课时 12

LSM 树在 Apache HBase 等存储系统中的应用

1. Log-Structured 结构的优化
2. SSTable 和 LSM 树
3. LSM 树的应用

前言

A: 1	A: 1	C: 1	A: 1	A: 1	B: 1	C: 1	A: 1
B: 1	B: 1	C: 1	C: 1	A: 1	C: 1	A: 1	A: 1

这种方法存在的一个问题

就是当我们不断地添加新数据进去之后，所占用的空间会越来越大

而且遍历整个数据结构的时间也随之越来越长

可以通过一种叫 Compaction 的方法把数据合并

Log-Structured 结构的优化

首先可以定义一个大小为 N 的固定数组，称它为 Segment

一个 Segment 最多可以存储 N 个数据，当有第 N+1 个数据需要写入 Log-Structured 结构的时候会创建一个新的 Segment，然后将 N+1 个数据写入到新的 Segment 中

Segment 1

A: 1	A: 1	C: 1	A: 1	A: 1	B: 1	C: 1	A: 1
B: 1	B: 1	C: 1	C: 1	A: 1	C: 1	A: 1	A: 1

Segment 2

B: 1	A: 1	C: 1	B: 1	C: 1

定义一个 Segment 的大小为 16

当 Segment 1 写满了 16 个数据之后会将新的数据写入到 Segment 2 里

Log-Structured 结构的优化

假设我们定义当 Segment 的数量到达两个的时候，后台线程就会执行 Compaction 来合并结果

Segment 1

A: 1	A: 1	C: 1	A: 1	A: 1	B: 1	C: 1	A: 1
B: 1	B: 1	C: 1	C: 1	A: 1	C: 1	A: 1	A: 1

Segment 2

B: 1	A: 1	C: 1	B: 1	C: 1	C: 1	C: 1	B: 1
B: 1	B: 1	C: 1	A: 1	A: 1	A: 1	A: 1	B: 1

Log-Structured 结构的优化

Segment 1

A: 1	A: 1	C: 1	A: 1	A: 1	B: 1	C: 1	A: 1
B: 1	B: 1	C: 1	C: 1	A: 1	C: 1	A: 1	A: 1

Segment 2

B: 1	A: 1	C: 1	B: 1	C: 1	C: 1	C: 1	B: 1
B: 1	B: 1	C: 1	A: 1	A: 1	A: 1	A: 1	B: 1

Compacted Segment

A: 13	B: 9	C: 10

SSTable 和 LSM 树

SSTable（Sorted String Table）数据结构是在 Log-Structured 结构的基础上，多加了一条规则

