项目申请书

一、项目信息

- 项目ID：228460418
- 项目名称：Doris Airbyte Connector
- 所属社区：Apache Doris
- 项目描述：Airbyte是当前数据整合（Data Integration）领域非常火的一款产品。其开源设置维护了数量庞大的数据源端和目的端的连接器，以方便用户在不同数据源之间同步或传输数据库。我们需要为Airbyte贡献Doris Connector。
- 产出要求：
  - 实现Doris的Source和Destination连接器。
  - 将连接器贡献给AirByte社区。
- 仓库地址：
  - https://github.com/apache/incubator-doris

二、技术方案

2.1 设计

Airbyte整合了多种数据源和目的端，通过阅读https://github.com/airbytehq/airbyte/tree/master/airbyte-integrations/connectors中源码，了解认识到mysql数据库如何接入Airbyte。我们需要实现的是Doris的Destination连接器。

doris的数据类型目前有17种：

如下表：

整数类型：

BIGINT	TINYINT	SMALLINT	LARGEINT	INT
8字节有符号整数	1字节有符号整数，范围[-128, 127]	2字节有符号整数，范围[-32768, 32767]	16字节有符号整数，范围[-2^127 + 1 ~ 2^127 - 1]	4字节有符号整数，范围[-2147483648, 2147483647]

布尔型：

BOOLEAN
0代表false，1代表true

字符型：

VARCHAR	STRING	CHAR
变长字符串	变长字符串，最大支持 2147483643 字节（2GB-4）	定长字符串，M代表的是定长字符串的长度。M的范围是1-255

日期：

DATETIME	DATE
日期时间类型，取值范围是['0000-01-01 00:00:00', '9999-12-31 23:59:59']. 打印的形式是'YYYY-MM-DD HH:MM:SS'	日期类型，目前的取值范围是['0000-01-01', '9999-12-31'], 默认的打印形式是'YYYY-MM-DD'

浮点型和定点型：

DOUBLE	DECIMAL(M[,D])	FLOAT
8字节浮点数	高精度定点数，M 代表一共有多少个有效数字(precision)，D 代表小数位有多少数字(scale)， 有效数字 M 的范围是 [1, 27]，小数位数字数量 D 的范围是 [0, 9]，整数位数字数量的范围是 [1, 18]， 另外，M 必须要大于等于 D 的取值。 默认值为 DECIMAL(9, 0)。	4字节浮点数

数组：

ARRAY
T支持的类型有：BOOLEAN, TINYINT, SMALLINT, INT, BIGINT, LARGEINT, FLOAT, DOUBLE, DECIMAL, DATE, DATETIME, CHAR, VARCHAR, STRING

其他类型：

BITMAP	HLL
BITMAP列只能通过配套的bitmap_union_count、bitmap_union、bitmap_hash等函数进行查询或使用。	HLL列只能通过配套的hll_union_agg、hll_raw_agg、hll_cardinality、hll_hash进行查询或使用

这里我们翻看doris的官方文档，将数据导入doris有使用jdbc导入数据，导入本地文件的方法等。这里我们选择使用导入本地文件的方法，因为，doris的官方有执行 Stream load 的一个简单的 JAVA 示例，我们可以参考官方的文档模仿。具体的流程如下：

第 1 步：使用模板创建目标：

Airbyte 提供了一个代码生成器，通过

```
$ cd airbyte-integrations/connector-templates/generator # assumes you are
starting from the root of the Airbyte project.
$ ./generate.sh
```

我们得到了一个`Java Destination`模板。

第 2 步：构建新生成的目的地

命令：

```
# Must be run from the Airbyte project root
./gradlew :airbyte-integrations:connectors:destination-<name>:build
```

第 3 步：实施`spec`

模板中写一个`airbyte-integrations/connectors/destination-<name>/src/main/resources/spec.json`. 生成的连接器将负责读取此文件并将其转换为正确的输出。

第 4 步：实施`check`

根据配置中的凭据报告我们是否能够连接到目的地。

第 5 步：实施`write`

实现`write`Airbyte 操作，`getConsumer`在生成的`<Name>Destination.java`文件中实现该方法。

我们通过**本地 CSV**的方式实现连接器。

通过参考airbyte/airbyte-integrations/connectors/destination-csv/src/main/java/io/airbyte/integrations/destination/csv/**CsvDestination.java**

翻看doris的官方文档，有通过本地文件导入数据。

在doris的官方有执行 Stream load 的一个简单的 JAVA 示例中，我们参考官方的文档模仿，改写本地CSV的连接器。

从而实现airByte的调用让上游到下游的一个导入。

其中的airbyte的api的调用参考如下：

```
import io.airbyte.integrations.base.AirbyteMessageConsumer;
import io.airbyte.integrations.base.CommitOnStateAirbyteMessageConsumer;
import io.airbyte.integrations.base.Destination;
import io.airbyte.integrations.base.IntegrationRunner;
import io.airbyte.integrations.base.JavaBaseConstants;
import io.airbyte.integrations.destination.StandardNameTransformer;
import io.airbyte.protocol.models.AirbyteConnectionStatus;
import io.airbyte.protocol.models.AirbyteConnectionStatus.Status;
import io.airbyte.protocol.models.AirbyteMessage;
import io.airbyte.protocol.models.AirbyteMessage.Type;
import io.airbyte.protocol.models.AirbyteRecordMessage;
import io.airbyte.protocol.models.ConfiguredAirbyteCatalog;
```

