**中图分类号：**

**论文编号：**



硕 士 学 位 论 文

**基于运行时间预测的并行图计算系统性能优化研究**

作者姓名 李东泽

学科专业 计算机技术

指导教师 樊文飞 教授

培养院系 计算机学院

**The Research and Implementation of Email Importance Evaluation Method based on Thread Network**

A Dissertation Submitted for the Degree of Master

**Candidate：Li Dongze**

**Supervisor：Associate Prof. Fan Wenfei**

School of Computer Science and Engineering

Beihang University, Beijing, China

**中图分类号：**

**论文编号：**

硕 士 学 位 论 文

基于运行时间预测的并行图计算系统性能优化研究

作者姓名 李东泽 申请学位级别 工学硕士

指导教师姓名 樊文飞 职 称： 教授

学科专业 计算技术 研究方向： 分布式图计算

学习时间自 年 月 日 起至 年 月 日 止

论文提交日期 年 月 日 论文答辩日期 年 月 日

学位授予单位 北京航空航天大学 学位授予日期 年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名： 日期： 年 月 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名： 日期： 年 月 日

指导教师签名： 日期： 年 月 日

摘 要

在社交网络服务走向多元化的今天，邮件由于其良好的隐私性和普及性仍然拥有庞大的用户群体。

**关键词：**邮件分类，特征评价，邮件会话网络，EmailRank

**Abstract**

Nowadays, with the diversification of social networking services, the email still maintains a large user base due to its good privacy and popularity.

**Key words:** email classfication, feature evaluation, email thread network, EmailRank

目 录

[摘 要 I](#_Toc23948477)

[**Abstract** II](#_Toc23948478)

[目 录 III](#_Toc23948479)

[图 目 V](#_Toc23948480)

[表 目 VI](#_Toc23948481)

[第一章 绪论 1](#_Toc23948482)

[1.1 研究背景及意义 1](#_Toc23948483)

[1.2 国内外研究现状 1](#_Toc23948484)

[1.2.1 邮件重要性定义 1](#_Toc23948485)

[1.2.2 研究现状小结 1](#_Toc23948486)

[1.3 研究目标和内容 1](#_Toc23948487)

[1.4 论文组织结构 1](#_Toc23948488)

[第二章 相关理论与技术研究 2](#_Toc23948489)

[2.1 邮件重要性评价方法 2](#_Toc23948490)

[2.1.1 基于社交网络的邮件重要性评价 2](#_Toc23948491)

[2.1.2 基于邮件特征的邮件重要性评价 2](#_Toc23948492)

[2.2 邮件会话网络的构建 2](#_Toc23948493)

[2.2.1 基于元数据的会话网络构建 2](#_Toc23948494)

[2.3 社交网络中节点重要性评价 2](#_Toc23948495)

[2.3.1 基于网络拓扑的节点重要性评价 2](#_Toc23948496)

[2.3.2 基于交互信息的节点重要性评价 2](#_Toc23948497)

[2.4 其他技术 2](#_Toc23948498)

[2.5 本章小结 2](#_Toc23948499)

[第三章 邮件重要性特征分析 3](#_Toc23948500)

[3.1 邮件特征 3](#_Toc23948501)

[3.2 特征有效性实验 3](#_Toc23948502)

[3.2.1 数据集及标注 3](#_Toc23948503)

[3.2.2 分类器选择 3](#_Toc23948504)

[第四章 邮件会话网络的构建 4](#_Toc23948505)

[4.1 邮件会话网络的改进 4](#_Toc23948506)

[4.2 邮件会话网络的生成 4](#_Toc23948507)

[4.3 邮件会话网络的存储 4](#_Toc23948508)

[4.4 实验评估 4](#_Toc23948509)

[4.4.1 实验方案 4](#_Toc23948510)

[第五章 基于会话网络的邮件重要性评价方法 5](#_Toc23948511)

