|  |  |
| --- | --- |
| **学校代码** | **10699** |
| **分 类 号** | **TP311** |
| **密 级** | **公开** |
| **学 号** | **2012201676** |

题目 大规模数据并行图处理算法与计算平台研究

|  |  |
| --- | --- |
| **作者** | **尤立** |

|  |  |
| --- | --- |
| **学科、专业** | **计算机软件与理论** |
| **指 导 教 师** | **陈群** |
| **申请学位日期** | **2015年2月** |

# 

西北工业大学

硕士学位论文

(学位研究生)

题目：大规模数据并行图处理算法

与计算平台研究

作 者： 尤立

学科专业：计算机软件与理论

指导教师： 陈群

二零一五年二月

Northwestern Polytechnical University

A Dissertation Submitted for the

Master Degree

Title: Research of parallel graph algorithms over big data and processing platform

Author YouLi

Specialty Computer Software and Theory

Supervisor CHENQun

February 2015

# 摘要

近年来，大数据已经成为各大公司追逐的热点领域。随着大数据时代的到来，大量、多样、变化的数据给业界带来诸多机遇和挑战。很多热门研究领域，如社交网络分析、Web文档聚类、实体识别、基因表达分析等应用中，都必须对海量数据进行分析和挖掘。这些应用中通常都包含较复杂的网络结构，实际生产过程中，人们通常使用图来抽象系统中的各个实体之间的关系结构，并通过对抽象的图挖掘从数据中提取有价值的知识。在这些应用中关于图的完全图和近似完全图挖掘已经成为不可或缺的重要组成部分。

完全图和近似完全图是经典的NP Complete问题，算法时空复杂度都很高。人们一直以来就致力于找出它们的高效解决算法。过去的研究已经提出了一些在较小数据集上表现良好的算法。但是在大数据环境下，单机算法受制于内存和磁盘的限制往往不能够满足现实生产需求，并行化是必然趋势，因此研究集群环境中大规模数据的并行图处理算法尤为重要。本文在充分了解现有的算法基础上，提出了一种基于图分割的完全图和近似完全图枚举算法（BGP，Binary Graph Partitioning），并分别在单机及并行环境中在多种实际和生成数据上，通过实验与经典算法（BK，Bron-Kerbosch）对比，验证了算法的可行性及高效性。在此基础之上，根据生物基因数据的特征提出了一种BK辅助BGP的Hybrid算法，在各种数据上都能够有更优的表现。

在并行环境中，本文选择利用当今流行的大数据计算框架MapReduce的开源实现Hadoop作为并行算法的运行平台。针对并行环境中完全图和近似完全图计算过程中出现的负载不均问题，本文提出了一种负载均衡的算法，实验证明此负载均衡算法得到了较好的均衡性。最后，本文进一步探讨了Hadoop系统中的负载均衡问题，提出静态和动态两种系统级负载均衡方案。

**关键词：**完全图，近似完全图，并行，负载均衡，MapReduce, Hadoop

# Abstract

The era of big data has come. A large,diverse data is generated so quickly that it presents a tremendous challenge to IT industry. MapReduce model which is proposed by Google becomes the initial model for the processing of large-scale data. Apache's Hadoop is an open source implementation of MapReduce model.Because of its high reliability, high scalability, high efficiency and high fault tolerance ,Hadoop becomes a hot platform rapidly. Hadoop cluster load balance has also become a hot topic in the field of distributed computing. This paper studies methods of Hadoop cluster load balance management in data skew environment.

**Keywords:**data skew,load balance,Hadoop,multi-table join,data localization.

# 目录

[摘要 I](#_Toc405367898)

[Abstract II](#_Toc405367899)

[目录 1](#_Toc405367900)

[第一章 绪论 1](#_Toc405367901)

[1.1 研究背景 1](#_Toc405367902)

[1.2 选题意义 1](#_Toc405367903)

[1.3 研究现状 1](#_Toc405367904)

[1.3.1 完全图单机算法 2](#_Toc405367905)

[1.3.2 完全图并行算法 2](#_Toc405367906)

[1.3.3 近似完全图单机算法 2](#_Toc405367907)

[1.3.4 近似完全图并行算法 2](#_Toc405367908)

[1.3.5 Hadoop系统负载均衡 2](#_Toc405367909)

[1.4 论文的内容及组织 2](#_Toc405367910)

[1.4.1 主要研究内容及成果 2](#_Toc405367911)

[1.4.2 论文组织结构 2](#_Toc405367912)

