Data structure

Xiaojie Zhu

February 12, 2018

1 Merkle Tree

Merkle tree is also called hash tree. It applies cryptographic hash function to construct a tree structure. Based on the property of cryptographic hash function, easy to compute from input and hard to reverse from result, it is mainly used to prove the data integrity. Specifically, every leaf node is labelled with the hash of a data block and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes.

2 Radix Tree

Radix tree is also called radix trie and compact prefix tree. It is a data structure that optimises the space utilisation. If a node is the only child, the child is merged with its parent. It results in each node having at least two children. In addition, in the radix tree edges can be labeled with sequences of elements as well as single element. Based on this design, it is more efficient for small sets and strings sharing long prefixes.

Radix tree supports insertion, deletion, and lookup operation.

The lookup operation determines whether a string exists in the tree or not. In total, the procedures are less than the number of bits. It starts from root and uses depth first search.

Insertion adds a new string to the tree while trying to minimise the amount of data stored. Specifically, it starts from root and traverses the tree. If there exists a shared prefix, the new edge is labelled with remaining element in the input string. Otherwise, it adds a new outgoing edge from the root.

Deletion is to remove a string from a tree. It first locates the leaf representing the target node. After that, it is removed and if its parent has only one child, then the child is merged with the parent node.

3 Merkle Patricia Trie

Trie is also called digital tree, radix tree or prefix tree, which is a kind of search tree. Merkle Patricia trie provides a cryptographically authenticated data structure. It has both property of Merkle tree and Radix tree. To be more precise, every node consists two parts. The first part represents the symbols of the alphabet and the second part is the terminal value at the node. Instead of using the memory address, it applies the cryptographic authentication to the data structure.

Compared with the radix tree, the Merkle Patricia Trie has better efficiency. It is mainly caused by extra data structure. For the leaf node, instead of traversing from root to the leaf, the path can be encoded at the beginning.

In the Ethereum, there are 3 Merkle tries, state trie, storage trie and transaction trie. State trie is one global state and updates over time. The storage trie is where all contract data lives and transaction trie is a separate transactions trie for every block.