# Hbase 日常工作文档

# 一：修改hbase表列族压缩格式

为了缓解磁盘空间的压力，需要将未压缩的表进行压缩，压缩前服务器需要安装snappy包，直接执行yum install snappy 即可，具体压缩单个表的例子如下：

1.disable表，在hbase shell 中执行

disable 'Your Hbase table'

2.修改hbase表列族的compression值，在hbase shell 中执行

alter  'Your Hbase table',NAME => 'f', COMPRESSION => 'snappy'

#解释：上面的脚本将压缩算法修改为snappy

3.enable表，在hbase shell 中执行

enable 'Your Hbase table'

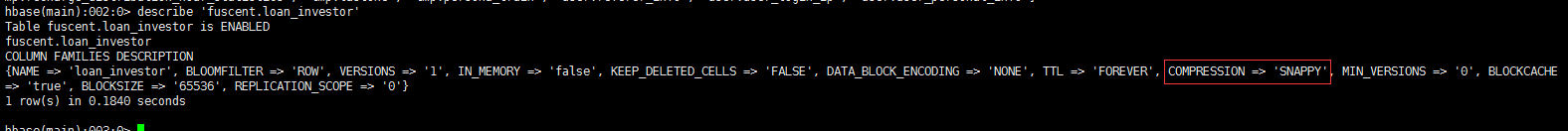
4.enable 表后，Hbase表的压缩格式并没有生效，还需要一个动作，即Hbase major\_compact,在hbase shell中执行

major\_compact 'Your Hbase table'

#该动作耗时较长，会对服务有很大影响，应该选择在一个服务不忙的时间来做。

5.验证，hbase shell中执行

describe  'Your Hbase table' 显示如下表示hbase表压缩格式修改完毕



# 二、同步一张表数据

同步数据实现方式为：从hadoop同步从库上拉存量数据，而后通过yooli-hadoop-sync-0.0.5.jar 同步后续增量数据。

1.首先停掉hadoop从库同步

stop salve；

2.在hbase里建相应表，在hbase shell内：

举个栗子

create 'autoplms.plms\_loan\_car', 'plms\_loan\_car '

3.生成sqoop脚本

sqoop import --connect 'jdbc:mysql://192.168.5.25:3307/autoplms?tinyInt1isBit=false' --username sqoop --password '\*\*\*\*\*' --hbase-table autoplms.plms\_loan\_car --column-family plms\_loan\_car --hbase-row-key rowkey --num-mappers 7 --split-by car\_id --target-dir /data/hbase/autoplms/plms\_loan\_car --hbase-bulkload --query 'select lpad(car\_id,12,"0") as rowkey,car\_id,contract\_id,plate\_numbers,frame\_number,engine\_number,registration\_num,register\_date,factory\_date,buy\_car\_date,brand\_code,brand\_name,style\_code,style\_name,storage\_date,storage\_personnel,outbound\_date,outbound\_personnel,create\_time,last\_update\_time,create\_user\_id,last\_update\_user\_id,remark from autoplms.plms\_loan\_car where $CONDITIONS'

--connect sqoop 需要连接hadoop同步从库，格式为： 'jdbc:mysql://从库IP:port/同步表所在库?tinyInt1isBit=false'

--username：sqoop（需要在从库上建立该账号）

--password：\*\*\*\*\*

--hbase-table：都是库名.表名，如上：autoplms.plms\_loan\_car

--column-family：列族名，可以理解为mysql的表名

--hbase-row-key：rowkey

--num-mappers：7

--split-by：mysql库中该表的pk列，如上为car\_id

--target-dir：该表物理文件存放路径

--hbase-bulkload

--query 'select lpad(car\_id,12,"0") as....

括弧内第一列：mysql库中该表pk列

第二列：hbase主键列的宽度，在此定义为12。

第三列：不足12位的用0补齐

as后，第一列固定的，剩下的就是原mysql该表各个字段。

4.执行完sqoop脚本之后，就可以等待存量数据同步完，然后启从库同步，同步程序会自动同步增量数据。

另外：如果用召静脚本执行，就是如下步骤了：

1.停同步

2.准备sqoop\_mysql2hbase\_print.py和generate\_selectforhbase.py、create\_hbase\_table\_new.py脚本内需要打开的文件。