[第二章 相关理论及技术 5](#_Toc405367913)

[2.1 完全图和近似完全图 5](#_Toc405367914)

[2.1.1 图论相关概念 5](#_Toc405367915)

[2.1.2 完全图 5](#_Toc405367916)

[2.1.3 近似完全图 6](#_Toc405367917)

[2.1.4 负载均衡的基本概念 6](#_Toc405367918)

[2.3 并行计算模型MapReduce 6](#_Toc405367919)

[2.4 开源分布式计算平台Hadoop 6](#_Toc405367920)

[2.5 图数据表示 6](#_Toc405367921)

[2.6 本章总结 7](#_Toc405367922)

[第三章 完全图和近似完全图检测算法 8](#_Toc405367923)

[3.1 完全图检测算法 8](#_Toc405367924)

[3.2 近似完全图检测算法 8](#_Toc405367925)

[3.3 Hybrid算法 9](#_Toc405367926)

[3.6 本章总结 9](#_Toc405367927)

[第四章 并行环境下的负载均衡 10](#_Toc405367928)

[4.1 完全图和近似完全图检测的负载均衡 10](#_Toc405367929)

[4.1.1中间结果集的膨胀问题 10](#_Toc405367930)

[4.2 并行平台Hadoop中通用的负载均衡 10](#_Toc405367931)

[4.3 本章总结 10](#_Toc405367932)

[第五章 实验部分 11](#_Toc405367933)

[5.1 实验条件及环境搭建 11](#_Toc405367934)

[5.1.1 实验条件 11](#_Toc405367935)

[5.1.2 搭建Hadoop集群 11](#_Toc405367936)

[5.2实验内容以及结果分析 14](#_Toc405367937)

[5.2.1 多表连接策略优化前后系统的性能对比 14](#_Toc405367938)

[5.2.3 Hadoop负载均衡系统参数调优及与原Hadoop的性能对比 16](#_Toc405367939)

[5.4 本章总结 16](#_Toc405367940)

[第六章 总结与展望 17](#_Toc405367941)

[6.1 本文研究总结 17](#_Toc405367942)

[6.2 课题研究展望 17](#_Toc405367943)

[参考文献 19](#_Toc405367944)

[在校期间发表论文 20](#_Toc405367945)

[致谢 22](#_Toc405367946)

[知识产权说明书 24](#_Toc405367947)

第一章 绪论

1.1 研究背景

随着互联网尤其是移动互联网的快速发展，社会化交际网络的快速普及，物联网、商业销售、Web语义分析、生物网络信息等相关应用的丰富，云计算相关技术的快速发展，更多的设备被添加到网络中，数据资源呈指数增长。正如人们所说：大数据时代到来了。在大数据时代，对于企业来讲最重要的是获得和使用数据，能否从现有的海量数据中快速、高效、准确地分析提取有用信息已经成为企业能否在新环境下取得成功的关键因素。早在2010CNNIC统计时，仅中国的网页规模就已经达到600亿，且还在以每年78.6%的比率增长；Web搜索领域Google索引了全球超过500亿的网页，每天有20PB数据要处理；同时社交网络的后来居上，Facebook已经超过7亿用户；在环境和气象领域，需要从数以万计的传感器中接受大量数据以进行监控和预测。正是有如此庞大和迫切的需求，快速高效的海量数据处理日益受到学术界和科技界的关注。

在各种各样的海量数据应用中普遍存在网络连接结构，如Web中网页的链接关系，社交网络中人与人之间的好友关系，基因表达数据中各个基因之间的协同表达关系等。而图正是数学上描述各个实体之间的网络关系的一种经典结构，如此总总应的用都可以抽象成关于图的计算。

