

基于染色的分布式弱连通分量求解算法^①

肖有诰, 谭菊仙, 李 璞

(江南计算技术研究所, 无锡 214083)

摘 要: 社交网络、生物信息网络等链接关系数据呈现爆炸性增加, 对亿万个顶级别的大图处理需求愈加迫切, 传统的图算法对单点内存的计算依赖性已经不能满足数据规模急速增长的需求. 本文针对图的弱连通分量求解问题, 提出了一种快速、可扩展的迭代染色算法 CR, 并建立了算法的 MapReduce 模型. 最后, 在 Hadoop 平台上, 对斯坦福大学社会网络分析实验室提供的四组通联数据进行测试, 并和开源的数据挖掘工具箱 XRIME 提供的算法进行了对比.

关键词: 弱连通分量; 染色算法; MapReduce; XRIME

Distributed Coloring Algorithm for Weakly Connected Components

XIAO You-Gao, TAN Ju-Xian, LI Pu

(Jiangnan Institute of Computing Technology, Wuxi 214083, China)

Abstract: With the explosively increase of social networking, bioinformatics network and other relational datas. It is urgent to improve the processing capability on large scale graphs with billions of vertices. The traditional graph algorithms can no longer meet the rapidly growing demand for the calculate dependence of the memory. To find the Weakly Connected Components of a graph, this paper presented a fast, scalable and iterable coloring algorithm CR and also established the MapReduce model. At last, we give out a contrast experiment between the open-source data mining toolbox XRIME and CR on four sets of communication data, which is provided by the Social Network Analysis Laboratory in Stanford University.

Key words: weakly connected components; CR; MapReduce; XRIME

无向图连通分量的挖掘可以用来实现社团划分、模式发现、地图导航等具体应用^[1-2]. 传统的求解无向图连通分量的方法主要有两种: 遍历方法和并查集方法^[3]. 遍历的方法通过遍历图中所有的节点, 进而得到独立的子图, 即连通分量. 遍历可以采取两种方法进行: 深度优先搜索(DFS)和广度优先搜索(BFS), 对于 V 点 E 条边的有向图, 其时间复杂度是 $O(V+E)$. 但搜索都需要将全图信息存储于内存中, 不能适应在数据爆炸环境下的大图处理, 同时搜索难以实现分布式化; 并查集方法是通过合并具有等价关系的节点对, 进而得到连通分量. 并查集方法是 on-line 的, 虽然单机算法同样依赖内存, 但这种方法容易转化为分布式计算模型. 为了降低时空开销, 设计高效可扩展的分

布式大图算法越来越受到人们的关注, 也取得了很多成绩. David 和 Sartaj[4]在网格并行机上寻找图的连通分量, 对于 (V,E) 无向图, 假设节点的最大度为 d 且 $V=n^q=2^p$, 则在一个 E 维的 $n \times n \times \dots \times n$ 的网格并行机上, 算法的时间复杂度是 $O(q^3(q+d)n \log n)$; 北京邮电大学的杜雅红在 Hadoop 平台上实现了分布式标签传递算法^[5], 通过多次迭代, 使得同一连通分量内部的节点具有相同的标签. 在机器数足够的情况下, 算法的运行时间不会随着数据量的增长而线性增长; 由于 MapReduce^[6]计算模式受限于易并行计算模型的约束而不易表达图算法, 文献[7]结合 MapReduce 计算模型和基于消息传递机制的块同步模型(BSP), 设计了基于 Hadoop 的支持块同步编程规范的并行编程模型, 大

^① 收稿时间:2013-09-12;收到修改稿时间:2013-11-18

幅降低了大图算法的处理时间;此外,还有 Apache 的 Hama^[8]和 Mahout、Google 的 Pregel^[9]等编程框架,都对大图算法的分布式实现进行了研究.为了更高效的挖掘的大图的连通分量,本文提出了一种快速的染色算法,该算法能够很好的适应 MapReduce 编程模型.最后,在 Hadoop 平台上,针对 SNAP 提供的四种网络联通数据,和开源的 XRIME 工具箱进行了对比实验.

