

第 3 章 HBase 数据结构

3.1 Row Key

与 nosql 数据库们一样,row key 是用来检索记录的主键。访问 HBASE table 中的行, 只有三种方式:

- 1.通过单个 row key 访问
- 2.通过 row key 的 range (正则)
- 3.全表扫描

Row key 行键 (Row key)可以是任意字符串(最大长度 是 64KB, 实际应用中长度一般为 10-100bytes), 在 HBASE 内部, row key 保存为字节数组。存储时, 数据按照 Row key 的字典序(byte order)排序存储。设计 key 时, 要充分排序存储这个特性, 将经常一起读取的行存储放到一起。(位置相关性)

3.2 Columns Family

列族: HBASE 表中的每个列, 都归属于某个列族。列族是表的 schema 的一部分(而列不是), 必须在使用表之前定义。列名都以列族作为前缀。例如 courses:history, courses:math 都属于 courses 这个列族。

3.3 Cell

由{row key, columnFamily, version} 唯一确定的单元。cell 中 的数据是没有类型的, 全部是字节码形式存贮。

关键字: 无类型、字节码

3.4 Time Stamp

HBASE 中通过 rowkey 和 columns 确定的为一个存贮单元称为 cell。每个 cell 都保存 着同一份数据的多个版本。版本通过时间戳来索引。时间戳的类型是 64 位整型。时间戳可以由 HBASE(在数据写入时自动)赋值, 此时时间戳是精确到毫秒 的当前系统时间。时间戳也可以由客户显式赋值。如果应用程序要避免数据版本冲突, 就必须自己生成具有唯一性的时间戳。每个 cell 中, 不同版本的数据按照时间倒序排序, 即最新的数据排在最前面。

为了避免数据存在过多版本造成的的管理 (包括存贮和索引)负担, HBASE 提供 了两种数据版本回收方式。一是保存数据的最后 n 个版本, 二是保存最近一段 时间内的版本(比

如最近七天)。用户可以针对每个列族进行设置。