图论（Graph Theory）作为数学的一个重要分支，以图为研究对象，研究顶点和边组成的图形的数学理论和方法。图论中所研究的图是对现实网络的抽象，由顶点和边构成，图中的顶点代表实现事物，用连接两点的边来表示顶点所代表的事物之间具有某种关系。早在1736年数学家欧拉关于柯尼斯堡七桥问题的论著中就有关于图论的研究记录，这个问题后来被推广为著名的欧拉回路问题。经过二百多年的发展，图论已经成为一个独立的数学研究分支。图论中的问题主要可以概括为子图问题、染色问题、路径问题、网络流与匹配问题以及覆盖问题。这些问题里各自出现了许多经典的代表，如哈密顿回路、最大团、四色问题、斯坦纳树、最短路径、中国邮递员问题、最小覆盖集等，也出现了许多如戴克斯特拉算法、克鲁斯卡尔算法、拓扑排序等等经典算法。然而经过研究证明，很多图相关的问题都是NP或者NP Complete问题，算法有很高的时间和空间复杂度。在过去的现实需求中数据量较小，传统算法可以具有较好的实用性。但在大数据时代，数据的采集、获取变得非常方便，数据中所蕴含的科学与商业价值推动着对于海量数据的图处理需求迅速提高。传统的单机算法受制于内存和磁盘的限制，尽管不断优化，还是不能够在有意义的时间内完成处理。同时现有的多机并行算法通常利用冗余来保证计算的完整和正确，很难达到较高的并行度，或者在计算过程中带来严重的数据膨胀问题。

1.2 选题意义

随着图在社交网络、生物信息、实体识别等领域广泛应用，业界对高效的图处理算法有着迫切的现实需求。完全图和近似完全图常常用于进行分类和聚类分析，例如在社交网络中，通过检测完全图和近似完全图，从而找到关系十分紧密的群体，群体之间往往具有相同的兴趣爱好或者消费习惯，因此在个性化服务和投放广告方面能做的更加地准确有效，从而增加用户的粘性，提升网站的流量和商业价值都；在生物信息领域，不同基因之间的协同表达是一个重要的研究问题，常常作为新型药物研制的突破口，是生物工程中常用的技术，完全图和近似完全图的检测则是分析基因协同表达中的重要技术，这些技术的应用能够有效缩短新药的研制时间和成本。

在大规模数据的背景下，TB级甚至PB级的大规模数据是比较常见的情况，这些问题难以在单机上进行计算，并行算法成为必然选择。并行算法的过程中经常会出现数据不均衡或者计算不均衡的情况，严重制约平台的吞吐量和计算效率，负载均衡已经成为并行和分布式计算平台研究中最重要的方面之一。

尽管对于完全图与近似完全图的枚举、并行环境的负载均衡等问题现在已经有许多优秀的算法和方案[10][12][14][16][18][20][33]，然而都存在性能瓶颈和难以适用情况，又或者是针对某些具有特定特征的数据定制，没有较好的通用性。文献[1]是Bron和Kerbosch完全图枚举的经典算法（BK），在BK的基础上发展、衍生了一系列算法，文献[3][5]通过实际应用显示BK算法的效率要普遍好于现有的其他算法。然而在本文的研究过程中发现BK算法在许多数据上依然存在搜索空间大，速度慢等等问题。文献[2],[4]等并行完全图算法虽然一定程度上缓解了单机计算能力的不足，但存在中间数据过度膨胀，冗余状态量大，需要通过大量的机器数目来存储和计算，资源利用效率低。另外现有的分布算法并行程度低，且当出现负载不均衡的情况，如社交网络数据中少数名人的好友关系非常大，而大部分普通用户关系简单，这种差异会导致系统少数机器有性能瓶颈从而影响整体执行效率，降低系统吞吐量。现有的近似完全图算法一般是借鉴完全图的解决方案稍作调整，但是目前还没有很好的并行算法。已有的近似完全图并行算法[6][7]是在特定的限制环境下的特殊解决方案。同时在近似完全图中同样也存在和完全图枚举过程中的负载不均衡问题。并行完全图和近似完全图枚举问题中所共有的负载不均衡的问题通常有两种解决思路：根据特定问题（本文中即完全图和近似完全图枚举）设计方案，在各个机器节点间互相分担高负载节点的任务量；对计算平台做相应修改，使计算平台本身可以达到自动均衡负载的功能。第一种方案具有简单、易实现的优势，但是需要设计者对输入数据特征和特定算法有较好的了解，而且这样的算法并没有通用性，需要为每个算法重新设计均衡策略；第二种方案，从平台上解决了负载均衡的问题，可以有较好的通用性，只要任务是可以切分的，系统可以自动去均衡各个机器节点的负载，而无需用户介入，缺点是实现比较复杂。

完全图枚举和近似完全图枚举都是经典的NP Complete问题一个问题的有效解决可以对其他NP Complete问题有借鉴意义，同时本文所使用的并行环境下的负载均衡方案既有针对完全图和近似完全图的特定应用方案，也有通用的Hadoop系统中负载均衡问题解决方案，可以用来解决分布式计算平台Hadoop中大部分负载不均衡问题。综上所述，本文的选题具有较大的研究意义。

