Hash Tree

2015/11/10
0x64 Tales
#02 Data Structure

Livesense Inc.
HORINOUCHI Masato

Hash Tree ってなに?

概要

暗号理論および計算機科学において、ハッシュ木(Hash tree, ハッシュツリー)またはマークル木(Merkle tree)とは、より大きなデータ（例えばファイル）の要約結果を格納する木構造の一種であり、データの検証を行う際に使用される。

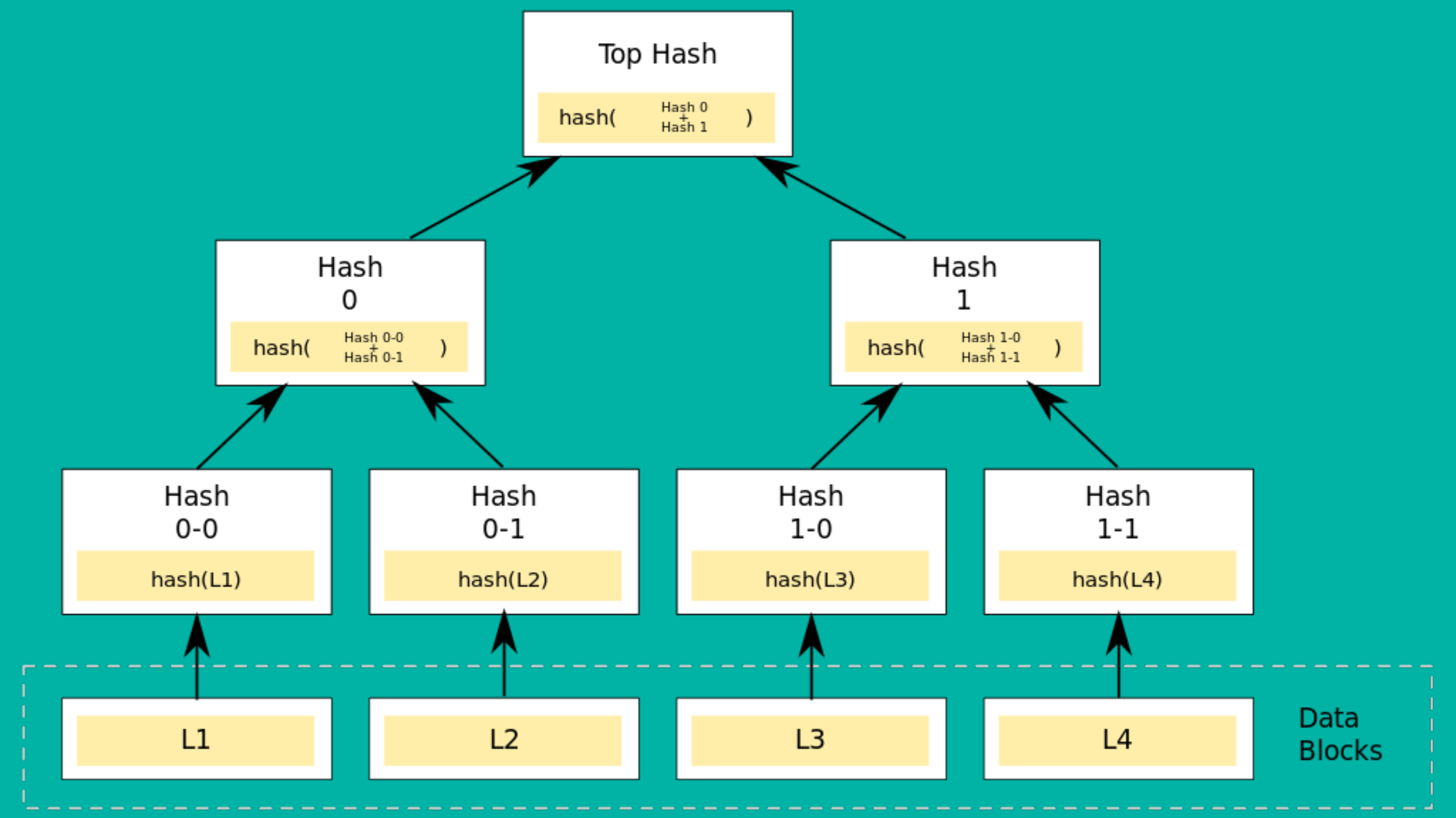
— ハッシュ木 - *Wikipedia*

要するに...

大きなデータを複数データブロックに分割し、破損や改竄の検証処理に利用できる。

動作原理

- 二分木
  - リーフ
    - データブロックのハッシュ値
  - (内部)ノード
    - 子ノードのハッシュ値を結合した結果のハッシュ値


Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks

デモ

元データ生成

```
$ dd if=/dev/urandom of=sample bs=1M count=4
$ split --bytes=1048576 --numeric=1 --suffix-length=1 sample L
$ ls -l
-rw-r--r-- 1 horinouchi horinouchi 1048576 Nov 10 18:37 L1
-rw-r--r-- 1 horinouchi horinouchi 1048576 Nov 10 18:37 L2
-rw-r--r-- 1 horinouchi horinouchi 1048576 Nov 10 18:37 L3
-rw-r--r-- 1 horinouchi horinouchi 1048576 Nov 10 18:37 L4
-rw-r--r-- 1 horinouchi horinouchi 4194304 Nov 10 18:34 sample
```

データ検証などで利用するMerkle Treeのメモ を参考にしました。

手作業

```
$ openssl dgst -sha256 -binary L1 > H00
$ openssl dgst -sha256 -binary L2 > H01
$ openssl dgst -sha256 -binary L3 > H10
$ openssl dgst -sha256 -binary L4 > H11

$ cat H00 H01 | openssl dgst -sha256 -binary > H0
$ cat H10 H11 | openssl dgst -sha256 -binary > H1

$ cat H0 H1 | openssl dgst -sha256 -binary > TOP

$ hexdump -C TOP
```

Ruby gem

```
$ gem install treehash
$ treehash sample
```

- Treehash
  - とはいえ、この gem の中身は全く二分木使ってない...。

ZFS

- Data integrity (データ完全性)
  - ディスクを直接書き換えてもブロック単位でチェックできるよ。
- Copy-On-Write
  - ブロック単位で更新箇所のみ write できるよ。

他の Hash Tree の利用

- P2P
  - BitTorrent
  - Bitcoin
- 分散VCS
  - Git

ご清聴ありがとうございました