

HBase schema design case studies

Organized by Evan/Qingyan Liu
qingyan123 (AT) gmail.com
2009.7.13

The Tao is ...

De-normalization

Case 1: locations

- China
 - Beijing
 - Shanghai
 - Guangzhou
 - Shandong
 - Jinan
 - Qingdao
 - Sichuan
 - Chengdu

In RDBMS

loc_id PK	loc_name	parent_id	child_id
1	China		2,3,4,5
2	Beijing	1	
3	Shanghai	1	
4	Guangzhou	1	
5	Shandong	1	7,8
6	Sichuan	1	9
7	Jinan	1,5	
8	Qingdao	1,5	
9	Chengdu	1,6	

In HBase

row		column families	
	name:	parent:	child:
<loc_id>		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
5	Shangdong	parent:1=nation	child:7=city child:8=city
8	Qingdao	parent:1=nation parent:5=state	

Case 2: student-course

- Student
 - 1 S ~ many C
- Course
 - 1 C ~ many S

In RDBMS

Students

id PK

name

sex

age

SCs

student_id

course_id

type

Courses

id PK

title

introduction

teacher_id

In HBase

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

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

In RDBMS

Actions
id PK
user_id IDX
name
time

- For fast `SELECT id, user_id, name, time FROM Action WHERE user_id=XXX ORDER BY time DESC LIMIT 10 OFFSET 20`, we must create index on `user_id`. However, indices will greatly decrease insert speed for index-rebuild.

In HBase

row	column families
	name:
<user><Long.MAX_VALUE - System.currentTimeMillis()> <event id>	

Case 4: user-friends

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

In RDBMS

Users
id IDX
name
sex
age

Friendships
user_id IDX
friend_id
type

- `SELECT * FROM friendships WHERE user_id='XXX';`

In HBase

row	column families	
	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.

Case 5: access log

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

In RDBMS

Accesslog
time
ip IDX
domain
url
referrer
browser_cookie IDX
login_id IDX

In HBase

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

INC_COUNTER is used to distinguish the adjacent same time values.