1.3 研究现状

本文主要涉及图论中的两个典型问题最大完全图枚举（Maximum Clique Enumeration，MCE）和最大近似完全图枚举（Maximum Qusi-Clique Enumeration）。这两个问题自提出以来就已被广泛研究，各有单机和并行算法两类算法。分布式计算平台Hadoop上的负载均衡问题近几年也有不少研究成果，下面将从这五方面介绍现有的研究现状做介绍。

1.3.1 完全图单机算法研究现状

完全图中的所有节点都与除自身外的其他节点相邻。在实际应用过程中尤其是计算科学相关领域中最有价值的是从杂乱无章的大量数据图中找出所有的最大完全图。这个问题也因Luce和Perry在社交网络分析中使用Clique表示团体中每一个人都认识其他所有人也成为最大完全图枚举（MCE）。

抽象的图论算法中有多种最大完全图枚举算法，以及一些与之相关的数学性质。这些算法主要分为三类：

第一类是以Bron-Kerbosch为典型代表的基于回溯搜索并应用剪枝算法降低搜索空间。这类算法[1][15]利用完全图必然是邻接点子集的特征，算法每次从候选点集中选取一个节点与现有的节点构成完全图，直到候选节点中不再有点可以与现有结果组成更大的完全图，最后在递归回溯搜索。

第二类类算法[9][11][13]使用反向搜索策略，这类算法的主要特征是可以较容易地得到算法关于结果集中完全图个数的复杂度上界。

第三类算法是基于“组装”的算法[21][22]，将简单的子图自动组装拼接成更大的单元，最终形成最大完全图。该算法首先将图分解，分解过程中如若产生完全图则直接输出，其他的子图按照特定的方式组装若能组装成完全图则输出。

另外还有如Thang Nguyen Bui等提出的基于遗传编程算法的完全图检测[17]。算法将网络连接关系图用树的形式表示，完全图即使从根到叶子节点的路径上所涉及到的点集，找出一个点的邻接节点非常方便。根据完全图的性质结合遗传编程的特点可以较快的剪去许多不符合条件的节点，减少候选节点数量提高搜索速度。同时由于图以树的形式表示可以借鉴许多在树上研究得较为成熟的技术。James Cheng等根据大规模图数据中完全图检测的前提下提出一种优化的存储结构H\*--graph[19]，该结构定义图的核心部分及核心部分所邻接的节点。将其他节点数据存在外存磁盘中，可以有效的限制大规模数据的情况下单机完全图检测过程中对内存的消耗。同时H\*--graph只保存了数据图中的核心部分，数据的更新代价大幅降低，避免了传统算法中数据更新时需要重新计算的缺点。

在图论领域有不少关于完全图的性质可以应用到实际的完全图检测算法中。Turon指出，如果一个图有足够多的边，那么它一定包含一个最大的完全图；若一个图中包含条边，则必然好汉一个三个节点的完全图。Moon和Moser指出，在一个包含3n个节点的图中，至多包含个最大完全图（Maximum Clique）。拉姆齐理论指出每个图或者其补图包含一个至少有对数个节点的完全图。

1.3.2 完全图并行算法研究现状

随着近十几年互联网、生物信息学、社交网络、移动互联网等众多新兴技术的迅速发展，数据的规模变得越来越大。为了适应变化的需求，原本在单机环境中的完全图检测算法也在近些年被人们扩展到并行环境中。这些并行完全图检测算法[23][24][25][26][27]的基本思想都是将图或者搜索空间切分为独立的、较小的子图或子空间，然后对于这些切分出来的子图或子空间将它们分散到多个机器节点上各自进行计算，以提高整体的计算效率。

S Szabo提出一种基于分解邻接矩阵的并行完全图检测算法[29]，该算法充分利用完全图的性质将邻接矩阵不断切分的数据单元，并将其分散到不同的节点上计算。

Matthew C.Schmidt等人提出一种基于多核并行的完全图检测算法[28]。该算法使用共享内存来存储全局信息，每个核负责处理部分图数据，各个核将处理出来的结果汇总合并成为最终结果。此算法可以处理较大数据量的图且可以通过核数量的扩展有接近线性的并行性，但是对系统的硬件性能有很高的要求。

