基于 Hive 的广电大数据编程应用研究

何 煌

广东创新科技职业学院 广东东莞

【摘要】Hive 提供了一种简化用户编写 MapReduce 程序工作量的方式来开发大数据的应用方式。本文以广电大数据应用为例，对如何搭建 Hive 开发环境和主要的配置过程以及如何使用 HiveSever2 和第三方语言（Java）进行开发，和通过 IDEA 开发工具进行程序运行调试主要方法进行了深入研究。

【关键词】大数据应用开发；Hive 应用编程

【收稿日期】2022 年 11 月 12 日 【出刊日期】2022 年 12 月 21 日 【DOI】10.12208/j.aics.20220079

Research on the Application of Broadcasting Big Data Programming based on Hive

Huang He

Guangdong Vocational College of Innovation and Technology Dongguan City, Guangdong Province

【**Abstract**】Hive provides a way to simplify the user's MapReduce program workload to develop big data applications. This paper takes radio and television big data application as an example to conduct in-depth research on how to build the Hive development environment, the main configuration process, how to use HiveSever2 and the third-party language (Java) for development, and the main methods of program debugging through the IDEA development tool.

【**Keywords**】Big data application development Hive application programming

1 引言

Hive 提供了一种简化用户编写 MapReduce 程序工作量的方式来开发大数据的应用方式。Hive 提供了 Thrift 服务，用于监听来自于其他进程的 Thrift 连接的一个守护进程。本文以广电大数据应用为例，对如何搭建 Hive 开发环境和主要的配置过程以及如何使用 HiveSever2 和第三方语言（Java）进行开发，和通过 IDEA 开发工具进行程序运行调试主要方法进行了深入研究。

2 Hive 远程服务

Hive 具有一个可选的组件 HiveServer，为 Hive 提供了一种允许客户端远程访问的服务，它基于 Thrift 协议，故也称 HiveServer 为 ThriftServer，它支持跨平台，跨编程语言对 Hive 访问；HiveServer2 支持多客户端的并发和认证，为客户端通过 API 访问 Hive 提供更好的支持。

3 通过 IDEA 搭建 Hive 远程连接环境

IDEA 全称 IntelliJ IDEA，是 Java 等编程语言开发的集成环境之一。该软件在智能代码助手、代码自动提示、重构、JavaEE 支持、版本工具、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面设计非常优秀。

下面以 IDEA 上搭建 Hive 开发环境为例，实现 Hive 远程连接环境的搭建。

3.1 搭建 IntelliJ IDEA 开发环境

可到 JetBrains 公司的官方网站自行下载相关版本的安装包。对于个人用户可下载 Community 版（社区版）。以下将以“IntelliJ IDEA Community Edition 2021.3.3”版本为例，进行开发环境的搭建。

3.2 创建 IDEA 项目

打开开发软件 IDEA，系统弹出欢迎界面，单击“New Project”，在跳出的页面中选择项目管理工具 Maven，选择 1.8 版本的 JDK。为 Project 命名为 HiveJavaAPI，并将该工程放置在本地磁盘目录下。单击“Finish”，系统会自生成一个项目框架。

在项目初始框架中，包含了以下项目元素。

（1）项目根目录：项目存贮的本地目录路径。

（2）idea 节点：主要保存 IDEA 项目的相关信息。

（3）src 节点：保存源代码（main 目录）和测试代码（test 目录）。

（4）External Libraries 节点：保存使用到的外部库文件链接。

（5）Scratches and Consoles：提供了两种临时的文件编辑环境，通过这两种临时的编辑环境，用户可以写一些文本内容或一些代码片段，主要用来测试小段代码的或 API 调用等。