在destination-Doris目录下应该实现大致三个java类如下：

DorisDestination	DorisSQLNameTransformer	DorisSqlOperations
模板生成，创建导入目标	sql语句的命名转换	对于sql操作的改写，兼容doris的sql操作

对于DorisSqlOperations类的sql的改写，同过模仿如下代码，使用本地的文件的导入：

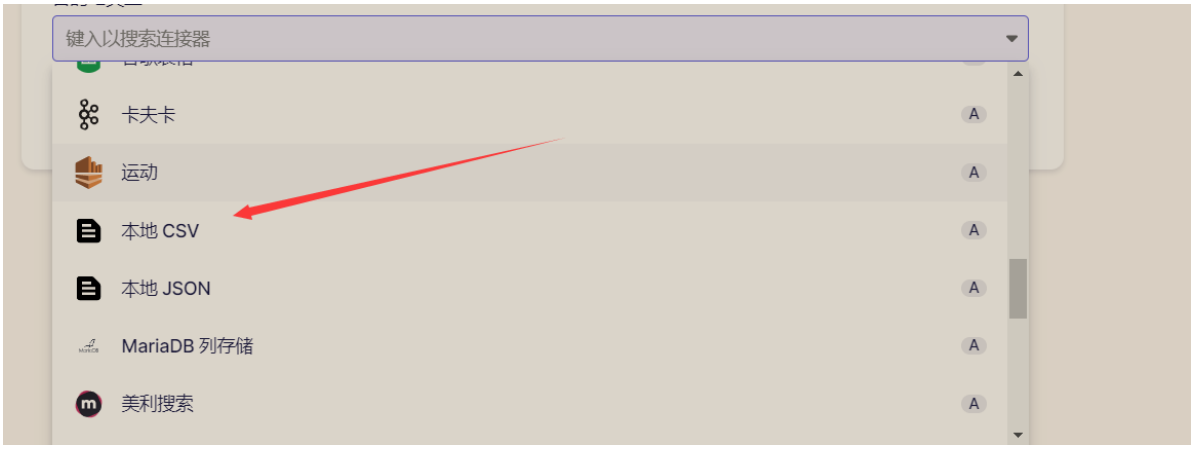
```
ClickHouseConnection conn = connection.unwrap(ClickHouseConnection.class);
ClickHouseStatement sth = conn.createStatement();
sth.write() // Write API entrypoint
    .table(String.format("%s.%s", schemaName, tmpTableName)) // where to write dat
    .data(tmpFile, ClickHouseFormat.CSV) // specify input
    .send();
```

2.2 测试

测试点1： 流程性是否能测通

代码是否不会报错，在ui操作界面是否会出现Doris的连接器图标.

例如：本地的csv



测试点2： 不同数据源类型的测试

数据源为mysql的话，导入的数据类型的转换是否匹配。如下表：

	mysql	doris
整数类型：	TINYINT、SMALLINT、MEDIUMINT、INT（INTEGER）和 BIGINT	BIGINT，TINYINT，SMALLINT，LARGEINT，INT
浮点型	FLOAT,DOUBLE	DOUBLE,FLOAT
定点型	DECIMAL	DECIMAL
布尔型	TINYINT（1）	BOOLEAN
日期类型	YEAR,DATE,TIME,TIMESTAMP,DATETIME	DATE,DATETIME
字符串	CAHR,VARCHAR,TEXT,TINYTEXT,BLOG,BIGARY等	VARCHAR,Char,STRING
数组	无，将数组元素按某个字符分割以字符串形式存储	array

这些数据类型都是常用的数据类型，mysql的数据类型相对比doris多很多。测试常用的数据类型的转化是否正确。

测试点3：性能的测试，是否能跑通多条数据

为提高性能，我们要测试海量数据的导入，我们可以导入10万条数据，通过查看响应时间，是否能同步接受到结果，资源使用率等性能指标来判断我们的程序的性能是否能支撑起海量数据。

2.3 文档

仿照README文件内容，添加的相关的类的说明以及示例。

三、时间规划

将时间划分为三个大阶段，每个大阶段含有若干个以周为单位的小时间段：

3.1 熟悉阶段

本阶段的主要任务是熟悉项目代码以及涉及的相关知识。

时间段	事项
06 月 16 日 - 06 月 23 日	通读项目源代码以及文档；复习 java和doris和连接器插件创建相关知识
06 月 24日 - 06 月 30 日	搭建 Doris Airbyte Connector开发环境，用于之后开发，调试与测试

3.2 实施阶段

本阶段的主要任务是编写代码，实现产出要求。

时间段	事项
7 月 1 日 -- 7 月 14日	完成模板创建和部分导入代码
7 月 15日 --8月1日	完成全部导入代码
8月2日 -- 8月11日	完成中期报告并提交
8月12日 -- 8月20日	完成三个测试点的测试

3.3 收尾阶段

本阶段的主要任务是测试、调试代码，完善文档。