之前的算法大多运行在多核机器或者MPI平台上，自从MapReduce计算模型的开源平台Hadoop兴起后，其良好的扩展性、稳定性，适应于异构集群、普通计算节点的特性带来了一波新的浪潮。Shengqi Yang等首先提出了使用MapReduce模型在Hadoop平台上进行完全图检测的并行算法[4]。该算法首先提取数据图中每个节点的两跳邻接点（相邻节点以及相邻节点的相邻节点），并将其发往不同的计算节点进行计算，各个计算节点上使用传统单机算法进行检测。

1.3.3 近似完全图单机算法研究现状

1.3.4 近似完全图并行算法研究现状

1.3.5 分布式计算平台Hadoop系统负载均衡研究现状

1.4 论文的内容及组织

1.4.1 主要研究内容及成果

本文相关研究受资助与国家自然基金重点—数据密集型计算环境下的数据管理方法与技术研究（基金号: 61033007）资助。笔者的主要研究方向是数据密集型计算环境下的图数据处理算法，以及并行处理平台的研究。本文主要关注大规模数据环境下并行图处理算法的研究以及并行环境中的负载均衡问题。主要包括完全图枚举、近似完全图枚举和负载均衡问题，期望能够为并行环境中高效解决图处理算法提供思路和经验。

1.4.2 论文组织结构

本文的组织结构如下：

第一章是绪论部分，从整体上介绍本文的研究背景，主要介绍了完全图枚举、近似完全图枚举和负载均衡三个问题的研究现状，最后简要介绍了本文的研究内容和研究成果。

第二章是相关理论及技术介绍，包含完全图、近似完全图、负载均衡相关概念，MapReduce并行计算框架和其开源实现Hadoop。

第三章对完全图和近似完全图枚举算法进行了深入分析，选取最具代表性的经典算法做为对比实验。分析了现有算法的优缺点。提出了一种新的完全图和近似完全图算法。并结合两种算法各自的优劣提出一种可以充分发挥两者优势的Hybrid算法。同时论述了这几种算法的并行实现。

第四章在第三章的基础之上分析了算法并行化实现过程中发现的负载均衡问题，提出了针对完全图和近似完全图算法的负载均衡方案。最后进一步深入探讨了并行平台中的负载均衡问题，提出静态和动态两种通用的负载均衡方案。

第五章是论文的实验部分。本章选取了一系列真实数据和生成数据通过多种维度的实验验证了本文提出的算法的高效性。同时通过实验验证了负载均衡方案的有效性。

第六章是对本文工作的总结以及未来工作的展望，总结全文的主要内容以及今后可以改进的地方及提升的空间。

第二章 相关理论及技术

2.1 完全图和近似完全图

Hadoop是一个开源项目，该项目由Apache倡导并开发的。HDFS是Hadoop平台的文件系

拷贝一个，来保证正确副本的数量。

(5)低成本。

2.1.1 图论相关概念

HDFS，全

NameNode，即名字节点，该节点时HDFS的主控节点，负责对文件和数据节点的管理。

2.1.2 完全图

**(1)MapReduce架构**

同HDFS一样，Hadoop MapReduce也采用了Master/Slave（M/S）架构，具体如图2-2所示。它主

同节点的数据进行排序，归并。Reduce函数运行完毕之后，所有的结果会被存储在系统的HD

**(2)MapReduce相关概念**

1)Client

Hadoop中所

duce Task两种，均由TaskTracker 启动并完成。

**(3)MapReduce执行机制**

MapReduce由两个阶段组成：Map和Reduce。Hadoop将HDFS中数据块切分成Split之后，被

化简操作（Reduce）主要负责将隶属于一个列表的元素进行合并，它的并行度没有map那么高

2.1.3 近似完全图

2.1.4 负载均衡的基本概念

负

单

2.3 并行计算模型MapReduce

2.4 开源分布式计算平台Hadoop

由于Hadoop的易扩

用到Hadoop集群环境中，通过对Hadoop集群中负载均衡机制的研究、设计和实现，达到平衡集群负载的目的，从而提升系统效率。

2.5 图数据表示

2.6 本章总结

本章主要介绍了相关的背景技术，主要有两方面的内容，一方面讲述了Hadoop的总体结构，对Hadoop中的分布式文件系统，即HDFS，以及MapReduce相关理论、架构进行了描述。另一方面介绍了负载均衡的基本概念，理论以及相关的算法。

第三章 完全图和近似完全图枚举算法

本章重点介绍Hadoop负载均衡系统的设计和实现，该系统包含了采样，信息整合，决策和消息传递描四个模块。本章从系统的总体设计和详细设计两方面入手，对系统的设计思路、实现方法以及相关算法做了详细的描述。

