Spark二次开发

www.huawei.com

HUAWEI

目标

- 学完本课程后，您将能够:
  - 了解**Spark**任务运行流程;
  - 搭建开发环境;
  - 运行程序;

目录

1. Spark部署原则

在FI集群中，Spark主要与以下组件进行交互：

1）HDFS：Spark在HDFS文件系统中读写数据(必选)

2）YARN：Spark任务的运行依赖Yarn来进行资源的调度管理(必选)

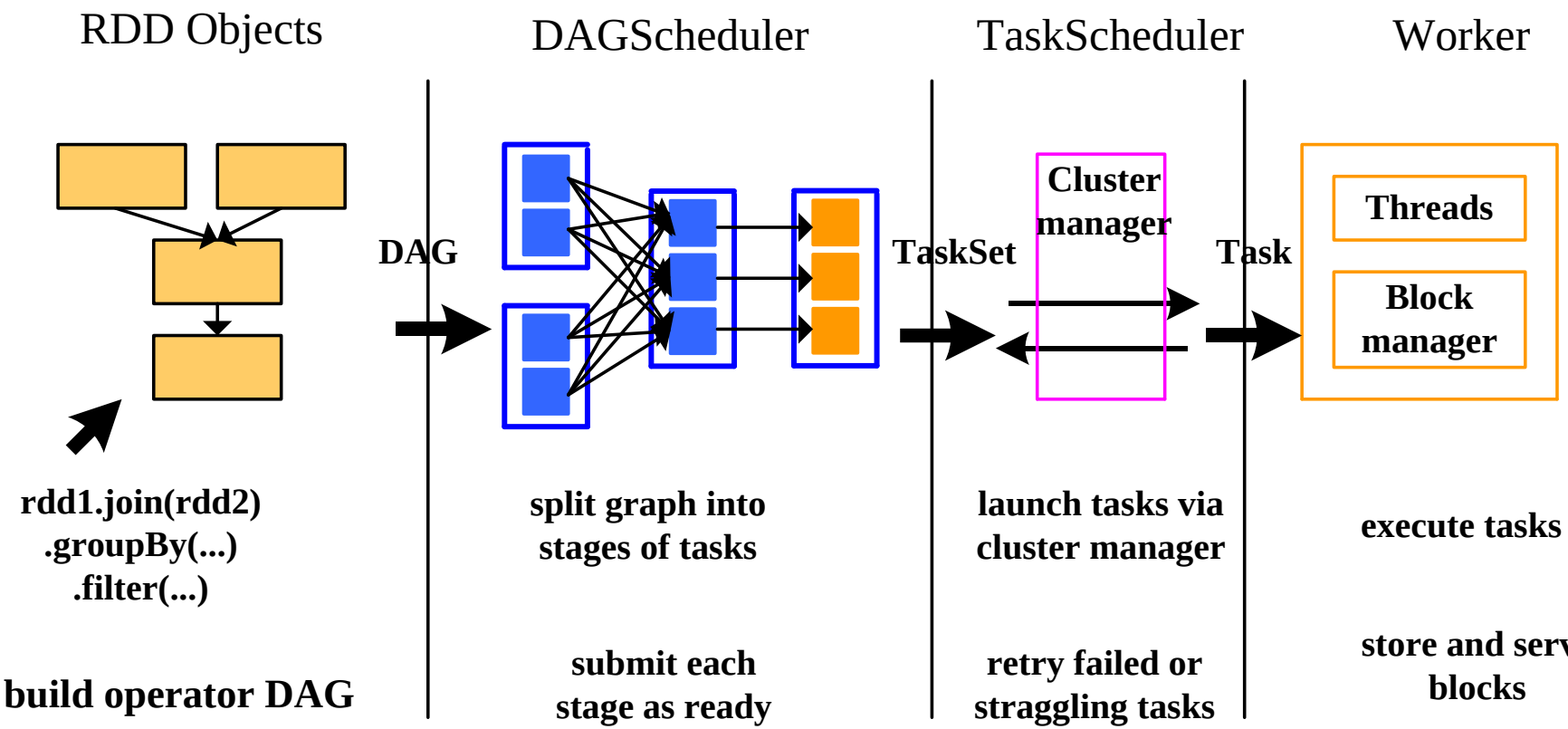
3）DBService：Spark-sql的表存储在Dbservice的数据库中(必选)

4）Zookeeper，JDBCServer的HA的实现依赖于Zookeeper的协调(必选)

5）Kafka：Spark可以接收Kafka发送的数据流(可选)

6）Hbase：Spark可以操作Hbase的表(可选)

