1 Spark 概述

1.1 Spark 定义

构建与计算集群之上支持大数据集的快速的通用的处理引擎

a)快速： DAG、Memory

b)通用：集成Spark SQL、Streaming、Graphic、R、Batch Process

c)运行方式：

StandAlone

YARN

Mesos

AWS

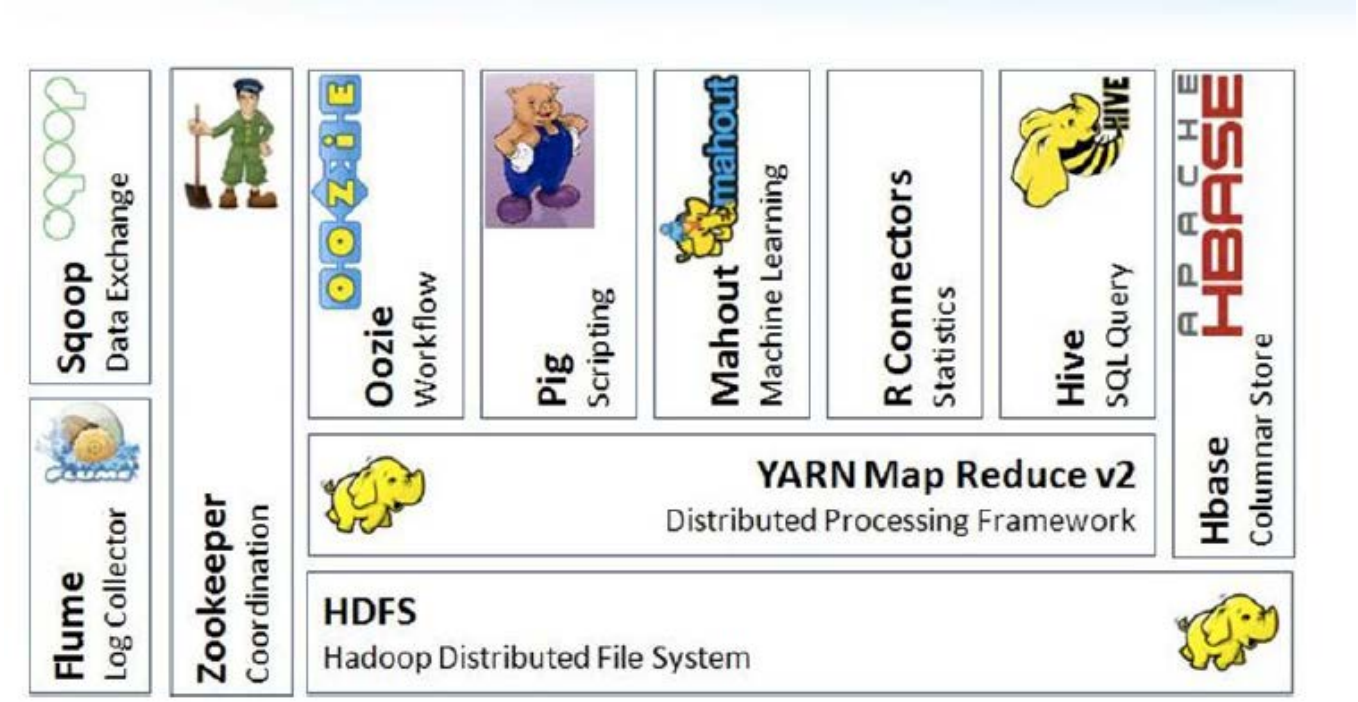
d)数据来源：

Hdfs Hbase Tachyon Cassandra Hive

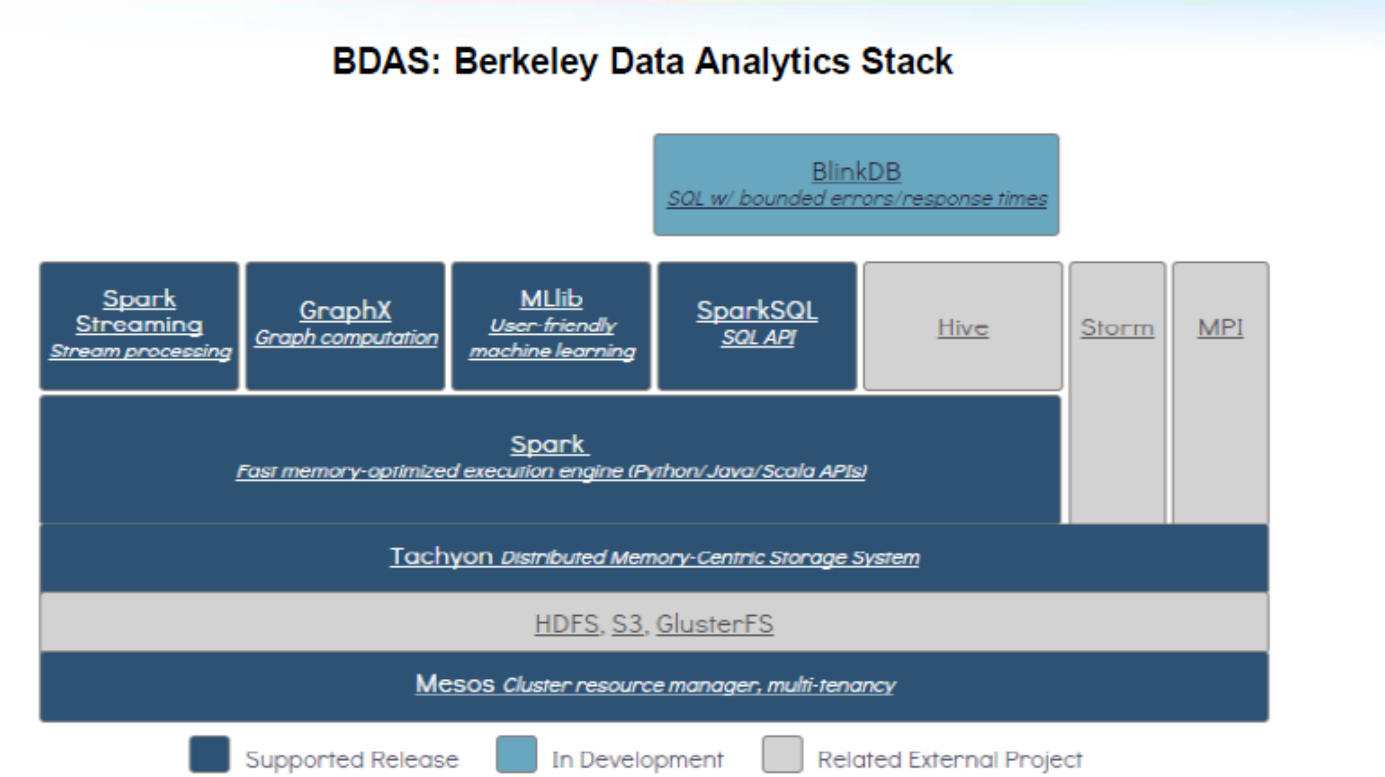
and Any Hadoop Data Source

1.2 Spark 协议栈

1.2.1 Hadoop 生态系统



1.2.2 Spark 协议栈



1.2.3 Spark VS Mapreduce

MapReduce	Spark
数据存储结构：磁盘hdfs文件系统的split	使用内存构建弹性分布式数据集RDD，对数据进行运算和cache
编程范式：Map + Reduce	DAG(有向无环图)：Transformation + action
计算中间数据落磁盘，io及序列化、反序列化代价大	计算中间数据在内存中维护，存取速度是磁盘的多个数量级
Task以进程的方式维护，任务启动就有数秒	Task以线程的方式维护，对小数据集的读取能达到亚秒级的延迟

MapReduce 与Spark比较
1.what? 处理对象
a)MapReduce:基于磁盘File的大数据处理系统
b）Spark：基于RDD(弹性分布式数据集)，可以显示的将RDD数据存储到磁盘和内存中
2.where(软硬件上下文)？
a)MapReduce: Disk
b)Spark: Mem
3.when? (应用场景)
a)MapReduce：可以处理超大规模数据，适合日志分析挖掘等迭代较少的长任务需求，结合了数据的分布式的计算
b)spark：适合数据的挖掘，机器学习等多伦迭代式计算任务

容错性：
a)数据容错性
MapReduce：容错性基于HDFS 冗余机制 ->安全模式->数据校验->元数据保护
spark：容错性基于RDD，spark容错性比mapreduce容错性低，但在处理效率上优势比较明显
b)节点容错性

