Merkle Patricia Tree 研究报告

姓名：何静　　学号：202000460142

摘要：Merkle Patricia Tree（MPT）是以太坊中的一种加密认证的，自校验防篡改的数据结构，融合了 Merkle tree 和前缀树两种树的结构优点。可用来存储键值对。以太坊区块的头部包括一个区块头，一个交易的列表和一个 uncle 区块的列表。在区块头部包括了交易的 hash 树根，用来校验交易的列表。

关键词：前缀树　默克尔树　以太坊

一、简介

Merkle Patricia Tree（MPT）是以太坊中的一种加密认证的，自校验防篡改的数据结构，融合了 Merkle tree 和前缀树两种树的结构优点。可用来存储键值对。以太坊区块的头部包括一个区块头，一个交易的列表和一个 uncle 区块的列表。在区块头部包括了交易的 hash 树根，用来校验交易的列表。在 p2p 网络上传输的交易是一个简单的列表,它们被组装成一个叫做 trie 树的特殊数据结构,来计算根 hash。值得注意的是，除了校验区块外，这个数据结构并不是必须的，一旦区块被验证正确，那么它在技术上是可以忽略的。但是，这意味着交易列表在本地以 trie 树的形式存储，发送给客户端的时候序列化成列表。客户端接收到交易列表后重新构建交易列表 trie 树来验证根 hash。

二、Merkle Patricia Tree

1、相关定义及术语

（1）空节点——NULL

空字符串

（2）叶节点/键值对节点——[key,value]

key 是 key 的一种特殊十六进制编码(MP 编码)， value 是 value 的 RLP 编码。

（3）分支节点——[v0 ... v15, vt]

因为 MPT 树中的 key 被编码成一种特殊的 16 进制的表示，再加上最后的 value，所以分支节点是一个 长度为 17 的列表， 前 16 个元素对应着 key 中的 16 个可能的十六进制字符 ，如果有一个[key,value]对在这个分支节点终止，最后一个元素代表一个值 ，即分支节点既可以搜索路径的终止也可以是路径的中间节点。

①附带字段 nodeFlag

I、节点哈希：若该字段不为空，则当需要进行哈希计算时，可以跳过计算过程而直接使用上次计算的结果（当节点变脏时，该字段被置空）。

II、脏标志：当一个节点被修改时，该标志位被置为 1。

III、诞生标志：当该节点第一次被载入内存中（或被修改时），会被赋予一个计数值作为诞生标志，该标志会被作为节点驱除的依据，清除内存中“太老”的未被修改的节点，防止占用的内存空间过多。

（4）扩展节点——[encodePath,key]

也是[key，value]的一个键值对 ，但是这里的 value 是其他节点的 hash 值 ，这个 hash 可以被用来查询数据库中的节点。

（5）世界状态

在以太坊中，所有账户的状态数据统称为世界状态。

（6）轻节点

只存储区块头数据的节点。

（7）区块链分叉

指向同一个父块的 2 个区块被同时生成的情况，某些部分的矿工看到其中一个区块，其他的矿工则看到另外一个区块。这导致 2 种区块链同时增长。

（8）区块头

指以太坊区块结构体的一部分，用于存储该区块的头部信息，如父区块哈希、世界状态哈希、交易回执集合哈希等。区块头仅存储一些“固定”长度的哈希字段。

2、思路

当出现一些元素的节点，但其路径较长，可以将这种层级关系缩减为一个键值对节点[key,value]。其中，键值为层级树的路径元素，值为节点的 hash 值。

3、三种 key 值编码方式

（1）Raw 编码

Raw 编码就是原生的 key 值，不做任何改变。这种编码方式的 key，是 MPT 对外提供接口的默认编码方式。

（2）Hex 编码

Hex 编码就是把一个 8 位的字节数据用两个十六进制数展示出来，编

码时，将 8 位二进制码重新分组成两个 4 位的字节，其中一个字节的低 4 位是原字节的高四位，另一个字节的低 4 位是原数据的低 4 位，高 4 位都补 0，然后输出这两个字节对应十六进制数字作为编码。Hex 编码后的长度是源数据的 2 倍。

（3）HP 编码

用于区分叶节点和扩展节点，把奇数路径变成偶数路径。如果有 terminator(16)那么就去掉 terminator。根据表格给 key 加上 prefix。

node type	path length	prefix	hexchar
extension	even	0000	0x0
extension	odd	0001	0x1
leaf	even	0010	0x2
leaf	odd	0011	0x3

