StreamNet原理说明

# 前言

DAG是基于POW的一套旨在提高吞吐，易于扩展的扩展分布系统。该系统解决了区块链低吞吐的问题。每当新增一笔新的块（这里的块是一个抽象概念，指一笔交易或者一笔证实记录）时，会为其选择两个Tips块。其中一个tip被称为“父块”，另一个则被称为引用块。父块通过主链算法实时计算得出，而引用块则使用蒙特卡洛随机游走算法得到。DAG通过pivot链算法获取一条主链，然后根据特定规则将其余块添加到pivot链中从而获得一条全图的拓扑序。

# 基本原理：

## 数据结构：

StreamNet协议的基本数据结构是一个有向无环图（DAG）, 可用公式表示如下：

***G = <B, g, P, E>***

其中，B是图G中所有块的集合；g是初始节点；P是块之间的证实（父子）关系，如下图1中，***P(1)=g，p(3)=1，***由于g是初始块***p(g)=null*** ; E是包含父子边和引用边的合集，同样下图中父子边<1,2>,引用边<2,6>都是E的元素 。

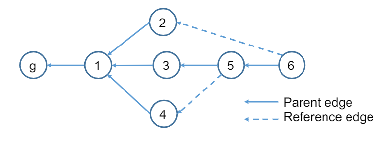


图1 数据结构示例

同样以上图为例介绍以下Tip块的含义。Tip块是指没有被任意块证实的块，如上图所示，块6没有被任意块证实所以块6是一个tip。再往前后退一步，假设现在块6刚刚产生并未建立到任意块的引用。此时图中有三个Tip块分别是2， 5，通过Pivot算法计算得出该图的pivot块是5，所以块6的父引用指向块5。由于只剩下Tip块2，所以块6的引用边指向块2。以上步骤完成后，该DAG的tip节点变化为块6.

## StreamNet架构图

图2表示了多机器节点下StreamNet的架构图。每一个StreamNet机器在本地增加块之后通过gossip协议广播给其他机器并保证所有节点的DAG图最终一致。

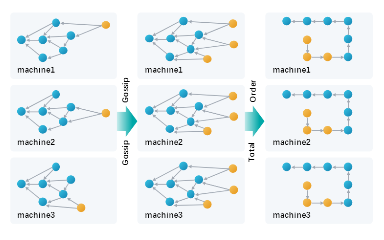
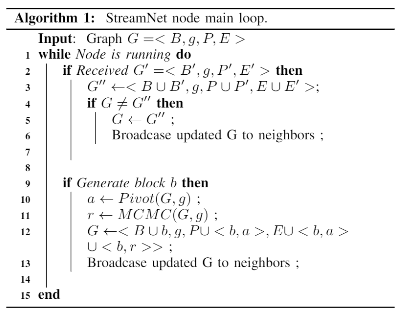


图2 StreamNet架构图

## 一致性协议

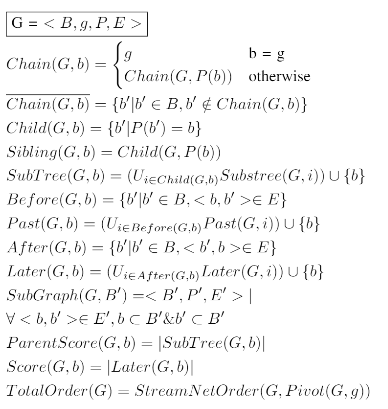
StreamNet的一致性协议是conflux算法（参见Conflux白皮书）的一种实现。为了讲解方便首先定义一些工具函数和表示符号。Chain()方法返回指定块到genesis块之间父引用边连接的所有块组成的链。以图1为例 Chain(G,6)返回 g<-1<-3<-5<-6. 算法实现如下：



Child()返回给定块的所有子块的集合，同样以图1做说明，Child(G,3)={5,6}. Sibling()返回返回指定块的姐妹块，所谓姐妹块是指父引用块相同的节点。Subtree() 返回直接或间接父块为指定块的块构成的子树。Before()返回指定块上的边指向的块的集合。如图1中，块5的边指向了3，4，所以Before(G,5) = {3,4}。Past()函数返回指定块上的边指向的块和它本身的集合。与Before不同点在于Past()包含指定块本身。After() 返回引用边指向该块的所有块的集合。Later()函数与After函数基本相同，除包含引用边指向该块的所有块的集合之外还包含当前你块本身。SubGraph() 返回指定的块集合所构成的子图。仍然以图1为例，当前图包含的块为{g,1,2,3,4,5,6}, 如果要获得只包含{g,1,2,3,4,5}的子图需要移除不包含在以上块中的引用边，即要移除边<3,6>,<5,6>。

ParentScore()是为了计算每个块作为父引用的权重而提出的，可以简单理解为有n个块把自己作为父边（包括父边的父边）那么该块的权重就是n。Score()表示每一个块的权重，与ParentScore()不同的是，前者不仅包含父引用也包含非父引用。

TotalOrder()是最终要得到的图的图的全拓扑序，由以上共同计算得到。其最终计算公式如下图：



## Gossip协议