

# 基于 MapReduce 的 HITS 算法的实现

余辉, 王笑梅\*

(上海师范大学 信息与机电工程学院 上海 200030)

**摘要:**在对 HITS 算法和基于 MapReduce 编程模型的云计算框架 Hadoop 的研究基础上, 利用 Hadoop 来重新设计并实现 HITS 算法. 同时, 在实验中分析了不同 blocksize 和集群规模对算法执行效率的影响. 实验表明: 当 blocksize 过大时, 由于没有充分利用集群的并行特性, 算法效率逐渐降低, 而适当扩大集群规模, 算法运行效率会逐渐提高.

**关键词:** HITS; MapReduce; Hadoop; 分块大小; 集群

**中图分类号:** TP 311.1   **文献标识码:** A   **文章编号:** 1000-5137(2013)05-0476-05

## 0 引言

HITS 算法是一种基于“导航页”和“权威页”的迭代算法<sup>[1]</sup>, 用于计算每个网页的重要性. 由于处理的网页数量巨大, 普通的串行实现效率太低, 在这种情况下, 考虑使用 MapReduce 编程模型来实现 HITS 算法成为必然. MapReduce 编程模型最早由 Google 公司提出<sup>[2]</sup>, 用于并行处理大规模数据集, 并逐渐成为云计算的核心技术之一. MapReduce 的开源实现框架 Hadoop<sup>[3-4]</sup> 是一个优秀的并行编程平台, 它克服了传统并行计算框架 MPI 的不足, 实现了计算向存储迁移<sup>[5]</sup>, 在处理密集型的大规模数据集时有很大的优势.

## 1 HITS 算法与 Hadoop 框架

### 1.1 HITS 算法

HITS 算法对网页重要性的计算有两个度量准则, 第一个是网页作为导航页的重要性, 即它的导航度; 第二个是网页作为权威页的重要性, 即它的权威度. 网页的导航度和权威度分别用  $n$  维向量  $h$  和  $a$  表示, 第  $i$  个网页的导航度为向量  $h$  的第  $i$  个分量, 权威度为向量  $a$  的第  $i$  个分量. 为了计算出向量  $h$  和  $a$ , 还需要一个  $n \times n$  维的连接矩阵  $L$  和它的转置矩阵  $L^T$ , 如果  $L_{ij} = 1$ , 则表示第  $i$  个网页到第  $j$  个网页存在链接, 如果  $L_{ij} = 0$ , 则表示两个网页之间没有关联.

一个网页的导航度依赖于这个网页所链出网页的权威度, 权威度依赖于这个网页所链入网页的导航度, 所以有以下公式:

$$h = La, \quad (1)$$

$$a = L^T h. \quad (2)$$

由权威度  $a$  计算出导航度  $h$ , 再由导航度  $h$  计算出权威度  $a$ . 显然, 这是一个迭代过程, 第一次迭代时取  $a$  为  $n$  维全 1 向量. 由于迭代过程中  $h$  和  $a$  的分量可能会急剧增大, 所以需要归一化处理:

收稿日期: 2013-06-17

作者简介: 余辉(1988-), 男, 上海师范大学信息与机电工程学院硕士研究生; 王笑梅(1970-), 女, 上海师范大学信息与机电工程学院副教授.

\* 通信作者

$$h = \xi La , \tag{3}$$

$$a = \delta L^T h. \tag{4}$$

其中  $\xi$  和  $\delta$  是归一化因子,它们将每次迭代计算出的  $h$  和  $a$  的最大分量变为 1. 以上公式(3)和(4)作为 1 次完整的迭代,当  $h$  和  $a$  分别达到一定精度要求时即停止迭代.

### 1.2 Hadoop 框架

基于 MapReduce 编程模型的云计算框架 Hadoop 是一种分布式并行计算框架,在其底层支撑 MapReduce 数据处理功能的是分布式文件系统 HDFS<sup>[6,7]</sup>,MapReduce 编程模型和 HDFS 构成了 Hadoop 最基本、最核心的组件. HDFS 的结构如图 1 所示.