1 图的连通性

在无向图中,若从顶点 u 到顶点 v 有路径,则称点 u 和 v 是连通的.如果无向图中任意一对顶点都是连通的,则称此图是连通图;相反,如果一个无向图不是连通图,则称非连通图.若非连通图 G 的子图 g 是连通的, g 中节点到图 G 中剩余节点之间都没有路径,则称子图 g 为 G 的弱连通分量.如下图 1 所示,由于节点 1 和节点 4 之间没有路径,为一个非连通图,其中子图 {1、2、3、5} 和子图 {4、6、7} 是图的弱连通分量.在由连接关系构成的社团分析应用中,一种方法是将连接图的弱连通分量看作相互之间独立的社团.

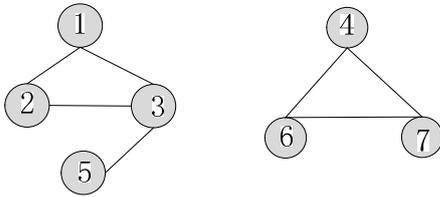


图 1 非连通图与弱连通分量

2 定义

定义 1. 等价关系

“同属一个集合”关系是一个等价关系,因为它满足等价关系(Equivalent Relation)的 3 个条件.

- (1) 自反性: 如 $X \equiv X$, 则 $X \equiv X$;
- (2) 对称性: 如 $X \equiv Y$, 则 $Y \equiv X$;
- (3) 传递性: 如 $X \equiv Y$ 且 $Y \equiv Z$, 则 $X \equiv Z$;

在图问题中,一个弱连通分量可视为一个等价的集合.若节点 X 和 Y 同属一个连通分量中,则 $X \equiv Y$, 即 X 和 Y 是一个等价对.

定义 2. 等价类

等价类(Equivalent class): 设 R 是集合 A 上的等价关系,对任何 $a \in A$, 集合 $[a]_R = \{x | x \in A, 且 aRx\}$, 称元素 a 形成的 R 等价类,其中, aRx 表示 a 与 x 等价. 所

谓 a 的等价类,就是所有跟 a 等价的元素构成的集合.

将等价类用于弱连通子图发现时候,同在一个弱连通分量的元素构成一个等价类,有联系的节点对称称为等价对.设初始时有一集合 $S = \{1,2,3,4,5,6,7,8\}$;依次读若干事先定义的等价对 $1 \equiv 5, 4 \equiv 2, 8 \equiv 5, 4 \equiv 6$;现在需要根据这些等价对将集合 S 划分成若干等价类.

在每次读入一个等价对后,把等价类合并起来.初始时,各个元素自成一个等价类(用 $\{ \}$ 表示一个等价类).在每读入一个等价对后,各等价类的变化一次如下:

初始: $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}$.

$1 \equiv 5$: $\{1,5\}, \{2\}, \{3\}, \{4\}, \{6\}, \{7\}, \{8\}$.

$4 \equiv 2$: $\{1,5\}, \{2,4\}, \{3\}, \{5\}, \{6\}, \{7\}, \{8\}$.

$8 \equiv 5$: $\{1,5,8\}, \{2,4\}, \{3\}, \{6\}, \{7\}$.

$4 \equiv 6$: $\{1,5,8\}, \{2,4,6\}, \{3\}, \{5\}, \{7\}$.

通过等价对信息逐步合并等价类的方法也叫做并查集方法.并查集方法需要根据等价对信息,遍历全图信息,进而合并相应的等价类,对于 n 点 m 条边和包含 k 个分量的图,其空间开销是 $O(m+n)$;同时,串行的并查集方法每次只读取一对等价对信息,时间开销也会很大.

定义 3. 着色函数

着色函数 $F: f(v_1, v_2, \dots, v_n) \rightarrow v$ 用来为一组具有等价关系的节点着色,是一个多对一的推荐函数,输入为点的向量,输出为一个唯一值作为这组节点的颜色.在下文中,用 $Color(v_i)$ 表征节点 v_i 的当前颜色.

同时,着色函数 F 满足如下约定:

- 1. $f(v_1, v_2, \dots, v_n)$ 有意义当且仅当对任意 $v_i \in (v_1, v_2, \dots, v_n)$, 存在 $v_j \in (v_1, v_2, \dots, v_n)$, 满足 $v_i \equiv v_j$;
- 2. V 和 X 为任意点向量,若 $f(V) = v'$ 且 $f(V, X) = x'$, 则 $f(v', X) = x'$. 即对于各组节点组合,推荐颜色具有传递性;
- 3. 若 $f(V_1, V_2) = v'$, 则 $f(V_1) = v'$ 或者 $f(V_2) = v'$.