项目具体结构如图 1 所示。

3.3 添加依赖

创建好项目工程后，在项目工程的 pom.xml 文件中添加 Hive 相关依赖，如代码 1 所示。

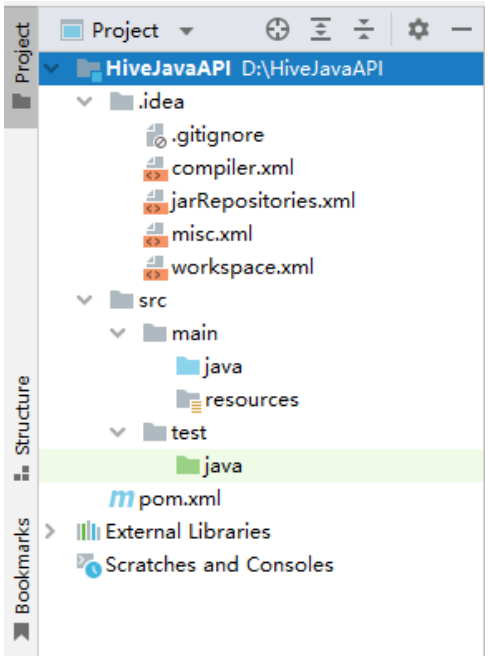

图 1 项目具体结构图

在 pom.xml 文档中，单击鼠标右键选择“Maven”，再单击“Reload Project”，可立即加载依赖。加载完成后，可在左边 Project 工具栏中单击“External Libraries”中查看。下载完成后，相关的依赖包会默认保存到系统用户目录下的.m2/repository 子目录下。

3.4 手动加载 MySQL 驱动

在选择 MySQL 驱动前，需要提前下载好驱动 JAR 包：“mysql-connector-java-8.0.20.jar”，并将其复制到本地磁盘的“HiveJavaAPI 连接驱动”目录下。单击菜单栏选项“File”，选择“Project Structure”，在跳出的页面中选择“Libraries”，单击加号“+”，单击“Java”可加载完成驱动。添加驱动后的项目视图会显示加载的新驱动程序列表。

3.5 JDBC 及其主要接口

JDBC 是 Java 数据库连接的简称，它是一套用于执行 SQL 语句的 Java API。应用程序可通过它连接到关系型数据库，并使用 SQL 语句来完成对数据库中数据的查询、新增、更新和删除等操作。JDBC 在应用程序与数据库之间起到了一个桥梁作用，当应用程序使用 JDBC 访问特定的数据库时，需要通过不同数据库驱动与不同的数据库进行连接，连接后即可对该数据库进行相应的操作。

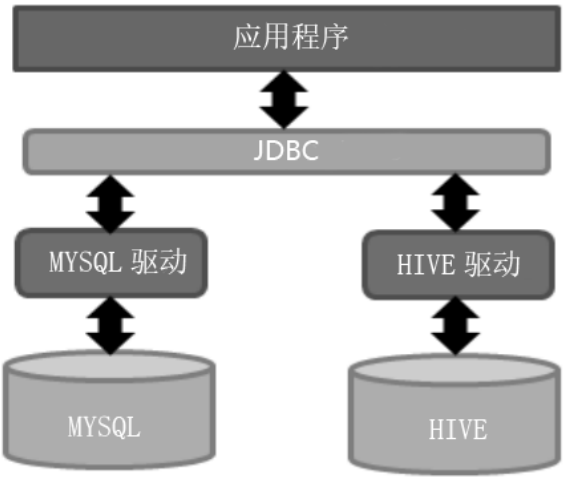

图 2 JDBC 结构示意图

Hive-JDBC 驱动是专用于 Hive 的 JDBC 驱动，是通过 JDBC 访问 Hive 的中介，如图 2 所示。前面已在 pom.xml 文件中添加了对 JDBC 驱动的依赖。JDBC API 主要位于 java.sql 包中，该包定义了一系列访问数据库的接口和类。JDBC 中提供了 Driver、DriverManager、Connection、Statement、PreparedStatement、ResultSet 等常用接口，以下作对这些接口作简要介绍。

