

基于交易预执行产生读写集的方式实现交易并行

本方案用于阐述一种交易并行执行的可行性方案。对于交易并行存在的最大的问题点在于：交易中可能存在多个账户同时共同操作同一个账户的情况，那么被操作账户的数据读写在没有明确的顺序保障的情况下，将出现脏读或脏写的情况。而交易在被真正执行之前是无法获取该笔交易真正操作了哪些账户，而被操作的账户又做了哪些操作均不可知，因此对于确定交易的读写情况，将变成解决交易并行的关键。

首先分方案需要对区块中打包的交易进行分组分批次，分组分批的目的是在于让不相关的交易得以并行执行。分组分批的依据在于对交易的交易发送者(from)以及交易接受者(to)的分析，规则如下：

- 1.相同的交易发送者的交易不能在同一个批次。
- 2.相同的交易接受者的交易不能在同一个批次。
- 3.在本批次交易中，有交易的发送者出现在交易接受者或交易的接受者出现在交易的发送者位置的交易不能出现在同一批次中。

以上规则可通过如下例子进行阐述，现假设区块中包含的交易如下：

交易哈希	交易发送者	交易接受者
0x0001	Address 1	Address a
0x0002	Address 2	Address b
0x0003	Address 3	Address d
0x0004	Address 1	Address c
0x0005	Address 1	Address a
0x0006	Address 4	Address 3
0x0007	Address 5	Address 6
0x0008	Address 6	Address e

首先对上面的交易片段进行分组画像，画像命名为：交易发送者标记画像。标记画像规则如下：

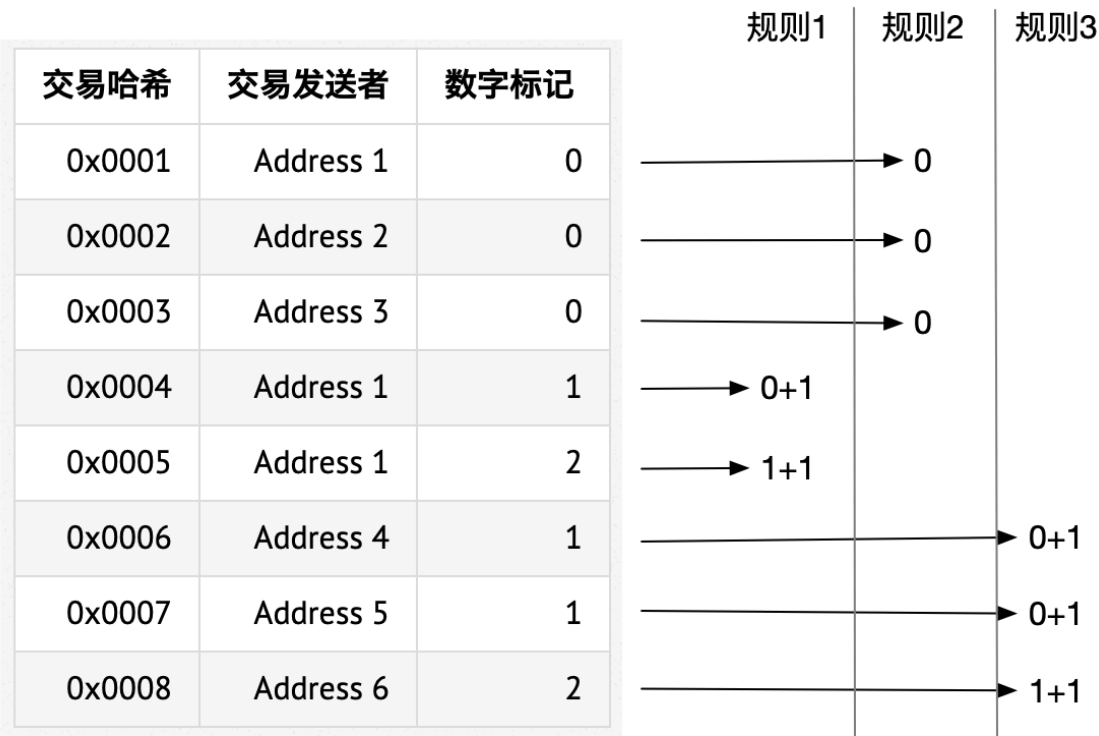
- 1.如果是相同的交易发送者不同的交易哈希，那么该条标记画像中的数字标记项按照升序逐笔递增，数字标 记从零开始。

2.如果是不同交易发送者不同的交易哈希，那么该条标记画像中的数字标记项统一设置为零。

3.如果在本批次的交易中发现了数字标记相同，但是交易的发送者出现在了交易接受者的位置或交易接受者出现在了交易发送者的位置，那么即可认定交易具有相关性，即可依靠原本的交易顺序对排位相对较后的交易对应的数字标记进行递增。

通过上述规则，绘制标记图如下：

交易哈希	交易发送者	数字标记
0x0001	Address 1	0
0x0002	Address 2	0
0x0003	Address 3	0
0x0004	Address 1	1
0x0005	Address 1	2
0x0006	Address 4	1
0x0007	Address 5	1
0x0008	Address 6	2



因此本批次交易片段分成三个批次，同一批次的交易可并行执行，不同批次交易

按照批次序号顺序执行。三个批次的交易如下：第一批：0x0001,0x0002,0x0003；第二批：0x0004,0x0006,0x0007；第三批：0x0005,0x0008。

往往交易的形式不会如上面的交易片段所示的那么简单，一般可以分成两种类型：1.本币类交易 2.合约类交易。在交易分组中，先执行本币类交易，然后再执行合约类交易，因为相较于合约类交易而言，本币类交易更加简单。在一个批次交易中执行完了所有的本币类交易，那么剩下的均为合约类交易。将剩下的所有合约类交易聚合将这个集合命名为tx contract list即(合约交易列表),简称CList。将Clist当中的交易同样进行按照批次进行拆分，拆分的依据是交易的批次，根据交易的数量以及机器的极限性能数进行拆分批次。

将分好批次的CList交易交给执行器并行预执行交易，在执行交易的过程中，每一笔交易肯定都会对某个或某几个账户的状态进行读取以及改写，而交易的最终结果无非就是对账户的数据状态的修改以及记录。因此，在对交易进行预执行的过程中需要暂存这个交易产生的读写集，读取到的账户地址记录在一个集合中，这个集合命名为读集；有涉及到修改的账户地址记录在另一个集合中保存，这个集合命名为写集；当区块包含的所有交易在分批次执行完了以后，所有交易的读写集情况也将产生。

针对产生的读写集以及交易进行如下操作：

1.创建一个全新的全局读写集；

2.遍历本批次交易，将遍历到的当前这笔交易中的读写集记录与全局的读写集记录进行比较。

2.1.如果当前这笔交易的读写集对象在全局读写集中的写集中没有找到，那么可以认定当前这笔交易目前为止没有与任何交易产生冲突，可以将本笔交易产生的读写集整合到全局读写集中；

2.2.如果当前遍历到的这笔交易的读写集对象在全局读写集中的写集中找到了，那么认定当前这笔交易与之前的某笔交易存在冲突，需要重新执行。重新执行本笔交易的读取数据优先从全局读写集中加载，全局读写集中不存在的再从状态库中加载，执行完产生的读写集继续更新到全局读写集中；

3.继续循环执行步骤2直到本区块所有批次的交易都遍历完。

完成后将全局读写集直接进行写入数据库，更新成当前区块的状态。