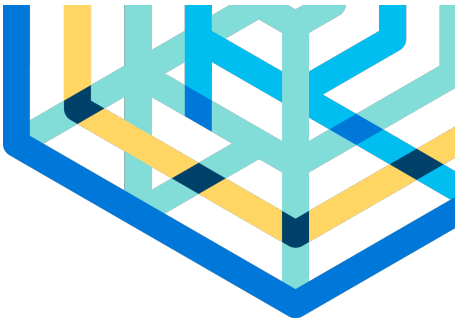
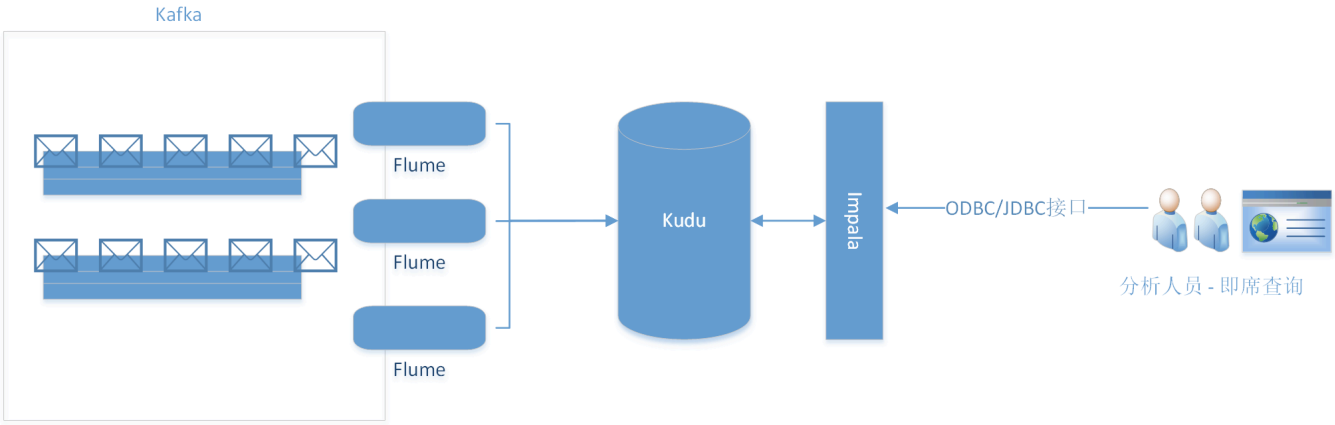
cloudera®



How-to：使用 Kudu+Impala 导入分析准实时数据

Impala 设计的初衷是为 Hadoop 上的海量数据提供交互式的分析功能。对于某些场景而言，数据并不是一次性全量导进 HDFS 的，而是通过实时、或者准实时的方式导入的，因此需要一种全新的存储系统，一方面支持数据的流式导入，另一方面支持数据的列式存储(例如 Parquet 存储格式)。Kudu 应运而生，目前 Cloudera 发布了 Beta 版(尚未提供技术支持)。

通过该文档，你将学习到如何使用 CDH5.4.7+实现该场景。以下是系统架构图：

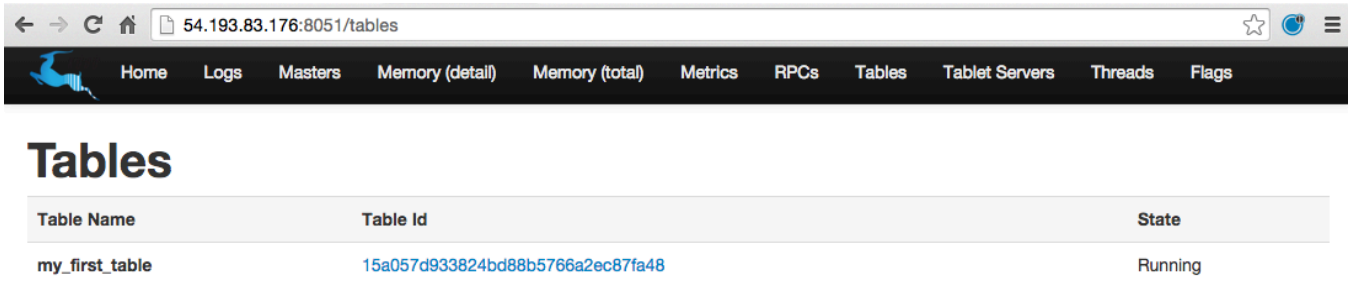


数据存储部分(Kudu)

