第十三周总结

首先是听课的一些笔记：

13.1 大数据计算引擎spark（上）

1. spark特点（spark为什么比MapReduce更快）

    1. DAG切分的多阶段计算过程更快速

    2. 使用内存存储中间计算结果更高效

    3. RDD编程模型更简单

2. 编程模型RDD

    1. RDD上一个数据集对象，spark每次计算都会形成一个RDD，spark计算过程即为RDD的转换过程。用户面向RDD进行编程，比裸写Map Reduce友好得多。

    2. 代码层面来看，spark的函数式编程中，每一步函数式计算都会产生新的RDD

    3. 纯粹行为类操作不会再产生RDD了，比如counts.saveAsTextFile

    4. 物理上，只有实际产生新的RDD的时候才会产生新的分分片。一般来说，使用了reduce行为都会产生新的RDD

    5. 在spark操作中，没有shuffle的操作叫做窄依赖，即每个机器在本机即可完成，有shuffle操作的叫做宽依赖，即需要不同机器合作。一把来说，窄依赖的各个步骤可以串联起来，连续计算，而不产生新的物理RDD。

13.2 大数据引擎spark（下）

1. Spark计算阶段：和shuffle强相关，产生shuffle时就会产生新的计算阶段。shuffle又依赖于窄依赖宽依赖。窄依赖不会产生shuffle，而groupBy join这样的宽依赖操作则有可能产生shuffle

2. Spark作业管理：spark的代码可以转换成DAG有向无环图。spark的计算中有两种函数，一种说转换函数，一种是action函数，前者会产生一个新的RDD，后者不会。每当遇到一个shuffle，spark就会创建一个计算阶段，每当遇到一个action，就会创建一个job。在一个计算阶段内，每个数据分片都会有一个计算的task，即一个计算分片可以包含很多task

3. Spark的执行过程：Spark支持YARN，Standalone，Mesos，Kubernates等多种部署方案。spark的整体执行过程和MapReduce很像，但是有一些区别。spark的计算结果不存磁盘，直接通过网络发给下一个executor。

13.3 流处理计算：Flink，Storm，Spark Streaming

1. 最早的流处理计算框架storm被认为是实时的hadoop：低延迟，高性能，分布式，可伸缩，高可用。

2. Storm基本概念：

1. Nimbers：负责资源分配和任务调度，相当于JobTracker

2. Supervisor：负责接受Nimbers分配的任务，启动和停止属于自己管理的worker进程，相当于TaskTracker

3. Worker：运行具体处理组件逻辑的进程，相当于Child

4. Spout/Bolt：Worker中的工作线程，相当于Map/Reduce

3. Spark stream实现原理：把数据分成源源不断的小批，形成RDD，交给spark引擎去计算。非常适合每秒点击量，每分钟销售额这一类数据的实时计算

4. Flink：流处理计算和批处理计算，以前者为主。其实流和批的区别只在于批的大小

13.4 大数据基准测试工具HiBench

1. HiBench是一个大数据的测试工具，可以直接在git上找到。可以用来生成大量的测试数据集，并内置了一些常用算法，用于测试。

13.5 大数据分析与可视化

1. 大数据分析可以分成对数据本身的分析以及机器学习

2. 数据大屏：简单但是耀眼展示大数据的一个应用

3. 互联网通过大数据分析的常用指标：

1. 新增用户数

2. 用户留存率

3. 活跃用户数

4. PV

5. GMV

6. 转换率：有购买行为的用户数/总用户数

4. 分析图表：折线图，散点图，热点图，漏斗图

13.6 网页排名算法PageRank

1. PageRank是一个投票算法。即互联网中页面之间的相互引用作为投票，进行排名。比如A页面引用了B页面，那么A页面就给B页面投了一票。通过分析页面之间的引用或者引用的引用而把最具价值的页面找出来。

2. 把PageRank算法公式通过MapReduce进行计算并得到结果，成就了早期Google的互联网霸主地位。

13.7 分类与聚类

1. KNN分类：算出目标与已分好类的N个样本之间的距离，取K个最近的进行投票决定。通过这种方式可以达到按类别自动推荐文章的目的

2. 特征词的提取：一个词在其他文章中越稀少，同时在当前文章中越多，则越有可能是特征词。

3. Kmeans算法：选定种子点 -> 根据到种子点的距离分类 -> 计算每一类的中心点 -> 中心点作为新的种子点进行分类 -> 循环直到种子点不再改变

13.8 推荐引擎算法

1. 基于人口统计的推荐：属于同一类的人，当A用户喜欢某个东西的时候，推测B也喜欢

2. 基于用户的协同过滤推荐：用户A和B都喜欢商品C和D，推测A和B是同一类人，然后发现A喜欢E，则猜测B也喜欢E

3. 基于商品的协同过滤：通过数据发现喜欢商品A的人也喜欢商品B，可以把A和B归为一类，对响应的用户进行推荐

13.9 机器学习与神经网络算法

1. 机器学习架构：

    1. 训练：样本数据 -> 学习算法 -> 模型

    2. 预测：预测数据 -> 预测系统 -> 预测结果

2. 机器学习的数学原理：本质是求函数的参数空间的一个过程。

学习总结：

如果说第十二周讲了一些大数据核心的架构与算法，第十三周则是基于大数据的各种使用场景的简介。基于大数据的计算能力，可以完成各式各样的应用场景。比如实时的流计算，各种数据的分析和可视化，搜索引擎排名的实现，分类与聚类，推荐引擎算法计算，机器学习与神经网络等等。这里面每一个话题都非常大，需要一个专门的团队来完成，个人觉得课程本身更多的是提供一些视野，让我们知道依托大数据技术可以完成哪些事情，如何适当的把大数据技术插入到日常业务中创造价值。