如果 prefix 是 0x0 或者 0x2，加一个 padding nibble 0 在 prefix 后面，所以最终应该是 0x00 和 0x20。原因是为了保证 key（path）的长度为偶数。

（4）转换方式

4、特点

（1）指向下一级节点的指针是使用节点的确定性加密 hash，而不是指向下一级节点的指针。

（2）融合了 merkle 树的优势：快速重哈希以及轻节点扩展。

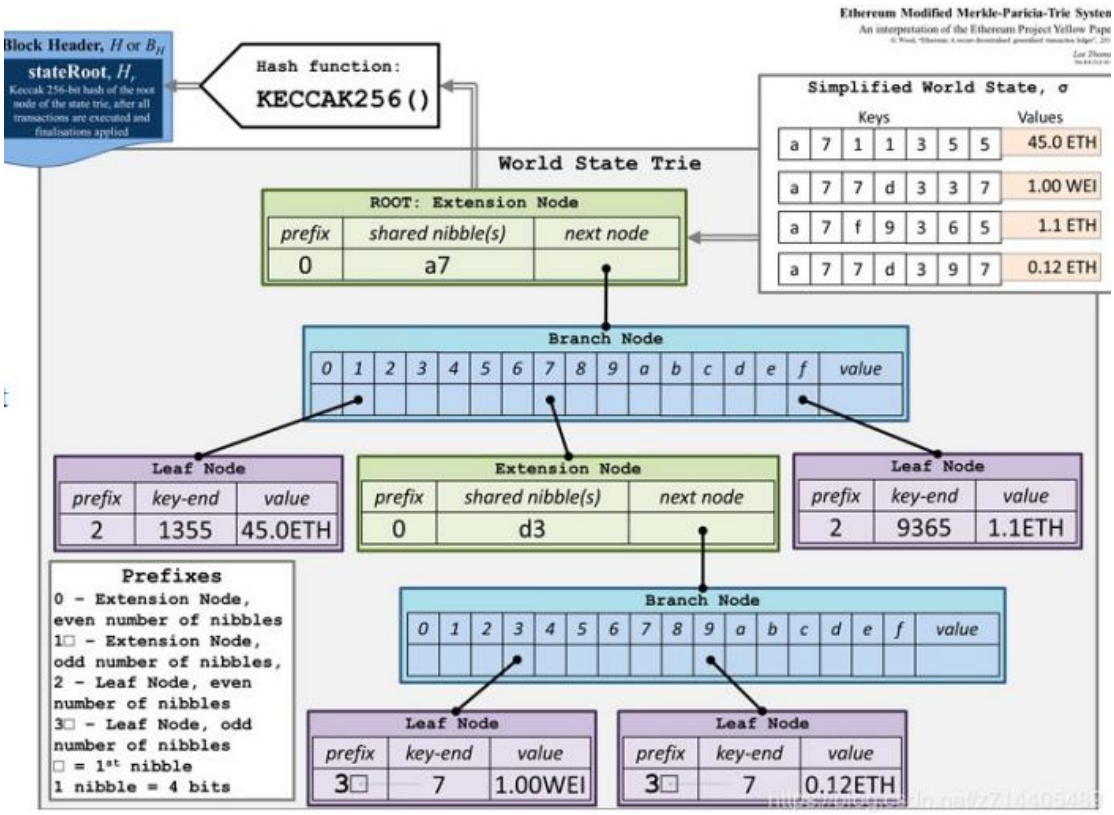
①快速重哈希

当节点内容发生变化时，可以在前一次哈希的基础上，只将被修改的树节点进行哈希重计算，就可以得到一个新的根哈希以代表整个树的状态。

②轻节点扩展

对于每个区块，不需要存储交易列表、回执列表等数据，只需存储区块头数据。可实现在非信任的环境中验证交易是否被收录在区块链账本上。

（3）结构图



三、操作

1、Get

（1）将需要查找 Key 的 Raw 编码转换成 Hex 编码，得到的内容称之为搜索路径；

（2）从根节点开始搜寻与搜索路径内容一致的路径；

①若当前节点为叶子节点，存储的内容是数据项的内容，且搜索路径的内容与叶子节点的 key 一致，则表示找到该节点；反之则表示该节点在树中不存在。

②若当前节点为扩展节点，且存储的内容是哈希索引，则利用哈希索引从数据库中加载该节点，再将搜索路径作为参数，对新解析出来的节点递归地调用查找函数。

③若当前节点为扩展节点，存储的内容是另外一个节点的引用，且当前节点的 key 是搜索路径的前缀，则将搜索路径减去当前节点的 key，将剩余的搜索路径作为参数，对其子节点递归地调用查找函数；若当前节点的 key 不是搜索路径的前缀，表示该节点在树中不存在。