1.3 Spark 安装配置

========== SetUp HDFS==========
Configuration

hadoop-env.sh

export JAVA_HOME=/opt/modules/jdk1.7.0_67

core-site.xml

```
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop-spark.dragon.org:8020</value>
</property>
<property>
    <name>hadoop.tmp.dir</name>
    <value>/opt/data02/hadoop-2.6.0-cdh5.4.0/data/tmp</value>
</property>
```

hdfs-site.xml

```
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>
```

slaves

hadoop-spark.dragon.org

Start HDFS

NameNode Format

bin/hdfs namenode -format

Start NN/DN

sbin/hadoop-daemon.sh start namenode

sbin/hadoop-daemon.sh start datanode

WEB UI

http://hadoop-spark.dragon.org:50070

========== SetUp Spark==========

Configuration

spark-env.sh

```
HADOOP_CONF_DIR=/opt/data02/hadoop-2.6.0-cdh5.4.0/etc/hadoop
JAVA_HOME=/opt/modules/jdk1.7.0_67
SCALA_HOME=/opt/modules/scala-2.10.4
##########################################################
SPARK_MASTER_IP=192.168.0.222
SPARK_MASTER_PORT=7077
SPARK_MASTER_WEBUI_PORT=8080
SPARK_WORKER_CORES=1
SPARK_WORKER_MEMORY=1000m
SPARK_WORKER_PORT=7078
SPARK_WORKER_WEBUI_PORT=8081
SPARK_WORKER_INSTANCES=1
```

slaves

hadoop-spark.dragon.org

spark-defaults.conf

spark.master spark://hadoop-spark.dragon.org:7077

Start Spark

Start Master

sbin/start-master.sh

Start Slaves

sbin/start-slaves.sh

WEB UI

http://hadoop-spark.dragon.org:8080

Start History Server:

sbin/start-history-server.sh

sc.stop()

========== Test Spark==========

```
scala> val rdd=sc.textFile("hdfs://hadoop-spark.dragon.org:8020/user/hadoop/data/wc.input")

scala> rdd.cache()

scala> val wordcount=rdd.flatMap(_.split(" ")).map(x=>(x,1)).reduceByKey(_+_)

scala> wordcount.take(10)

scala> val wordsort=wordcount.map(x=>(x._2,x._1)).sortByKey(false).map(x=>(x._2,x._1))

scala> wordsort.take(10)
```

1.4 开发环境搭建

2 spark 编程模型

2.1 Spark Application

Spark应用程序由两部分组成

1)Driver

2)Executor

spark应用程序基本概念

Application：基于Spark的用户程序，包含了一个Driver Program和集群中多个的executor

Driver Program：:运行Application的main()函数并且创建**SparkContext**，通常用SparkContext代表Driver Program.

Executor:是为某个Application运行在worker node上的一个进程，该进程负责运行Task，并且负责将数据持久化到内存或者磁盘上。每个Application都有各自独立的executors.

Cluster Manager：集群的资源调度服务

Worker Node： 集群中任何可以运行Application代码的节点

Task：executor的工作单元

Job：包含多个Task组成的并行计算任务，由Spark Action触发任务的执行

Stage：Job的调度单位，每个job被拆分成多组task,每组任务被称为stage，可也称为taskSet

RDD：Spark的基本计算单元，可以通过一系列算子进行操作(Transformation Action)