3.1 完全图枚举算法

Hadoop负载均衡系统，即Hadoop\_Load\_Balance，主要包含采样，信息整合，决策控制和消息传递

要完成的任务，并即刻调集数据开始执行。整个过程的详细流程图如图3-2所示。

3.2 近似完全图枚举算法

采样模块主要是对系统的输入数据以及Map任务输出的数据进行采样。该部分在Map阶段进行。采样结果作为决策模块的输入。

具体要对那些数据进行采样，用户可以自定义，本文设计的Hadoop负载均衡系统主要是采集元组记录个数和所占用的字节数。

|  |
| --- |
| **Sampler** |
| **Input:**  *M*:Map Task’s Output  *N*:Job’s split data  *Threshold*:timer’s Threshold  **Output:**  *Scale[M/N]*:*M/N*’s scale（The number of tuples and the size of the space occupied）  *Threshold=0；*  Pick *M[j]* update to *Scale[M/N]*;  Pick *N[i]*;  **If** *Scale[M/N] don’t contain N[i]***do**  将*N[i]* update to *Scale[M/N]；*  **end if**  **end if**  **end for**  **end for**  **return** *Scale[M/N]*; |

3.3 Hybrid算法

信息整合模块主要负责

程。

3.6 本章总结

本章主要介绍了Hadoop负载均衡系统，即Hadoop\_Load\_Balance系统的总体设计和详细设计，对系统中的数据采样模块，信息整合模块，决策模块和消息传输模块的功能，原理，架构和涉及到的相关算法都做了详细的介绍。

1. 并行环境下的负载均衡

4.1 完全图和近似完全图枚举的负载均衡

Hadoop\_Load\_Balance系统在处理多表连接问题时，大规模的中间结果加剧了集群负载不均衡的状况，从而严重降低了系统效率。为解决该问题，笔者提出一种“替换-查询”方法，该方法通过对连接表建立索引，将预输出的元组集替换为索引信息输出到中间结果，以索引的形式参与多表连接，以此减少中间结果规模，；并运用缓冲池、二次排序和多线程技术对索引信息进行优化管理，加快索引的查询速度，提升系统的整体性能。

4.1.1中间结果集的膨胀问题

多表

4.2 并行平台Hadoop中通用的负载均衡

4.3 本章总结

本章对Hadoop\_Load\_Balance系统做了两方面的优化。首先，针对系统处理多表连接问题时，中间结果极速膨胀导致降低系统性能下的问题，本章详细介绍了“替换-查询”的优化方法。其次，数据不在本地所引发了集群内大规模的数据迁移，大量的网络带宽被占用导致系统能明显下降，针对该问题，本章介绍了数据本地化的优化方法，并对Reduce阶段本地化的优化方法，算法，设计和实现都做了详细的介绍。

第五章 实验部分

5.1 实验条件及环境搭建

5.1.1 实验条件

软件环境： Ubuntu10.10，JDK1.6，Hadoop-0.20.205.0以及Hadoop负载均衡系统（Hadoop\_Load\_Balance）。

开发环境：Eclipse3.6，JDK1.6，Shell脚本语言，TPC-H，Hadoop集群环境。

硬件环境：5台1.6GHz的八核CPU，16G内存，150G硬盘的曙光系列服务器。

5.1.2 搭建Hadoop集群

利用试验环境中介绍的各种软件和硬件资源，搭建一个拥有五个节点的Hadoop集群环境，该集群中包含一个Master节点，四个Slave节点。Hadoop版本为Hadoop-0.20.205.0。这5个节点的IP地址是172.19.0.118~122，对应的节点的主机关闭Hadoop系统：

$ bin/stop-all.sh

5.2实验内容以及结果分析

[5.2.1 多表连接策略优化前后系统的性能对比](#_Toc380793448)

（1）实验目的：验证Hadoop负载均衡系统“替换-查询”的多表连接策略的有效性。

（2）实验数据：TPC-H的基准数据集，选用nation, region，customer，orders四张表作为输入数据，数据规模为397.6M。

（3）实验用例：多表连接。

（4）Hadoop系统：基于Hadoop的负载均衡系统。

（5）实验内容及及结果分析。

优化前后的两套系统分别执行以上四张表的连接操作，连接的具体过程如图5-1所示。连接算法是经典的Reduce Join算法，两个系统分别执行了五遍该算法，将各个阶段的统计信息整合求取平均值，并将其填入表5-1中，由于优化后系统的最终结果是一系列NIHDFS索引信息，需要将其恢复为记录元组，该阶段消耗的时间是t4，产生的中间结果是size4，原系统在该阶段不需要做任何操作。

