第1章Spark概述

(源自:https://biglab.site)

(版本:Ver1.2-20230828)

学习重点

理论学习

- （1）Spark简介。
- （2）搭建Spark环境。
- （3）了解Spark运行架构与原理。

实验学习

- （1）搭建Spark集群。
- 蓝桥课程，4820
-

学习视频

- Spark应用场景
- Spark架构原理
- Spark完全分布式搭建

任务1.1,认识Spark

- Spark的发展
- Spark的特点
- Spark的生态圈
- Spark的应用场景

任务1.2搭建Spark环境

搭建单机版环境

软件准备

spark

- 官网下载: http://spark.apache.org
  - https://spark.apache.org/downloads.html
  - ~ https://www.apache.org/dyn/closer.lua/spark/spark-3.1.3/spark-3.1.3-bin-without-hadoop.tgz
- 国内镜像
  - https://mirrors.tuna.tsinghua.edu.cn/apache/
  - ~ https://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.1.3/spark-3.1.3-bin-without-hadoop.tgz

scale

- 根据配套关系,Spark3.1.2->Scala2.12
- 官网下载:https://www.scala-lang.org/download/2.12.16.html
- wget https://downloads.lightbend.com/scala/2.12.16/scala-2.12.16.tgz

pyspark

- pip install pyspark -i http://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.aliyun.com

百度网盘:

链接:https://pan.baidu.com/s/1FF7HHQi9y2kVBXom_Omd_g
提取码:8ft4

软件列表:

软件	版本	安装包称
Spark	3.2.1	spark-3.2.1-bin-hadoop2.7.tgz
JDK(WIN)	1.8	jdk-8u333-windows-x64.exe
JDK(LINUX)	1.8	jdk-8u281-linux-x64.rpm
IDEA	2018.3.6	ideaIC-2018.3.6.exe
Scala插件	2018	scala-intellij-bin-2018.3.6.zip
Scala	2.12.15	scala-2.12.15.tgz
Hadoop	3.1.4	hadoop-3.1.4.tar.gz
IntelliJ IDEA	2018.3.6	ideaIC-2018.3.6.exe
Git	2.39.2	Git-2.39.2-64-bit.exe
TortoiseGit	2.14	TortoiseGit-2.14.0.0-64bit.msi

软件版本配套

- 参考:https://spark.apache.org/docs/3.1.2/

  Spark runs on
  Java 8/11,
  Scala 2.12.x,
  Python 3.6+
  R 3.5+.
  Java 8 prior to version 8u92 support is deprecated as of Spark 3.0.0.

软件安装

- 1,通过 secureCRT进入Hadoop集群
- 2,运行批命令安装Spark

```
cd /root
wget https://repo.huaweicloud.com/apache/spark/spark-3.1.3/spark-3.1.3-bin-hadoop3.2.tgz  --no-check-certificate
 tar -xvf ./spark-3.1.3-bin-hadoop3.2.tgz
 mv ./spark-3.1.3-bin-hadoop3.2 /usr/local/spark
 chown hadoop:users -R /usr/local/spark
```

- 3,运行批命令安装Scala

```
cd /root
 wget https://downloads.lightbend.com/scala/2.12.16/scala-2.12.16.tgz
 tar -xvf ./scala-2.12.16.tgz
 mv ./scala-2.12.16 /usr/local/scala
```

- 4,安装pyspark

```
pip install pyspark -i http://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.aliyun.com
```

- 5,配置Scala&Spark

  环境变量

  vi /etc/profile 输入i进入编辑状态,然后粘贴下面几行到文件尾部:

```
export SCALA_HOME=/usr/local/scala
 export SPARK_HOME=/usr/local/spark
 export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin:$JAVA_HOME/bin:$SCALA_HOME/bin:$SPARK_HOME/bin:$SPARK_HOME/sbin
 export HADOOP_CLASS=$(hadoop classpath)
 export SPARK_DIST_CLASSPATH=$HADOOP_CLASS
```

  使配置生效

```
source /etc/profile
```

- 6,验证scala版本

```
[root@master sbin]# java -version
openjdk version "1.8.0_322"
OpenJDK Runtime Environment (build 1.8.0_322-b06)
OpenJDK 64-Bit Server VM (build 25.322-b06, mixed mode)
[root@master sbin]#

[root@master sbin]# scala -version
Scala code runner version 2.12.16 -- Copyright 2002-2022, LAMP/EPFL and Lightbend, Inc.
[root@master sbin]#
```

