和回复。 它还处理控制类 TX 请求，并在必要时调用与智能合约相关的操作。

4. 智能合约层： 该层执行虚拟机内的智能合约执行，并保持临时合约状态。

5. API 层：API 用于处理终端用户输入并获取下层的输出及返回。

**2. P2P 网络**

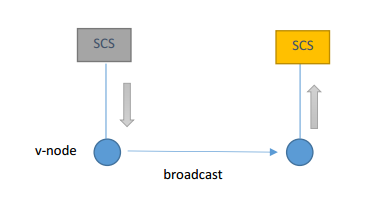
P2P是运行相同协议的节点之间的基础通信层， 节点通过使用经过加密和认证的传输协议的RLPx发送消息进行通信。

对等体可以在任何TCP端口上进行通告和接受连接，但是默认端口连接可能被监听。

每个节点和他配对的对等体连接，保留了一个节点列表。我们连接到三个不同类别的节点：引导节点，静态节点和可信节点。

每个节点可以设置最大对等体数量，一个典型的数字是25。广播节点将发送pkt给所有的对等体。这些同行将进一步发送给同行。 在多跳中，pkt将到达网络中的所有节点。

这个协议提供了基本的网络通道，下层共识就这样建立的。此外，SCS层将形成另一个共识层。 但是，SCS之间没有直接通信。 因此，SCS将其协议打包通信到较低的pkt中。 V-node负责向同行广播SCS的pkt。每个同伴将重新播出，并将pkt传递给其SCS。 因此，SCS将与底层的p2p网络形成逻辑共识。

****

**3.基本分片**

目前的区块链具有可扩展性的问题, 区块链处理能力限于单个节点的处理能力, 这限制了区块链的处理速度,也浪费了整个网络的大量处理能力，随着更多的节点加入到网络，这不会增加系统处理能力, 而是随着网络流量增加会使系统性能降低。

分片是解决这个问题的方法。 它将整个网络分为多个分片，并行处理事务,只要每个分片中有足够的节点,这个系统仍然是非常安全的。

每个分片随机选择会员处理单个智能合同。同一合同的调用将由相同的分片成员处理。适当的合约可以频繁或者明确调用分片内的成员。

请注意 ，即使在任何给定的时间里，每个片内只有几个节点验证和创建区块，安全级别实际上并不比每个节点正在验证和创建区块的情况要低得多。 详细计算见后面部分。

**4. 分布式随机发生器**

分片成员是从所有注册的智能合同服务器随机挑选的，为此我们将使用确定性阈值签名并且验证协作生成随机值。该方案将抵抗绝大多数的操纵，除非大多数验证器相互串通（在某些情况下，根据实施情况，33-50％的验证者可能会干扰执行，导致协议具有67％的活性假设）

阈值签名也用于分片成员试图将合约状态刷新到主块的提交阶段， 我们稍后会介绍。

**5.分片成员的选择**

分片成员的选择分为以下几个步骤：

1）所有SCS在系统合约中注册。

2）系统合同将随机抽取k％的具有足够保证金的SCS节点。 选择算法是公开的，每个v-node可以验证其SCS的ID是否被选中，并通知其SCS节点。

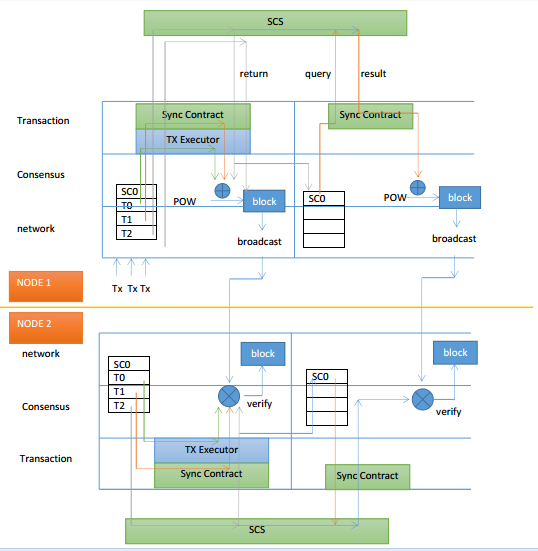
3）如果SCS有效并且愿意参与，它会用带有签名的Ack回复到v节点， V节点将广播其注册。