[5.2.3 Hadoop负载均衡系统参数调优及与原Hadoop的性能对比](#_Toc380793450)

1. **Hadoop负载均衡系统参数调优实验**

实验目的：Hadoop负载均衡系统决策时机的参数调优，即map运行到什么进度时执行

越明显。良好的系统性能，对代价模型的有效性也进行了验证。

5.4 本章总结

本章主要介绍了实验条件，以及Hadoop集群环境的搭建步骤，并设计实验验证了Hadoop\_Load\_Balance系统的有效性。经验证，该Hadoop\_Load\_Balance能够有效的平衡Hadoop集群任务负载，系统的性能得到了明显的提升。

第六章 总结与展望

6.1 本文研究总结

Hadoop系统处理用户提交的工作时，常出现集群负载不均衡的问题，瓶颈任务大大拖长了整个任务的执行周期。针对这个问题本文提出了一套基于Hadoop的负载均衡系统，该系统通过数据采样来探知系统中的数据分布信息，样本数据经过信息分类，去重，归并之后被传输到的系统的决策模块中，该模块对其进行分析，得知系统中是否存在数据倾斜以及数据倾斜的严重程度，并据此对分区进行重构，通过降低重构后的逻辑分区负载的方差值，来达到均衡各个Reduce任务负载的目的。

针对Hadoop\_Load\_Balance系统处理连接问题时，因中间结果爆炸，导致的负载不均衡问题，通过对中间结果建立索引，用小巧的索引替代冗长的记录元组，从而大大降低了系统I/O代价。在系统上实现了Reduce端的数据本地化技术，减少了系统Shuffle阶段远程拷贝的数据量，提升了系统性能。

6.2 课题研究展望

本文提出的负载代价模型，忽略了集群中各个机器节点的异构性，也没有关注机器CPU，Memory等资源的占用率与系统性能的变化关系。在以后的研究中可以将异构性考虑在内，在为系统分配任务之前，根据机器节点的状态确定是否为其分配新任务，从而对集群负载做到更加精细的控制。另外，本文中只考虑了Map和Reduce任务一轮就执行完毕的情况，针对Map和Reduce任务需要多轮才能执行完毕的情况，将作为下一步要研究的点。

