作业7：SPARK 史健均 171870632

1. 简述为什么会有Spark

由于Hadoop框架对很多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计算模式外，人们开始关注大数据处理所需要的其他各种计算模式和系统。

Hadoop只提供了Map和Reduce两种操作，流计算和其他模块支持比较缺乏，所以，在后Hadoop时代，新的大数据计算模式和系统出现，其中尤其以内存计算为核心，集诸多计算模式之大成的Spark生态系统的出现为典型代表。

Spark推出了很多组件，SparkSQL, Spark Streaming, MLlib和GraphX等，逐渐形成了大数据处理一站式解决平台。

Spark是基于内存计算思想提高计算性能：

- 提出了一种基于内存的弹性分布式数据集（RDD），通过对RDD的一系列操作完成计算任务，可以大大提高性能。
- 同时采用一组RDD形成可执行的有向无环图DAG，构成灵活的计算流图
- 覆盖了多种计算模式

Spark是MapReduce的替代方案，而且兼容HDFS、Hive，可融入Hadoop的生态系统，以弥补MapReduce的不足。

2. 对比Hadoop和Spark

综述：

- Spark把中间数据放在内存中，迭代运算效率高；MapReduce中计算结果需要落地，保存到磁盘上。Spark支持DAG图的分布式并行计算，减少了迭代过程中数据的落地，提高了处理效率。
- Spark容错性高。Spark引入了RDD的抽象，它是分布在一组节点中的只读对象集合，这些集合是弹性的。在RDD计算时可以通过CheckPoint来实现容错，而CheckPoint有两种方式：CheckPoint Data和Logging The Updates。
- Spark更加通用。Hadoop只提供了Map和Reduce两种操作，Spark提供的数据集操作类型很多。另外各个处理节点之间的通信模型不再像Hadoop只有Shuffle一种，用户可以命名、物化、控制中间结果的存储、分区等

标准	Map Reduce	Spark
简洁	复杂 需要样本	几乎没有样本
性能	高延迟	非常快
可测试性	通过库包，但很麻烦	非常容易
迭代处理	非微不足道	直接
数据探索性	不容易	Spark shell允许快速和简单的数据探索
SQL接口等	通过Hive	建立在SparkSQL
容错	每个阶段的处理结果存盘保障容错	利用RDD的不变性激活容错
生态系统	很多工具，但并不完全无缝集成	统一的接口和SQL一样，流处理等单一抽象的RDD
在内存中计算	不可能	可能

各方面比较：

- Spark对标于Hadoop中的计算模块MR，但是速度和效率比MR要快很多；Spark是由于Hadoop中MR效率低下而产生的高效率快速计算引擎，批处理速度比MR快近10倍，内存中的数据分析速度比Hadoop快近100倍（源自官网描述）；
- Spark没有提供文件管理系统，所以，它必须和其他的分布式文件系统进行集成才能运作，它只是一个计算分析框架，专门用来对分布式存储的数据进行计算处理，它本身并不能存储数据；
- Spark可以使用Hadoop的HDFS或者其他云数据平台进行数据存储，但是一般使用HDFS；
- Spark可以使用基于HDFS的HBase数据库，也可以使用HDFS的数据文件，还可以通过jdbc连接使用Mysql数据库数据；Spark可以对数据库数据进行修改删除，而HDFS只能对数据进行追加和全表删除；
- Spark处理数据的设计模式与MR不一样，Hadoop是从HDFS读取数据，通过MR将中间结果写入HDFS；然后再重新从HDFS读取数据进行MR，再刷写到HDFS，这个过程涉及多次落盘操作，多次磁盘IO，效率并不高；而Spark的设计模式是读取集群中的数据后，在内存中存储和运算，直到全部运算完毕后，再存储到集群中；
- Spark中RDD一般存放在内存中，如果内存不够存放数据，会同时使用磁盘存储数据；通过RDD之间的血缘连接、数据存入内存中切断血缘关系等机制，可以实现灾难恢复，当数据丢失时可以恢复数据；这一点与Hadoop类似，Hadoop基于磁盘读写，天生数据具备可恢复性；
- Spark引进了内存集群计算的概念，可在内存集群计算中将数据集缓存在内存中，以缩短访问延迟，对7的补充；
- Spark中通过DAG图可以实现良好的容错。

3.简述Spark的技术特点

3.1 高效性：

运行速度提高将近100倍，使用DAG调度程序，查询优化程序和物理执行引擎，可以实现批量和流式数据的高性能。

3.2 易用性：

Spark支持Java、Python和Scala的API，还支持超过80种高级算法，使用户可以快速构建不同的应用。而且Spark支持交互式的Python和Scala的shell，可以非常方便地在这些shell中使用Spark集群来验证解决问题的方法。

3.3 通用性：

Spark提供了统一的解决方案。Spark可以用于批处理、交互式查询（Spark SQL）、实时流处理（Spark Streaming）、机器学习（Spark MLlib）和图计算（GraphX）。这些不同类型的处理都可以在同一个应用中无缝使用。Spark统一的解决方案非常具有吸引力，毕竟任何公司都想用统一的平台去处理遇到的问题，减少开发和维护的人力成本和部署平台的物力成本。

3.4 兼容性：

Spark可以非常方便地与其他的开源产品进行融合。比如，Spark可以使用Hadoop的YARN和Apache Mesos作为它的资源管理和调度器，并且可以处理所有Hadoop支持的数据，包括HDFS、HBase和Cassandra等。这对于已经部署Hadoop集群的用户特别重要，因为不需要做任何数据迁移就可以使用Spark的强大处理能力。Spark也可以不依赖于第三方的资源管理和调度器，它实现了Standalone作为其内置的资源管理和调度框架，这样进一步降低了Spark的使用门槛，使得所有人都可以非常容易地部署和使用Spark。此外，Spark还提供了在EC2上部署Standalone的Spark集群的工具。

3.5 生态圈：

Spark生态圈以HDFS、S3、Techyon为底层存储引擎，以Yarn、Mesos和Standlone作为资源调度引擎；使用Spark，可以实现MapReduce应用；基于Spark，Spark SQL可以实现即席查询，Spark Streaming可以处理实时应用，MLib可以实现机器学习算法，GraphX可以实现图计算，SparkR可以实现复杂数学计算。

- **SparkCore**：将分布式数据抽象为弹性分布式数据集（RDD），实现了应用任务调度、RPC、序列化和压缩，并为运行在其上的上层组件提供API。
- **SparkSQL**：SparkSQL 是Spark来操作结构化数据的程序包，可以让我使用SQL语句的方式来查询数据，Spark支持多种数据源，包含Hive表，parquest以及JSON等内容。
- **SparkStreaming**：是Spark提供的实时数据进行流式计算的组件。
- **MLlib**：提供常用机器学习算法的实现库。
- **GraphX**：提供一个分布式图计算框架，能高效进行图计算。
- **BlinkDB**：用于在海量数据上进行交互式SQL的近似查询引擎。
- **Tachyon**：以内存为中心高容错的的分布式文件系统。

----------END----------