Apache HBase 快照（Snapshots）介绍

在《HDFS 快照编程指南》文章中，我简单介绍了 HDFS 的快照功能。本文将介绍 HBase 快照功能，因为 HBase 的底层存储是基于 HDFS 的，所以 HBase 的快照功能也是依赖 HDFS 快照的知识。HBase 快照功能是从 HBase 0.95.0 开始引入的，详见 HBASE-50。

HBase 快照功能允许管理员在不复制数据的情况下克隆一张表，同时对 RegionServers 影响很小。HBase snapshots 的使用场景主要包括以下几种情况：

- 对重要表定期进行快照操作，这样可以在出现问题时将影响减小到最少；
- 在重大应用升级或改变之前保存一个快照，这样如果应用升级或改变发生问题，可以快速回滚到操作之前的状态。
- 将数据迁移到其他集群，我们可以创建一个 snapshot，然后使用 ExportSnapshot 工具迁移到另外一个集群，并在另外一个集群进行恢复，这样可以做一些离线分析或数据备份功能等；
- 构建测试环境数据。

什么是Snapshot

对 HDFS 上的文件进行快照时，其实是快照文件记录 block 的列表和文件大小，并不做数据的拷贝。HBase 底层用的也是 HDFS 的快照功能，所以对 HBase 表进行快照也不会发生数据的备份。HBase 快照其实是一系列的 metadata 信息组合，利用这些

信息，管理员可以将一个表恢复到之前的状态，任何在创建快照之后插入的数据都会丢失。

快照操作

和 HDFS 快照一样，HBase 快照也支持一系列的操作。主要有以下几种。

- 创建快照：这个操作尝试在给定表上创建快照。如果表中的 Region 由于发生 balance、split 或 merge 而产生移动的时候可能会导致创建快照失败。
- 克隆快照：这个操作将会使用特定快照中的相同模式（schema）和相同数据来创建一张新表，这个操作执行之后会创建一个全新的表，对这个表进行的任何操作将不会对原表挥着快照有任何影响。
- 恢复快照：这个操作将表的模式和数据还原到快照状态。注意，所有在创建快照之后的修改将会丢失。
- 删除快照：这个操作将从系统中删除快照。释放未共享的磁盘空间，而且不会影响其他克隆或快照。
- 导出快照：这个操作将快照的数据和元数据拷贝到其他集群。这个操作只用到了 HDFS，并不会和 Master 或 RegionServer 发生通讯，所以我们可以直接把 HBase 集群关闭。

快照使用

在进行下面操作之前，确保你集群 hbase.snapshot.enabled 参数设置为 true。为了方便下面的操作，我们先创建一张 HBase 表，并往里面插入几条测试数据，如下：

```
hbase(main):010:0> create 'iteblog_table', 'cf'
Took 0.1013 seconds
hbase(main):010:0> put 'iteblog_table','1', 'cf:name','wyp'
Took 0.2110 seconds
hbase(main):012:0> put 'iteblog_table','2', 'cf:name','iteblog'
Took 0.0083 seconds
hbase(main):013:0> put 'iteblog_table','3', 'cf:name','iteblog_hadoop'
Took 0.0066 seconds
hbase(main):014:0> scan 'iteblog_table'
ROW             COLUMN+CELL
 1              column=cf:name, timestamp=1547646979906, value=wyp
 2              column=cf:name, timestamp=1547646995691, value=iteblog
 3              column=cf:name, timestamp=1547647005389, value=iteblog_hadoop
3 row(s)
Took 0.0346 seconds
```

我们创建了名为 iteblog_table 的表，并插入了几条数据。

创建快照

创建快照可以使用 snapshot 命令，具体如下：

```
hbase(main):016:0> snapshot 'iteblog_table', 'snap_iteblog_table'
Took 1.4361 seconds
```

显示快照

使用 list_snapshots 命令可以显示系统里面的所有快照， 具体如下：

```
hbase(main):017:0> list_snapshots
SNAPSHOT                                    TABLE + CREATION TIME
 snap_iteblog_table                           iteblog_table (2019-01-16 21:57:31 +0800)
1 row(s)
Took 0.0245 seconds
=> ["snap_iteblog_table"]
```

克隆快照

正如前面说的，这个操作会使用快照里面的信息创造一个全新的表，我们对这个新表的操作不会影响旧的表。

```
hbase(main):018:0> clone_snapshot 'snap_iteblog_table', 'iteblog_table_v1'
Took 2.4177 seconds
hbase(main):019:0> scan 'iteblog_table_v1'
ROW                 COLUMN+CELL
 1                  column=cf:name, timestamp=1547646979906, value=wyp
 2                  column=cf:name, timestamp=1547646995691, value=iteblog
 3                  column=cf:name, timestamp=1547647005389, value=iteblog_hadoop
3 row(s)
Took 0.0784 seconds
hbase(main):020:0> put 'iteblog_table_v1','4', 'cf:name','iteblog.com'
Took 0.0121 seconds
hbase(main):021:0> scan 'iteblog_table_v1'
ROW                 COLUMN+CELL
 1                  column=cf:name, timestamp=1547646979906, value=wyp
 2                  column=cf:name, timestamp=1547646995691, value=iteblog
```

```
 3                column=cf:name, timestamp=1547647005389, value=iteblog_hadoop
 4                column=cf:name, timestamp=1547647268467, value=iteblog.com
4 row(s)
Took 0.0395 seconds
hbase(main):022:0> scan 'iteblog_table'
ROW                  COLUMN+CELL
 1                column=cf:name, timestamp=1547646979906, value=wyp
 2                column=cf:name, timestamp=1547646995691, value=iteblog
 3                column=cf:name, timestamp=1547647005389, value=iteblog_hadoop
3 row(s)
Took 0.0698 seconds
```

恢复快照

快照的一个目的就是可以用来恢复数据，我们现在来演示如何通过快照还原备份的数据。假设我们误删除了 RowKey 为 1 的数据，这时候我们可以通过使用 restore_snapshot 来恢复快照。

```
hbase(main):023:0> delete 'iteblog_table','1','cf:name'
Took 0.0431 seconds
hbase(main):024:0> scan 'iteblog_table'
ROW                  COLUMN+CELL
 2                column=cf:name, timestamp=1547646995691, value=iteblog
 3                column=cf:name, timestamp=1547647005389, value=iteblog_hadoop
2 row(s)
Took 0.0371 seconds
hbase(main):026:0> disable 'iteblog_table'
Took 1.7439 seconds
hbase(main):028:0> restore_snapshot 'snap_iteblog_table'
Took 2.5920 seconds
hbase(main):029:0> enable 'iteblog_table'
Took 1.2940 seconds
hbase(main):030:0> scan 'iteblog_table'
ROW                  COLUMN+CELL
 1                column=cf:name, timestamp=1547646979906, value=wyp
 2                column=cf:name, timestamp=1547646995691, value=iteblog
 3                column=cf:name, timestamp=1547647005389, value=iteblog_hadoop
3 row(s)
Took 0.0272 seconds
```

当然，我们也可以使用快照来分析 HBase

里面的数据，这个后面会再起一篇文章进行介绍，敬请期待。