# 参考文献

1. C.Bron and J.Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. Communications of the ACM, 16(9):575–577, 1973.
2. Li Lu and Yunhong Gu. D MaximalCliques: A distributed algorithm for enumerating all maximal cliques and maximal clique distribution. IEEE International Conference on Data Mining Workshops. 2010.
3. F.Cazals and C.Karande. A note on the problem of reporting maximal cliques. Theoretical Computer Science, 407(1):564–568, 2008.
4. Shengqi Yang, Bai Wang and Haizhou Zhao. Efficient dense structure mining using MapReduce. IEEE International Conference on Data Mining Workshops. 2009.
5. D.Eppstein and D.Strash. Listing all maximal cliques in large sparse real-world graphs. In 10th International Symposium on Experimental Algorithms, 2011.
6. Bin Wu and Xin Pei. A parallel algorithm for enumerating all the maximal k-plexes. PAKDD 2007 workshops, LNAI 4819, 476-4831.
7. Arash Khosraviani and Mohsen Sharifi. A distributed algorithm for γ-Quasi-Clique extractions in massive graphs. INCT, 422-431, 2011.1
8. E.Tomita, A.Tanaka, and H.Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. Theoretical Computer Science, 363(1):28–42, 2006.
9. S.Tsukiyama, M.Ide, and I.Shirakawa. A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing, 6(3):505–517, 1977.
10. F.Cazals and C.Karande. A note on the problem of reporting maximal cliques. Theoretical Computer Science, 407(1-3):564–568,2008.
11. N.Chiba and T.Nishizeki. Arboricity and subgraph listing algorithms. SIAM Journal on Computing, 14(1):210–223, 1985.
12. E.A.Akkoyunlu. The enumeration of maximal cliques of large graphs. SIAM Journal on Computing, 2(1), 1973.
13. K.Makino and T.Uno. New algorithms for enumerating all maximal cliques. In SWAT, Lecture Notes in Computer Science 3111, pages 260–272, 2004.
14. V.Stix. Finding all maximal cliques in dynamic graphs. Computational Optimization and Applications, (27):173–186, 2004.
15. Robert E. Osteen and Julius T. Tou. A clique detection algorithm based n neighborhoods in graphs. International Journal of Computer and Information Sciences, Vol.2, No.4, 1973.
16. D. Eppstein, M. L ofﬂer, and et al. Listing all maximal cliques in sparse graphs in near-optimal time. In ISAAC(1), pages 403–414,2010.
17. Thang Nguyen Bui and Paul H. Eppley. A hybrid genetic algorithm for the maximum clique problem. In Larry Eshelman, editor, Proceedings of the Sixth International Conference on Genetic Algorithms, 478-484, 1995.
18. S.Tsukiyama, M.Ide, and I.Shirakawa. A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing,6(3):505–517, 1977.
19. James Cheng, Yiping Ke, Ada Wai-Chee Fu and Jeffrey Xu Yu. Finding maximal cliques in massive nerworks by h\*-graph. SIGMOD, June 6-11, 2010.
20. S. C. Racha. Load balancing map-reduce communications for efficient executions of applications in a cloud. 2012.
21. R.Rama, Suresh Badarla and Kamala Krithivasan. Clique detection algorithm using clique-self-assembly. 6th International Conference on Bio-Inspired Computing: Theories and Applications, 2011.
22. Jose L. Walteros and Panos M. Pardalos. A decomposotion approach for solving critical clique detection problems. SEA 2012, LNCS 7276, 393-404, 2012.
23. B. Wu, S. Yang, and et al. A distributed algorithm to enumerate all maximal cliques in mapreduce. In International Conference on Frontier of Computer Science and Technology, pages 45–51, 2009
24. L. Lu, Y. Gu, and et al. dmaximalcliques: A distributed algorithm for enumerating all maximal cliques and maximal clique distribution. In IEEE International Conference on Data Mining Workshops, pages 1320–1327, 2010.
25. S. Yang, B. Wang, and et al. Efﬁcient dense structure mining using mapreduce. In IEEE International Conference on Data Mining Workshops, pages 332–337, 2009.
26. M.C.Schmidt, N.F.Samatova, K.Thomas, and B.H.Park. A scalable, parallel algorithm for maximal clique enumeration. Journal of Parallel and Distributed Computing, (69):417–428, 2009.
27. J. Cheng, L. Zhu, and et al. Fast algorithms for maximal clique enumeration with limited memory. In KDD, 2012
28. Mattew C.Schmidt. A scalable, parallel algorithm for maximal clique enumeration. Journal of Parallel and Distributed Computing, Vol 69, pp 417-428, 2009.
29. S Szabo. Parallel algorithms for finding cliques in a graph. 5th International Workshop on Multi-Rate Processes and Hysteresis. Journal of Physics: Conference Series, 2011.
30. 1
31. 1

# 在校期间发表论文

# 致谢

在本文完成后，我即将结束为期三年的研究生生涯，我衷心地感谢在我硕士期间，所有对我的学习、工作及生活给予帮助的人。

首先，感谢我的硕士生导师陈群教授，感谢他在我研究生期间对我的关心和帮助。生活中，陈老师像家长一样，常常与我进行沟通与交流，并给予我鼓励与安慰，使我倍感温暖；工作中，陈老师一丝不苟，从严治学，他严谨的工作风格给我留下了深刻的印象，是我以后科研及工作的榜样。我毕业论文的顺利完成，与陈老师的悉心指导是分不开的。再一次向陈老师致以真诚的敬意。

我还要感谢我的另一位导师，潘巍老师。在进行项目研究的过程中，潘老师丰富的经验以及扎实的理论功底，使我受益匪浅，收获颇丰。无论是研究点的设立、算法的设计，还是科研文章的撰写、做科研的态度，潘老师都会一一耐心地教导我，使我逐步走进科研的殿堂，同时，潘老师活跃的思想，谦逊的性格以及追求真理的态度是我以后科研道路上的榜样。

同时，我要感谢教研室的 ，是他们在我遇到技术难关时，及时给予帮助，在我困惑于一个科研问题时，牺牲时间与我讨论，再一次感谢他们。

最后，我要特别感谢我的父母，我的家人，他们是我学习、科研的坚强后盾，一直在为我默默地付出，无私地奉献。无论我遇到什么困难，什么挫折，他们总会站在我身边，给我最强有力的支持，他们的支持是我顺利完成学业的保证。谨以此文献给他们。

# 知识产权说明书