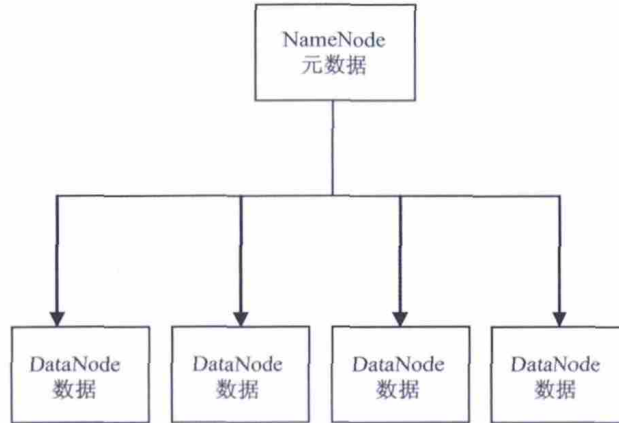


图 1 HDFS 结构

在分布式环境中,存在着一个主节点 NameNode 和多个从节点 DataNode. 主节点存储着文件系统的元数据,比如文件系统的名字空间等,用于向用户映射文件. 但实际的数据并非存储于主节点,而是分布存储在多个从节点上. Hadoop 在 MapReduce 编程模型上的主程序被称为 Jobtracker,子程序被称为 Tasktracker. 主程序负责整个 MapReduce 执行流程的控制工作,由于需要读取文件系统的相关信息,所以它通常位于主节点. 主程序启动之后,会创建位于从节点的子程序. 子程序 Tasktracker 每隔一段时间向主程序 Jobtracker 发送心跳,返回自己的执行状态. 子程序会在从节点创建两种类型的子任务,分别是 map 任务和 reduce 任务. 在 Hadoop 中由于有 HDFS 文件系统的支持,数据是分布式存储在各个节点的,map 任务在并行运行时,各节点读取存储在自己节点的数据进行处理,从而避免了大量数据在网络上的传递,实现了计算向存储的迁移. map 任务生成中间的 <key, value> 对集合,将其存储在本地,并将具有相同 key 的中间值合并发送给 reduce 任务; reduce 任务将合并的值进行相关的计算后产生最后的 <key, value> 对集合. MapReduce 编程模型如图 2 所示.

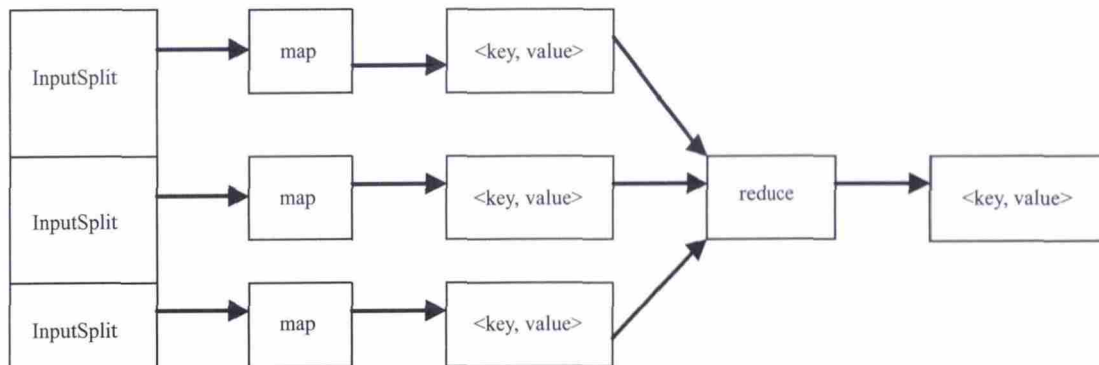


图 2 MapReduce 编程模型

## 2 算法设计

### 2.1 基本思想

从公式(3)和(4)可以看出,HITS算法可以简化为矩阵和向量的乘法.若有 $n \times n$ 维矩阵 $A$ 和 $n$ 维向量 $b$ ,则

$$(Ab)_i = \sum_{j=1}^n A_{ij}b_j \quad i = 1, 2, \dots, n.$$

基于MapReduce的矩阵和向量乘法的算法思想如下:

