## 概述：

由于匿名性，比特币被广泛用于洗钱和违禁物品的交易。但是由于每一笔比特币交易历史记录，都保存在一个叫做“区块链（Blockchain）”的公共记录中，包括管理账户的信息以及交易的数量。所以比特币的匿名性只是“伪匿名”，比特币的交易仍然可以追溯到交易者本身。

但是由于区块链非常庞大（目前为80GB量级），对其进行实时查询、实时统计的过程，需要耗费较多的时间。最近我将区块链放置在微软开发的分布式内存图数据库GraphEngine中，加快了实时查询和实时统计的速度。

## 背景：

Blockchain是记录所有比特币交易记录的一个数据集合。在blockchain的结构中，从内向外以此为：支付记录(Transaction)，区块(Block)，区块链(Blockchain)。区块链由多个区块拼接而成，每个区块中保存着多条支付记录，每条支付记录中包含交易的详细信息。区块链示意图如下：



支付记录的结构如下：

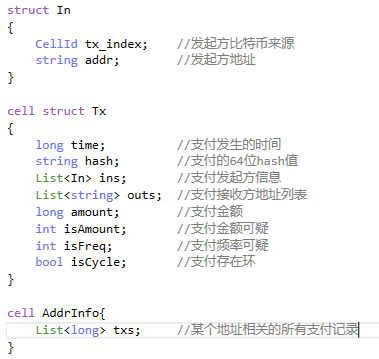
|  |  |
| --- | --- |
| Time | 支付发生的时间 |
| Hash | 支付的64位hash值 |
| Inputs(可以有多个) | |  |  | | --- | --- | | Addr | 发起方地址 | | tx\_index | 发起方比特币来源 | |
| outs(可以有多个) | 接收方地址列表 |
| amount | 支付金额 |

可以将支付记录看做图的一个节点，二发起方比特币来源则是指向前一笔支付记录的边。

## 设计与实现：

首先下载所有的区块，在下载的过程中，删除无关内容，并转化为容易反序列化的json格式。

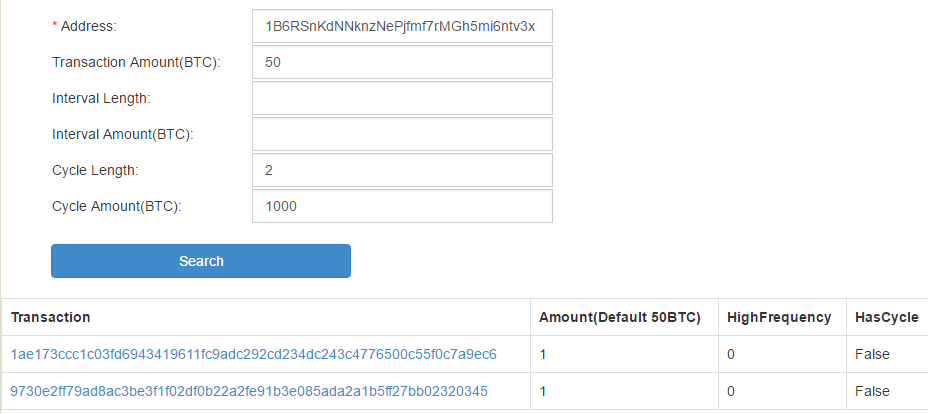
然后逐行读取支付记录，将支付记录存储在GraphEngine的存储模型中，对于此程序设计的存储模型如下：



然后进行可疑支付追踪，追踪包含两个部分：**静态追踪与实时追踪**。静态追踪指的是根据一些公认的指标，对于所有的支付记录预先进行可疑判别。静态追踪可用于生成统计结果，预测可疑支付等操作。实时追踪指的是用户输入某个地址，以及自定义判别条件，系统就能给出此地址相关的可疑支付记录。

在静态追踪中，对于比特币的可疑支付判断标准尚未确定，目前满足以下任一条件则判别为可疑交易：1)交易金额大于50个比特币; 2)在1秒之内多条支付记录汇向同一个地址; 3)存在A->B->A的环形支付记录。

在动态追踪中，用户可以输入如限制金额，交易间隔等参数进行搜索，界面如下：



## 结果：

目前下载的block容量为21GB，height为34万（共40万+），完成静态追踪的初始化耗时1小时。动态查询一个Address（局域网下）耗时为毫秒级别。

## 备注：

难度主要在于静态处理，需要对千万级别的节点进行预处理，而预处理的过程中往往会涉及大量与此节点相关的节点（平均达到100个），所以至少进行billion级别的遍历，导致耗时的增加。

对于动态处理，相比基于文件系统的搜索，通过GraphEngine，将数据放入内存中，能真正做到常数级别的搜索。

### 测试：

见压缩文件。