TRANSWARP
星 环 科 技

Hyperbase的使用

陈晓勇
Xiaoyong.chen@transwarp.io
13801795827
星环科技
www.transwarp.io

Hyperbase用途

TRANSWARP 星环科技

- Hyperbase是一个提供高并发，大容量查询为主的数据库平台，Hyperbase中的表概念和关系型数据库的二维表不同。是一个“稀疏的，分布式的，持久的，多维度有序map”，以下是Hyperbase概念。
  - 表（Table）：Hyperbase以表为单位组织数据。表名的数据类型为string。
  - 行（Row）：表中数据以行存储。每行数据都有一个独特的RowKey。表中各行数据按RowKey排序。Row key没有数据类型，以byte[]（字节数组）存储。
  - 列族（Column Family）：行中数据以列族分组。各行数据拥有的列族必须相同。但是并 不是每个列族中都需要有数据。列族名的数据类型为string。
  - 列限定符（Column Qualifier）：列族中可以有一列或者多列数据。各列根据列限定符识 别。各行的拥有的列不一定需要相同。列名没有数据类型，以byte[]存储。
  - 单元格（Cell）：行、列族和列限定符的组合指向独特的单元格。单元格中存放的数据成为 单元格的值。单元格的值没有数据类型，以byte[]存储。
  - 时间戳（Timestamp）：单元格的值可以有不同版本。各个版本由版本号区分。默认版本 号为单元格值被写入时的时间戳。

Hyperbase的操作

TRANSWARP 星环科技

- Hyperbase通过hbase shell方式建表，其使用同传统数据库有很大不同。
  - 创建/显示/删除/修改表：CREATE/LIST/DISABLE/ENABLE/DROP/
  - 填入数据/扫描数据/删除数据：PUT/SCAN/DELETE/TRUNCATE/DROP TABLE
  - 添加一个列族： alter ‘table_name’, {NAME=> ‘column_family_name,...}
  - 描述表和数据库：DESCRIBE
  - 建立Hyperbase索引：add_index/rebuild_index
  - Inceptor over Hyperbase操作

基础语句：STATUS/VERSION

TRANSWARP 星环科技

- 举例

```
hbase(main):001:0> status
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
4 servers, 0 dead, 4.0000 average load
```

```
hbase(main):002:0> version
0.98.6-transwarp, r, Tue Oct 13 06:46:40 EDT 2015
```

基础建表语句：CREATE TABLE

TRANSWARP 星环科技

- 语法

```
hbase(main):011:0>create ‘Tables_name',’table_items1', ’table_items2', ’table_items3',…
```

- 举例

```
hbase(main):011:0>create ‘stock_hbase1',’F01', ’F02', ’F03', ’F04’,’F05’, ’F06’, ’F07’, ’F08’, ’F11’, ’F13’, ’F88’, ’F99’
```

基础表使用语句：list

TRANSWARP 星环科技

- 获取表的列表语法

```
hbase(main):011:0>list
```

- 举例

```
hbase(main):001:0> list
TABLE
hbase_stock_date
stock_hbase
stock_hbase2
stock_hbase2_index_stock_symbol
stock_hbase_index_date
stock_hbase_pjj_index
stock_hbase_test
table001
8 row(s) in 2.0090 seconds

=> ["hbase_stock_date", "stock_hbase", "stock_hbase2", "stock_hbase2_index_stock_symbol",
"stock_hbase_index_date", "stock_hbase_pjj_index", "stock_hbase_test", "table001"]
hbase(main):002:0>
```

基础表使用语句：describe

TRANSWARP 星环科技

- 获取表的列表语法

```
hbase(main):011:0>describe ‘table_name’
```

- 举例

```
hbase(main):004:0> describe 'stock_hbase2'
DESCRIPTION                                                ENABLED
 'stock_hbase2', {NAME => 'cf1', DATA_BLOCK_ENCODING => 'PREFIX', B true
 LOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', CO
 MPRESSION => 'SNAPPY', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP
 _DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'fal
 se', BLOCKCACHE => 'true'}, {NAME => 'cf2', DATA_BLOCK_ENCODING =>
  'PREFIX', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSION
 S => '1', COMPRESSION => 'SNAPPY', MIN_VERSIONS => '0', TTL => 'FO
 REVER', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_ME
 MORY => 'false', BLOCKCACHE => 'true'}, {NAME => 'cf3', DATA_BLOCK
 _ENCODING => 'PREFIX', BLOOMFILTER => 'ROW', REPLICATION_SCOPE =>
 '0', VERSIONS => '1', COMPRESSION => 'SNAPPY', MIN_VERSIONS => '0'
 , TTL => 'FOREVER', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '6
 5536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}, {NAME => 'cf4'
 , DATA_BLOCK_ENCODING => 'PREFIX', BLOOMFILTER => 'ROW', REPLICATI
 ON_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'SNAPPY', MIN_VER
 SIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'false', BLO
 CKSIZE => '65536', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
```