map阶段:对于矩阵 $A$ 的每个元素 $A_{ij}$ ,生成键值对 $\langle \langle i, 1 \rangle, \langle A_{ij}, A_{ij} \rangle \rangle$ ;对于向量 $b$ 的每个元素 $b_j$ ,生成键值对 $\langle \langle i, 1 \rangle, \langle b_j, b_j \rangle \rangle$ ,其中 $i = 1, 2, \dots, n$ .

reduce阶段:根据MapReduce编程模型的原理,具有相同key的键值对合并后发送给reduce任务,此时即可计算出 $(Ab)_i$ ,并输出键值对 $\langle \langle i, 1 \rangle, (Ab)_i \rangle$ .

### 2.2 算法实现

为了实现基于MapReduce的HITS算法,需要编写3个函数,即map函数、reduce函数和main函数.map和reduce函数分别实现上述算法思想中的map阶段和reduce阶段,main函数则负责向量的精度控制.当同时满足 $\max_{1 \leq j \leq n} |h_j - h'_j| \leq 0.00001$ 和 $\max_{1 \leq j \leq n} |a_j - a'_j| \leq 0.00001$ 两个条件时,就停止迭代.基于MapReduce的HITS算法的伪代码描述如下:

```
map(key, value, context)
{
    if( input is  $A_{ij}$  )
        context.write(  $\langle i, 1 \rangle, \langle A_{ij}, A_{ij} \rangle$  );
    else if( input is  $b_j$  )
    {
        for(  $i = 1; i <= n; i++$  )
            context.write(  $\langle i, 1 \rangle, \langle b_j, b_j \rangle$  );
    }
}

reduce(key, list( value ), context)
{
    //key is  $\langle i, 1 \rangle$ 
    for(  $j = 1; j <= n; j++$  )
        sum + =  $A_{ij}b_j$ ; //  $A_{ij}$ 
    context.write(  $\langle i, 1 \rangle, sum$  );
}

main()
{
    while(  $\max_{1 \leq j \leq n} |h_j - h'_j| > 0.00001 \&\& \max_{1 \leq j \leq n} |a_j - a'_j| > 0.00001$  )
        start an iteration for new  $h$  and new  $a$ ;
    System.exit( 0 );
}
```

## 3 实验

### 3.1 实验平台

采用一台 Intel(R) Xeon(R) cpu E5 - 16070@3.0GHZ 8G 内存 pc 作为主机,在 pc 上安装 5 台 VMware 虚拟机,其中 1 台作为 NameNode(Jobtracker) 节点,另外 4 台作为 DataNode(Tasktracker) 节点. 每台虚拟机安装 linux Red Hat 9.0 操作系统,分配 1G 内存;Hadoop 和 java 的版本分别为 hadoop - 0.20.2 jdk1.6. 实验数据采用来自斯坦福大学的 StanfordBerkeleyWeb 数据集,其中包含 685230 个网页,页面之间存在 7636535 条链接,数据集大小为 143M.

### 3.2 实验分析

#### 3.2.1 分块大小对算法运行时间的影响

在 Hadoop 框架中,分块大小(blocksize) 会直接影响算法执行的效率. 在集群中, map 任务的数量和对输入文件进行分块后的块数是相等的,所以输入文件的大小和 blocksize 决定了集群中 map 任务的数量. 如果 blocksize 太小,则启动的 map 任务过多,就会导致集群负载剧增,过多地损耗资源,降低算法性能;反之,如果 blocksize 太大, map 任务太少,算法就不能充分利用集群资源,不能充分发挥出并行的特性. 因此,在执行算法时,需要合理的配置集群中 blocksize 的大小,在不增加集群负载的情况下充分发挥出并行处理的优势. 实验中采用数据集 StanfordBerkeleyWeb,将 blocksize 分别设置为 16、32、64、128、256M,在不同的 blocksize 下分别执行算法,结果如表 1 所示.

