

# 公有链分叉解决方案及分析

作者：何正军

日期：2018.1.30

在区块链的实现中，因为众多挖矿节点的存在，在同一个区块高度可能挖掘出不同的区块，而矿工又会各自将其进行广播，其他矿工在接收到不同的块之后，在进行有效性的验证之后，又会各自基于其接收到块进行后续块的挖掘。这样最后就会产生分叉，我们需要分析如何解决分叉，使得所有的矿工能够一起努力沿着一致的区块链进行后续区块的挖掘。

## 一、数据结构定义

### 1. bestChain \*chainView

该数据用于记录当前主链（最大累计工作量）最新的块索引数据，用于能够快速查找链的最新数据，同时对其进行维护，保持其往最大工作量的方向发展。

其中的 chainView 定义如下：

```
type chainView struct {  
    mtx    sync.Mutex  
    nodes map[int]*blockNode  
}
```

其中的 nodes 的容量需要作一定的限制，防止长时间运行，保存的区块节点会越来越多，同时较早前经过确认的块不会再产生分叉，浪费不必要的内存，当前设计为保存一天的出块量。

```
const blockNodeCacheLimit = 24*60*4
```

### 2. index \*blockIndex

该数据用于保存本节点接收到的，同时本节点也存在父区块的块，因为块的父区块的传递性，即所有在其中的区块都可以通过父区块回溯到创世区块。

其中 blockIndex 的定义如下：

```
type blockIndex struct {  
    db          database.DB
```

```

chainParams *chaincfg.Params

sync.RWMutex
index map[chainhash.Hash]*blockNode
}

```

同时由于节点会长时间运行，同时资源使用需要受限原因，需要对 index 的容量进行限制。

```
const indexCacheLimit = 24*60*4
```

### 3. orphans map[chainhash.Hash]\*orphanBlock

当存在一些网络延迟的原因或者是区块链分叉原因，当接收到辛苦在本地不存在其父块时，该区块就被定义为孤块，即 orphan。孤块不能参与链的生长，但是当期父块被本节点接收到之后，如果其代表了链最大工作量的延伸方向，就可能被重新接纳为最新块。

```

type orphanBlock struct {
    block      *btcutil.Block
    expiration time.Time
}

```

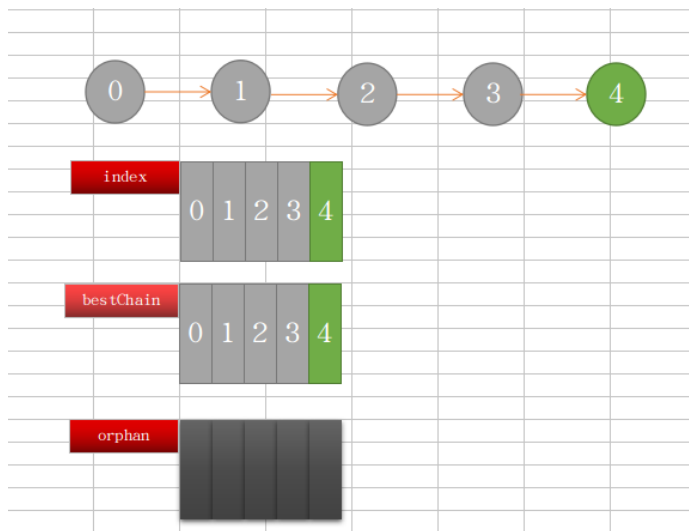
其中孤块数据结构设计中包含有超时丢弃的参数，当孤块没有超时丢弃时间范围内被主链接纳为新区块，其需要被删除。当前设计的超时时间为 6\*15 秒=90 秒，即 6 个块确认的时间

```
const orphanExpirationTime = time.Second * 90
```