2. Spark任务执行流程



目录

2 搭建开发环境

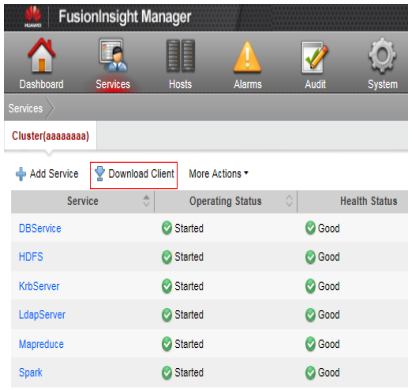
1.确认Spark组件和Yarn组件已经安装，并正常运行。

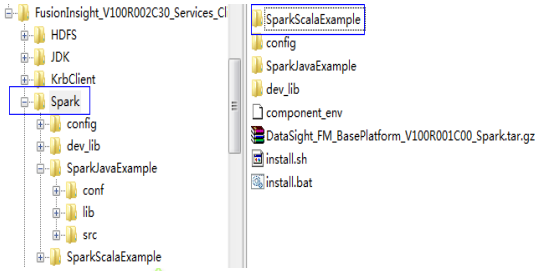
2.客户端机器使用JDK1.7(或1.8)，intellij idea使用13.1.4版本，Scala(2.10.4)版本。

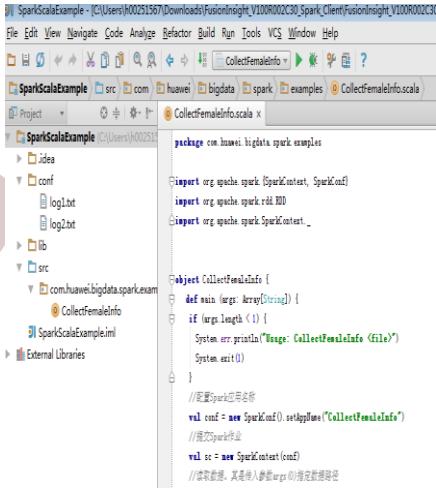
3.客户端机器的时间与FusionInsight集群的时间要保持一致，时间差要小于5分钟，FusionInsight集群的时间可通过登录主管理节点（集群管理IP地址所在节点）运行date命令查询。