Driver 接口是所有 JDBC 驱动程序必须实现的接口，该接口专门提供给数据库厂商使用。需要注意的是，在编写 JDBC 程序时，必须要把所使用的数据库驱动程序或类库加载到项目的 classpath 中。DriverManager 是 JDBC 提供的工具类，用于加载 JDBC 驱动、创建与数据库的连接。在 DriverManag

er 接口中，DriverManager 类中的方法都是静态方法，所以在程序中无须对它进行实例化，直接通过类名即可调用。主要方法：getConnection(String url, String user, String password)，它的主要作用：建立到给定数据库 URL 的连接，返回类型 static Connection。Connection 主要作用是与特定数据库的连接，在连接上下文中执行 SQL 语句并返回结果。Statement 用于执行静态 SQL 语句并返回它所生成结果的对象，Statement 接口对象可以通过 Connection 实例的 createStatment( )方法获得。PreparedStatement 表示预编译的 SQL 语句的对象，它是 Statement 的子接口，该接口扩展了带有参数 SQL 语句的执行操作，应用该接口中的 SQL 语句可以使用占位符“？”来代替参数，然后通过 setXxx( )方法为 SQL 语句的参数赋值。ResultSet 表示数据库结果集的数据表，通常通过执行查询数据库的语句生成，它主要用于保存 JDBC 执行查询时返回的结果，该结果集封装在一个逻辑表格中，在 ResultSet 接口内部有一个指向表格数据行的游标。

3.6 创建连接测试程序

JDBC 连接数据库时通常需要提供如下 4 个必要参数。驱动类名、连接地址、端口号：“jdbc:hive2://HOSTNAME:10000”。用户名：使用默认用户 root。密码：使用默认密码。对 Hive 数据库的操作均基于 JDBC 编程接口实现。在新建的 Java 类 Connection.java 进行与 Hive 默认数据库 default 连接，并创建新的数据库 test，如代码 2 所示。单击右键选择“Run’Connect.main( )’”运行代码，在下方工具栏“RUN”运行结果出现“Process finished with exit code 0”表示运行无误。

代码 1 pom.xml 核心代码

```
<dependencies>
<dependency><groupId>org.apache.hadoop</groupId>
<artifactId>hadoop-common</artifactId>
<version>3.1.4</version>
</dependency>
<!-- hive-->
<dependency>
<groupId>org.apache.hive</groupId>
<artifactId>hive-exec</artifactId>
<version>3.1.2</version>
</dependency>
<dependency>
<groupId>org.apache.hive</groupId>
<artifactId>hive-jdbc</artifactId>
<exclusions>
<exclusion>
<groupId>org.glassfish</groupId>
<artifactId>javax.el</artifactId>
</exclusion>
<exclusion>
<groupId>org.eclipse.jetty</groupId>
<artifactId>jetty-runner</artifactId>
</exclusion>
</exclusions>
<version>3.1.3</version>
</dependency>
</dependencies>
```

代码 2 Connection.java 核心代码

```
String driver = "org.apache.hive.jdbc.HiveDriver";
String url = "jdbc:hive2://master:10000/default";
String username = "root";
String password = "123456";
Class.forName(driver);
java.sql.Connection connection = DriverManager.getConnection(url,username,password);
Statement stmt = connection.createStatement();
stmt.execute("create database test");
stmt.close();
connection.close();
```

4 程序实现广电数据的存储

广电大数据系统是基于双向广电有线网络，使用大数据技术，对用户信息进行采集、存储，并进行有效分析与处理，以便及时了解市场需求并为用户提供具有针对性的产品与服务的一套系统。它促进平台的不断完善，使管道化传输变为平台化传输、单向传播变为双向互动，从而真正实现广电用户从看电视到用电视的转变，推动广电行业进一步发展，也将为社会信息化、政府信息化等提供全面支撑。以下通过程序调用的方式，将广电大数据系统采集到的数据存储 Hive 中，并结合具体案例展示在 IDEA 开发环境如何创建项目并进行调试。

4.1 创建项目

打开开发软件 IDEA，单击“File”->“New”，创建项目 ZJSM，并将该工程放置在本地目录（如：“D:\Hive\ZJSM”）下，在项目工程的 pom.xml 文件中添加 Hive 相关依赖。

4.2 创建 HiveHelper 类和 JDBC 连接

在 src/main/java 目录下创建新的类：HiveHelper，用于处理数据库相关操作，在 HiveHelper 类中创建创建一个新方法：getConn()，用于创建 JDBC 数据库连接，需要使用异常处理，抛出异常（throws ClassNotFoundException, SQLException），如代码 3 所示。

