

# 交易分组并行可行性阐述 V0.2

本方案阐述一种可以通过一种方式将一个区块中打包的交易，进行分组、分批然后并行执行的方式提高区块链中交易的执行效率以及提升区块链的整体效率和速度。下面是对于本方案的详细阐述和示意图。

本阐述方案的实现基础是基于一个已实现的或可实现的技术前提，即合约注册关系表。合约注册关系表登记记录的是一个智能合约中可能调用到的所有非本合约的其他智能合约。这种调用关系可以是一个也可以是多个，由合约的编写者来完成本操作，因为相较于其他人，合约的编写者更加熟悉最新部署的这一智能合约调用到了哪些其他的智能合约。合约注册关系表中的注册关系形式可以是一对一，也可以一对多，并且合约默认对合约自己依赖。

当节点在接收到一个新的区块或触发打块逻辑从交易池中打包一批交易之后，所需要做的动作是统一的，即将交易进行逐一执行。这是区块链的统一行为，而这也正是限制区块链吞吐量的十分重要的难点之一。为了更好地描述本方案的实现细节，预先阐述一批待执行交易的形式如下：

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	null
0x0004	Address 1	Address c
0x0005	Address 1	Address a
0x0006	Address 4	Address A
0x0007	Address 5	Address B
0x0008	Address 6	Address C
0x0009	Address 7	Address F
0x00010	Address 8	Address G
0x00011	Address 9	Address F
0x00012	Address 10	Address J

0x00013	Address 11	Address I
0x00014	Address 12	Address d
0x00015	Address 9	Address C
0x00016	Address 13	Address D
0x00017	Address 14	Address H
0x00018	Address 13	Address I
0x00019	Address 15	Address B
0x00020	Address 16	Address D
0x00021	Address 17	Address H
0x00022	Address 18	Address 1
0X00023	Address 19	Address a
0x00024	Address 20	Address c
0x00025	Address 21	Address 4
0x00026	Address 22	Address erc20 contract X
0x00027	Address 23	Address erc20 xontract Y
0x00028	Address 24	Address G
0x00029	Address 22	Address H
0x00030	Address 15	Address d

其中交易哈希为伪哈希值仅用于区分交易，交易发送者用Address加数字的方式表示，代表普通账户即from。交易的接受者用Address加小写字母表示其他普通账户（用以区别交易发送者from）；用Address加大写字母表示智能合约地址；用Address加erc20 contract加大写字母表示erc20类及token类非本币的智能合约。null表示部署智能合约的交易。

根据交易的接受者类型会发现交易大致可以分为两类：1.账户与账户交互 2.账户与合约交互。其中第二类又可以分成两类：2.1 不涉及token及本币交易的智能合约 2.2 token及本币类的智能合约。

那么可以定义合约注册关系如下：

1.当智能合约中涉及到token、非本币类型的功能时，该智能合约的注册关系为与token

- 有关。
- 2.当智能合约中涉及本币类的转账类型的功能时，该智能合约的注册关系为与coin有关。
- 3.当智能合约中涉及到了其他智能合约的功能或与其他智能合约有交互时，但是合约编写者没有手动注册合约的调用关系，那么该智能合约的注册关系为与sync有关。
- 4.当智能合约中涉及到了其他智能合约的功能或与其他智能合约有交互时，那么该智能合约的注册关系为与其使用到或间接使用到的智能合约有关，如Address A,Address B等。

现假设：智能合约A中使用到了智能合约B,C;智能合约B使用到了智能合约C,F;智能合约E使用到了智能合约F,G;智能合约C使用到了智能合约I;智能合约F使用到了H,I;合约注册关系表可表示为如下形式：

智能合约地址	合约注册关系
Address A	Address B,Address C
Address B	Address C
Address E	Address F,Address G
Address F	Address H,Address I

智能合约注册关系是正向相关，逆向相关性不成立，即Address A与Address B，Address C有关，但是Address B与Address A有关系性推测不成立，Address C与Address A有关系性推测也不成立。而且智能合约注册关系式可传递性的，即Address E与Address F,Address G有关，而Address F与Address I和Address H有关，那么Address E与Address H,Address I有关。合约注册关系图应如下显示形式：

智能合约地址	合约注册关系
Address A	Address B,Address C
Address B	Address C
Address E	Address F,Address G,Address H,Address I
Address F	Address H,Address I

- 那么根据合约注册关系表，构建出当前这批交易的全交易画像，全交易画像分为两种，
- 第一种：对交易发送者构建标记画像，即from标记画像。包含交易发送者地址、交易哈希、数字标记三项。
- 第二种：对交易的接收者构建标记画像,即to标记画像。包含所有合约的地址，交易哈

希，发送者地址，token，sync以及coin。

标记画像构建规则如下：

- 对于from标记画像
- 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。

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	0
0x0007	Address 5	0
0x0008	Address 6	0
0x0009	Address 7	0
0x00010	Address 8	0
0x00011	Address 9	0
0x00012	Address 10	0
0x00013	Address 11	0
0x00014	Address 12	0
0x00015	Address 9	1
0x00016	Address 13	0
0x00017	Address 14	0