4.下载Spark客户端程序到客户端机器中。

开发环境准备







编码

导入Idea

将样例工程导入到Idea中。

同时导入spark依赖的hadoop的jar包。

生成样例工程

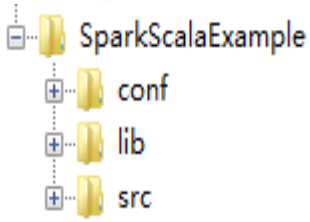
执行安装目录中的install.bat生成工程。

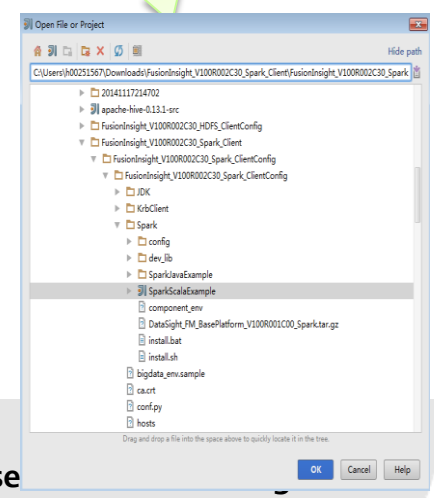
获取样例工程

解压客户端包，从中获取Sample样例工程

下载客户端

登录FusionInsight HD spark下载客户端工具安装包

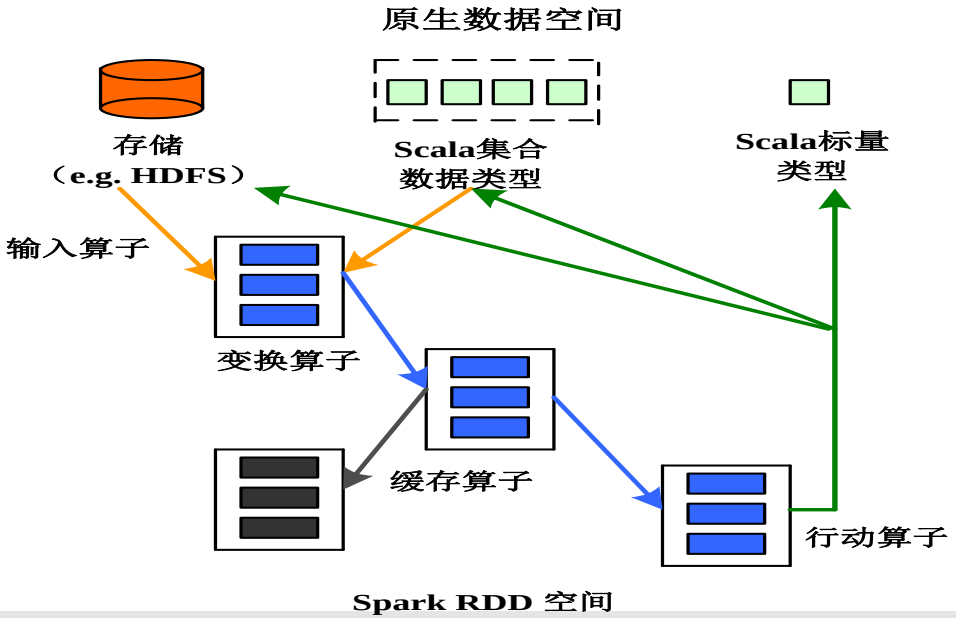




HUAWEI

Spark核心概念 - RDD（Resilient Distributed Datasets）

- 定义：只读的，可分区的分布式数据集；数据集可全部或部分缓存在内存中，在一个**App**多次计算间重用，**RDD**是**Spark**的核心。
- 血统容错：根据血统（父子间依赖关系）重计算恢复丢失数据
- **RDD**操作： **Transformation**和**Action**。



Spark核心概念

RDD的生成：

从Hadoop文件系统（或与Hadoop兼容的其它存储系统）输入（例如HDFS）创建 。

从父RDD转换得到新RDD。

从集合转换而来。

RDD的存储：

用户可以选择不同的存储级别（例如DISK_ONLY，MEMORY_AND_DISK）存储RDD以便重用。

当前RDD默认只存储于内存，当内存不足时，RDD也不会溢出到磁盘中。

简单示例

```
Val file = sc.textFile("hdfs://...")
Val error = file.filter(_.contains("ERROR"))
errors.cache()
error.count()
```

textFile算子从HDFS读取日志文件，返回file（初始RDD）。

filter算子筛出带“ERROR”的行，赋给errors（新RDD）。filter算子为一个Transformation操作。

cache算子把它缓存下来以备未来使用。

count算子返回errors的行数。count算子为一个Action操作。

Spark Application的结构

一个Spark Application的结构主要可分为两部分：初始化SparkContext和主体程序。

初始化SparkContext：构建Spark Application的运行环境。
构建SparkContext对象，如: new SparkContext (master, appName, [SparkHome], [jars])

参数介绍： master：连接字符串，连接方式有local, yarn-cluster, yarn-client等 appName：构建的Application名称 SparkHome：集群中安装Spark的目录 jars：应用程序代码和依赖包。

主体程序：主要是对RDD进行操作以满足应用需求。

Spark shell命令

Spark基本shell命令，支持提交Spark应用。命令为：

./bin/spark-submit \ --class <main-class> --master <master-url> \ ... # other options <application-jar> \ [application-arguments]

参数解释： --class：Spark应用的类名
--master：Spark用于所连接的master(如yarn-client，yarn-cluster)
application-jar：Spark应用的jar包的路径
application-arguments：提交Spark应用的所需要的参数(可以为空)

目录

3运行应用

Spark提供了多中服务供用户使用，用户可通过spark-shell,spark-sql,spark-submit等接口向集群提交任务，与集群交互，也可以通过beeline连接到集群的JDBCServer，提交任务。在客户端使用首先要执行source #{client_home}/bigdata ; kinit user等操作

1）spark-shell使用简单示例（文件读写）：

a:执行 ./spark-shell --master yarn-client进入spark-shell交互式终端

b：读hdfs系统文件，val rdd = sc.textFile("/sparkJobHistory/application_1441091820457_0002")

c: 打印文件行数： println(rdd.count)

在b中，filename可通过fi界面的HDFS->NameNode(Active)->Utilities->Browse the file system中选取文件读取。

3运行应用

2） 运行sparkpi任务

执行sparkpi任务 ./spark-submit --class org.apache.spark.examples.SparkPi --master yarn-client ../lib/spark-examples*.jar

以yarn-cluster模式运行sparkpi任务时，由于driver在集群中的某个节点上启动，需要修改#{client_home}/Spark/spark/conf/spark-defaults.conf中的spark.driver.extraJavaOptions

3运行应用

3）执行beeline连接JDBCServer

a): cd #{client_home/Spark/spark/bin}

b): 执行 ./beeline 进入beeline命令行

c): 执行!connect jdbc:hive2://ha-cluster/default;user.principal=spark/hadoop.hadoop.com@HADOOP.COM;sasl.qop=auth-conf;auth=KERBEROS;principal=spark/hadoop.hadoop.com@HADOOP.COM

或者直接执行spark-beeline

谢谢