④若当前节点为分支节点，若搜索路径为空，则返回分支节点的存储内容；反之利用搜索路径的第一个字节选择分支节点的孩子节点，将剩余的搜索路径作为参数递归地调用查找函数。

2、Insert

（基于查找过程完成）

（1)首先找到与新插入节点拥有最长相同路径前缀的节点,记为Node;

（2）若该 Node 为分支节点：

①剩余的搜索路径不为空，则将新节点作为一个叶子节点插入到对应的孩子列表中；

②剩余的搜索路径为空（完全匹配），则将新节点的内容存储在分支节点的第 17 个孩子节点项中（Value）；

（3）若该节点为叶子 / 扩展节点：

①剩余的搜索路径与当前节点的 key 一致,则把当前节点 Val 更新即可；

②剩余的搜索路径与当前节点的 key 不完全一致，则将叶子 / 扩展节点的孩子节点替换成分支节点，将新节点与当前节点 key 的共同前缀作为当前节点的 key，将新节点与当前节点的孩子节点作为两个孩子插入到分支节点的孩子列表中，同时当前节点转换成了一个扩展节点（若新节点与当前节点没有共同前缀，则直接用生成的分支节点替换当前节点）；

（4）若插入成功，则将被修改节点的 dirty 标志置为 true，hash 标志置空（之前的结果已经不可能用），且将节点的诞生标记更新为现在。

3、Delete

（与插入操作类似，需借助查找过程完成）

(1)找到与需要插入的节点拥有最长相同路径前缀的节点,记为Node;

(2)若 Node 为叶子 / 扩展节点：

①若剩余的搜索路径与 node 的 Key 完全一致，则将整个 node 删除；

②若剩余的搜索路径与 node 的 key 不匹配，则表示需要删除的节点不存于树中，删除失败；

③若 node 的 key 是剩余搜索路径的前缀，则对该节点的 Val 做递归的删除调用；

(3)若 Node 为分支节点：

①删除孩子列表中相应下标标志的节点；

②删除结束，若 Node 的孩子个数只剩下一个，那么将分支节点替换成一个叶子 / 扩展节点；

(4)若删除成功，则将被修改节点的 dirty 标志置为 true，hash 标志置空（之前的结果已经不可能用），且将节点的诞生标记更新为现在。

4、Update

（插入与删除操作的结合）

当用户调用 Update 函数时，若 value 不为空，则隐式地转为调用 Insert；若 value 为空，则隐式地转为调用 Delete。

5、Commit

Commit 函数提供将内存中的 MPT 数据持久化到数据库的功能。在 Commit 完成后，所有变脏的树节点会重新进行哈希计算，并且将新内容写入数据库；最终新的根节点哈希将被作为 MPT 的最新状态被返回。

四、作用

1、提供一种快速计算所维护的数据集合 hash 标识的机制

2、提供了快速回滚机制

3、提供了默克尔证明方法，可进行轻节点的扩展，实现简单支付验证

五、参考资料

[1] 以太坊 Merkle Patricia Tree 全解析 - 知乎 (zhihu.com)

[2] Merkle Patricia Tree (MPT) 树详解 - 风之舞 555 - 博客园 (cnblogs.com)

[3] Merkle Patricia Tree (梅克尔帕特里夏树) 详解 | yangcl's (yangchenglong11.github.io)

[4] (32 条消息) Merkle Patricia Tree 详解_tianlongtc 的博客-CSDN 博客_patricia tree

[5] (32 条消息) MPT 树详解_weixin_30832405 的博客-CSDN 博客

[6] (32 条消息) Merkle Patricia Tree 梅克尔帕特里夏树（MPT）详细介绍_跨链技术践行者的博客-CSDN 博客