4.3 创建测试类

在 src/test/java 目录下创建新的测试类：HiveTest，用于调用 HiveHelp 类，完成相关的数据库操作，如代码 4 所示。

4.4 创建 Hive 数据库

创建一个新方法：createDatabase(String dbName)，用于创建系统数据库，需要使用异常处理，如代码 5 所示。

4.5 创建 Hive 表

创建一个新方法：createTable(String dbName)，用于创建系统数据库，需要使用异常处理，由于数据表较多，仅以其中一个表 mediamatch_usermsg 的创建为例，如代码 6 所示，其他表的创建，读者可参照案例自行补充完成。

4.6 装载数据

创建一个新方法：loadData(String localFile,String tbName)，用于从本地 CSV 文件加载数据到系统数据库，需要使用异常处理，如代码 7 所示。

4.7 程序运行与调试

在菜单栏选择“Run”->“Run HiveTest”运行新的测试类。假如程序运行错误，可依据错误提示的行号，可在相应的代码前面设置断点，当程序执行到断点时侯，系统会自动暂停，用户可以去观察程序运行的状态和相关变量的值，当需要继续运行时，可单击下一步按扭。

在调试模式下，程序遇到断点会自动中断暂停运行。在菜单栏选择“Run”->“Debug”，可进入程序调试模式。程序在中断情况下，可通过切换到 Debugger 窗口，从而查看变量值，查找程序是否存在拼写或逻辑错误。

5 查询与处理

在完成了创建数据库、数据表、导入数据等操作后，需要创建一个程序测试类来驱动运行相关程序。

5.1 查询数据

由于表中数据较多，仅以查询表中一些字段数据为例，如代码 8 所示。在 HiveTest 类调用 selectAll()方法，并使用该方法查询表中的数据。

代码 3 使用 JDBC 连接 Hive 核心代码

```
public class HiveHelper {
    private static String driverName="org.apache.hive.jdbc.HiveDriver";
    private static String url="jdbc:hive2://master:10000";
    private Connection conn=null;
    private static String username="root";
    private static String password="123456";
    private Statement stmt=null;
    private ResultSet rs=null;
    public Connection getConn() throws ClassNotFoundException, SQLException {
        if (null==conn)
        {
            Class.forName(driverName);
            conn=DriverManager.getConnection(url,username,password);
        }
        return conn;
    }
    public void close() {
        try {
            if (null != conn && !conn.isClosed())
                conn.close();
        } catch(SQLException e){
            e.printStackTrace();
        }finally {
            conn=null;
        }
}
}
```

代码 4 HiveHelp 类

```
public class HiveTest {
    public static void main(String[] args) {
        String dbName="ZJSM2";
        String localFile1="/opt/data/mediamatch_userevent.csv";
        String tbName1 = "mediamatch_userevent";
        HiveHelper helper=new HiveHelper();
        helper.createDatabase(dbName); //1.创建广电数据库
        helper.createTable1(dbName);//2.1.创建用户基本信息表
        helper.createTable2(dbName);//2.2.创建用户状态变更信息表
        helper.createTable3(dbName);//2.3 创建账单信息表
        helper.createTable4(dbName);//2.4 创建订单信息表
        helper.createTable5(dbName);//2.5 创建用户收视行为信息表
        helper.showTables(dbName);//3 显示数据表名称
        helper.loadData(localFile1,tbName1);//4.加载 CSV 数据到 Hive 表
        helper.selectAll(tbName1);//5.显示表内容
    }
}
```

代码 5 创建数据库

```
public void createDatabase(String dbName){
    try {
        stmt=getConn().createStatement();
        stmt.execute("create database if not exists "+dbName);
} catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
```

代码 6 创建表

```
public void createTable1(String dbName){
    try {
        stmt=getConn().createStatement();
        stmt.execute("use "+dbName);
String sql;
        sql = "create table if not exists mediamatch_usermsg(\n"+
                " terminal_no string, phone_no string,    \n"+
                " sm_name string, run_name string,    \n"+
                " sm_code string, owner_name string,    \n"+
                " owner_code string,    run_time string,    \n"+
                " addressoj string, open_time string,    \n"+
                " force string)    row format delimited fields terminated by ';'";
        stmt.execute(sql);
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
```