2.2 Spark RDD

RDD三大特性:
弹性分布式数据集
1.分区
2.计算
3.依赖
 a)宽依赖
 b)窄依赖

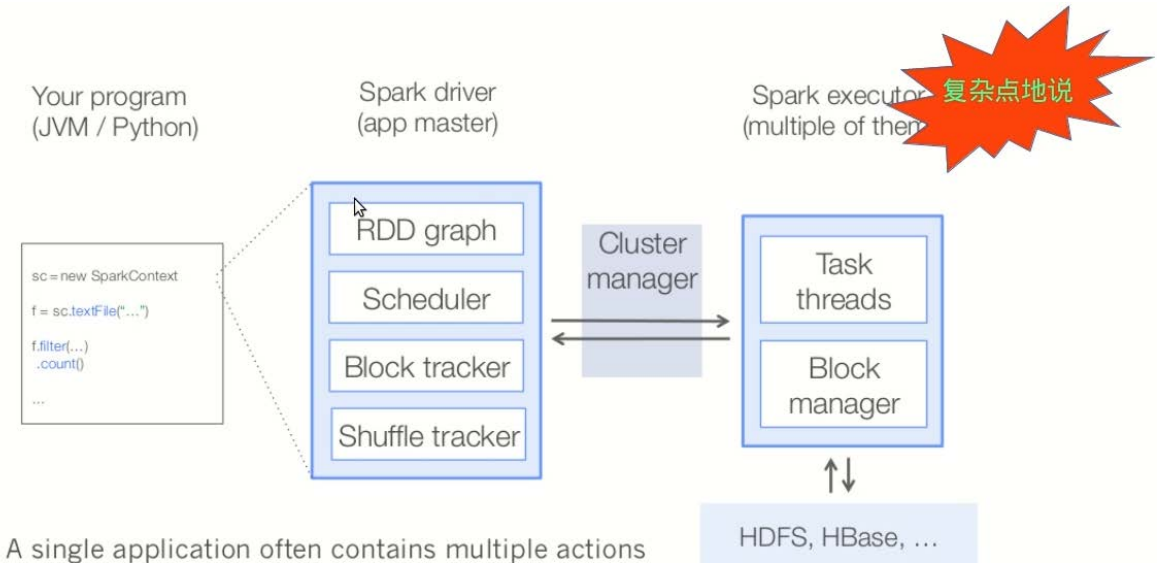
1、RDD 创建
a）外部数据源
HDFS
Spark支持textFile sequenceFile
b）集合Parallelized Collections
RDD 分区partition
sc.textFile("",4)：RDD有四个分区，手动的指定分区个数

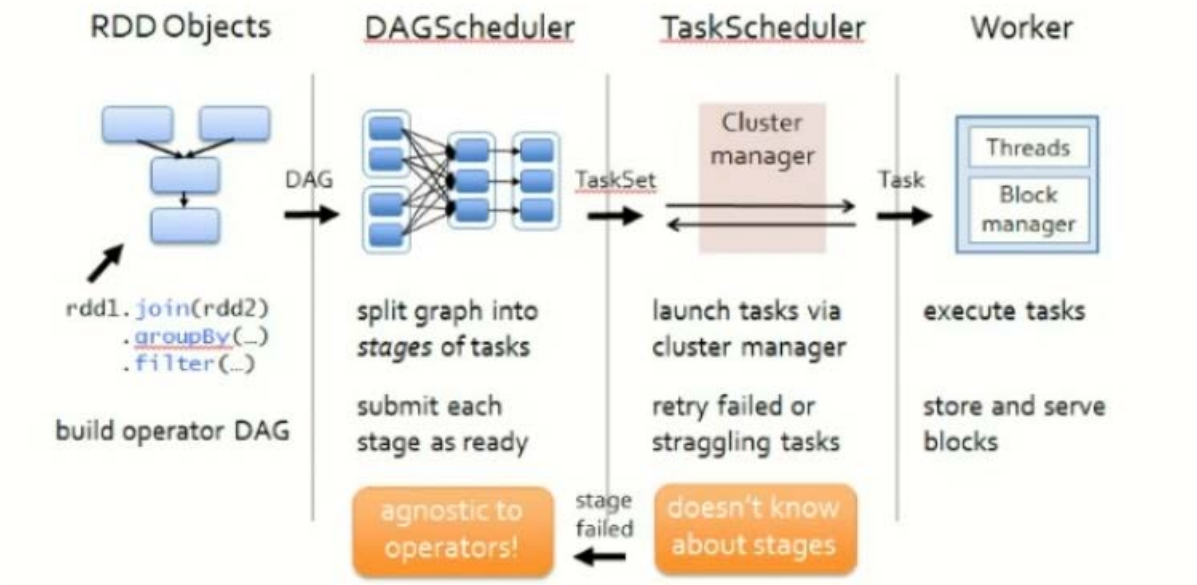
2、RDD Tansformation
支持 map flatMap filter union distinct join groupByKey reduceByKey sortByKey 从一个RDD变为另外一个RDD，
WordCount:map reduceByKey text -> RDD[String] -> RDD[(String,Int)] ->
RDD[(String,Int)] -> collect wordRDD kvRDD resultRDD 一行一行的数据
lazy，懒执行
lineage，生命周期，转换