就是所有保存在 Log-Structured 结构里的数据都是键值对，并且键必须是字符串

在经过了 Compaction 操作之后，所有的 Compacted Segment 里保存的键值对都必须按照字符排序

SSTable 和 LSM 树

假设现在想利用 Log-Structured 结构来保存一本书里的词频

为了方便说明，把 Segment 的大小设为 4

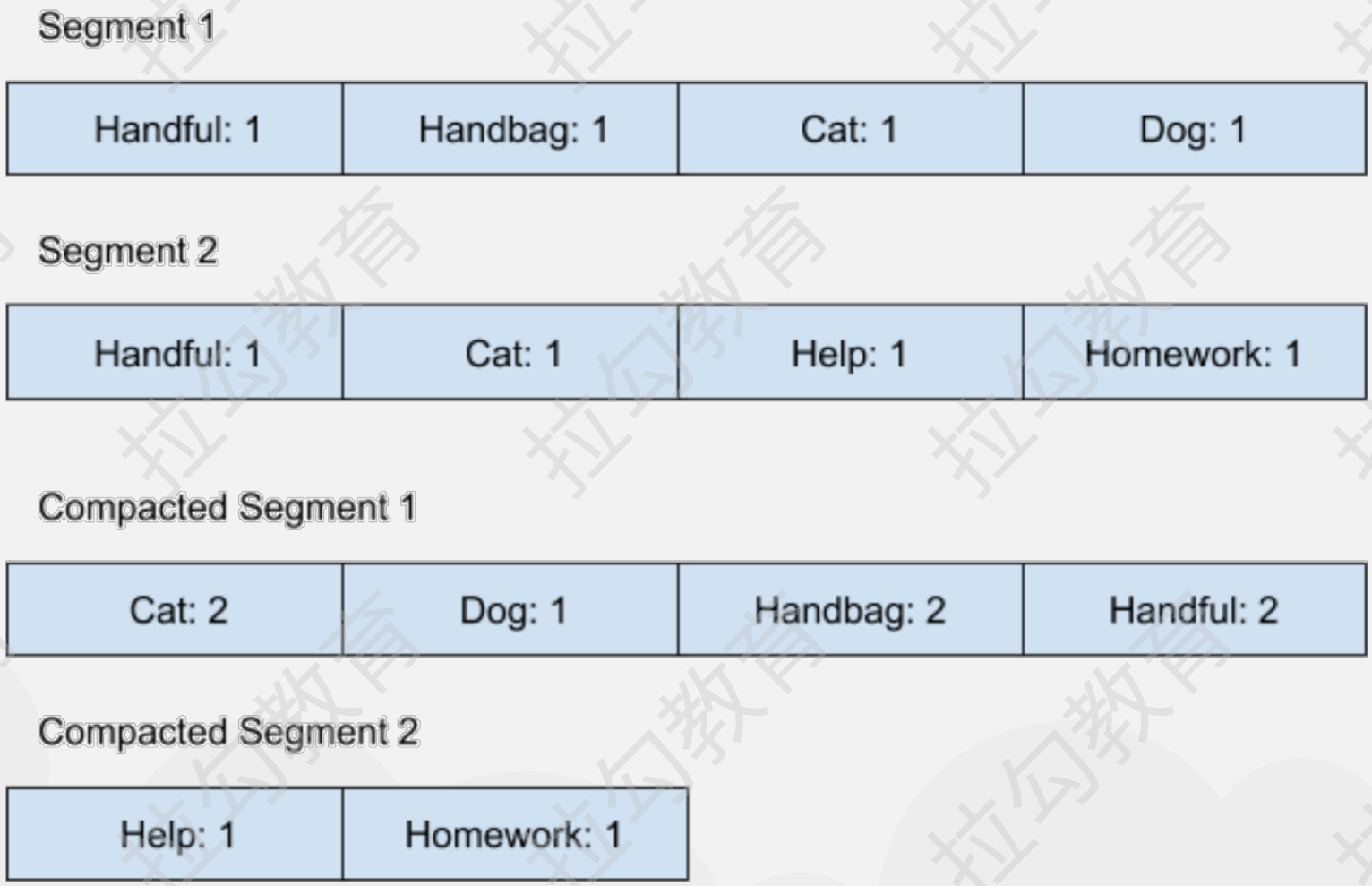
Segment 1

Handful: 1	Handbag: 1	Cat: 1	Dog: 1

Segment 2

Handful: 1	Cat: 1	Help: 1	Homework: 1

SSTable 和 LSM 树



SSTable 和 LSM 树

二叉查找树里的一个特性：

二叉查找树的任意一个节点都比它的左子树所有节点大，同时比右子树所有节点小

在业界上，我们为了维护数据结构读取的高效，一般都会维护一个平衡树

比如，在上一讲中说到的红黑树或者 AVL 树

而这样一个平衡树在 Log-Structured 结构里通常被称为 memtable

上面所讲到的概念，通过内部维护平衡树来进行 Log-Structured 结构的 Compaction 优化

这样一种数据结构被称为是 LSM 树（Log-Structured Merge-Tree）

它是由 Patrick O'Neil 等人在 1996 年所提出的

LSM 树的应用

在数据库里面，有一项功能叫做 Range Query，用于查询在一个下界和上界之间的数据

比如查找时间戳在 A 到 B 之内的所有数据

许多著名的数据库系统，像是 HBase、SQLite 和 MongoDB，它们的底层索引因为采用了 LSM 树

所以可以很快地定位到一个范围

LSM 树的应用

Compacted Segment 1

Cat: 2	Dog: 1	Handbag: 2	Handful: 2

Compacted Segment 2

Help: 1	Homework: 1	Man: 1	Men: 1

Compacted Segment 3

No: 4	Television: 1	Victor: 1	Victory: 1

LSM 树的应用

采用 Lucene 作为后台索引引擎的开源搜索框架，像 ElasticSearch 和 Solr，底层其实运用了 LSM 树

因为搜索引擎的特殊性，有可能会遇到一些情况

那就是：所搜索的词并不在保存的数据里，而想要知道一个数据是否存在 Segment 里面，必须遍历一次整个 Segment，时间开销还并不是最优化的，所以这两个搜索引擎除了采用 LSM 树之外，还会利用 Bloom Filter 数据结构，它可以用来判断一个词是否一定不在保存的数据里面