Mysql 迁移数据

1. 迁移表格
   * 1. Udcweb.organization;
     2. Udcweb.Entity
     3. Udcweb.node
     4. Udcweb.sensor
     5. Udcweb.historydata

2. 迁移要求

# 表一：Entity

合并Udcweb.organization，Udcweb.Entity => Entity

1. **Rowkey:**

ID+time

ID从1自增；预留8位；

1. **Column family**

F

1. **Column**

timestamp统一为importtime\*1000，转化成毫秒)

Name

每次操作（import/update/delete）产生一条记录

import: 值为importuserid，时间戳为importtime\*1000，value=userid

update(if exist): 值为updateuserid, 时间戳为updatetime\*1000，value=userid

delete(if exist): 值为deleteuserid 时间戳为deletetime\*1000，value=userid

多个值，用不同版本表示

Description

Memo

Fast

Isfrozen

Left

Right

Level

Pre-order 算法，计算 left , right, level

ParentID: 保留org表中的parentid；Entity的parentId为orgid

OE=ORG|ENT; 标注是org还是entity

PreID: 在mysql中的ID，用于对应旧表

# 表二: udcweb.node => Node

1. Rowkey

Mac

1. Column family

F

1. Column

(timestamp统一为importtime\*1000)

ID: previouse id as older version

Name

Type

Manufacture

SensorsReportInterval

Model

Username

Password

Protocol

Runmode

Position

Memo

Status

import: 值为importuserid，时间戳为importtime\*1000，value=userid

update(if exist): 值为updateuserid, 时间戳为updatetime\*1000，value=userid

delete(if exist): 值为deleteuserid 时间戳为deletetime\*1000，value=userid

多个值，用不同版本表示

Fast

OfflineTime

LastReportTime

没有entityid，如何根据mac查找entity？

Mac->nodeid+(sensorNum/channelID)->entityid

# 表三：udcweb.sensor=> sensor

1. Rowkey

NodeID+Num

= udcweb.node.nodeid(8位)+udcweb.sensor.num(3位);

1. Column family

F

1. Column:

Name

Type

Lowerlimit

Upperlimit

Threshold

Entity

EntityID,用于查询sensor所属entity

历史entity放在旧版本

Description

fast

# 表四：udcweb.historydata=> historydata

1. Rowkey:

EntityID+time

udcweb.historydata.Entityid (8位)+udcweb.historydata.time(10位)

1. Column Family

F

3. Colunm

nodeid：

verson1(sensor1.data),时间戳为sensorNum

verson2(sensor2.data),

verson3(sensor3.data),

# 表五：HBase: entitynode

用于描述entity和node 的关系

1. Rowkey

Udcweb.node.Entityid+ Udcweb.node.nodeid(entityid保留8位，nodeid保留8位)

1. Column Family

F

1. Column

mac:

import: 值为importuserid，时间戳为importtime\*1000，value=userid

update(if exist): 值为updateuserid, 时间戳为updatetime\*1000，value=userid

delete(if exist): 值为deleteuserid 时间戳为deletetime\*1000，value=userid

多个值，用不同版本表示

# 表六：nodeentity – generate this table automatically

1. Rowkey

Udcweb.node.nodeid(mac)+udcweb.node.entityid

1. Column Family

F

1. Column

Id (current version is current ID, previous ids are store in older versions)

import: 值为importuserid，时间戳为importtime\*1000，value=userid

update(if exist): 值为updateuserid, 时间戳为updatetime\*1000，value=userid

delete(if exist): 值为deleteuserid 时间戳为deletetime\*1000，value=userid

sensors: timestamp is importtime\*1000, value=1:2:4 for sensors 1, 2,4

# 表7：badindex–关于超过阈值的异常数据

1. Rowkey

Entityid+first\_errtime (entityid保留8位，nodeid保留8位)

1. Colunm Family

F

1. Column

Err:

verson1(ent\_errtime),时间戳为sensorNum

verson2(ent\_errtime),时间戳为sensorNum