实验指导（二）使用 Flume、Thrift、Kafka、HBase 进行数据收集、传输与存储

实验指导

本次实验可以遵循以下流程进行：

1、 启动 HBase，并创建 HBase 表。

完成 HBase 的安装后，修改配置文件 hbase-env.sh、hbase-site.xml、regionservers 使其以伪分布式方式启动运行。

进入 HBase Shell 交互式命令终端，使用 create 命令创建名为 “HBase_Orders”的表，该表包括两个列族，“Order Detail”与“Transaction”。此步骤可选做使用 Java API 创建“HBase_Orders”表。

2、 启动三个 Kafka 服务器。

完成 Kafka 的安装后，启动 Kafka 服务器的流程包括以下步骤：

2.1 启动 Kafka 自带的 Zookeeper。由于 HBase 也启动了自带了 Zookeeper，为了区分两个服务，需要将 Kafka 的 Zookeeper 服务器端口设置为 2281（默认为 2181）。此步骤需要修改 Kafka/config 目录下 zookeeper.properties 文件。

2.2 将 Kafka/config 目录下 server.properties 文件复制三份，分别命名为 server0.properties, server1.properties, server2.properties，并修改相应的配置。

2.3 启动三个 Kafka 服务器。

2.4 创建一个名为“Kafka_Orders”的 Topic。

3、 编写 Kafka 消费者。

为了方便后续的运行与打包，推荐使用 IDEA 集成环境软件创建 Maven 项目，并修改 pom.xml 文件，引入以下依赖：

```
<dependencies>
    <dependency>
        <groupId>org.apache.kafka</groupId>
        <artifactId>kafka_2.13</artifactId>
        <version>2.7.0</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hbase</groupId>
        <artifactId>hbase-client</artifactId>
        <version>2.3.4</version>
    </dependency>
    <dependency>
        <groupId>org.apache.hbase</groupId>
        <artifactId>hbase-server</artifactId>
        <version>2.3.4</version>
    </dependency>
    <dependency>
        <groupId>org.apache.thrift</groupId>
        <artifactId>libthrift</artifactId>
        <version>0.13.0</version>
    </dependency>
</dependencies>
```

仿照 PPT 中程序示例，编写 Kafka 消费者端，代码包括以下步骤：

3.1 连接 Kafka 服务器。

3.2 连接 HBase 服务器。

3.3 不断读取 Kafka 消息队列中的事件。

3.4 对于每个事件，将其转化为 HBase Put 类的形式，并提交到 HBase 服务器中。完成后回到步骤 1.3。

完成 Kafka 消费者编写后，可以在 IDEA 中启动该程序。

4、 启动 Flume

完成 Flume 的安装后，进入 apache-flume-1.9.0-bin/conf，复制 flume-env.sh.template 文件并命名为 flume-env.sh，修改其中 JAVA_HOME 参数。在 Flume 的安装目录下打开终端，运行./bin/flume-ng ve rsion 命名，如果出现 Flume 版本则安装正确。

复制 flume-conf.properties.template 文件并命名为 flume-kafka-hbase.conf，修改其中 Source、Channel、Sink 的参数。在本次实验中，Source 使用 HTTP Source，Sink 使用 Kafka Sink。

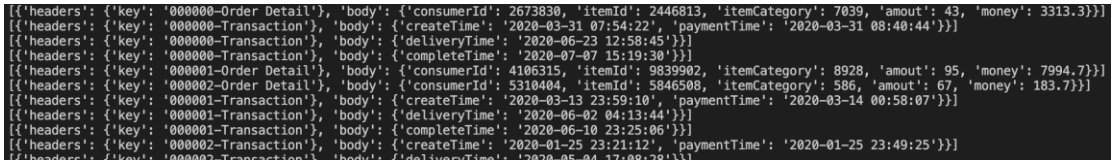
运行所编写的 flume-kafka-hbase.conf，启动 Flume 服务器进行数据收集。

5、 启动具有数据生成功能的服务器

运行所提供的服务器 generatorData.py，开始产生数据并发送到 Flume 进行数据收集。

generatorData.py 产生的数据如图所示，该服务器会自动将产生的每一条数据中“headers”作为报文头字段（也就是 HTTP 报文中的“header”），将“body”作为报文

内容字段（也就是 HTTP 报文中的“data”）。



运行方法：在脚本所在目录下打开终端，以下命令：

python3 generatorData.py -h xxx.xxx.xxx.xxx -p yyyyy

其中 xxx.xxx.xxx.xxx 为安装 Flume 的机器的 ip 地址，yyyyy 为 Flume 收集端监听的端口号。如，在步骤 4 中，安装 Flume 的机器的 ip 地址为 10.105.242.50，收集监听的端口为 44444，则命令为：

python3 generatorData.py -h 10.105.242.55 -p 44444

6、 查看 HBase 表

进入 HBase Shell 交互式命令终端，使用 scan 命令查看表的数据，验证是否正确写入。

选做内容 1：

使用 Java API 替代 Java shell 操作 HBase，并完成以下内容：

1、 创建本次实验中需要使用到的“HBase_Orders”表。

2、 定时查看“HBase_Orders”表内的数据量，完成最简单的统计。注意，此步骤重点在于考察 Java API 操作，只需要完成最基础的定时统计功能，不要求对统计结果进行图形化等其他操作。

选做内容 2：

在上述流程中，数据以字符串和 JSON 格式进行传输，传输效率不高。针对这个问题，可以使用数据序列化技术，即在 Flume 中将数据序列化为字节流，在写入 HBase 前对字节流进行反序列化，再进行存储。

1、 使用 Thrift，分别定义两个类型“OrderDetail”，“Transaction”，两者的定义如下：

```
struct OrderDetail {
        1: required i32 consumerId;
        2: required i32 itemId;
        3: required i32 itemCategory;
        4: required i32 amount;
        5: required double money
}
```

```
struct Transaction {
        1:required string createTime
        2:required string paymentTime
        3:required string deliveryTime
        4:required string completeTime
}
```

使用 thrift 命令生成对应的 JAVA 类代码。

2、 在 Flume 中 HTTP Source 允许用户提供数据处理器类 Handler 以格式化收集到的数据。自定义 Handler 类，读取接收到的报文内容字段的字符串信息，并转化为字节流

3、 在 HBaseConsumer 端,反序列化接收到的字节流,将其还原为类对象,再存储到 HBase 表中。

参考链接：

HBase 官方手册：
https://hbase.apache.org/book.html
Kafka 官方中文手册：
https://kafka.apachecn.org/
Flume 官方手册：
http://flume.apache.org/FlumeUserGuide.html
Thrift 官方手册：
https://thrift.apache.org/