## 二、主链扩展延伸分析

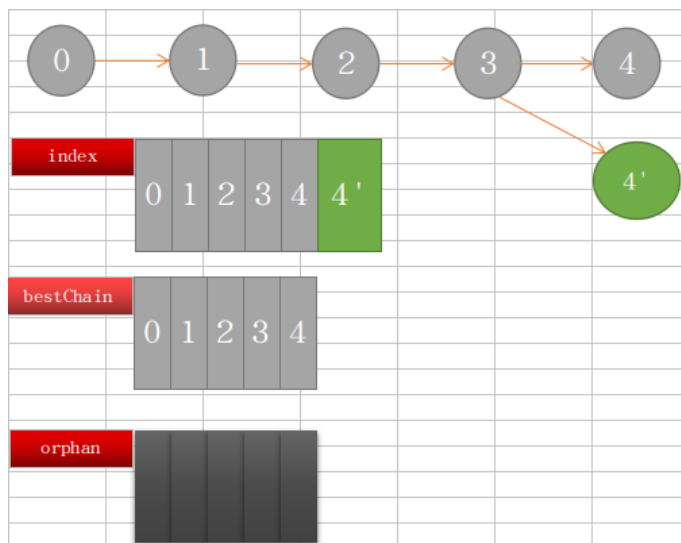
### 1. 正常添加新块

一般情况下，矿工挖到的新块是基于主链（bestChain）的，其可以直接被添加到主链，同时其索引信息也可以从 index 中找到。



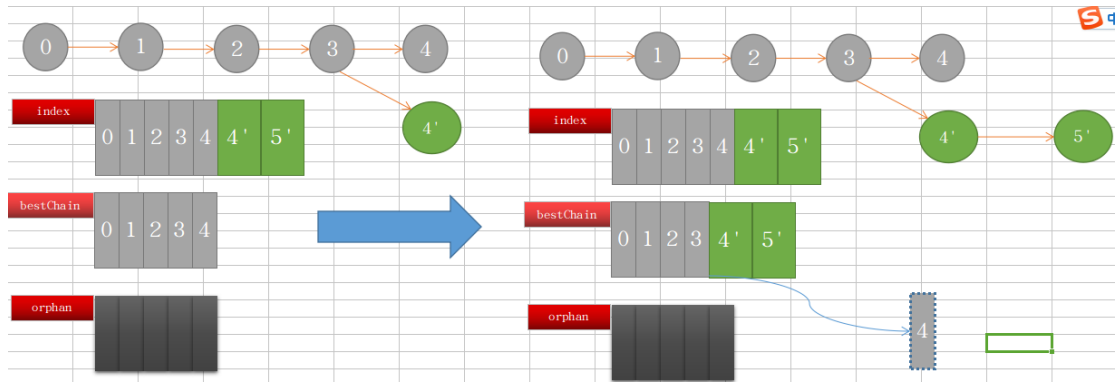
## 2. 生成分叉

当新接收到块与当前主链的最新高度相同，且其工作量没有超过主链，则会形成分叉。



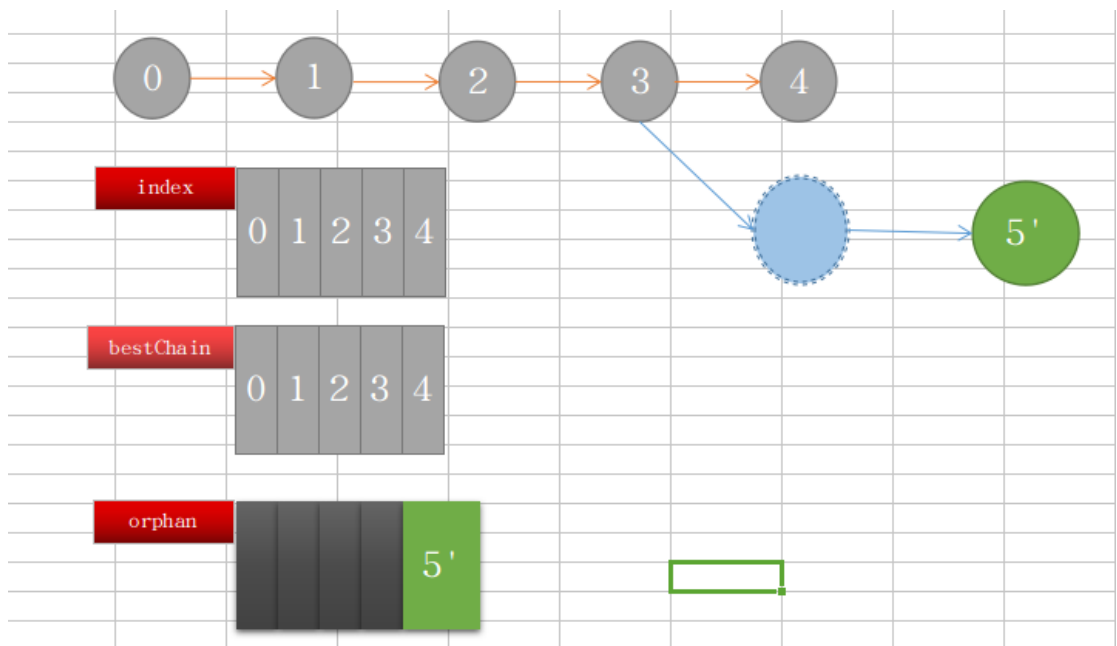
## 3. 侧链变成主链

当最新收到的块是基于侧链进行延伸扩展的，同时其工作量超过了当前存在的主链工作量，则该侧链就升级为主链，而主链降级为侧链。同时更新用于维护主链的块索引，增加新主链的部分块索引，如下图中的 4' 和 5' 的索引，而删除其原有部分块的索引，如块索引 4。



## 4. 接收到孤块（orphan）

当接收到的新块不能从本地的 **index** 中找到其 **parent hash** 指向的块，即该块不能通过前向 **hash** 链接其他的块，则该块就被认为是孤块 **orphan**。同时该块的索引也不会保存到 **index** 和 **bestChain** 中，但是该块的索引会被保存至 **orphan** 指向的 **map** 中，用于后续接收到其父块时，可以将其进行 **chain** 的扩展。



## 5. 链接孤块

当接收到的新块链接到当前链路后，都需要检查孤块池中的所有孤块，查找是否存在这样的孤块，其父区块刚好是当前接收到的区块。如果是，则启动对其添加到相应分支的操作。如果存在分叉的情况，添加了孤块的分支可能成为主分支，如下图所示：

