

Hbase schema 表设计教程整理版

近期在互联网上搜集了大量的关于 Hbase Schema 设计相关的教程, 现综合在一起, 希望给向我一样在摸索 hbase Schema 设计的人一些启发。Hbase、Nutch、Hadoop、Zookeeper 方面相关的讨论希望大家跟我交流, beyiwork@gmail.com。

声明: 此文章中涉及的资料部分来源于网上, 本自由之精神传播于大众, 若有侵权, 请及时联系我, 即刻删除。

第一部分: Hbase Schema design 总述

HBase 与 RDBMS 的区别在于: HBase 的元 (Cell, 可理解为每条数据记录中的数据项) 是具有版本描述的 (versioned), 行是有序的, 列 (qualifier) 在所属列簇 (Column families) 存在的情况下, 由客户端自由添加。以下的几个因素是 Hbase Schema 设计需要考虑的问题:

1. Joins

Hbase 中没有 joins 的概念, 但是, 大表的结构可以使得不需要 joins 的存在而解决这一问题, 你要考虑的是, 一条行记录, 加上一个特定的行关键字, 实现把所有关于 joins 的数据并在一起。

2. Row keys

对你的 Rowkey 要做一番思考, 它非常重要。以存储天气数据为例, 复合的 Rowkey 由监测站 (station) 作为前缀 (方便把某监测站的天气数据聚在一起), 倒置的时间串作为 Rowkey 的后缀可以使温度等天气数据从新到旧排列。

如果你的 Rowkey 是整型的, 用二进制的方式应该比用 String 来存储一个数据更节约空间。

第二部分：Case Study

Case1.Locations

#1.数据需求

- ◆ China
 - Beijing
 - Shanghai
 - Guangdong
 - Shangdong
 - jinan
 - Qingdao
 - Sichuan
 - chengdu

#2. In RDBMS

Loc_id PK	Loc_name	Parent_id	Child_id
1	China		2,3,4,5
2	Beijing	1	
3	Shanghai	1	
4	Guangdong	1	
5	Shangdong	1	7,8
6	Sichuan	1	9
7	Jinan	1,5	

8	Qingdao	1,5	
9	Chengdu	1,6	

#3.In Hbase

row	Column families		
<loc_id>	Name:	Parent	child
		parent:<loc_id>	child:<loc_id>
1	China		child:1=state child:2=state child:3=state child:4=state child:5=state child:6=state
3	Shangdong	parent:1=nation	child:7=city child:8=city
8	Qingdao	parent:1=nation parent:5=state	

Case 2:student-course

#1. 数据需求说明

● Student

1 S ~ many C

● Course

1 C ~ many S

#2. RDBMS

Students表
id PK
name
sex
age

SCs表
student_id
course_id
type

Courses表
id PK
title
introduction
teacher_id

#2. Hbase 表设计

row	Column families	
	info:	course:
<student_id>	info:name info:sex info:age	course:<course_id>=type

row	Column families	
	info:	student:
<course_id>	info:title info:introduction info:teacher_id	student:<student_id>=type

Case 3: user-action

#1.数据需求说明

- users performs actions now and then
 - store every events
 - query recent events of a user

#2.RDBMS表设计

Actions表
id PK
user_id IDX

name
time

说明: SELECT id, user_id, name, time FROM Action WHERE user_id=XXX ORDER
BY time DESC LIMIT 10 OFFSET 20

加速查询需要对user_id IDX建立索引,但是建立索引的代价是由于索引的重
建,插入速度减慢。

#3.Hbase表设计

Row	Column Families
	Name:
<user><Long.MAX_VALUE - System.currentTimeMillis()> <event id>	

Case4. User-friends表

#1.数据需求描述

- ◆ 1 user has 1+ friends
- ◆ will lookup all friends of a user

#2.in RDBMS

Users表
id IDX
name

sex
age

Friendships表
user_id IDX
friend_id
type

#3.In HBase

row	column	
	info:	friend:
<user_id>	info:name info:sex info:age	friend:<user_id>=type

- ◆ actually, it is a graph can be represented by a sparse matrix.

- ◆ then you can use M/R to find sth interesting.

e.g. the shortest path from user A to user B.

Case5. Access log

#1.数据需求描述

- ◆ each log line contains time, ip, domain, url, referer, rowser_cookie, login_id, etc
- ◆ will be analyzed every 5 minutes, every hour, daily, weekly, and monthly

#2.in RDBMS

Accesslog
time
ip IDX
domain
url
referer
browser_cookie IDX
login_id IDX

#3.in Hbase

row	column	
	http:	user
<time><INC_COUNTER>	http:ip http:domain http:url http:referer	user:browser_ cookie user:login_id

整理 by beyiwork

Isiteam