3、RDD action 触发计算，进行实际的数据处理
reduce collect count countByKey first take foreach saveAsTextFile saveAsSequenceFile

4、RDD cache
cache方法，是延迟执行，需要在一个action执行以后，进行缓存RDD。
cache是persistent特殊缓存方式，将RDD放到内存中

2.3 Spark 运行架构





2.3.1 DAGScheduler

1. 接收用户提交的job
2. 构建Stage，记录哪个RDD或者Stage输出被物化
3. 重新提交shuttle输出丢失的stage
4. 将Taskset传输给底层调度器

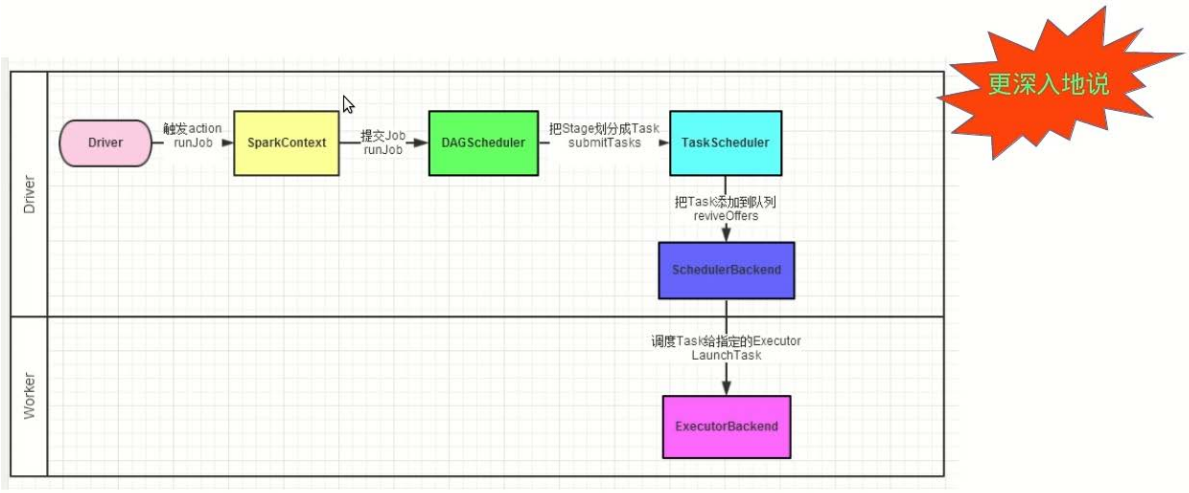
2.3.2 TaskScheduler

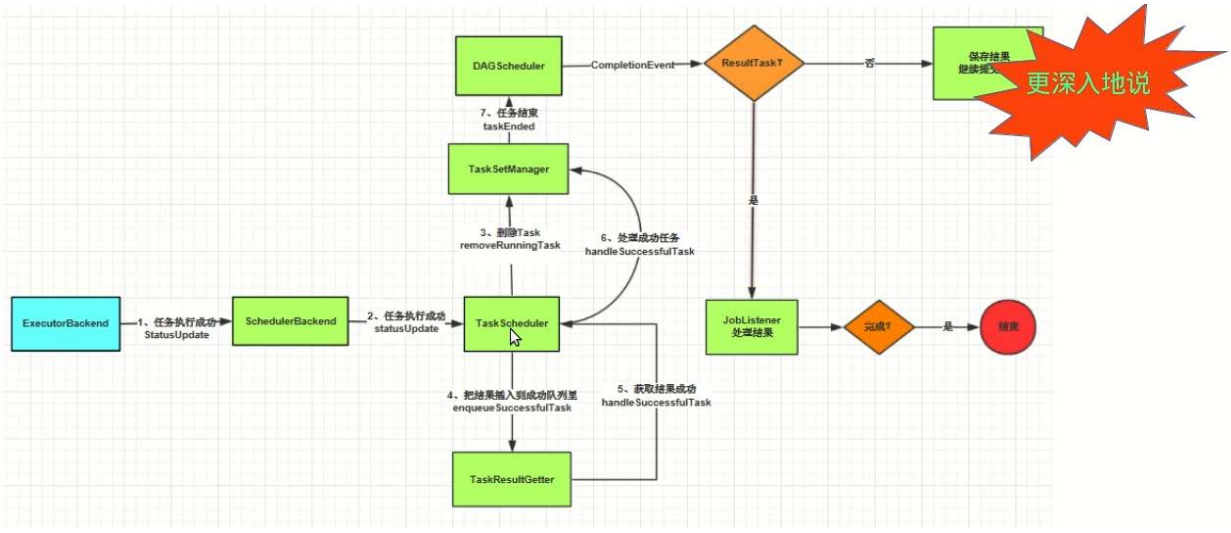
1. 提交tasket到集群运行并监控
2. 为每一个tasket构建一个TaskSetManager实例管理这个TaskSet的生命周期