基础表统计：count

TRANSWARP 星环科技

- 获取表的列表语法

```
hbase(main):011:0>count count '<tablename>', CACHE => 1000
```

- 举例

```
hbase(main):004:0> count 'stock_hbase1’

1 row(s) in 0.1600 seconds
```

基础表使用语句：disable, enable, alter

TRANSWARP 星环科技

- 删除一个列族，alter，disable，enable语法
- 举例

```
hbase(main):004:0>disable ’stock_hbase'
0 row(s) in 2.0390seconds
hbase(main):005:0>alter’stock_hbase',{NAME=>’F01',METHOD=>'delete'}
0 row(s) in 0.0560seconds
该列族已经删除，我们继续将表enable
hbase(main):008:0> enable ‘stock_hbase'
0 row(s) in 0.0420seconds
```

基础表使用语句：Drop

TRANSWARP 星环科技

- 删除一个列族，alter，disable，enable语法
- 举例

```
hbase(main):004:0>disable ’stock_hbase'
0 row(s) in 2.0390seconds
hbase(main):005:0>drop ’stock_hbase’
0 row(s) in 0.0560seconds
//查询表是否存在
hbase(main):006:0>exists ’stock_hbase’
Table stock_hbase doesnot exist
0 row(s) in 0.0410 seconds
```

插入/删除记录语句：PUT

TRANSWARP 星环科技

- 插入几条记录语法: PUT, DELETE
- 举例

```
hbase(main):004:0> put 'stock_hbase','7268337','cf1:F01','800001'
0 row(s) in 0.0380 seconds


delete 命令
删除表“stock_hbase”中行为“7268337”, 列族“cf1:F01”中的“'800001”。
 hbase(main):001:0> delete 'stock_hbase', '7268337', 'cf1:F01'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
0 row(s) in 0.3900 seconds
```

查看表数据语句：GET/SCAN

TRANSWARP 星环科技

- 插入几条记录语法: GET, SCAN
- 举例

```
hbase(main):006:0> get 'stock_hbase', '600030'
COLUMN                  CELL
 cf1:F02                timestamp=1449559651834, value=\xE4\xB8\xAD\xE4\xBF\xA1\xE8\xAF\x81\xE5\x88
                        \xB8
 cf1:F99                timestamp=1449559651834, value=2015-12-04
 cf2:F03                timestamp=1449559651834, value=18.74
 cf2:F04                timestamp=1449559651834, value=18.52
 cf2:F05                timestamp=1449559651834, value=2.770569479E9
 cf3:F06                timestamp=1449559651834, value=18.62
 cf3:F07                timestamp=1449559651834, value=18.16
 cf3:F08                timestamp=1449559651834, value=18.24
 cf3:F11                timestamp=1449559651834, value=1.50895837E8
 cf3:F13                timestamp=1449559651834, value=18.24
 cf4:F88                timestamp=1449559651834, value=SH
11 row(s) in 0.0310 seconds

scan ‘table_name',{COLUMNS =>’cf1:keyrow',LIMIT =>10, STARTROW => ’start_row', STOPROW=>’end_row'}
```

事实上，inceptor的表达能力远远强于hbase shell，所以一般都使用inceptor中的insert命令

条件查询表数据语句：Filter

TRANSWARP 星环科技

- 获取表的列表语法

```
hbase(main):011:0>scan
'tablename',STARTROW=>'start',COLUMNS=>['family:qualifier'],FILTER=>SingleColumnValueFilter.new(Bytes.to
Bytes('family'),Bytes.toBytes('qualifier'),CompareFilter::CompareOp.valueOf('EQUAL'),SubstringComparator.new(
'value')),LIMIT=>1
```

- 举例

```
hbase(main):008:0> scan 'stock_hbase1', { COLUMNS => 'cf1:F01', LIMIT=>5,FILTER
=>SingleColumnValueFilter.new(Bytes.toBytes('cf1'),Bytes.toBytes('F01'),CompareFilter::CompareOp.valueOf('E
QUAL'),SubstringComparator.new('600030'))}
ROW                COLUMN+CELL
 3828891           column=cf1:F01, timestamp=1107694454448, value=600030
 3828892           column=cf1:F01, timestamp=1107694454448, value=600030
 3828893           column=cf1:F01, timestamp=1107694454448, value=600030
 3828894           column=cf1:F01, timestamp=1107694454448, value=600030
 3828895           column=cf1:F01, timestamp=1107694454448, value=600030
5 row(s) in 8.2450 seconds
```