代码 7 装载数据

```
public void loadData(String localFile,String tbName){
    String sql= " load data local inpath "'+ localFile +"' overwrite into table "+tbName;
    System.out.print(sql);
    try {
   stmt=getConn().createStatement();
        stmt.execute(sql);
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
```

代码 8 查询数据

```
public void selectAll(String tbName){
    String sql="select * from "+tbName;
    if (tbName.equals("mediamatch_userevent")) {
        try {stmt = getConn().createStatement();
            rs = stmt.executeQuery(sql);
            while (rs.next()) {
                System.out.println(
                        rs.getString("phone_no") + "\t" +
                                rs.getString("run_name") + "\t" +
                                rs.getString("run_time") + "\t" +
                                rs.getString("owner_name") + "\t" +
                                rs.getString("owner_code") + "\t" +
                                rs.getString("open_time")
                 ); }
        } catch (SQLException e) { e.printStackTrace(); }
catch (ClassNotFoundException e) {e.printStackTrace(); }
    }
}
```

代码 9 清洗用户基本信息表

```
public void cleanTable1(String dbName){
    try {
        stmt=getConn().createStatement();
        stmt.execute("use "+dbName);
        String sql;
        sql = "CREATE TABLE IF NOT EXISTS mediamatch_usermsg_clean\n"+
 "AS    SELECT * FROM mediamatch_usermsg \n "+
"WHERE    owner_code NOT IN ('2','9','10')   \n "+
 "AND   owner_name NOT IN ('EA 级','EB 级','EC 级','ED 级','EE 级') \n "+
 "AND   sm_name IN ('数字电视','互动电视','珠江宽频','甜果电视') \n "+
"AND   run_name IN ('正常','欠费暂停','主动暂停','主动销户');";
        stmt.execute(sql);
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
```

5.2 清理无效数据

此任务是清洗无效数据，以用户信息为例，任务实现主要步骤如下。创建一个 mediamatch_usermsg_clean 表，将无效的数据剔除，将有效的数据导入表中，实现数据清洗，如以下代码 9 所示。在 HiveTest 类调用 cleanTable1()方法，并使用该方法清理 mediamatch_usermsg 表的无效数据。

6 结语

本文主要研究如何进行 Hive 应用开发，先介绍了配置和启动 Hive 远程服务，接着讲述了如何搭建

Hive 开发环境以及通过 JDBC 访问后台数据，最后程序实现了广电数据的存储，并对广电数据进行了查询和数据清理的操作。在操作过程中还对于创建项目、测试类、数据库、表以及删除无效数据、程序调试等进行了探讨。

参考文献

[1] 卡普廖洛等. Hive 编程指南[M].曹坤,译. 北京：人民邮电出版社,2013.

[2] 孙帅,王美佳. Hive 编程技术与应用[M]. 北京：水利水电出版，2018.

[3] 黑马程序员. Java 基础案例教程[M]. 北京:人民邮电出版社，2021.

[4] 黑马程序员. Hive 数据仓库应用[M]. 北京：清华大学出版社,2021.

[5] 中国广电有线网络技术及年度发展报告(2019) 2020.10.全国互联网与音视频广播发展研讨会暨中国数字广播电视与网络发展年会(2020 年特辑)

[6] 杨力.大数据 Hive 离线计算开发实战[M]. 北京:人民邮电出版社,2020.

[7] 王宏志,李春静. Hadoop 集群程序设计与开发[M]. 北京：人民邮电出版社，2018.

[8] 米洪,张鸽. Hadoop 平台搭建与应用[M]. 北京：人民邮电出版社，2021.

[9] 肖芳,张良均. Spark 大数据技术与应用[M]. 北京：人民邮电出版社，2018.

[10] 李俊杰,谢志明. 大数据技术与应用基础项目教程[M]. 北京：人民邮电出版社，2017.