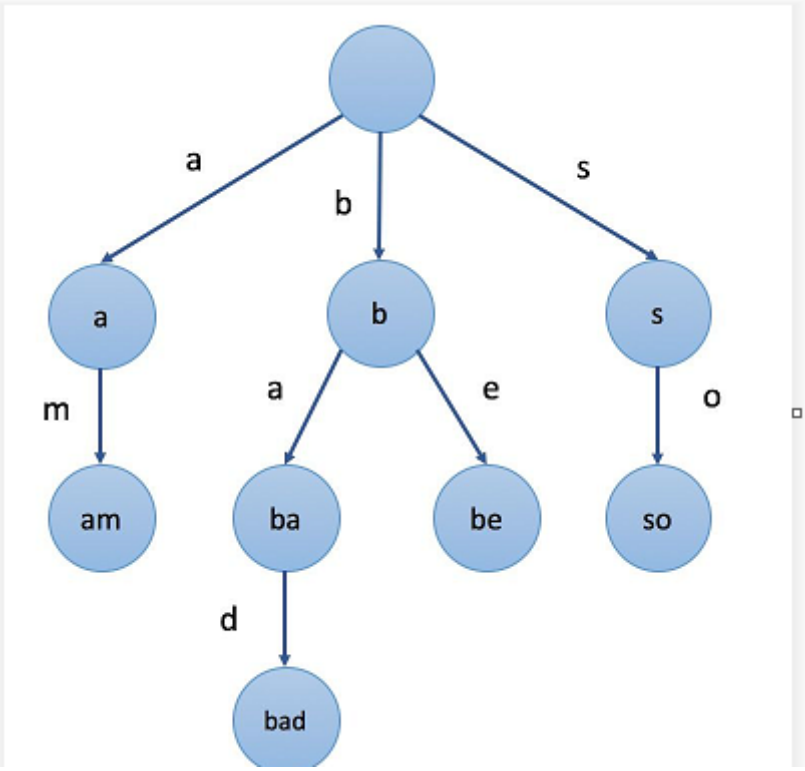
Report on MPT

前言

在之前的项目中，学习了数据结构merkle树，了解了这种其计算机领域，用来比对及其验证处理的作用，接下来介绍前缀树和另一种更优的数据结构——Merkle Patricia Trie.

前缀树

典型示例



应用场景

Trie树的典型应用有：

- 用于统计、排序和保存大量的字符串（但不局限于字符串）.
- 常被搜索引擎系统用于文本词频统计.

优点

利用字符串的公共前缀来减少查询时间，最大限度地减少无谓的字符串比较，查询效率比哈希树高.

基本性质

- 根节点不包含字符，除根节点外每个节点都只包含一个字符.
- 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串.

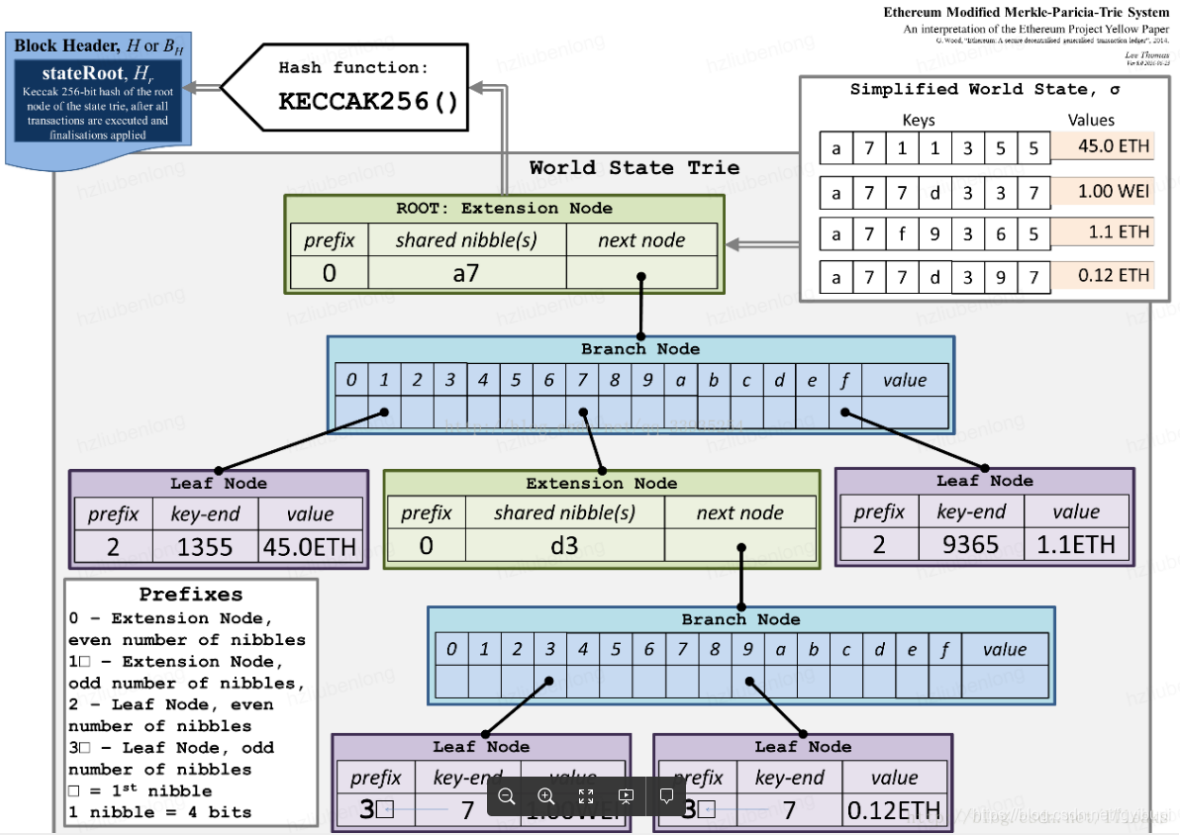
- 每个节点的所有子节点包含的字符都不相同.