Hbase shell总结

TRANSWARP 星环科技

	Hbase Shell 命令	描述
1	version	查询Hbase版本信息
2	status	Hbase集群的状态
3	list	列出Hbase中的所有表（包括索引）
4	create	创建表命令
5	count	count统计表中的行数
6	describe	显示表中的详细信息，诸如column family等
7	alter	提示Hbase修改column family的模式
8	delete	删除制定的表对象(表，行，列等)
9	disable	提示Hbase该表无效
10	Drop	删除表
11	enable	提示Hbase该表有效
12	put	向Hbase的指定表单元插入数据
13	exists	查询表是否在Hbase中存在
14	deleteall	删除指定行的所有数据

Hbase shell总结

TRANSWARP 星环科技

	Hbase Shell命令	描述
15	get	获取行或cell的值
16	scan	扫描指定表获取对应的值
17	truncate	重新创建指定表
18	incr	增加指定表，行或列的cell值
19	shutdown	关闭Hbase集群
20	exit	退出Hbase shell

Hyperbase的问题

TRANSWARP 星环科技

- 事实上，inceptor的表达能力远远强于hbase shell，所以一般都使用inceptor中的insert命令
- 与HBase表查询相关的操作，均在inceptor shell中。
- 所有有关inceptor表的查询操作，均可使用在HBase表中

通过Inceptor对Hyperbase进行建表

TRANSWARP 星环科技

- 要在Inceptor中对Hyperbase表进行交互式查询，要先在Inceptor中建一张外表，然后将Hyperbase表通过映射建立和一张二维表的对应关系。映射时，只有最新版本的单元格值会被保存。
- 语法：

```
CREATE EXTERNAL TABLE table_name (row_key_column data_type,
column_name_1 data_type, column_name_2 data_type, ...)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler|
io.transwarp.hyperbase.HyperbaseStorageHandler' WITH SERDEPROPERTIES
("hbase.columns.mapping" = ":row_key_column, column_family:column_qualifer_1,
column_family:column_qualifier_2, ...") TBLPROPERTIES ("hbase.table.name" =
"hbase_table_name")
```

- 示例：

```
transwarp> create external table stock_hbase(
         > F01 string, F02 string, F03 double, F04 double, F05 string, F06 double,
         > F07 double, F08 double, F11 string, F13 double, F88 string, F99 date)
         > row format delimited fields terminated by ","
         > stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
         > with
serdeproperties('hbase.columns.mapping'=':key,cf1:F02,cf2:F03,cf2:F04,cf2:F05,cf3:F06,cf3:F
07,cf3:F08,cf3:F11,cf3:F13,cf4:F88,cf1:F99')
         > tblproperties('hbase.table.name'='stock_hbase');
Time taken: 4.022 seconds
```

通过Inceptor对Hyperbase进行插入操作

TRANSWARP 星环科技

- 在Inceptor中，我们可以对一张映射表做所有除了分区和分桶外所有的InceptorSQL操作。
- 插入语法：

```
INSERT INTO table_name (column_name1, column_name2, ...) VALUES (value1, value2, ...)
```

- 示例：

```
transwarp> insert into table stock_hbase select
F01,F02,F03,F04,F05,F06,F07,F08,F11,F13,F88,tdh_todate(F99) from daily_stock;
7268337 rows affected.
Time taken: 76.906 seconds
[tdh-1:10000] transwarp> select count(*) from stock_hbase;
9350
Time taken: 2.889 seconds
```

通过Inceptor对Hyperbase进行更新操作

TRANSWARP 星环科技

- 使用UPDATE语句可以在Inceptor Shell里对Hyperbase表进行更新。
- 更新语法：

```
UPDATE table_name SET column_name = value, column_name = value, ... WHERE
filter_condition;
```

- 示例：

```
[localhost:10000] transwarp>INSERT INTO stock_hbase2 (F01,F02,F03) VALUES('7268338','
600001','10.02');
1 rows affected.
Time taken: 1.01 seconds
[localhost:10000] transwarp> select * from stock_hbase2 where F02=' 600001';
7268338        600001        10.02          NULL          NULL          NULL          NULL
               NULL          NULL           NULL          NULL          NULL          NULL
Time taken: 1.848 seconds
[localhost:10000] transwarp>UPDATE stock_hbase2 SET F02 = ' 600002' where F02 = '
600001';
1 rows updated.
Time taken: 1.85 seconds
```

通过Inceptor对Hyperbase进行删除操作

TRANSWARP 星环科技

- 删除语法：

```
DELETE FROM table_name WHERE filter_condition
```

- 示例：

```
[localhost:10000] transwarp> delete from stock_hbase2 where F02 = ' 600002';
1 rows deleted.
Time taken: 1.882 seconds
[localhost:10000] transwarp> select * from stock_hbase2 where F02=' 600002';
Time taken: 1.542 seconds
```

