作业7 Spark Intro

1. 简述为什么会有Spark
2. 对比Hadoop和Spark
3. 简述Spark的技术特点

1、简述为什么会有Spark

由于Hadoop计算框架对很多非批处理大数据问题的局限性，除了原有的基于Hadoop HBase的数据存储管理模式和MapReduce计算模式以外，人们开始关注大数据处理所需的其他各种计算模式和系统，探索一个综合计算框架，Spark是这个方向的探索的产物。

Hadoop计算框架其实存在着很多实现，即使它已经隐藏了很多MapReduce内部的处理细节，程序员还是需要处理很多接口调用的规则，编程负担依然较大。

由于MapReduce计算模型更适合大型的批处理任务，运行过程开销大，且不容易完成实时查询、实时处理等任务，这在实际使用中不总是很方便。

有一个综合性的，既继承了MapReduce大数据方面的优点，又更提高了抽象层次，弥补了Map Reduce速度性能、易读易写方面的缺失的综合框架出现是大势所趋。

2、对比Hadoop和Spark

项目	Hadoop	Spark
分布模型	分布式储存+分布式计算	分布式计算
计算模型	MapReduce框架	通用型计算框架
中间数据	存于磁盘或HDFS，效率较低	存于内存，迭代效率较高
计算速度	慢	快
迭代计算	不擅于进行	特别适合
容错处理	存储阶段计算结果的方式	利用RDD不变性
操作通用性	仅Map和Reduce两种操作	多种数据操作类型
节点通信模型	仅Shuffle一种	可以命名、物化、控制中间结果的储存、分区等
SQL查询工具	Hive	Spark SQL
流处理工具	Storm/Kafka	Spark Streaming
机器学习	Mahout	Spark ML Lib
各自优势	无限规模；企业级可靠性；应用面广	易于开发；内存级性能；集成工作流

3、简述Spark的技术特点

综合来说，Spark是一种基于内存的迭代式分布式计算框架，适用于完成迭代式、关系查询、流式处理等计算密集型任务。

弹性分布式数据集RDD**：Spark的核心分布式数据抽象，是Spark许多特性的核心。

Transformation和Action运算机制：Transformation运算并不真正执行，Action运算时才提交作业一次性完成前面内容。

Lineage血统关系：RDD中数据鲁棒性的保证，确定RDD的演变流程，以便数据丢失时可从父级恢复。

调度模式：使用事件驱动的Scala库类Akka来完成任务启动，通过复用线程池的方式来取代Map Reduce进程或减少线程启动/切换的开销。

API优势：以Scala语言开发并默认以Scala作为编程预言，由于Scala语言本身的简洁特性，Spark程序编写起来简洁于MapReduce。同时Spark也支持使用Java、Python开发。

Spark生态：Spark SQL、Spark Streaming、GraphX等组件使得Spark可以适应多种不同场景和计算任务。

Spark部署：可以以Standalone、Mesos、YARN等多种模式部署，且可直接部署在底层平台上。

适用数据场景：适于多次操作特定数据集的任务，且反复操作次数越多、数据量越大优势越明显；不适用于异步细粒度更新状态的应用、增量修改的应用模型，如Web上的储存、爬虫、索引等。