表 1 不同 blocksize 下算法执行时间

blocksize/M	16	32	64	128	256
time/s	79	94	110	124	155

由于集群有 5 台虚拟机,其中 4 个是 DataNode 节点,所以 blocksize 为 16、32、64、128、256M 时,平均每个 DataNode 节点分配到的 map 任务为 2~3 个,1~2 个,0~1 个,0~1 个,0~1 个. 从表 1 可以看出,当 blocksize 为 16M,即每个 DataNode 节点分配 2~3 个 map 任务时,算法运行的效率最高;当 blocksize 增大时,由于没有充分利用集群的并行特性,算法效率逐渐降低.

#### 3.2.2 集群规模对算法运行时间的影响

将 blocksize 设置为 64M,分别在 3 机集群、4 机集群、5 机集群的环境中执行算法,结果如表 2 所示.

表 2 不同集群下算法执行时间

cluster scale	3	4	5
time/s	98	90	79

从表 2 可以看出,适当扩大集群规模,算法运行效率逐渐提高. 这是因为随着集群规模的扩大,集群中可利用的处理器和内存也会随之增加,从而集群也获得了更强大的并行处理能力.

## 4 结 语

提出了一种基于 Hadoop 框架下的 HITS 算法的设计与实现. 算法的设计主要涉及 map 阶段和 reduce 阶段,其中 map 阶段是算法并行化的关键. 在试验中深入分析了集群配置参数 blocksize 和集群规模对算法执行效率的影响,探索了执行 map 任务期间的集群负载和集群的可伸缩性. 在后续工作中,将继续探索云计算环境下的搜索引擎相关算法的设计与实现,并就如何提高算法的性能展开研究.

## 参考文献:

- [1] RAJARMAN A ,ULLMAN J D. Mining of Massive Datasets [M]. Cambridge: Cambridge University Press 2011.
- [2] DEAN J ,GHEMAWAT S. Mapreduce: simplified data processing on large clusters [J]. Communications of the ACM 2008 , 51( 1) : 1 - 13.
- [3] WHITE T. Hadoop: The Definitive Guide [M]. Newton: O' Reilly Media 2010.
- [4] WIKIPEDIA. Hadoop [EB/OL]. [2011 - 10 - 15 ]( 2013 - 04 - 15) . <http://en.wikipedia.org/wiki/Hadoop>.
- [5] 王鹏. 云计算的关键技术与应用实例 [M]. 北京: 人民邮电出版社 2010.
- [6] SHVACHKO K ,KUANG H. The Hadoop Distributed File System [C]//Proceedings of the 26<sup>th</sup> IEEE Symposium on Massive Storage Systems and Technologies ,Piscataway: IEEE Press 2010.
- [7] GHEMAWAT S ,GOBIOFF H ,LENGG S. The google file system [J]. ACM Sigops Operating Systems Review ,2003 ,37 ( 5) : 29 - 43.

## The realization of HITS algorithm based on MapReduce

YU Hui , WANG Xiaomei

( College of Information ,Mechanical and Electrical Engineering ,Shanghai Normal University ,Shanghai 200234 ,China)

**Abstract:** In this paper ,we use Hadoop to redesign HITS algorithm according to the research of HITS algorithm and the cloud computing framework Hadoop which base on MapReduce programming model. Meanwhile ,we analyze the effect of different block-size and cluster scale in influence on the efficiency of the algorithm. The results show that the efficiency reduced with the blocksize increased ,while the efficiency increased gradually when cluster scale had been expanded properly.

**Key words:** HITS; MapReduce; Hadoop; blocksize; cluster

( 责任编辑:包震宇)