通过Inceptor对Hyperbase进行JOIN操作

TRANSWARP 星环科技

- 删除语法：

```
SELECT COLUMNS_NAME FROM TABLE_NAME1 JOIN TABLE_NAME2 ON condition1 =
condition 2;
```

- 示例：

```
[localhost:10000] transwarp> select * from stock2 join stock_hbase2 on
stock2.F02=stock_hbase2.F02;
```

Hyperbase的索引

TRANSWARP 星环科技

- Hyperbase支持两种索引：
  - 组合索引|COMBINE_INDEX： COMBINE_INDEX使用一列或者多列生成索引|
  - 结构索引|STRUCT_INDEX： STRUCT_INDEX对STRUCT类型中的一个字段生成索引|
- 这两种索引分别可以是全局的（global）和局部的（local）。全局的索引与与原表独立,以一张表（ 索引表 ）形式存在。局部的索引就在原表中，以一个新的列（ 索引列 ）的形式存在。

语义：

```
COMBINE_INDEX|INDEXED=cf1:cq1:n1|cf1:cq2:n2|...|rowKey:rowKey:m,[UPDATE=true]
```

索引的创建(4.2 or older version)

TRANSWARP 星环科技

- 语句为table_name表添加全局索引，生成一张名为’table_name_index_name’的索引
- 建索引

```
hbase(main):025:0>add_index 'table_name', 'index_name', 'index_definition_clause'
```

- 生成索引

```
hbase(main):025:0>rebuild_index 'table_name','index_name'
```

- 例子：

```
Hbase(mail):004:0>add_index 'stock_hbase2’, 'index_stock_symbol’,
'COMBINE_INDEX|INDEXED=cf1:F02:7|rowKey:rowKey:6’
rebuild_index ’stock_hbase2’,'index_stock_symbol’
hbase(main):004:0> rebuild_index 'stock_hbase2', 'index_stock_symbol’
2016-01-09 17:26:55,069 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
2016-01-09 17:26:59,812 INFO mapreduce.JobSubmitter: number of splits:3
2016-01-09 17:27:00,053 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1452153464656_0003
2016-01-09 17:27:00,380 INFO impl.YarnClientImpl: Submitted application application_1452153464656_0003
2016-01-09 17:27:00,442 INFO mapreduce.Job: The url to track the job: http://TDH1-
1:8088/proxy/application_1452153464656_0003/
2016-01-09 17:27:00,443 INFO mapreduce.Job: Running job: job_1452153464656_0003
```

索引的创建(4.3 or above)

TRANSWARP 星环科技

- 语句为table_name表添加全局索引，生成一张名为’table_name_index_name’的索引
- 建索引

```
hbase(main):025:0>add_index 'table_name', 'index_name', 'index_definition_clause'
```

- 生成索引

```
hbase(main):025:0>rebuild_global_index 'table_name','index_name'
```

- 例子：

```
Hbase(mail):004:0>add_index 'stock_hbase2’, 'index_stock_symbol’,
'COMBINE_INDEX|INDEXED=cf1:F02:7|rowKey:rowKey:6’
hbase(main):004:0> rebuild_global_index 'stock_hbase2', 'index_stock_symbol’
```

比较一下有无索引的区别？

TRANSWARP 星环科技

- 示例

```
[localhost:10000] transwarp> select * from stock_hbase2 where F02=' 600030' limit 10;
3828891    600030    中信证券    4.5    0.0    0.0    0.0    0.0    4.5    150.0
           SH        2002-12-31
3828892    600030    中信证券    4.5    0.0    0.0    0.0    0.0    4.5    150.0
           SH        2003-01-02
3828893    600030    中信证券    4.5    0.0    0.0    0.0    0.0    4.5    150.0
           SH        2003-01-03
3828894    600030    中信证券    4.5    5.53   9.98892578E8   5.58   4.97   5.01   1.94288556EN
           SH        2003-01-06
3828895    600030    中信证券    5.01   4.96   2.9062243E8    5.05   4.82   4.85   5.8977717E7N
           SH        2003-01-07
3828896    600030    中信证券    4.85   4.83   3.68411113E8   5.34   4.81   5.34   7.2064078E7N
           SH        2003-01-08
3828897    600030    中信证券    5.34   5.65   6.36358017E8   5.87   5.61   5.87   1.10086868EN
           SH        2003-01-09
3828898    600030    中信证券    5.87   6.19   1.300495478E9  6.46   6.11   6.21   2.06393422EN
           SH        2003-01-10
3828899    600030    中信证券    6.21   6.3    6.85740331E8   6.49   6.1    6.3    1.09178302EN
           SH        2003-01-13
3828900    600030    中信证券    6.3    6.27   9.77541645E8   6.9    6.18   6.54   1.49481283EN
           SH        2003-01-14
Time taken: 2.216 seconds
```