作业八

林恺越 181098163

1、简述Spark的技术特点

RDD：Spark提出的弹性分布式数据集，是Spark最核心的分布式数据抽象，Spark的很多特性都和RDD密不可分。

Transformation & Action：Spark通过RDD的两种不同类型的运算实现了惰性计算，即在RDD的Transformation运算时，Spark并没有进行作业的提交；而在RDD的Action操作时才会触发SparkContext提交作业。

Lineage：为了保证RDD中数据的鲁棒性，Spark系统通过血统关系（lineage）来记录一个RDD是如何通过其他一个或者多个父类RDD转变过来的，当这个RDD的数据丢失时，Spark可以通过它父类的RDD重新计算。

Spark调度：Spark采用了事件驱动的Scala库类Akka来完成任务的启动，通过复用线程池的方式来取代MapReduce进程或者线程启动和切换的开销。

API：Spark使用Scala语言进行开发，并且默认Scala作为其编程语言。因此，编写Spark程序比MapReduce程序要简洁得多。同时，Spark系统也支持Java、Python语言进行开发。

Spark生态：Spark SQL、Spark Streaming、GraphX等为Spark的应用提供了丰富的场景和模型，适合应用于不同的计算模式和计算任务。

Spark部署：Spark拥有Standalone、Mesos、YARN、K8S等多种部署方式，可以部署在多种底层平台上。

Spark适用于需要多次操作特定数据集的应用场合。需要反复操作的次数越多，所需读取的数据量越大，受益越大，数据量小但是计算密集度较大的场合，受益就相对较小。由于RDD的特性，Spark不适用那种异步细粒度更新状态的应用，例如web服务的存储或者是增量的web爬虫和索引。就是对于那种增量修改应用模型不适合，适用于数据量不是特别大，但是要求实时统计分析需求的任务。

综上所述，Spark是一种为大规模数据处理而设计的快速通用的分布式计算引擎，适合于完成一些迭代式、关系查询、流式处理等计算密集型任务。

2、简述Spark的基本构架和组件功能

基本构架：Spark 架构采用了分布式计算中的Master-Slave模型。Master 是对应集群中的含有Master进程的节点，Slave 是集群中含有Worker 进程的节点。Master 作为整个集群的控制器，负责整个集群的正常运行；Worker相当于是计算节点，接收主节点命令与进行状态汇报；Executor负责任务的执行；Client作为用户的客户端负责提交应用，Driver负责控制一个应用的执行。Spark 集群部署后，需要在主节点和从节点分别启动Master进程和Worker 进程，对整个集群进行控制。在一个Spark 应用的执行过程中，Driver 和Worker 是两个重要角色。Driver 程序是应用逻辑执行的起点，负责作业的调度，即Task任务的分发，而多个Worker用来管理计算节点和创建Executor 并行处理任务。在执行阶段，Driver 会将Task 和Task所依赖的file和jar 序列化后传递给对应的Worker 机器，同时Executor 对相应数据分区的任务进行处理。

组件：

Master node：是集群部署时的概念，是整个集群的控制器，负责整个集群的正常运行，管理Worker node。

Worker node：是计算节点，接收主节点命令与进行状态汇报。

Executors：每个Worker上有一个Executor，负责完成Task程序的执行。

Spark集群部署后，需要在主从节点启动Master进程和Worker进程，对整个集群进行控制。

在Spark应用程序执行过程中，Driver和Worker扮演着最重要的角色。

Driver ： 是应用执行起点，负责作业调度。

Worker：管理计算节点及创建并行处理任务。

Cache：存储中间结果等。

Input Data：为输入数据。

3、简述何为“数据编排”以及Alluxio的特点

数据编排是一个相对较新的概念，用于描述一组技术，这些技术可抽象跨存储系统的数据访问，虚拟化所有数据，并通过标准化 API 将数据呈现给数据驱动的应用程序。“数据编排平台”，架构在计算框架和存储系统之间。数据编排平台跨存储系统将数据访问抽象出来，虚拟化所有数据，并通过具有全局命名空间的标准化API将数据呈现给数据驱动的应用程序。同时，它还应该具有缓存功能，以支持快速访问热数据。总之，数据编排平台为数据驱动的应用程序提供了数据可访问性、数据本地性和数据可伸缩性。

Alluxio是一个开源的基于内存的分布式存储系统，现在成为开源社区中成长最快的大数据开源项目之一。数据存储与计算分离，两部分引擎可以进行独立的扩展。计算引擎(如Hadoop, Spark)可以访问不同数据源(Amazon S3, HDFS)中的数据。

特点：

1、内存I/O速度（可作为分布式共享缓存服务）。

2、采用简化的云和对象存储。
3、 简化数据管理， Alluxio提供对多个数据源的单点访问。除了连接不同类型的数据源之外，Alluxio还使用户能够同时连接到同一存储系统的不同版本，例如多个版本的HDFS，而无需复杂的系统配置和管理。
4、简单的应用程序部署， Alluxio管理应用程序与文件或对象存储之间的通信，将数据访问请求从应用程序转换为底层存储接口。Alluxio兼容Hadoop，Spark和MapReduce程序，可以无需修改任何代码在Alluxio之上运行。