Lab2 Pagerank on the Wikipedia Corpus 实验报告

2013011326　计 32　周建宇

【实验内容】

本次实验以维基百科提供的 1800 多万文章数据（abstr-enwiki-articles.xml）作为程序输入，要求对这 1800 多万篇文章进行 pagerank。最终给出每一个页面（page）相应的打分和他所链接的文章标题列表。

【实验步骤】

按照实验指导的提示，完成实验可以分为以下三个步骤：

1. 对输入数据进行预处理，抽取出关键信息，具体为一篇文章的标题（title），该文章所链接到的标题列表（link_list）和该页面初始的 page_rank 值（人为给出，在此我们设置为 1000）
2. 将处理好的数据作为输入，进行 20 轮迭代，每轮迭代中计算一次所有页面的 page_rank 值，其他内容不更新，按照输入格式输出。
3. 对最终的数据进行一次格式整理，整理成按照 page_rank 值从小到大排列的顺序。

【实现难点】

1. 数据预处理阶段，需要抽取重要信息。在 MapReduce 框架中，我们需要自定义输入类（默认为 TextInputFormat），这里我们自定义 MyInputFormat 类，该类将一行一行不停地读取文本内容直到碰到</page>停止，<page>与</page>之间的内容保存在 content 中，我们要抽取出<title>与</title>之间的 title 作为 key，另外，我们需要抽取出 content 中所所有它所链接到的 link_list, 每一个 link_title 是位于[[和]]之间的内容。这样 map 函数得到的数据就比较规整了，几乎不用做任何事情，只需要在 reduce 输出阶段加上一个我们赋予的 page_rank 默认值即可。

2. 迭代计算阶段，存在一个比较棘手的问题是因为数据非常大，我们一开始只能从原始数据集中抽取一小部分作为输入，这样方便调试。但由于输入数据不全，使得很多文章的 title 根本就不存在于<title>与</title>之间，另一些存在于<title>与</title>的 title 却根本没有指向它的 page，这使得 page_rank 值由很多为 0，而其他相当一部分的值都一样。最后只能用自己人为构造的小数据来验证算法的正确性。

【实验总结】

这次实验让自己总算知道了什么是“大数据”，当将所给文件解压出来的时候简直惊呆了，38G 将磁盘完全写满……，一时间觉得根本不可能处理这些数据。从中读取了一小部分之后也是让我第一时间手足无措，原本以为是像实验指导中说的那样数据已经比很好的 formatted 了，没想到原始的 xml 最多只能算半结构化。因此用 mapreduce 进行数据抽取占据了自己很大一部分时间。

由于时间紧迫，关于进行的迭代是否收敛等工作其实并没有做，只是直观规定了迭代次数，因此效果好不好自己也不知道，而且在自己的机子上 mapreduce 跑的出奇的慢。或许只有在真正的分布式环境下才能发挥 mapreduce 的威力吧。