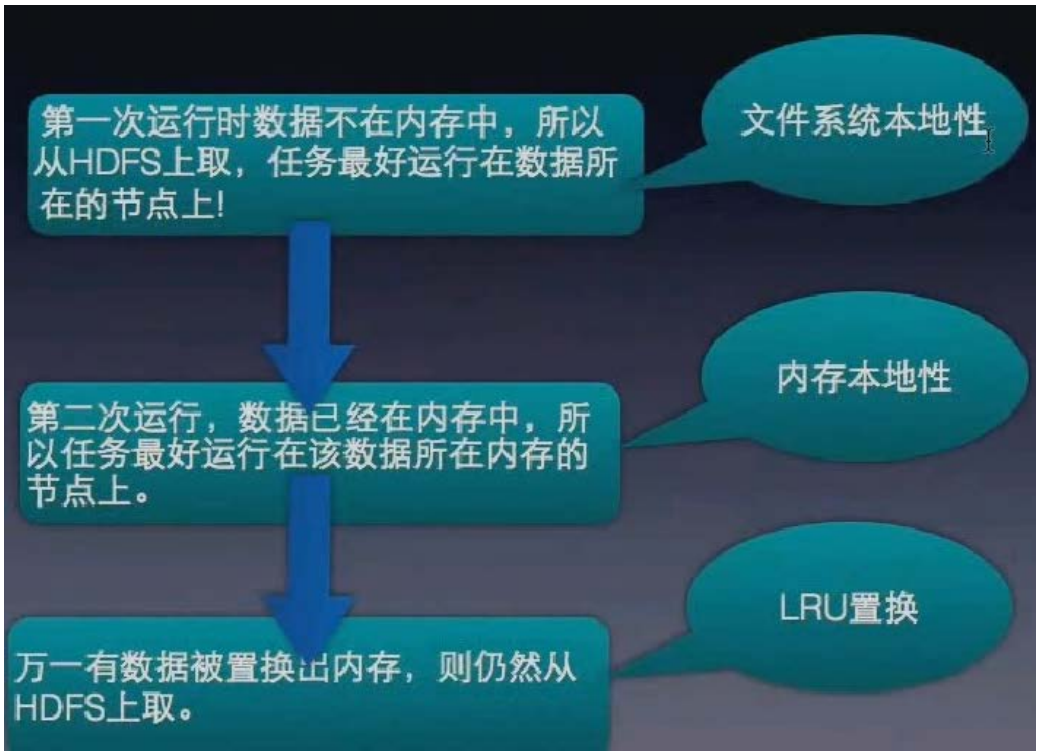
3. 数据本地性决定每个task最佳位置(process-local node-local rack-local and then any)
4. 推测执行，碰到straggle任务需要放到其他的节点重试出现shuffle输出lost要报告fetch failed错误

2.3.3 Task

1. Task是Executor中的执行单元
2. Task处理数据的两个来源：外部存储以及shuffle数据
3. Task可以运行在集群中的任意一个节点上
4. 为了容错，会将shuffle输出写到磁盘或者内存中







2.3.4 SparkContext 初始化过程

1. 依据SparkContext的构造方法中的参数SparkConf创建一个SparkEnv
2. 初始化，SparkUI，以便Spark Application在运行时，方便用户监控，默认端口为4040
3. 创建和启动Scheduler
   1. 创建TaskScheduler、SchedulerBackend
   2. 创建DAGScheduler
   3. 启动TaskScheduler、DAGScheduler
4. 启动Executors