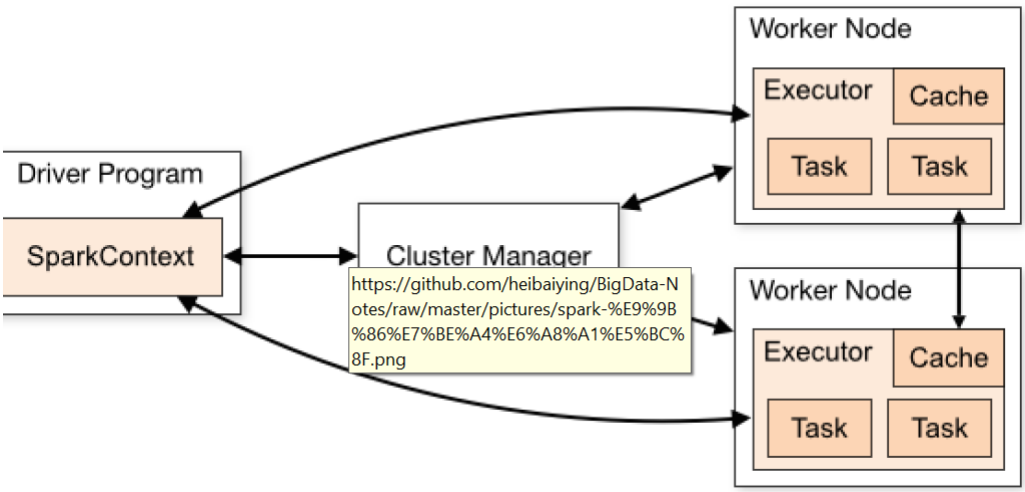
7.spark

学生：张帅豪 18030100101　　老师：李龙海

简介

> - Spark是一个快速、通用、可扩展的分布式计算平台(引擎)。
> - 相对于MapReduce的批处理计算，Spark 可以带来上百倍的性能提升，因此成为继MapReduce之后，最为广泛使用的分布式计算平台。
> - 使用先进的DAG(有向无环图) 调度程序，查询优化器和物理执行引擎，以实现性能上的保证；
>   多语言支持，目前支持的有Java，Scala，Python 和R；
> - 丰富的部署模式：支持本地模式和自带的集群模式，也支持在Hadoop，Mesos，Kubernetes 上运行；
> - 多数据源支持：支持访问HDFS，Alluxio，Cassandra，HBase，Hive 以及数百个其他数据源中的数据。

体系结构

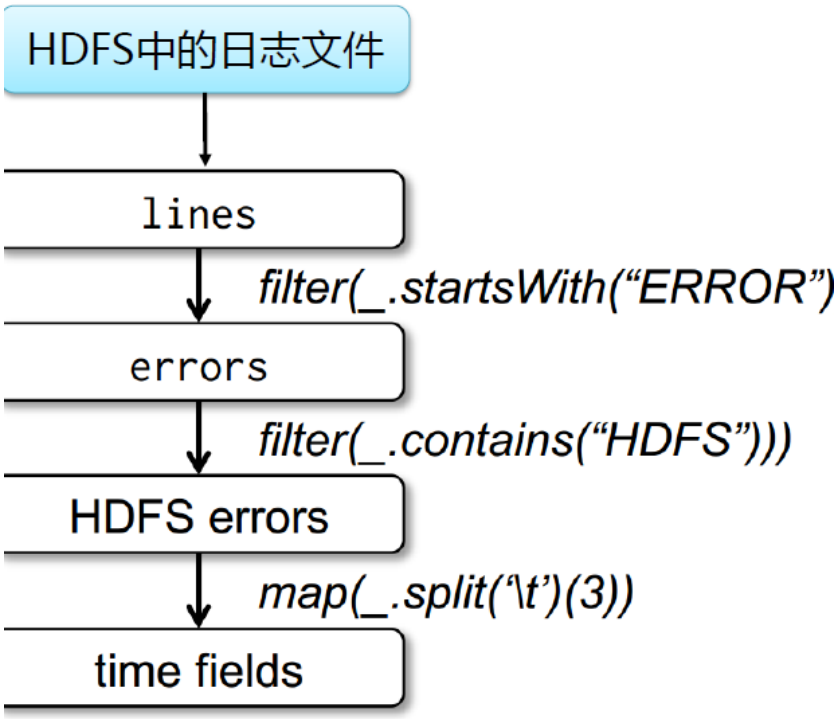


分布式弹性数据集RDD

1. RDD全称为Resilient Distributed Datasets，是Spark 最基本的数据抽象，它是只读的、分区存储的、分布式的数据集合。
2. RDD可以基于外部持久化存储系统中的数据集创建，也可以其他RDD转换而来。
3. 一个RDD由一个或者多个分区（Partitions）组成。对于RDD来说，每个分区会被一个计算任务所处理，用户可以在创建RDD时指定其分区个数，如果没有指定，则平台会根据数据分布存储、CPU资源等情况自己决定。
4. 关于RDD的一个分区的计算任务失败后，Spark平台会自动在其他计算节点上回复该任务。（容错性）
5. 可以将RDD看成是一个分布式存储的“大数组”。应用程序只需关心如何由一个RDD转换为另一个RDD，不用关心RDD在底层是如何分区、如何分布到多个节点上、如何在内存中缓存、内存缓存丢失后如何重新生成。

RDD和DAG

1. 一个具体的大数据处理任务可以表达为一系列RDD之间的转换。
2. 一个分布式计算任务中涉及到的不同RDD之间存在依赖关系，RDD的每次转换都会生成一个新的依赖关系，这种RDD之间的依赖关系就像流水线一样。RDD(s) 及其之间的依赖关系组成了DAG(有向无环图)。
3. 一个分布式计算任务可以表达为一个DAG。
4. DAG 定义了这些RDD(s) 之间的Lineage(血统) 关系，通过血统关系，如果一个RDD的部分或者全部计算结果丢失了，也可以重新进行计算。
5. Spark 可以根据DAG对某些计算子任务进行合并。



- 创建RDD的算子：
  - 基于外部存储系统上的文件创建：
    lines = sc.textFile（“hdfs://localhost:9000/log.txt”）
  - 基于驱动程序(Driver Program)的一个本地数组创建：
    val list = List(3, 6, 9, 10, 12, 21)
    lines = sc.parallelize(list)
- Transformation算子：
  - 将一个RDD转换成一个新的RDD。
  - Transformation算子的动作是“惰性执行”的，即不是在定义时刻执行，只在必要时才执行。
- Action算子：
  - 在RDD上运行计算后将结果返回到驱动程序本地；或者在RDD上运行计算后将结果保存到外部存储系统上。
  - Action算子定义的动作一般会立即执行，进而触发其它的惰性Transformation算子的执行。

11

Spark实现单词计数

```
from pysparkimport SparkConf,SparkContext
conf= SparkConf().setMaster("local").setAppName(“wordcount")
sc= SparkContext(conf=conf)
textData= sc.textFile("./readme.txt")
splitData= textData.flatMap(lambda line:line.split(" "))
flagData= splitData.map(lambda word:(word,1))
countData= flagData.reduceByKey(lambda x,y:x+y)
countData.saveAsTextFile("./result")
```