3 染色算法

并查集方法是串行的将等价类合并成最终的社团,染色算法 CR 通过着色函数,将等价类中所有的节点染成同一种推荐颜色,进而标记每个社团.通过将数据分块,首先块内染色,然后块间调色,迭代的进行染色调色操作,直到同一连通分量内部的成员都被染

成同一个颜色。

染色算法的具体步骤为:

- ①根据原则具有相同颜色的节点即互为邻接点, 规约出各个节点的邻接点, 初始状态每个节点及其直接连接点;
- ②染色: 利用色彩函数, 计算各个节点及其邻接点的推荐颜色值, 作为这组节点的颜色;
- ③调色: 由于单个节点在每次迭代过程中可能归

属多个等价类, 每个节点在染色阶段可能会被染成多种颜色并被标记, 调色即为这些具有混合颜色的节点及以它为颜色的子节点推荐出新的颜色;

- ④判断调色阶段有无颜色变化. 若有, 迭代的进行步骤 1, 2, 3; 若无, 算法迭代结束;
- ⑤聚集: 按照各个节点的颜色进行分发, 具有同一颜色的节点即归属于同一个弱连通分量。

下图 2 用一个实例演示了染色算法的具体过程:

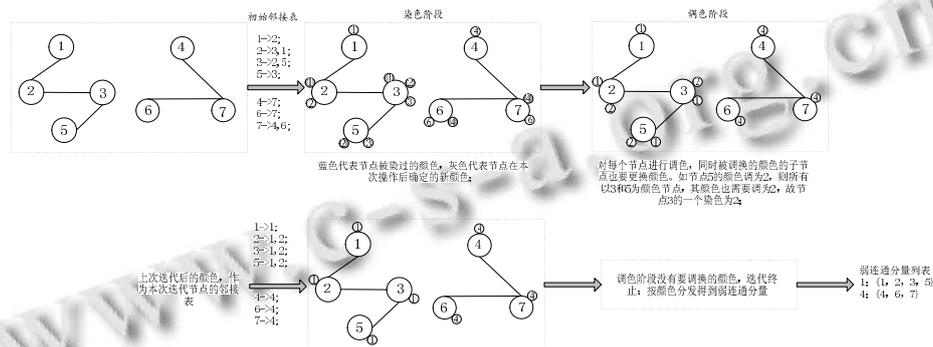


图 2 染色算法图示

4 染色算法伪代码

染色算法的伪代码如下所述:

符号说明: C 指图中的节点集合, $T(V)$ 指全图的邻接关系表, $t(v)$ 指节点 v 的邻接表, C_v 指节点 v 的颜色值, $S(v, c_v)$ 指每次染色后, 颜色有变化的节点及颜色, F 指推荐函数。

Step1:初始化邻接表;

Step2:染色阶段

for each v in V

$C_v = F(t(v));$ //从邻接链表内部成员中推荐

颜色

for each v_i in $t(v)$

$S.add(v_i, C_{v_i});$ //记录推荐色有变化的成员

员

end

end

Step3:调色阶段

for each v_i in S //节点颜色 v_i 有变化

for each v_j in V

if $(C_{v_j} == v_i)$

$C_{v_j} = C_{v_i};$ //更新原具有相同颜色的

节点

end

end

end

Step4:更新邻接表

for each v in V

if $(C_{v_i} == C_{v_j})$ //若两节点颜色值相同

$t(v_i).add(v_j);$ //将对方添加到邻接表中

$t(v_j).add(v_i);$

end

end

Step5:判断迭代是否结束

if $(S.size > 0)$ //若 S 非空, 返回染色阶段; 否则算法结束.

goto Step2;

else

get the v that has the same c_v together as a component; //迭代结束后, 具有相同 c_v 值的节点即同属一个连通分量.

