Ssh框架与大数据平台（Spark）集成 读取mysql关系型数据库关键技术说明(http://m.blog.csdn.net/aneok/article/details/52667367)

**Ssh框架与大数据平台（Spark）集成**

**读取mysql关系型数据库关键技术说明**

1、  所需要框架组件及版本要求

ü  spark-1.6.2-bin-hadoop2.6.tgz

ü  jdk-1.7(1.8)-windows-x64.exe

ü  mysql-installer-community-5.6.32.0.msi

2、  将spark1.6.2解压至硬盘，并在操作系统环境变量中配置其路径

3、  基于Maven的SSH项目POM.xml配置

<dependency>

<groupId>org.apache.spark</groupId>

<artifactId>spark-core\_2.11</artifactId>

<version>1.6.0</version>

</dependency>

<dependency>

<groupId>org.apache.spark</groupId>

<artifactId>spark-mllib\_2.11</artifactId>

<version>1.6.0</version>

</dependency>

<dependency>

   <groupId>org.apache.spark</groupId>

   <artifactId>spark-sql\_2.11</artifactId>

    <version>1.6.0</version>

</dependency>

<dependency>

<groupId>org.scala-lang</groupId>

<artifactId>scala-library</artifactId>

<version>2.11.0</version>

</dependency>

<dependency>

<groupId>org.scala-lang</groupId>

<artifactId>scala-compiler</artifactId>

<version>2.11.0</version>

</dependency>

<dependency>

<groupId>org.scala-lang</groupId>

<artifactId>scala-reflect</artifactId>

<version>2.11.0</version>

</dependency>

4、  基于Maven的SSH项目spring(application).xml配置

<bean id="sparkConf"class="org.apache.spark.SparkConf">

           <property name="AppName"value="SparkForSpring" />

           <!—如果在集群环境中，Master的值应当是实际的集群IP地址 -->

           <property name="Master"value="local" />

</bean>

<bean id="javaSparkContext"class="org.apache.spark.api.java.JavaSparkContext">

           <constructor-argtype="SparkConf" ref="sparkConf" />

</bean>

<bean id="sqlContext"class="org.apache.spark.sql.SQLContext">

           <constructor-argtype="JavaSparkContext" ref="javaSparkContext" />

</bean>

5、  SSH项目添加spark的jar包依赖

将spark解压目录下LIB子目录中的spark-assembly-1.6.0-hadoop2.6.0.jar添加至工程项目，后期需要逐一排除不需要的jar包，达到全部以maven配置的方式实现

6、  调用示例

packagecom.ssmm.service;

importjava.util.Properties;

importjavax.annotation.Resource;

importorg.apache.log4j.Logger;

importorg.apache.spark.api.java.JavaRDD;

importorg.apache.spark.sql.DataFrame;

import org.apache.spark.sql.SQLContext;

importorg.springframework.beans.factory.annotation.Value;

importorg.springframework.stereotype.Service;

//@Scope("prototype")

@Service

publicclass SparkUpper {

private Logger logger =Logger.getLogger(SparkUpper.class);

@Value("${spark.master}")

private String master;

@Resource

private SQLContext sqlContext;

public String upper(String inputFilee) {

           JavaRDD<String> jdbcDF = null;

           DataFrame DF = null;

//以下数据库连接内容请使用实际配置地址代替

           String url ="jdbc:mysql://192.168.8.62:3306/test1?useUnicode=true&characterEncoding=UTF-8";

           String table = "book";

           Properties connectionProperties = newProperties();

           connectionProperties.setProperty("dbtable",table);

connectionProperties.setProperty("user","admin");//数据库用户connectionProperties.setProperty("password","admin"); //数据库用户密码

           DF = sqlContext.read().jdbc(url,table, connectionProperties);

           DF.registerTempTable("testtable");

           return sqlContext.sql("select \*from testtable").javaRDD().collect().toString();

}

}