[5.1 PageRank算法 5](#_Toc23948512)

[5.1.1 实验方案 5](#_Toc23948513)

[第六章 EmailRank算法在实际系统中的应用 6](#_Toc23948514)

[6.1 应用于邮件客户端 6](#_Toc23948515)

[6.1.1 需求分析 6](#_Toc23948516)

[6.2 本章小结 6](#_Toc23948517)

[总结与展望 7](#_Toc23948518)

[论文总结 7](#_Toc23948519)

[未来工作展望 7](#_Toc23948520)

[参考文献 8](#_Toc23948521)

[攻读硕士学位期间取得的学术成果 9](#_Toc23948522)

[致 谢 10](#_Toc23948523)

图 目

[图 1邮件信息过载问题是一项价值6500亿美金的经济累赘 1](#_Toc531617249)

[图 2 ERIC邮件客户端的用户界面 4](#_Toc531617250)

[图 3信任网络结构 8](#_Toc531617251)

[图 4 EMIRT算法信任分数计算原理图 9](#_Toc531617252)

[图 5重要邮件分类方法的架构 11](#_Toc531617253)

[图 6邮件会话的两种网络结构 13](#_Toc531617254)

[图 7 Webmagic系统架构 18](#_Toc531617255)

[图 8 Lucene的架构和全文检索过程 19](#_Toc531617256)

[图 9 Spring Data Neo4j的应用架构 21](#_Toc531617257)

[图 10不同分类器实验效果对比 27](#_Toc531617258)

[图 11基于特征方法与基准方法分类效果对比 28](#_Toc531617259)

[图 12邮件的两种引用方式 31](#_Toc531617260)

[图 13改进后邮件会话网络构建示意图 32](#_Toc531617261)

[图 14邮件关系匹配流程图 34](#_Toc531617262)

[图 15 Neo4j浏览器界面中的邮件会话网络 36](#_Toc531617263)

[图 16改进和未改进会话网络对比图 36](file:///C:\Users\jiang\Desktop\最终论文\毕业论文_蒋坤_SY1606301.docx#_Toc531617264)

[图 17某用户邮件关系网络中的重要节点识别 44](#_Toc531617265)

[图 18 EmailRank和基于邮件特征方法实验结果对比 45](#_Toc531617266)

[图 19原始Nylas-Mail客户端界面 48](#_Toc531617267)

[图 20插件ThreadNetwork效果图 48](#_Toc531617268)

[图 21 W3C会员影响力评价系统架构图 50](#_Toc531617269)

[图 22分布式爬虫架构图 51](#_Toc531617270)

[图 23 W3C会员影响力评价系统截图：影响力排名 53](#_Toc531617271)

[图 24 W3C会员影响力评价系统截图：影响力对比 53](#_Toc531617272)

表 目

[表 1节点中心度的度量方法 16](#_Toc531563451)

[表 2四类邮件特征列表 22](#_Toc531563452)

[表 3 SemanticDistance的算法描述 24](#_Toc531563453)

[表 4邮件重要性特征重要权值 30](#_Toc531563454)

[表 5改进和未改进会话网络下的方法性能对比 37](#_Toc531563455)

[表 6 EmailRank和基于网络拓扑方法的实验结果对比 45](#_Toc531563456)

