计算机科学与技术学院 大数据管理与分析 课程实验报告

|  |  |  |
| --- | --- | --- |
| 实验题目：PageRank算法实现 | | 学号：201605130116 |
| 日期：2019.5.5 | 班级：2016级泰山学堂 | 姓名：杜洪超 |
| Email：[1503345074@qq.com](mailto:1503345074@qq.com) | | |
| 实验目的：  PageRank 网页排名的算法，曾是 Google 发家致富的法宝。用于衡量特定网页相对于搜索引擎索引中的其他网页而言的重要程度。通过对 PageRank 的编程在 Hadoop 和 Spark 上的实现，熟练掌握 MapReduce 程序与 Spark 程序在集群上的提交与执行过程，加深对 MapReduce 与 Spark 的理解。 | | |
| 实验软件和硬件环境：  软件环境：  系统：Ubuntu16.04 LTS 64位  软件：openjdk-7-jre,openjdk-7-jdk,java1.7.0\_95  Hadoop 2.9.2  Spark 2.3.0  Scala 2.11.8  Eclipse,ssh  硬件环境：  CPU: Intel® Core™ i5-6260U CPU @ 1.80GHz × 4  磁盘：121.8 GB  内存：7.7 GiB | | |
| 实验原理和方法：   1. 分别Hadoop和Spark实现PageRank算法，对比两种实现 | | |
| 实验步骤：（不要求罗列完整源代码）   1. Hadoop实现 2. GraphBuilder类   对数据进行处理，将（url，links）格式的数据转换为（url，rank.lists）  Map函数负责处理初始数据，去掉多余的tab和分隔符，并将初始rank值赋给每一个url；  Reduce函数不做处理，直接输出key和value     1. PageRankIter   用于生成每次迭代结果。  Map函数对key所对应的links中的每一个链接，计算当前url对它的PR值的贡献，最后生成（key，links）维护链接关系  Reduce函数将所有url的PR值加起来，再次生成（url，rank，links）的key-value对，供下一次迭代使用；     1. RankViewer类   用于生成最后结果  Map函数把rank作为key，key作为value，这样可以实现按pr值排序  通过重载比较函数可以实现从大到小排序  Reduce函数格式化输出结果     1. PageRankDriver类   总控程序，负责控制迭代次数和调用各类的次序，以及对应的目录参数   1. Spark实现 2. 用scala语言实现PageRank，实现思路和MapReduce类似， | | |
| 结论分析与体会：  通过对 PageRank 的编程在 Hadoop 和 Spark 上的实现，掌握了 MapReduce 程序与 Spark 程序在集群上的提交与执行过程，加深对 MapReduce 与 Spark 的理解。  Spark用scala实现相比Hadoop的实现，代码长度显著减少，但需要学习scala的基本语法和适应编程思路，在实际编程中二者花费时间相差不多，但前者有相当一部分是学习成本。  实际计算中，Spark要快于Hadoop，体现了Spark的优势。 | | |
| 就实验过程中遇到和出现的问题，你是如何解决和处理的，自拟1－3道问答题：   1. Scala语言直接打包成jar比较困难，最终选择借助Intellij IDEA实现打包 2. 不知道因为什么原因，spark所输出的结果精度不够，不能实现double的精度，只能保证float精度，目前仍没有解决，不过在另一台电脑上执行结果正确，猜测可能是电脑或者环境配置甚至IDE的原因。 | | |