单机测试

- 1,运行批命令测试

```
cd /usr/local/spark/bin
 ./run-example SparkPi 2
```

  - ~运行结果

搭建单机伪分布式环境

软件准备

spark

- 官网下载: http://spark.apache.org
  - https://spark.apache.org/downloads.html
  - ~ https://www.apache.org/dyn/closer.lua/spark/spark-3.1.3/spark-3.1.3-bin-without-hadoop.tgz
- 国内镜像
  - https://mirrors.tuna.tsinghua.edu.cn/apache/
  - ~ https://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.1.3/spark-3.1.3-bin-without-hadoop.tgz
- 内网下载
  - http://home.hddly.cn:90/soft/hadoop/spark-3.1.3-bin-without-hadoop.tgz

scale

- 根据配套关系,Spark3.1.2->Scala2.12
- 官网下载:https://www.scala-lang.org/download/2.12.16.html
- wget https://downloads.lightbend.com/scala/2.12.16/scala-2.12.16.tgz

pyspark

- pip install pyspark -i http://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.aliyun.com

软件版本配套

参考：https://spark.apache.org/docs/3.1.2/

Spark runs on
Java 8/11,
Scala 2.12.x,
Python 3.6+
R 3.5+.
Java 8 prior to version 8u92 support is deprecated as of Spark 3.0.0.

软件安装

- 1,通过 secureCRT进入Hadoop集群
- 2,运行批命令安装Spark

```
cd /root
wget https://repo.huaweicloud.com/apache/spark/spark-3.1.3/spark-3.1.3-bin-hadoop3.2.tgz  --no-check-certificate
 tar -xvf ./spark-3.1.3-bin-hadoop3.2.tgz
 mv ./spark-3.1.3-bin-hadoop3.2 /usr/local/spark
 chown hadoop:users -R /usr/local/spark
```

- 3,运行批命令安装Scala

```
cd /root
 wget https://downloads.lightbend.com/scala/2.12.16/scala-2.12.16.tgz
 tar -xvf ./scala-2.12.16.tgz
 mv ./scala-2.12.16 /usr/local/scala
```

- 4,安装pyspark

```
pip install pyspark -i http://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.aliyun.com
```

- 5,配置Scala&Spark

  环境变量

  vi /etc/profile 输入i进入编辑状态,然后粘贴下面几行到文件尾部:

```
export SCALA_HOME=/usr/local/scala
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin:$JAVA_HOME/bin:$SCALA_HOME/bin:$SPARK_HOME/bin:$SPARK_HOME/sbin
export HADOOP_CLASS=$(hadoop classpath)
export SPARK_DIST_CLASSPATH=$HADOOP_CLASS
```

  使配置生效

```
source /etc/profile
```

- 6,验证scala版本

```
-bash-4.2# java -version openjdk version "1.8.0_322" OpenJDK Runtime Environment (build 1.8.0_322-b06) OpenJDK 64-Bit Server VM (build 25.322-b06, mixed mode)
 -bash-4.2# scala -version Scala code runner version 2.12.16 -- Copyright 2002-2022, LAMP/EPFL and Lightbend, Inc.
```

软件配置

修改conf下的配置文件

```
cd /usr/local/spark/conf
cp ./spark-env.sh.template ./spark-env.sh
vi ./spark-env.sh

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.322.b06-1.el7_9.x86_64/jre
 export HADOOP_HOME=/usr/local/hadoop-3.3.1
 export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
 export SPARK_MASTER_HOST=master
 export SPARK_MASTER_PORT=7077
 export SPARK_LOCAL_IP=master
```

检查进程

jps查看进程

!搭建完全分布式环境!

软件准备

spark

- 国内镜像

https://repo.huaweicloud.com/apache/spark/spark-3.1.3//spark-3.1.3-bin-hadoop3.2.tgz

- 备用下载

http://home.hddly.cn:90/soft/hadoop/spark-3.1.3-bin-hadoop3.2.tgz

scale

根据配套关系,Spark3.1.2->Scala2.12

官网下载:https://www.scala-lang.org/download/2.12.16.html

```
wget https://downloads.lightbend.com/scala/2.12.16/scala-2.12.16.tgz
```

pyspark

```
pip install pyspark -i http://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.aliyun.com
```

软件版本配套

参考：https://spark.apache.org/docs/3.1.2/

Spark runs on
Java 8/11,
Scala 2.12.x,
Python 3.6+
R 3.5+.
Java 8 prior to version 8u92 support is deprecated as of Spark 3.0.0.