4）在一定时间内，如果确认的注册率达到最低百分比要求，那么注册成功。

5）每个节点会被秘密的si通知。

6）如果注册超时，则启动另一轮的选择。

**6. 异步的工作流程**



**7.异步合约服务器的调用条件**

在下列条件下，不能以异步执行合约：

a.合约的输出会改变外部状态。

b.在合约中嵌入其他合约调用。

C.合约取决于输入外部状态。这种条件可以放宽，如果时间宽松，让每个调用通过引入

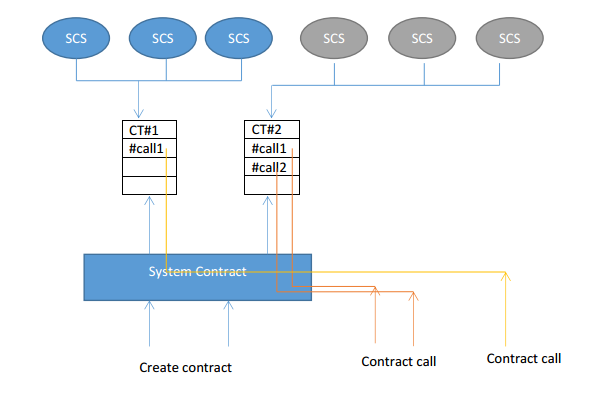
输入状态的方式执行。

**8.分片与合约调用**

a. 运用随机生成器选择执行单一合约的一定比例的节点。系统合约（SC）将记录该选定合同的参与者。

b.特殊分区会定向执行同样的合约。

c. 如果必须根据项目#b进行同步调用，必须在该调用之前进行提交。



d. 合同的执行结果不能立即由合同退还; 也没有额外的系统合将查询分片合约调用结果的结果。执行结果通过显式提交返回。

**9.分片的架构**

在分片的情况下，每个合约执行的结果以Merkel树的形式存储在每个参与的节点。每个分片可以继续增长其区块链，直到达到明确提交请求或达到预定的提交计划。

同一分片中的节点将形成某种类型的共识。 这个协议非常灵活，可以通过合同创建设置指定。

每个SCS连接到一个低一层的共识节点（v-node）,这个SCS完全信任从V-node传来的信息。

每个SCS将首先注册到系统合约，以表明它将参与智能合约处理。 它需要提交一定数量的保证金，如果发现任何欺诈行为，该债券会被扣除。

每个合约的处理，所需的保证金是b，如果SCS有足够的处理能力和带宽支持更多的合约处理，它可以提交n x b的保证金。请注意SCS执行同步合约调用无需保证金。

当用户提交创建一个合约时，他将需要设定下面的参数：

模式：异步或同步

分片比例：10%，最小分片大小的20%

自动提交频率：块号

典型的合约调用工作流程如下：

a. 用户提交启动全部TX的合约，这个TX在v-node中处理，并且启动时通知系统合约。

b. 以确定性的方式随机选择节点。 选择算法将在每个v-node运行，从而决定当前的SCS节点是否是选择的一部分。

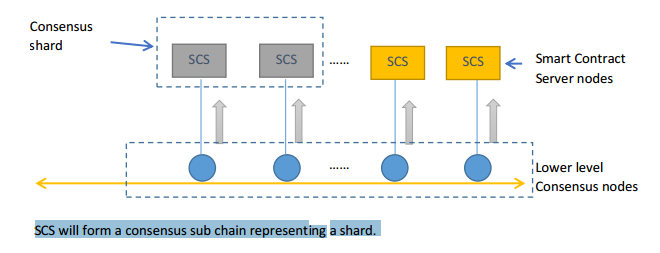
c. 每个SCS节点会发送登记保证金到系统合约来确认它同意参加这个分片。保证金是由系统定义的，合约创建者可以根据需要选择更高的保证金。

d. 每个选定的节点会通过连接的v-node发送广播pkt识别自己，广播是通过低层的P2P网络，这个过程只是在在合约创建时或更新合约时需要。请注意，由于多跳连接，一个v-node可能不知道对方的IP地址，这将防止小分区直接的DoS。

e.每个节点这时开始执行启动合约并生成本地状态。

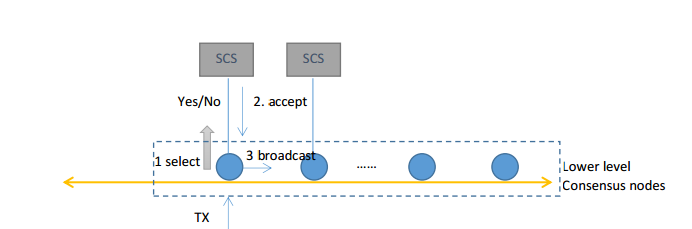
f. 合约上的任何异步调用都会被排队指定到到每个SCS节点，未执行调用的顺序需要节点达成共识，这可以通过POS协议来完成。频率完全由接收的合约调用决定，并且有最小的阈值。