Merkle Patricia Trie

有了哈希树和前缀树的知识点，接下来介绍一下MPT，Merkle Patricia Trie是默克尔树和帕特里夏树的结合缩写，是一种经过改良的、融合了默克尔树和前缀树两种树结构优点的数据结构，经典的应用场景——以太坊，使用Merkle Patricia Trie来存储所有账户状态，以及每个区块中的交易和收据数据，它是一种典型的用空间换时间的数据结构.

基本结构

官方结构：



节点

从前面结构图可以看出，Merkle Patricia Tree有4种类型的节点：

叶子节点，表示为[key,value]的一个键值对。和前面的英文字母key不一样，数据库中的key是节点的RLP编码的sha3哈希，value是节点的RLP编码.
扩展节点，也是[key，value]的一个键值对，但是这里的value是其他节点的hash值，通过hash链接到其他节点.
分支节点，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list，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。分支节点的父亲必然是extension node。扩展节点合并相同的前缀，扩展节点下面是分支节点。由于分支节点是16长度数组，故该节点减少了层高和存储空间.
空节点，代码中用null表示.

KEY值编码

在以太坊中，MPT树的key值共有三种不同的编码方式，以满足不同场景的不同需求，在这里对每一种进行介绍三种编码方式分别为：Raw编码（原生的字符）、Hex编码（扩展的16进制编码）、Hex-Prefix编码（16进制前缀编码）.

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.

Hex编码

从Raw编码向Hex编码的转换规则是：

- 将Raw编码的每个字符，根据高4位低4位拆成两个字节.
- 若该Key对应的节点存储的是真实的数据项内容（即该节点是叶子节点），则在末位添加一个ASCII值为16的字符作为终止标志符.
- 若该key对应的节点存储的是另外一个节点的哈希索引（即该节点是扩展节点），则不加任何字符.

Hex-Prefix编码

HP编码的规则如下：

- 若原key的末尾字节的值为16（即该节点是叶子节点），去掉该字节.
- 在key之前增加一个半字节，其中最低位用来编码原本key长度的奇偶信息，key长度为奇数，则该位为1；低2位中编码一个特殊的终止标记符，若该节点为叶子节点，则该位为1.
- 若原本key的长度为奇数，则在key之前再增加一个值为0x0的半字节.
- 将原本key的内容作压缩，即将两个字符以高4位低4位进行划分，存储在一个字节中（Hex扩展的逆过程).

主要功能

1、存储任意长度的key-value键值对数据.

2、快速计算所维护数据集哈希标识.

3、快速状态回滚.

4、默克尔证明的证明方法，进行轻节点的扩展，实现简单支付验证.

四大操作

Insert Key Value

操作步骤

1、如果待插入的key的长度为0，那么意味着在当前节点上更新待插入的value.

2、判断当前节点n的类型，针对不同类型的结点有不同类型的操作：

1、压缩节点：计算当前节点n的key与待插入的key的相同前缀下标+1（返回的是不一致的第一个下标）.如果相同前缀下标与当前节点n的key长度一致，也就意味着待插入的key与当前节点n的key完全匹配，就是更新当前节点n的value。递归调用当前方法，参数为当前节点n的value，prefix+之前计算的相同前缀，待插入的key剩下的部分，以及value。完成后将返回的节点作为当前节点的value返回。如果不一致，也就意味着有了分支。新建fullNode，分别对 当前节点n的key 和 待插入的key 的不一致下标开始递归调用插入后续key及value，返回值为fullNode的两个分支。节点n插入的是n的value，另一分支为待插入的value。递归完成后，当前调用返回shortNode，key为相同前缀，value为新建的fullNode。还有一种情况，如果最开始节点key就不匹配，直接就返回fullNode，因为没有共同前缀key.

2、分支节点：因为是分支节点，那么每个child的key只有一位，那么只要将value插入跟待插入的key的第一位相同的child位置就可以了.

3、哈希节点：哈希节点先去数据库中load相关节点的数据，之后再递归调用.

Delete Key

与Insert key类似.

Update Key Value

相当于Insert key与Delete key的整合.

Get Key Value

valueNode：直接返回节点n的value.

shortNode: 如果剩余的key的长度小于节点n的key的长度或者两个key的前缀不匹配，表示在树种没有找到对应的key，直接返回。到这里了表示待查找的key与该节点n的key是匹配的，那么只需要将节点n的value和剩余的待查找的子key带入递归.

结语

以上就是对Merkle Patricia Trie的学习.