软件安装

- 1,通过 secureCRT进入Hadoop集群
- 2,运行批命令安装Spark

```
cd /root
wget https://repo.huaweicloud.com/apache/spark/spark-3.1.3/spark-3.1.3-bin-hadoop3.2.tgz  --no-check-certificate
tar -xvf ./spark-3.1.3-bin-hadoop3.2.tgz
mv ./spark-3.1.3-bin-hadoop3.2 /usr/local/spark
```

- 3,运行批命令安装Scala

```
cd /root
wget https://downloads.lightbend.com/scala/2.12.16/scala-2.12.16.tgz  --no-check-certificate
tar -xvf ./scala-2.12.16.tgz
mv ./scala-2.12.16 /usr/local/scala
```

- 5,配置Scala&Spark

  环境变量

  vi /etc/profile 输入i进入编辑状态,然后粘贴下面几行到文件尾部:

```
export SCALA_HOME=/usr/local/scala
export SPARK_HOME=/usr/local/spark
export PATH=$PATH:$SCALA_HOME/bin:$SPARK_HOME/bin:$SPARK_HOME/sbin
export HADOOP_CLASS=$(hadoop classpath)
export SPARK_DIST_CLASSPATH=$HADOOP_CLASS
```

使配置生效

```
source /etc/profile
```

- 6,验证scala版本

```
[root@master sbin]# java -version

java version "1.8.0_281"
Java(TM) SE Runtime Environment (build 1.8.0_281-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)
[root@master sbin]#

[root@master sbin]# scala -version

Scala code runner version 2.12.16 -- Copyright 2002-2022, LAMP/EPFL and Lightbend, Inc.
[root@master sbin]#
```

- 7,运行批命令测试

```
cd /usr/local/spark/bin
 ./run-example SparkPi 2
```

Spark配置

1,修改conf下的配置文件

```
cd /usr/local/spark/conf
cp ./spark-env.sh.template ./spark-env.sh
vi ./spark-env.sh
```

spark-env.sh 内容如下

```
export JAVA_HOME=/usr/java/jdk1.8.0_281-amd64
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop/
export SPARK_MASTER_IP=master
export SPARK_MASTER_PORT=7077
export SPARK_WORKER_MEMORY=512m
export SPARK_WORKER_CORES=1
export SPARK_EXECUTOR_MEMORY=512m
export SPARK_WORKER_INSTANCES=1
```

```
cp ./workers.template ./workers
vi ./workers
```

workers内容配置如下：

```
slave1
slave2
```

```
cp ./spark-defaults.conf.template ./spark-defaults.conf
vi ./spark-defaults.conf
```

spark-defaults.conf的内容配置如下：

```
spark.master            spark://master:7077
spark.eventLog.enabled      true
spark.eventLog.dir        hdfs://master:8020/spark-logs
spark.history.fs.logDirectory   hdfs://master:8020/spark-logs
```

2,复制spark到从机

```
 scp -r /usr/local/scala/ slave1:/usr/local/
 scp -r /usr/local/scala/ slave2:/usr/local/
 scp -r /usr/local/scala/ slave3:/usr/local/
```

```
 scp -r /usr/local/spark/ slave1:/usr/local/
 scp -r /usr/local/spark/ slave2:/usr/local/
 scp -r /usr/local/spark/ slave3:/usr/local/
```

3,复制系统环境配置文件到从机

```
 scp  /etc/profile slave1:/etc/
 scp  /etc/profile slave2:/etc/
 scp  /etc/profile slave3:/etc/
```

然后到各从机使用source /etc/profile生效配置

```
source /etc/profile
```

4,在hdfs 上创建spark日志目录

```
hdfs dfs -mkdir /spark-logs
```

启动Spark

主机，正常master主机启动时会将所有从机一起启动

```
cd /usr/local/spark/sbin
 ./start-all.sh
```

从机，仅在发现从机没有启动时使用此方法启动。通过查看是否有worker进程来查看是否从机有启动。

```
./start-worker.sh master:7077
```

快速安装

```
wget https://biglab.site/files/scala.tar.gz
tar -xzvf ./scala.tar.gz
mv ./scala /usr/local/

wget https://biglab.site/files/spark.tar.gz
tar -xzvf ./spark.tar.gz
mv ./spark /usr/local/

wget https://biglab.site/files/hive.tar.gz
tar -xzvf ./hive.tar.gz
mv ./hive /usr/local/

