Spark

Spark系统是分布式批处理系统和分析挖掘引擎

Spark能做什么:  
数据处理:可以用来快速数据，兼具容错性和可扩展性

迭代计算:支持迭代计算，有效应对多步数据的处理逻辑

数据挖掘:在海量数据基础上进行复杂的挖掘分析，可支持各种数据挖掘和机器学习算法

计算分为两大类:离线(批处理)和实时(流式处理，在线)

Spark数据处理(从一大堆数据中找到或者统计出想要的结果):

容错性:mr的错误点:在计算的过程中，某一台的计算服务挂掉了(网线掉了，死机，内存不足，timeout超时等等)，对于mr来说，整个程序是不是挂了?

并不会，会将这台服务器上处理的块数据，迁移到另一个块的主机上，继续进行计算(会影响job的整体时间)，但是对于job的结果还是会正常计算的

Spark的容错性与mr一致(一台主机挂掉，就计算迁移到另一台接着计算)

扩展性:mr增加nodemanager节点(slaves文件中增加)。就等于加了计算单元，spark与mr一样，就可以动态的加节点了(服务器)

迭代计算:  
Data=>A(会产生一个结果)=>B(基于A的结果接着计算)=>C 如果中间的某一步出现了问题，将会”影响结果”或者重头再来

数据分析:机器学习和图计算(是大数据挖掘工程师需要掌握的

为什么很多国内的公司采用的yarn而不是mesos?  
Spark解决的是什么问题:计算

在大数据里面hdsf（文件存储）已经被认为是文件存储比较稳定的文件系统，所以大部分的公司采用hadoop中的hdsf来存储相应的文件，hadoop在hadoop2.x里将yarn引入到了hadoop架构里，所以公司里面就自带有yarn了。而且yarn也是一个比较优秀的资源调度管理框架

Spark架构中一共有5块内容:spark core,spark sql,spark streaming ,图计算，机器学习

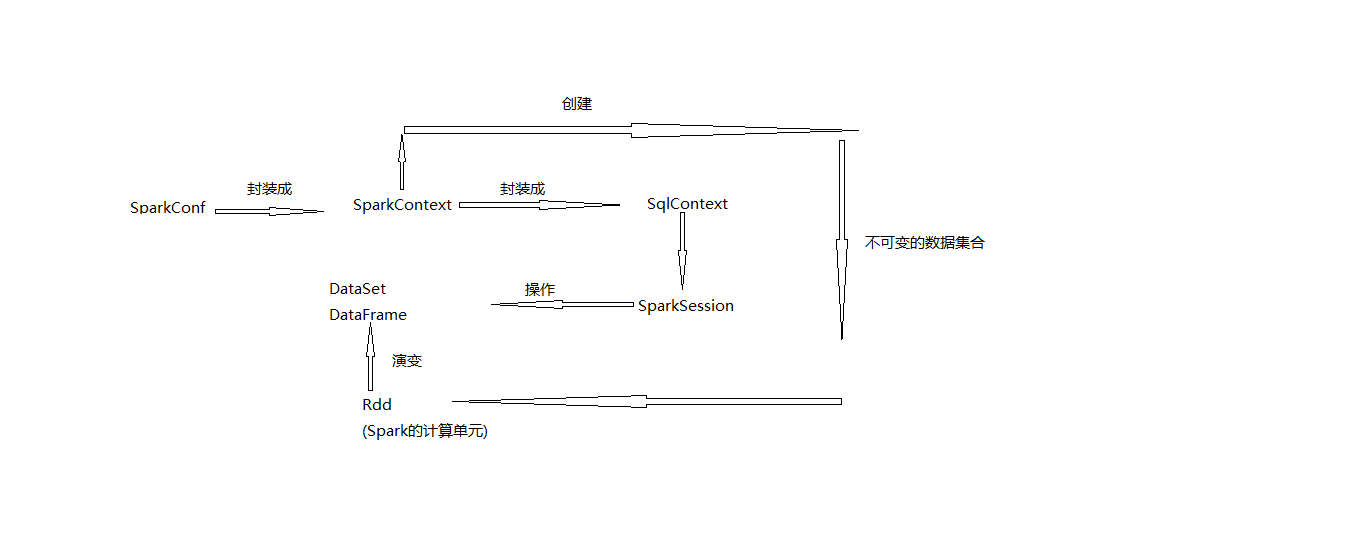
Spark对Kafka有很好的集成，有相应的API可以对kafka的数据流进行处理。

Spark将Mysql表或者Hbase表注册成Spark表，然后进行SparkSql操作

Spark的两种模式:

单机模式：本地模式

多机模式:Spark on yarn



SparkSession实质上就是SqlContext和HiveContext的组合，内部封装了SparkContext，所以计算实际上还是SparkContext完成创建的

用例:

<dependency>

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

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

<version>2.3.0</version>

</dependency>

<dependency>

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

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

<version>2.3.0</version>

</dependency>

<dependency>

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

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

<version>2.3.0</version>

</dependency>

SparkConf conf=new SparkConf();

Conf.setMaster(“local[32]”);

Conf.setAppName(“mc-task”);

....

SparkSession sparkSession=SparkSession.builder().config(conf).getOrCreate();

DataSet<Row> c=sparkSession.read().format(“jdbc”)

.option(“url”,env.getProperty(“spring.dataSource.url”))

.option(“user”,””)

.option(“password”,””)

.option(“dbtable”,”(select \* from table)a”).load();

这时可以将c看成本地的映射零时表