5 染色算法的MapReduce模型

染色算法是由染色和调色组成的迭代算法, 算法从单个节点与其邻接点分析, 在具有直接联系的节点间进行染色, 在调色阶段将间接联系的节点对转化为

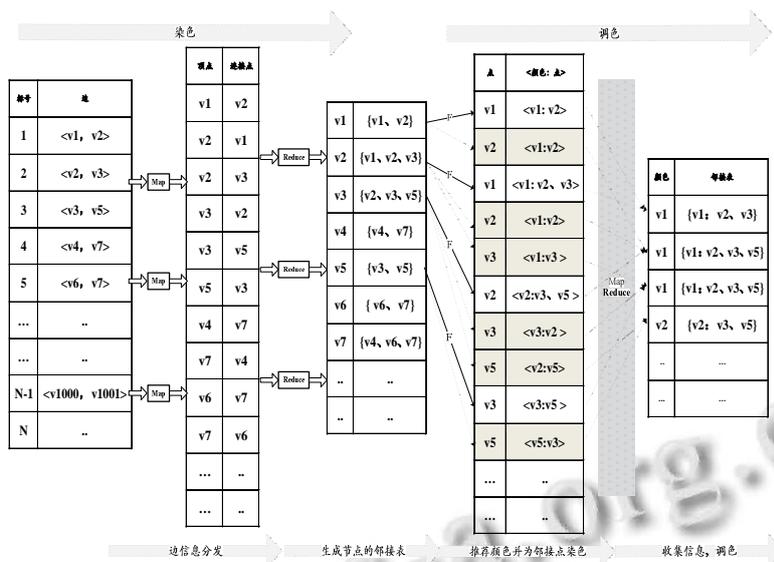


图 3 染色算法的 MapReduce 模型

直接节点对. 如此, 迭代的进行染色调色操作, 直到同一颜色扩散到弱连通分量中的所有节点, 进而达到查找所有弱连通分量的目的.

文章前面对算法步骤进行详细简单的图示并且给出了算法的伪代码, 可以看出, 着色函数的计算限于对同一弱连通分量内部的节点之间, 并且染色依赖的 <节点、邻接边>和调色依赖的<颜色、成员>结构都非常适合 MapReduce^[2-3,5]的键值对分发规约处理模型. 图 3 展示了一种染色算法的 MapReduce 处理模型.

6 实验结果与分析

6.1 实验环境

实验是在 5 个节点构成的 Hadoop 环境中进行, 硬件环境如表 1 所示.

表 1 Hadoop 实验环境

	CPU	Memory/GB	Map Task Capacity	Reduce Task Capacity
Master Node(1)	X5660 6(24) @ 2.80GHz	16		
Slave Node(4)	X5660 6(24) @ 2.80GHz	16	24	24

1 个管理节点与 4 个工作节点上都配置了 Hadoop 0.20.2 版本.

6.2 数据集

测试数据源之斯坦福大学社会网络分析实验室提

供的网络数据, 其中, soc-Epinions*是 Epinions.com 网站提供的人们通过购物关系信任信息形成的关系网络, LiveJournal 是网络社区通联数据, 此外还有 Slashdot 科技资讯信息. 这几种数据集都以 <FromNodeId ToNodeId>二元组集合的形式提供网络的无向图拓扑. 处理后拓扑图, 其主要 WCC 信息如下表 2:

表 2 测试数据信息

数据集名称	点数	边数	WCC 个数	最大 WCC 点数	最大 WCC 边数
soc-Slashdot0811	77360	905468	1	77360	905468
soc-Slashdot0922	82168	948464	1	82168	948464
soc-LiveJournal1	4847571	68993773	4	4843953	68983820
soc-Epinions1	75879	508837	2	75877	508836

6.3 实验设计与结果分析

针对上面介绍的测试数据, 在相同的 Hadoop 环境中, 分别用 XRime 提供的算法和分布式 CR 算法求解弱连通分量. XRime^[10]是北京邮电大学和 IBM 合作在 Hadoop 平台上开发的图算法工具箱, 它分别提供了用深度优先遍历(DFS)和广度优先遍历(BFS)求解弱连通分量的接口, 实验中采用的是深度优先遍历方法. XRime 工具包需要并提供了将二元组<FromNodeId ToNodeId>类型的测试数据转化为<node,List<edge>>数据类型的接口. 下图 4 为两种算法的迭代次数和运行时间对比图.