通过 Impala 创建 Kudu 表，要求构成 Primary Key 的列必须排在前面：

```
create TABLE my_first_table (
  id BIGINT,
  name STRING
)
TBLPROPERTIES(
  'storage_handler' = 'com.cloudera.kudu.hive.KuduStorageHandler',
  'kudu.table_name' = 'my_first_table',
  'kudu.master_addresses' = 'ip-172-31-28-144:7051',
  'kudu.key_columns' = 'id'
);
```

创建表后，可以通过 Kudu 管理界面进行查看



数据导入部分

通过 Kafka 采集外部数据源。首先利用 Kafka 工具创建主题(Topic)：

```
#!/bin/sh

ZOOKEEPER='ip-172-31-28-144'
TOPIC='test'
kafka-topics --create --zookeeper $ZOOKEEPER --topic $TOPIC --partitions 1 --replication-factor 1
```

使用 Kafka 自带的示例 producer 程序，通过命令行窗口交互式输入测试数据：

```
#!/bin/sh

KAFKA='ip-172-31-28-145:9092,ip-172-31-28-146:9092'
TOPIC='test'
kafka-console-producer --broker-list $KAFKA --topic $TOPIC
```

利用 Kudu Java API 定制 Flume Sink 将从 Kafka 中读取的数据直接写入 Kudu，Java API 示例：

```
String master = " ip-172-31-28-144";
String tableName = "java_kudu_table";

KuduClient client = new KuduClient.KuduClientBuilder(master).build();
KuduTable table = client.openTable(tableName);
KuduSession session = client.newSession();

Insert insert = table.newInsert();
PartialRow row = insert.getRow();

row.addInt(0, 11);
row.addString(1, "hello world");
 session.apply(insert);
```

修改 Flume 配置文件，定制 Flume agent：

```
a1.sinks = k1
a1.channels = c1

# 定制的 Flume Sink，直接将数据写入 Kudu 表 my_first_table
a1.sinks.k1.type = com.cloudera.example.KuduSink
a1.sinks.k1.kuduMaster = 172.31.28.144
a1.sinks.k1.kuduTable = my_first_table

# Kafka Channel 定义
a1.channels.c1.type = org.apache.flume.channel.kafka.KafkaChannel
a1.channels.c1.capacity = 10000
a1.channels.c1.transactionCapacity = 1000
a1.channels.c1.brokerList = ip-172-31-28-145:9092,ip-172-31-28-146:9092
a1.channels.c1.topic = test
a1.channels.c1.zookeeperConnect = ip-172-31-28-144:2181
a1.channels.c1.parseAsFlumeEvent = false

# 将 channel 与 sink 绑定起来
a1.sinks.k1.channel = c1
```

数据访问部分

直接使用 Impala，利用 SQL 对数据进行查询分析。

DEMO 示例

1. 创建 Kafka 主题 test，创建完毕后查看 test 主题的相关信息：

```
Topic:test    PartitionCount:1    ReplicationFactor:1        Configs:
Topic: test        Partition: 0        Leader: 130        Replicas: 130    Isr: 130
```

2. 启动 Flume agent，从 Kafka 读取数据、写入 Kudu
3. 启动 Kafka Producer 脚本，在命令行窗口输入测试数据

```
WARN Property topic is not valid (kafka.utils.VerifiableProperties)
4,kip
5,laura
6,Lee
```

4. 打开 Impala-shell，访问表 my_first_table

```
select * from my_first_table;
Query: select * from my_first_table
+----+-------+
| id | name  |
+----+-------+
```

```
| 5  | laura |
| 6  | Lee   |
| 4  | kip   |
+----+-------+
Fetched 5 row(s) in 1.44s
```

参考文档：

http://archive.cloudera.com/cdh5/cdh/5/flume-ng/FlumeUserGuide.html

http://www.cloudera.com/content/www/en-us/documentation/kafka/latest/topics/kafka_flume.html

https://github.com/cloudera/kudu-examples/tree/master/java/java-sample