```

查看版本

```
spark-shell
```

可支持多种运行模式

- 本地运行模式 （单机）
- 本地伪集群运行模式（单机模拟集群）
- Standalone Client模式（集群）
- Standalone Cluster模式（集群）
- YARN Client模式（集群）
- YARN Cluster模式（集群）

如何选择

- 从对比上看，mesos似乎是Spark更好的选择，也是被官方推荐的
- 但如果你同时运行hadoop和Spark,从兼容性上考虑，Yarn是更好的选择。
- 如果你不仅运行了hadoop，spark。还在资源管理上运行了docker，Mesos更加通用。
- Standalone对于小规模计算集群更适合！

任务1.3了解Spark运行

Spark集群架构

Spark作业运行流程

Standalone模式

- 它是Spark实现的资源调度框架
- 它主要的节点有：Client节点、Master节点、Worker节点
- Driver可以运行在 Client节点上也可以运行在Master 节点上
- Driver运行节点
    - 使用spark-shell交互工具时在Master节点
    - 使用spark-submit工具提交Spark的Job时，在Client节点
    - 使用IDEA,Eclipse使用new SparkConf().setMaster(spark://master:7077),运行在Client节点
- 运行 Spark-shell结果界面

YARN模式

根据Driver在集群中的位置

分为两种

YARN-Client模式(客户端模式)

```
spark-shell --master yarn --deploy-mode "client"
```

YARN-Cluster模式(集群模式)

```
spark-submit --driver-memory 1G  --master yarn --deploy-mode "cluster" --class
src.traind.StreamingKafka  /root/train.jar
```

Mesos模式

略

Spark核心数据集RDD

RDD:Reslient Distributed Datasets,弹性分布式数据集

RDD分成多个分区，每个分区存储在不同的机器上、内存、HDFS、或其他分布式文件系统

RDD两大操作

转换(Transformations)

把原始的数据集加载到RDD,以及把一个RDD转换为另一个RDD

常用转换函数

- map(func)
  - 对RDD数据集中的每个元素都使用func,返回一个新的RDD
- filter(func)
  - 对RDD数据集中的每个元素都使用 func，返回使用 func为 true的元素构成的RDD
- flatMap(func)
  - 和map类似，但是flatMap生成的是多个结果
- union(otherDataset)
  - 返回一个新的dataset,包含源dataset和给定dataset的元素的集合
- groupByKey(numTasks)
  - 返回(K,Seq[V]),根据相同的key分组
- ruduceByKey(func,[numTasks])
  - 用一个给定的func作用在groupByKey而产生的(K,Seq[V]),比如求和

所有的转换都是懒惰(Lazy)操作,只有等到 Actions操作时才真正启动计算

操作(Actions)

把RDD存储到硬盘或触发转换执行

常用的

操作函数

- reduce(func)
  - 通过func聚集数据集中的所有元素
    func接收两个参数，返回一个值
- collect()
  - 返回数据集中的所有元素
- count()
  - 返回数据集中所有元素的个数
- first()
  - 返回数据集中的第一个元素
- take(n)
  - 返回前n个元素
- saveAsTextFile(path)

    - 将数据集以textfile形式保存到本地或hdfs等地方
- foreach(func)
    - 对数据集中的每个元素都执行函数 func

Spark核心原理

概念

宽依赖和窄依赖

- 窄依赖:子RDD的一个分区只依赖于某个父RDD的一个分区
- 宽依赖:子RDD的每一个分区只依赖于某个父RDD的一个以上分区

Stage

- 一个Job会分成一定数量的Stage,各个Stage之间按照顺序执行
- 一个Job会分成多组Task,每组任务就是一个Stage
- Stage有两类Task:ShuffleMapTask和ResultTask
- Stage是以Shuffle和Result这两种类型划分的

DAG有向无环图

常见问题

spark文件清理

针对于Work的配置可在Spark安装目录下conf目录下的spark-env.sh文件中进行配置,以下配置来源于线上生产环境

#对WORK目录进行清理一小时检查一次,对于标准输出/标准错误的数据每小时一个文件保留最近的10个
SPARK_WORKER_OPTS='-Dspark.worker.cleanup.enabled=true -Dspark.worker.cleanup.interval=3600 -Dspark.executor.logs.rolling.strategy=time -Dspark.executor.logs.rolling.maxRetainedFiles=10 -Dspark.executor.logs.rolling.time.interval=hourly'

参考资料

Hadoop和Spark的对比

- ~参见

版本历史

- Ver1.1-20220121
    - 初始版本
- Ver1.2-20230828
    - 增加markdown版本

下一章节