Solana 中的异步程序执行（APE）

作者：toly，Solana Labs 联合创 来源：X，@aeyakovenko 翻译：善欧巴，本站

在Solana中，状态组织为平坦的键值存储，程序在此状态上执行以更新值，包括用于共识的关键投票程序。异步执行的主要目的是允许投票程序独立于其他所有程序运行。为了始终如一地实现这一目标，我们需要确保几个关键原则。

## 执行域

执行域是一组程序及其操作的键和值，这些程序在执行时彼此独立。执行域可以在不同的线程和核心上运行，并且在各种机器上以不同的时间完成。重要的是，一个执行域（例如，域A）不能读取或写入另一个域（例如，域B）中的任何值。然而，域可以共享在任何一个域执行期间保持一致的状态。需要一个协议来同步域之间的状态，并促进键和值的移动。

交易费用支付者决定交易执行的域。

## 关键域：VED和UED

在Solana中，我们主要关注两个域：投票执行域（VED）和用户执行域（UED）。目标是使验证者能够在所有用户程序执行完成之前进行投票。

VED的组成部分

系统程序：用于转账

投票程序系统变量：用于投票的投票程序系统变量

投票程序：核心投票程序。该程序是静态且一致的，必须在UED和VED中都有。

投票权限：有权投票的账户

投票费用支付者：支付与投票相关费用的账户

费用支付者基金：可用于将sol移动到VED的可更新账户

所有其他账户放在UED中

## 计算VED状态

在N-1和N的纪元边界，运行时计算N+1域的VED状态。此计算必须在纪元N结束前完成。UED计算必须在纪元N-1完成后才能开始。

VED状态的标准

投票账户：质押超过5000 sol且处于活跃状态的账户。

费用支付者和投票权限，费用支付者基金：与上述投票账户关联。

新创建的投票账户最初设置为不活跃。用户必须激活这些账户，一旦激活，它们将在下次VED状态计算中包含在内。用户可以停用账户，该账户将在N+1纪元中从VED状态中移除。

在VED中移动资金

投票账户中声明的费用支付者资金账户可用于在VED中移动资金。更新投票账户中的费用支付者资金账户后，当前的费用支付者基金将在下一个纪元结束前保持活跃。一旦重新计算的VED激活，新的费用支付者基金将进入VED状态，并可用于向费用支付者添加额外的sol。旧的费用支付者基金将进入UED状态，因此可以将资金转回UED中的账户。

虽然可以将其与费用支付者合并，但为了最小化每次投票使用的签名数量，费用支付者通常设置为与投票权限相同的值，因此需要一个单独的账户在VED和UED之间转移sol。

## 通用执行域

上述方法非常针对投票程序。更进一步可以将该方法推广到任意数量的执行域。账户跟踪它们被映射到的执行域，就像操作系统管理程序跟踪物理页面中的VMID一样。对于这个特定项目，有一小部分映射是有用的：

RX-UED：在UED中只读和可执行

RX-UED-VED：在UED和VED中只读和可执行

RW-VED：在VED中读写

RW-UED：在UED中读写

由于账户不能在多个域中同时读写，其他映射是无效的。

程序：只能映射为RX-UED或RX-UED-VED，即在用户或两个域中可读和可执行，但不可写。任何将程序包含为可写的交易必须失败。

账户：只能映射为RW-VED或RW-UED。

系统快照将投票程序和系统程序配置为RX-UED-VED。系统程序提供将系统账户和投票程序账户移动到VED和从VED移动的接口。实际更新仍需一个完整的纪元才能激活。

只有当所有者程序在目标域中时，账户才能在域之间重新映射。因此，只有投票程序和系统程序账户可以在VED和UED之间移动。

采用通用方法后，不再需要显式的费用支付者资金账户。开发人员可以在VED和UED之间重新映射任何系统账户，从而在域之间移动资金。

## 计算VED哈希

在接收到区块后，节点执行所有交易。过程如下：

执行仅涉及VED账户的交易。

含有VED费用支付者但混合账户的交易失败。

跳过所有其他交易。

生成的VED状态更新用于计算银行哈希，即VED哈希。验证者现在可以使用VED哈希进行投票，而不是旧的银行哈希。如果自上次对该分叉进行投票以来有UED哈希更新，则该哈希和插槽将包含在投票中。

## 计算UED哈希

在接收到区块后，节点处理所有交易：

不在VED中的账户被视为UED的一部分。

执行仅涉及UED账户的交易。

含有UED费用支付者但混合账户的交易失败。

跳过所有其他交易。

生成的UED状态更新用于计算银行哈希，即UED哈希。

## 最终性

最终性保持不变。

## 交易执行状态代码

异步执行的副作用是非确定性可能不会在投票分叉时同步立即检测到。如果1/3或更多验证者提交不同的UED哈希，则所有节点必须停止确认和投票，并警告操作员。

要获得状态代码，RPC API请求必须允许用户指定计算相同UED哈希的观察到的质押。执行状态可以表示为ExecutionStatus（签名，质押=X），其中X可以为0或某个默认值。在具有多个节点和客户端的设置中，X值为0是可以接受的，这给用户带来了非确定性极不可能的信心。

非确定性是唯一需要在向用户返回已确认交易的状态代码之前检测的故障，因此X的默认值通常可以低至乐观确认阈值或约5％。

## 领导者和区块生成

领导者必须能够在VED完成后立即开始创建区块。这意味着领导者将对任何UED费用支付者的状态有部分和不完整的信息。

领导者仅对其创建的区块执行VED域

UED重放必须跳过无效的费用支付者

## 投票程序的功能激活

投票程序必须在假设VED投票可能在功能激活时跨越纪元边界的情况下设计，以不同于UED中接触投票程序的其他交易的时间操作。

## 奖励

只有处于VED状态的投票账户才应获得奖励。

## 惩罚

所有投票账户无论处于活跃状态还是VED状态，都有被惩罚的风险。

## 全局时钟更新

时钟系统变量由投票更新。应该使用替代方法。相反，每个区块领导者可以将时钟最多向前移动1秒。至少40%的网络需要恶意，时钟才能比实际时间更快地增长。时钟也可以默认向前移动400毫秒。假设50毫秒的漂移，只有12.5%的领导者需要调整时钟时间以保持同步。