# 绪论

## 研究背景及意义

在社交媒体、社交网络和即时通讯飞速发展的如今，随着“互联网+”以及各式互联网应用等新型产品，如抖音、猿辅导的兴起和云计算等新相关技术的飞速发展，人类获取数据的途径和规模正以前所未有的速度快速增长，于此同时，与大数据相关的技术变革于理论发展成为了当今世界的热点话题。大数据在环境科学、超级计算、生物学、互联网经济学等领域以及通讯、航天、金融等行业普遍存在，并引起人们的广泛关注。在这些领域或行业中所涉及的数据通常具有某种关联性，从而产生了大图数据。图作为计算机数据结构中一种基本且常用的数据类型，是表示实体与实体间关系的方法，一张图由若干顶点和顶点间所连接的边组成，其中根据连接边是否具有方向的不同又分为有向图与无向图。无论哪种表示方式，其在表达的语义甚至结构方面都比线性表和树更为复杂。社会或科学领域中的许多应用场景都可以通过转化成图计算模型从而得到解决，与图相关的处理和应用几乎无所不在。比如，社会安全部门需要对整个社交网络进行精准的模式匹配，从而增加寻找贩毒、犯罪集群的可能性；交通运输部门需要确定最优的运输路线；生物信息领域需要对蛋白质进行子图匹配，分析蛋白质间的相互作用从而开发出更有效的临床药物[1]。不仅如此，互联网行业中的用户账号匹配、社交网络分析、风险控制模型的训练都可以看作是大图数据的应用。因此，都大图数据的分析与计算具有很重要的现实意义。

早期传统的图数据管理与图分析计算技术通常只针对“小图”进行处理，整个计算过程中不需要过多甚至复杂的迭代过程，更不会产生大量需要信息通讯的消息，时间与空间的开销相对较低。但是，现实世界中实体规模的持续扩大，导致相应图模型的数据规模迅速增长，甚至有数十亿顶点以及上万亿条边的图数据，而这些只靠顶点和边数量组成的图数据结构还只是大图数据的冰山一角。很多应用中涉及的复杂图数据为了表达其现实语义，往往在顶点以及边上附加一定的属性信息，如表示人的节点通常含有姓名、电话、地址等相关属性，这些属性的存储与计算代价通常是点和边的数十倍乃至数百倍。而且，相比于图数据上的搜索查询，大图上的分析计算算法往往需要基于图的结构进行递归迭代操作，直至满足收敛条件或达到预设的最大迭代轮数，因此会产生巨大的通讯数据，并需要频繁的处理中间结果。此外，除了静态数据与离线计算外，很多情况下实时处理情景需要图数据是动态变化的，如淘宝“双十一”实时分析风控数据等，这种动态往往以流式载图作为扩图途径，不断改变着大图数据的规模和结构，因此面对如此大规模的静态和动态图数据，其底层存储、索引建立、统计分析、实时计算等处理的时间开销和空间开销远远超出了早期传统图处理的能力，因此如何解决大图数据的存储技术与查询计算技术，已经成为数据库领域需要迫切解决的问题，也是大数据领域极具挑战的工作。

面对上述大图数据存储与计算的诸多问题，研究者们已经不满足于仅通过传统的方法探讨基本的图管理与分析问题，一些新兴的研究热点也在不断涌现，如图查询语言、图计算框架等。对于这些研究方向，虽然早期也有一些相关的探讨，但随着计算机软硬件的以及云技术的不断发展，如何提出更准确高效的方法来解决这些问题，为图管理与分析工作带来了新的机遇与挑战。Parallel BGL[7]最为最早一代的并行图计算系统，其在经典的图应用算法基础上开发了并行版本，但此类系统大多数本身不具备一定的稳定性与容错性，并且部分图应用算法的正确性也得不到保证，为了解决稳定性的问题，一部分研究团队专门为特定的图计算应用开发了分布式图计算架构来加速图数据的并行计算，虽然针对性的架构处理可对特定图应用算法进行深度的代码优化以达到更好的处理性能，但毕竟过于局限，针对某一新的图应用算法，算法的移植甚至重开发操作需要重写大量的代码，不仅代码的复用率极低，框架的通用型、可拓展性以及可维护性也很差，对编程人员与使用人员的挑战性极大；MapReduce[8]作为一个并行的计算框架，及其开源实现Hadoop[9]为海量的数据处理提供了便利，为此，用户只需要定义Map和Reduce函数就可完成相应的逻辑计算，不需要关心并发、容错、一致性等复杂问题。但是由于MapReduce框架的自身特点，任务的执行划分为多次Map和Reduce阶段，每个阶段都需要多次的读写分布式文件系统，并且，任务的执行期间无法共享更新数据，再加上图计算的过程通常涉及多轮的迭代，各迭代涉及顶点间的复杂联系，且迭代完成时还要依赖对点的消息传递以满足图计算正确性，因此若将每轮迭代抽象成MapReduce作业不能高效的进行并行图计算，虽然Haloop[10]等处理框架对MapReduce过程进行许多改进，但是并没有从根本上解决基于分布式文件系统所带来的I/O开销问题。针对上述图计算框架的问题，为解决对大规模图数据计算的迫切需要和传统的计算平台无法满足当今的图处理需求的问题，面向大规模并行图计算框架的研究吸引了越来越多的研究者投身其中。