0x00018	Address 13	1
0x00019	Address 15	0
0x00020	Address 16	0
0x00021	Address 17	1
0x00022	Address 18	0
0x00023	Address 19	0
0x00024	Address 20	0
0x00025	Address 21	0
0x00026	Address 22	0
0x00027	Address 23	0
0x00028	Address 24	0
0x00029	Address 22	1
0x00030	Address 15	1

所有标记项中标记为数字0的账户交易为第一批次可优先并行执行的交易。而后根据数字标记的数字遴选出剩下批次的可并行执行的交易。

- 对to标记画像

1.首先将所有在本批次交易中调用到的合约地址进行统计，并将所有涉及本币及本币类的合约引申为coin类型；所有erc20token类合约引申为Token类；没有注册合约调用关系但是合约里面调用了其他合约的合约引申为sync类型；将to为空的交易(即部署合约交易)同样引申为coin类型；

2.将当前批次交易中的当前某笔交易调用到的智能合约中所能涉及到的所有智能合约(无论是直接或者间接的涉及到)标记栏里标记为1，没有涉及到的其他智能合约标记栏标记0；

3.当前批次交易中的当前某笔交易调用到的智能合约以及这个智能合约中调用到的其他合约，如果与之前的交易接受者或交易接受者中涉及到的智能合约有关，那么本笔交易调用到的智能合约以及使用到的其他智能合约标记数值在之前有关的智能合约数字标记数值的基础上加一；

		Address	Address	Address	Address	Address
--	--	---------	---------	---------	---------	---------

交易哈希	交易发送者	A	B	C	D	E
0X0001	Address 1	0	0	0	0	0
0x0002	Address 2	0	0	0	0	0
0x0003	Address 3	0	0	0	0	0
0x0004	Address 1	0	0	0	0	0
0x0005	Address 1	0	0	0	0	0
0x0006	Address 4	1	1	1	0	0
0x0007	Address 5	0	2	2	0	0
0x0008	Address 6	0	0	3	0	0
0x0009	Address 7	0	0	0	0	0
0x00010	Address 8	0	0	0	0	0
0x00011	Address 9	0	0	0	0	0
0x00012	Address 10	0	0	0	0	0
0x00013	Address 11	0	0	0	0	0
0x00014	Address 12	0	0	0	0	0
0x00015	Address 9	0	0	4	0	0
0x00016	Address 13	0	0	0	1	0
0x00017	Address 14	0	0	0	0	0
0x00018	Address 13	0	0	0	0	0
0x00019	Address 15	0	3	5	0	0
0x00020	Address 16	0	0	0	2	0
0x00021	Address 17	0	0	0	0	0
0x00022	Address 18	0	0	0	0	0
0x00023	Address 19	0	0	0	0	0
0x00024	Address 20	0	0	0	0	0

0x00025	Address 21	0	0	0	0	0
0x00026	Address 22	0	0	0	0	0
0x00027	Address 23	0	0	0	0	0
0x00028	Address 24	0	0	0	0	0
0x00029	Address 22	0	0	0	0	0
0x00030	Address 15	0	0	0	0	0

将通过from标记画像中所有标记为0的交易开启并行处理，在执行的过程中根据to标记画像进行执行，执行的前提是对账户进行上锁，账户对应的是合约账户。之后的并行执行依靠当前全局唯一性标记记录，全局唯一性标记记录包含的是当前批次中所有使用到的智能合约账户(无论是直接使用还是间接使用)，以及智能合约当前的数字标记。并行执行的方式如下：

可并行的所有交易开启并行执行，当运行到交易接受者的时候首先根据全局唯一性标记记录中对应项中标记的数字标记，如果to标记画像中的对应合约数字标记与其相同，那么可以继续执行；如果不相等那么需要对合约账户进行上锁操作，等待合适的数字标记数字小的优先执行完，全局唯一性标记中对应的合约地址项的数字标记增加以后再判断。如果还不相等就继续等待直到全局唯一性标记中对应的合约地址项数字标记增加到与to标记画像中对应的合约数字标记相等。那么被锁定的合约账户可以解锁并继续执行。

全局唯一性标记形式如下：

合约地址	Address A	Address B	Address C	Address D	Address E	Address F	Address G
数字标记	1	2	3	3	1	2	3

全局唯一性标记的数字标记数值在每执行完一笔交易以后，对应项的智能合约账户的数字标记会在原有基础上增一。待第一批次的可优先执行的交易执行完毕，第二批可执行的交易遵循上述规则继续并行执行。

当当前批次交易都执行完毕以后再进行下一批次的交易执行的时候，无论是from标记记录以及to标记记录和全局唯一性编辑记录都要清空重新开始根据交易绘制。