解释一下：

generate\_selectforhbase.py：该脚本生成mysql下该表所有列。

脚本内容如下：

 Expand source

[root@ser4-61 importtask]# cat generate\_selectforhbase.py

\_\_author\_\_ = 'Administrator'

import MySQLdb

def get\_table\_list():

f = open("/root/importtask/enterprise\_info.txt")

table\_list = f.readlines()

tb\_list = []

f.close()

for l in table\_list:

tb\_list.append(l.split(' ')[0]+'.'+l.split(' ')[1])

return tb\_list

def generate\_selectforhbase(table\_name):

ip\_address = "192.168.1.40"

db\_host = ip\_address

db\_port = 3307

db\_user = 'zhaojing.peng'

db\_password = '\*\*\*\*\*\*'

conn = MySQLdb.connect(host=db\_host, port=db\_port, user=db\_user, passwd=db\_password, charset='utf8')

cur = conn.cursor()

sql = 'show columns from ' + table\_name

cur.execute(sql)

select\_str = ''

for data in cur.fetchall():

select\_str = select\_str + data[0] + ','

select\_str = select\_str[0:-1] + ' from ' + table\_name +' where $CONDITIONS' + "'"

select\_file\_name = "/root/importtask/mysql\_select/"+table\_name.split('.')[1]+'.txt'

f = open(select\_file\_name, "w")

f.write(select\_str)

f.close()

if \_\_name\_\_ == '\_\_main\_\_':

# generate\_selectforhbase()

table\_list=get\_table\_list()

for table\_name in table\_list:

generate\_selectforhbase(table\_name)

create\_hbase\_table\_new.py：生成hbase建表语句。

脚本如下：

 Expand source

[root@ser4-62 importtask]# cat create\_hbase\_table\_new.py

#!/usr/bin/python

import os

import commands

class mysql2hbase:

def check\_hbase\_etc\_result(self):

f = open("/root/importtask/jianwei.txt")

table\_list = f.readlines()

f.close()

dict\_hbase\_table = {}

for list in table\_list:

line = list.replace('\n', '').split(' ')

if line[4] in dict\_hbase\_table.keys():

dict\_hbase\_table[line[4]].append(line[5])

else:

dict\_hbase\_table[line[4]] = [line[5]]

print dict\_hbase\_table

return dict\_hbase\_table

def hbase\_create\_table(self, table\_dict):

sql = ""

for key, value in table\_dict.items():

sql = sql + " create " + "'" + str(key) + "'"

for i in value:

sql = sql + ", '" + str(i.replace('\r','')) + "'"

sql = sql + "\n"

sql = sql + "exit\n"

f = open("/root/importtask/create\_hbase\_table", "w")

f.write(sql)

f.close()

#status, info = commands.getstatusoutput("hbase shell " + "/root/importtask/create\_hbase\_table")

#print status, info

def hive\_create\_table(self):

create\_dir = '/root/importtask/hive2hbase\_hql'

hql\_list = os.listdir(create\_dir)

for i in hql\_list:

cmd = "hive -f " + create\_dir + "/" + i

print cmd

status,info = commands.getstatusoutput(cmd)

print status,info

if \_\_name\_\_ == '\_\_main\_\_':

myhbase = mysql2hbase()

table\_list = myhbase.check\_hbase\_etc\_result()

myhbase.hbase\_create\_table(table\_list)

#myhbase.hive\_create\_table()

sqoop\_mysql2hbase\_print.py：生成sqoop语句。

脚本如下：

 Expand source

[root@ser4-62 importtask]# cat sqoop\_mysql2hbase\_print.py

#!/usr/bin/python

import commands

class mysql2hbase:

def sqoop\_mysql2hbase(self):

mysql\_host = "192.168.5.25"

mysql\_port = "3307"

mysql\_user = "sqoop"

mysql\_password = "\*\*\*\*\*\*\*"

f = open("/root/importtask/auto.txt")

table\_list = f.readlines()

f.close()

for line in table\_list:

sqoop\_cmd = ""

list = line.replace("\n", "").split(" ")

f = open("/root/importtask/mysql\_select/" + list[1] + ".txt")

select\_sql = f.readlines()[0].replace("\n", "")

sqoop\_cmd = "sqoop import --connect 'jdbc:mysql://" + mysql\_host + ":" + mysql\_port + "/" + \

list[0] + "?tinyInt1isBit=false' --username " + mysql\_user + " --password '" + mysql\_password + \

"' --hbase-table " + list[4] + " --column-family " + list[5].replace("\r","") + " --hbase-row-key rowkey --num-mappers 7 --split-by " + \

list[2] + " --target-dir /data/hbase/" + list[0] + "/" + list[1] + " --hbase-bulkload --query '" + "select lpad(" + \

list[2] + ',' + list[3] +","+'"'+"0" + '"' + ") as rowkey," + select\_sql

print sqoop\_cmd

#status,info = commands.getstatusoutput(sqoop\_cmd)

#print status,info

if \_\_name\_\_ == '\_\_main\_\_':

myhbase = mysql2hbase()

myhbase.sqoop\_mysql2hbase()

3.运行create\_hbase\_table\_new.py，生成hbase建表语句，并在hbase shell内执行

4.运行generate\_selectforhbase.py，生成mysql下该表所有列。

5.运行sqoop\_mysql2hbase\_print.py，执行sqoop脚本

6.启从库同步。