。

## 国内外研究现状

目前面向大规模图数据计算的分布式并行图计算模型及其相关优化研究取得了不错的结果，依据调度方式的不同，大致分为同步计算模型、异步计算模型、混合计算模型、自适应模型四类。

### 同步计算模型

为改进MapReduce计算模型在大规模图数据上的计算效率低等问题，1990年，图领奖获得者Valiant提出了基于消息传递的BSP(bulk synchronous parallel)计算模型[11]，Google也在2010年推出了基于该计算模型的大规模图数据并行计算框架Pregel[11]，这种计算模型区分于现有的MapReduce计算框架在于其适合做数据的迭代计算，其采用一种“块”同步的思想，即通过消息传递机制实现块间数据同步、块内并行计算。其核心思想是将每一次迭代称为一个超步(Superstep)，整个图计算划分为多个超步，而一个BSP作业由一系列顺序执行的并通过全局同步路障分开的超步组成，并行任务时每个节点按照各自的超步组织，接收来自上一个超步的全局消息，执行本地计算并将本次计算生成的消息发送给下一次超步，计算过程如图2.1所示，并行迭代直到全部顶点收敛，即每个顶点相邻两次迭代的变化量小于给定的常量或达到预设的最大迭代次数。

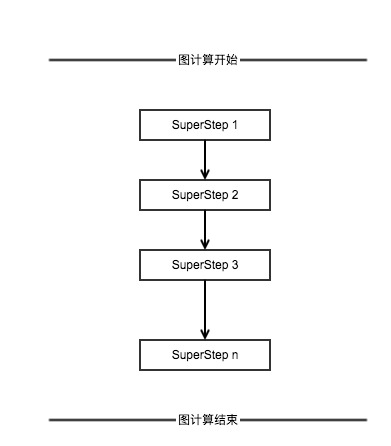


图2.1 图计算垂直结构

在每一次的超步计算中（图体现出超步的内部计算），又会分为本地计算、消息传递、全局同步三个阶段，图2.2、2.3（这两个这图感觉不太好）分别展示了每个超步过程中的计算过程及同步模型的整体计算流程。

1. 本地计算: 计算节点间相互独立，每个节点负责读取本地内存数据执行并行计算。

2. 消息传递: 本地计算阶段完成后，每个计算节点将生成消息通过通信传递给与之相关联的其他节点，以便交换彼此计算所需要的数据。

3. 全局同步: 用于数据间的整体同步，即块见同步。各个计算节点在完成本地任务后进入全局同步阶段，先完成本地任务的节点需要等待没完成任务的节点，直到最后一个节点完成本地任务，所有节点统一进入下一轮的本地计算。