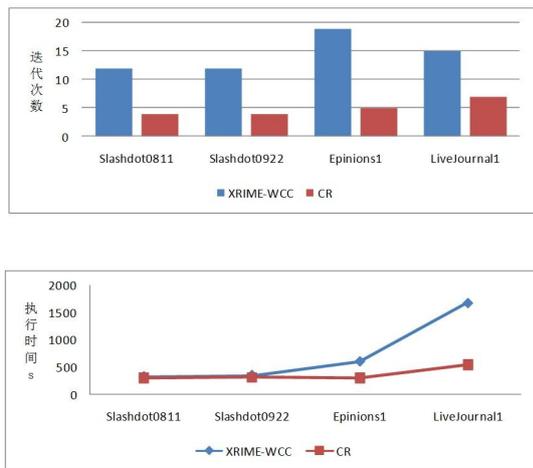


图 4 XRIME-WCC 和 CR 的实验对比

如图 4 所示, 针对实验数据, 分布式 CR 算法的迭代次数和执行时间要比 XRIME 要好, 因为 CR 算法在每次迭代时将节点的所有等价类颜色信息都发布出去, 通过对节点调色能够将多个子等价类进行合并, 同时对所有节点颜色标记更新. 同时由图 4 可以推断, 在数据量继续增长的情况下, 分布式 CR 算法的时间性能提高会更好.

7 结语

本文结合目前分布式大图算法的实现困难和执行性能低的现状, 针对非连通图的弱连通分量求解问题, 提出了一种快速的染色算法. 并在开源的 Hadoop 基础上实现了分布式染色算法, 通过在每次染色后对节点的颜色信息进行冗余分发, 能够有效的降低算法迭代次数和执行时间. 实验证明, 和开源的 XRIME 工具箱提供的弱连通分量算法, 分布式 CR 算法具有较好的时间性能, 并且随着数据量的增加, 分布式 CR 算法的时间性能提高会更好. 为了提高算法的时间效率, 分布式 CR 算法在每次分发颜色信息的时候会有冗余, 其需要的 Map 和 Reduce 任务会比 XRIME 多. 寻找更加合理高效的染色调色原则, 降低中间数据量将是下一步的工作重点.

参考文献

- 1 丁国栋, 王斌, 白硕. Web 超链挖掘: 中国境内 Web 图结构研究. 计算机工程, 2005, 7: 31(14): 24-26.
- 2 田晶, 宋子寒, 艾廷华. 运用图论进行道路网网络模式提取. 武汉大学学报(信息科学版), 2012, 6: 37(6): 724-727.
- 3 王贵平, 王衍, 任嘉辰. 图论算法理论、实现和应用. 北京: 北京大学出版社, 2011.
- 4 Nassimi D, Sahni S. Finding connected components and connected ones on a mesh-connected parallel computer. SIAM Journal on Computing, 1980, 9: 744-757.
- 5 杜雅红. 基于云计算平台的图算法研究[学位论文]. 北京: 北京邮电大学, 2011: 14-19.
- 6 Amazon Elastic MapReduce. <http://aws.amazon.com/elasticMapReduce/>. 2010-09-26.
- 7 潘巍, 李战怀, 伍赛, 陈群. 基于消息传递机制的 MapReduce 图算法研究. 计算机学报, 2011, 10: 34(10).
- 8 Seo S, Yoon EJ, Kim J, Jin S. HAMA: An efficient matrix computation with the MapReduce framework. Proc. of the Cloud Computing Technology and Science Conference (CloudCom'10). Indianapolis, USA. 2010. 721-726.
- 9 Malewicz G, Austern MH, Bik A, Dehnert JC, Horn I, Leiser N, Pregel. A system for large-scale graph processing. Proc. of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD'10). Indianapolis, Indianam, USA. 2010. 135-146.
- 10 X-RIME: Hadoop based large scale social network analysis. <http://xime.sourceforge.net/>.
- 11 Amazon Introduces Elastic MapReduce Service. <http://www.byteonic.com/2009/amazon-introduces-elastic-MapReduce-Hadoop-framework-service/>.
- 12 Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters. Google, Inc. OSDI 2004[To Appear].