**9.1共识分片**



SCS将形成共识子链代表一个分片。

**9.2创建新分片**

****

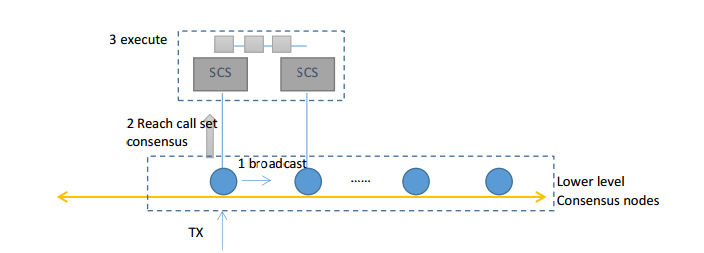
在创建的第一阶段，用户要提交TX来启动合约创建。 V-node将首先执行地址生成，配置等必要的工作，它要明确的决定连接的SCS是否被选择去处理这个合约，如果是的话，通过授权SCS，并通知并验证其参与。SCS也会利用底层的P2P网络广播它参与了该合约。通信pkg包括 (SCS\_ID, ContractAddr, Config)等信息。请注意，此通信可以与其他合同创建相结合，以减少传输pkts。 其他选定的SCS将执行相同的操作。

此通所需要的pkt总数为

𝐶 = ∑𝑘 S，

k: 执行广播的合约总数。 S: 分片大小

**9.3合约的函数调用**

****

用户向v-node提交合同调用。 v-node将通过底层的p2p网络广播到所有分片内其它成员。 V-node会检查当前SCS是否需要执行此调用。 如果是的话，它将传递这个调用到SCS，SCS将以异步方式执行它。

如果多个合约调用到达同一个块中的v-node，这些调用可能会以不同的顺序到达不同的节点， 为了确保所有节点具有相同的顺序，需要共识确保每个分片的成员都具有相同的tx组。

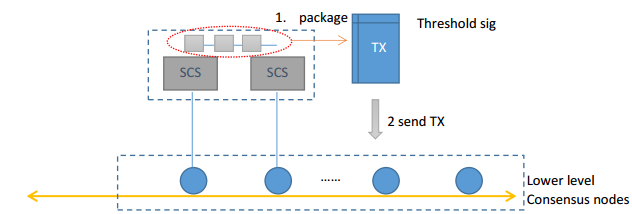
**9.4查询合约状态**

同步合约和分片（异步）合约的合约状态检查略有不同。在同步合约调用中，所有信息将在主/根链中可查，但是，在分片的合约根链中不会有最新的合约状态，而是存储在共识的片中。

共识片状态检查是分两个步骤中完成：

a. 任何想要检查合约状态的节点都需要注册系统合约观察者， 它不需要执行合约或达成共识， 它只是接收来自分片活动的信息，如共识结果，区块的更新。

**9.5预提交**

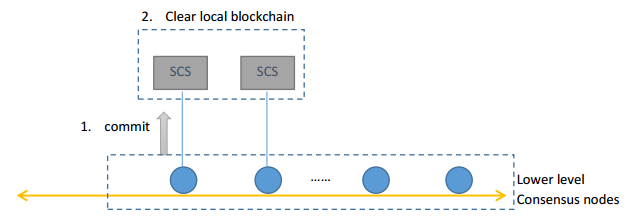
****

共识片会保留对合约执行的区块链，定期地将区块链的这些合约调用结果提交到主/根链。

可以通过对分片的合约调用来调用预提交。或者，如果预定义的话，则可以通过系统合约来调用它。 所有诚实的SCS将发送其分享的TX并广播给其他分片成员。在这些情况下，共识分片将达到阈值签名协议， 并且每个SCS将打包所有未提交的TX封装在一个提交事务中， 提交事务将被发送到其v-node并进一步广播到网络。因为提交TX可以被任何一个节点验证，它将最终被包含到根块链中。

一旦这个TX被包含在根块链中，v-node将通知其SCS提交。

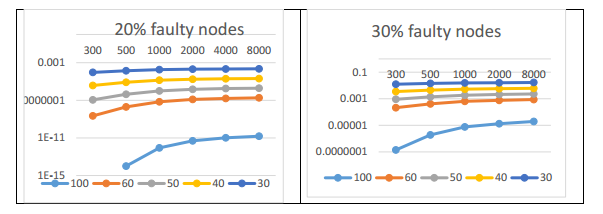
**9.6 提交**



V-node将使用正确的根区块链merkle树哈希向SCS提交提交命令。由于SCS信任其v-node，SCS在信息已经提交到根区块链后，清除本地块链。

**9.7 阈值签名使用**

在提交阶段,我们运用用阈值签名签署共识TX。对于1000个节点,有30％的拜占庭故障的典型设置, 每个碎片都是100,阈值大小51.没有获得51个有效签名的概率为10e-5。



因此，分片大小不能太小,每个分片至少最小需要100个节点，这个应该容易得到满足，因为MOAC是为更广泛的互联网应用设计的应用。此外，SCS的进入壁垒不是很高，而且SCS的处理激励会鼓励更多的SCS加入。