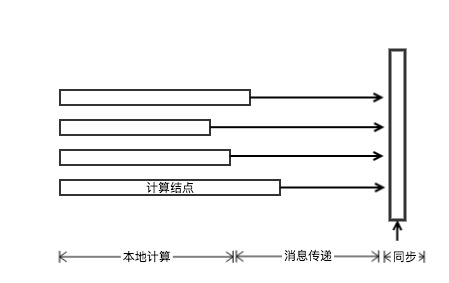


图2.2 超步内计算过程



图2.3 同步计算模型

从上图中我们可以清楚的看出，每轮计算中所有顶点只能使用上一轮迭代计算的更新消息，而通过计算得到的最新消息也只能通过同步在下一轮的计算中使用。因为全局同步的存在，使得整个图计算的调度十分简单，同时，每一轮的消息会在全局的同步下批量的传送，极大限度的利用了网络的流量。

为高效的处理大规模图数据的计算问题，在上述BSP计算模型的启发下，Google首次提出了以点为中心的计算框架Pregel[12]。在点中心的计算模型中，用户只需要从点的角度出发，重写一个与具体应用紧密相关的计算函数Compute，而不需要考虑数据的处理过程。在Compute函数内，涉及1) 处理上轮迭代的更新消息; 2) 本地计算逻辑，并更新本节点的值或节点状态信息; 3) 消息传递逻辑。 Pregel的具体执行过程如下: 首先，选定一个图分区策略[13-15]进行图分区操作，划分后的不同分区分配到不同的计算节点上，然后在每一个图节点上执行Compute函数，即用户自定义的超步处理逻辑，每一次的迭代后，通过全局同步操作保证所有计算节点统一进入下一轮的迭代计算，整个迭代过程直到没有任何消息需要同步或所有的图节点达到收敛状态时停止，输出结果。Pregel框架从容错性、一致性、同步控制等方面给出了可行的解决方案，整个计算模型简单、易用、可拓展性强。受Pregel框架的启发，相继涌现出很多基于BSP模型的大规模并行图计算系统，如GRAPE[18], Gemini、GPS[16]、Giraph[17]、Hama、Giraph++、GiraphX、Pregel+、X-Pregel、Pregellix、Mizan、MOCgraph、Blogel、Nscale、SC-BSP、LFGraph等，是对Pregel开源实现的改进与扩展，在性能与功能上不断完善了基于BSP同步计算模型。

GRAPE[18]是以子图[11]为中心基于MPI[12]消息传递的分布式并行图计算引擎，采用独创的PIE(PEval、IncEval、Assemble)编程接口，该接口易于使用并保证了并行计算过程对用户完全透明。用户只需要为特定的图应用算法编写这三个函数便可执行期望的图计算任务；Giraph作为点中心模型[12]遵循了Pregel的机制、基于Hadoop平台的开源分布式图计算引擎，其利用Zookeeper解决了单节点故障问题、采用Netty网络通信的消息传递机制将每轮超步串行化，解决了Hadoop平台本身在图计算过程中低效率问题；GPS类似于Pregel系统，拥有容错机制、扩展性强且易于编程的优势，保证其可在超大规模图数据上执行图计算任务。其于Pregel系统的不同在于1) 通过master.compute函数扩展了Pregel的API，该函数可轻松高效地实现图应用算法；2) GPS具有可选的动态重分区功能，该功能在图计算过程中根据通信状态以决定是否将顶点重新分配给不同的机器，以提高性能；3) GPS拥有LALP的优化功能可减少网络通信量；Hama[]也是基于Hadoop平台于BSP模型下的图计算系统，其专注于解决机器学习和图算法领域的大规模计算任务；Gemini[114] 作为一个分布式的图计算系统，其关注计算和通信组件的性能，采用高效的划分，积极压缩计算时间，同时隐藏通信成本，从而达到了在保持高效的同时实现可扩展，弥补了高效共享内存系统可扩展性差和可扩展分布式系统性能差的不足。

相对来说，针对需要多轮迭代处理的图计算任务而言，基于BSP的同步计算模型比MapReduce模型的大数据处理平台具有更大的性能优势，也极大的简化了图计算任务，充分利用了网络流量，但是该模型仍有以下几方面限制：