**10.跨链支持**

本节将介绍Moac 区块链和Ethereum区块链如何协同工作，以实现这两个区块链之间的两种令牌的原子交换。

a) 假设在MOAC网络中，有个用户下单要用MOAC的代币J交换 Ethereum 的代币 T，价 格、数量和收件人地址的指定格式: Order: J->T, p, v, T2.

b) 此请求经适当授权提交给智能合约 SC1，并且因此提交到订单簿.

c) Ethereum 中的某用户想要卖出代币 T 来获得MOAC网络的代币 J，参数如下: Order: T->J, p, v, J2.

d) 该订单提交部署到 Ethereum，智能合约 SC2，带有适当的授权。

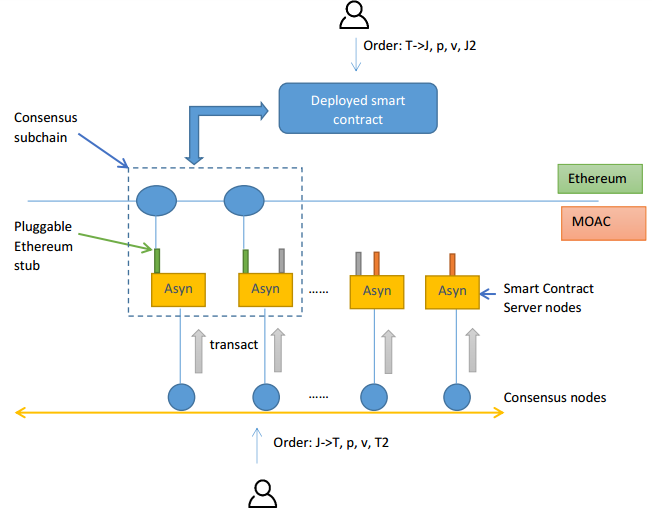
e) MOAC的异步合约服务器可以有连接到 Ethereum 公共链的可插拔存根。这些服务 器将使用优化共识算法形成一个共识子链。这些合约服务器将监视 Ethereum 的 活动并将订单添加到订单簿。

f) MOAC系统中的验证节点将找到匹配的事务（T < - > J）。它会生成一个提案{（J1-> Js，int（Js-> J2），（T1-> Ts，int（Ts-> T2）），Js 是 SC2 的地址，Ts 是 SC1 的地 址。 int（）是有待执行的意图事务

g) J1> Js 将在MOAC网络中成交，T1-> Ts 将在 Ethereum 网络成交。子链将确保这两个交易发生过并且可以验证。这可以通过异步调用的性质完成。通常需要等待块确认的时间段。这个时间段用户可以定义。

h) 每个合约都要等待确保两边的交易都完成了。然后每个都将完成 Ts-> T2 或 Js-> J2 交易。这是以确定性的方式完成的

i) 无论由于任何原因，步骤#g 无法完成，则合约可以给原用户退款 。



在MOAC网络中的异步服务器有几个目的:

a. 形成BFT子链，用在MOAC网络和其他区块链系统之间通信.

b. 启动对其它区块链智能合约的调用。

c. 对订单执行监控并返回相关的状态 。

d. 为发送MOAC基础代币定义逻辑。

类似的，在 Ethereum 系统合约会定义发送Ethereum基础代币逻辑。

**11. 系统合约**

系统合约在工作流程中起关键作用， 它具有以下属性：

a）能被每个节点首先执行的内置合约；

b）拥有SCS注册信息w/bond。

c）如果需要，拥有异步合约存根的内部TX调用。

d）启动合约查询，预提交，提交调用。

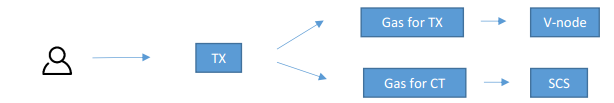
e）从TX中抽取燃料，并向SCS参与者分配燃料。

f）基于分布式表决方案处理版本控制过程。

在每个v-note中执行系统合约，并为每个节点送达相同的结果。

12. 挖矿和奖励

挖矿的报酬分为两部分，一个是底层的POW（现在）。另一个是执行合同的智能合约服务器。



POW部分非常简单， 谁找到了该区块谁就会获得该区块奖励以及TX燃料。由于异步合约执行在SCS节点中完成，相应的奖励将被分发给SCS。所以当用户提交交易时，支付燃料包括两个部分：一个是TX的固定值部分，另一个是基于合同代码计算的变量值。如果合同以同步方式执行，则将被视为TX部分并属于找到该块的矿工。 如果以异步和分片方式执行合同，该部分燃料将分配给分片成员。特别是，每个子链块的共识的分片成员会给予燃料奖励。份额由系统合同确认和记录，只有当区块被提交给根链时才会被分配。

由SCS分片发送的封装的TX中包括向每个分片成员提供的所属燃料。