1. BSP模型具有木桶效应，因为全局同步的存在，每一轮迭代的计算时间取决于最慢的计算节点。因此，节点的计算能力不同或图数据的分区不合理都会导致计算节点间的负载不均衡，严重影响图计算效率。

2. BSP模型每轮迭代只能使用上一轮迭代的更新信息，导致迭代次数较多，收敛速度较慢。

3. 消息量大，导致通信代价昂贵。在分布式的图处理框架中，为了满足大规模图数据处理，通常需要将原始的图数据划分为多个子图，并把每个子图分配到不同的计算节点上，因为处于不同计算节点之间的顶点具有一定的关联关系，这就导致不同计算节点间需要进行消息传递以满足图计算的正确性。然而在实际的图数据中，尤其是社交网络等图应用，图中顶点的度数服从幂律分布，从而使得通信的代价成为影响系统性能的瓶颈之一。

4. 某些需要协调图中邻接点的算法无法在BSP模型下实现，例如图着色问题[19],该问题意在使用最少的颜色为图着色，保证相邻的顶点颜色不同。这类问题对于贪心的解决方案在BSP模型下不会收敛，因为图节点试图获得非邻居节点的颜色值，只能通过两者的共同邻居节点来获得，因此同步操作的过程中可能会出现这两个节点颜色不断翻转的情况，本轮迭代具有相同颜色的邻接顶点在下轮的迭代仍会选取相同的颜色，从而导致无法收敛。

### 异步计算模型

为解决上述BSP同步计算模型的诸多问题，科研工作者提出了异步计算模型AP(Asynchronous Parallel)，如图2.4所示。



图2.4 异步计算模型

异步模型中，相邻两次迭代没有明显的界限，任何计算节点随时接收来自其它节点的消息，并直接接受调度器的调度执行; 同样，本地计算生成的最新消息也随时传递给其它需要的节点。因为异步模型摒弃了全局同步的操作，使得每个计算节点不用关心其它节点的进度、状态，只要消息队列存在消息就可以进行下一轮的本地计算，因此消除了同步计算模型下木桶效应问题。经大量的理论分析和实验表明，异步计算模型在执行效率以及资源的利用率方面相对于同步计算模型具有较大的优势，尤其是在图处理系统负载不均衡时，这种优势体现的更加明显[20-22]。

目前采用异步图计算模型的系统包括GRAPE+、Maiter、PowerGraph、GraphLab、GraphX、PowerLyra。其中Maiter框架作为完全异步的大规模图数据处理框架，其采用了DAIC计算模型，避免了同步开销，加快收敛速度，极大的提高了大规模图处理效率；GraphLab主要是针对MLDM应用开发的支持同步与异步的图计算系统，其对图计算中数据一致性模型做了详细分析；PowerGraph针对图数据详从工作负载、图分割、存储、通信和计算等多方面带做了详细的分析，提供了完善的图分割数学理论支撑，证明切点法比切边法能提高一个数量级的图计算性能。其采用了以顶点为中心的GAS编程模型，增加了细粒度并发性同时支持同步和异步模型。但它不支持图的动态修改，容错机制未能充分利用顶点副本；PowerLyra从图分割方面对PowerGraph进行了改进。提出了一种混合图分割方法hybrid-cut，即出入度高的顶点采用切点法反之出入度低的顶点采用切边法，经过试验对比性能提高了至少1.24倍。

虽然异步计算模型相比于同步计算模型省去了全局同步所带来的开销，但是也同样存在着以下方面的不足之处：

1. 异步模型可以加速图计算的收敛速度，但由于没有全局同步的限制，导致迭代轮数的增多, 造成了大量的冗余计算。

2. 异步模型在分布式图计算系统的设计方面相对于同步模型来说更加复杂，在设计过程中，不仅要设计正确的调度器，还需要考虑数据的一致性问题，因为异步模型下的任何一个计算节点随时接收来自其它计算节点的消息，因此必须设计一套额外的机制来保证相同数据对象可以被不同的进程和线程互斥访问，例如GraphLab[23]采用分布式锁的形式来保证数据的一致性，而GRAPE+[24]采用了聚合函数Aggr()来保证数据的一致性。

3. 异步模型的计算过程与计算结果具有不确定性，如果算法设计的不合理，迭代计算可能无法达到预期的效果，甚至不能收敛，即使能够收敛，也同样会在迭代计算的最终收敛前，造成大量的冗余计算。不仅如此，对于采用异步模型的分布式图计算系统来说，编程难度有所增加，调试十分困难[25]。

### 混合计算模型

最近研究表明不同计算模型下，图算法的执行效率受到图应用算法、图分区方式、数据图的特性与规模、分布式集群的规模等多方面因素的影响，不仅如此，在整个迭代周期的不同阶段，可能也需要不同的计算模型以达到全局最优[26]，因此为了解决上述问题，科研工作这提出了混合计算模型。

延迟计算模型[27]作为一种轻量级的混合计算模型，简称SSP模型。其通过引入一个固定的限定值bound来保证最快与最慢的计算节点间相对的迭代轮数不会超过给定的bound值。因此，其允许计算能力强计算节点采用异步的方式执行下去，来减缓同步模型下的木桶效应; 同时通过bound值的限定强制要求计算能力强的节点需等待计算能力差的节点以减少异步模型下的冗余计算。

除此之外，混合计算模型[26]，其通过一组启发式的算法建立代价收益模型来动态预测同步模型和异步模型两种调度方式的性能，并实现了在计算过程中对两种计算模型自由的切换，如图2.5所示。



图2.5 混合计算模型

[26]的实验结果表明，采用了上述混合计算模型的PowerSwitch分布式图计算引擎可以准确预测两种模型的性能，并完成调度模式的切换，相对于使用单一的调度模式，在其上执行了大量的图算法，如PageRank算法、单源最短路径算法、图着色算法等，都在执行效率上得到了不同程度的提高。

### 自适应计算模型

自适应计算模型[24]计算模型，简称AAP(Adapt Aysc Parallel)模型。这种模型摒弃了全局同步的操作，因此从本质上来看其属于异步模型。但与异步计算模型不同的是，每个计算节点在自适应计算模型下均保留一些参数来测量与其它计算节点的相对执行进度并根据当前接收到的消息决定是否进入下一轮迭代。上述整个过程只在本计算节点下完成，不用关心其它任何节点的执行状态。该模型不同与其它计算模型主要体现在以下几个方面:

1. 同步计算模型：AAP模型通过异步的消息传递机制来消除BSP模型下木桶效应问题，摒弃了全局同步机制从而消除了这一操作所带了的额外开销。

2. 异步计算模型：AAP模型通过引入限定值Delay Stretch使得各个计算节点自己决定每一轮迭代需等待的时间，聚集更多的消息传递量从而达到执行更有效的计算，因此很大程度上减少了AP模型下的冗余计算问题。

3. 延迟计算模型：延迟计算模型下通过预设的限定值bound，使得在bound值允许的范围内，各计算节点采用要异步AP模型进行计算；一旦到到达bound限定的范围，计算快的节点就需等待，直到计算慢的节点完成当前轮数的计算，才可进行下一轮的计算。但AAP模型不同于延迟计算模型在于其通过限定值Delay Stretch动态调整各计算节点间的相对进程，当出现计算落后的节点时，其允许该节点等待一段时间以便接受更多的消息，这样可以最大化的利用其它计算节点快的优势，使得慢节点的每一轮计算更有效率。

4. 混合计算模型：相对于Hsync模型，AAP模型不需要全局的预测开关在BSP和AP模型进行切换，相反，AAP模型允许每个计算节点决定各自的计算模式，因此在图算法的执行期间，某些计算节点采用BSP模式，与此同时，其它节点采用AP模式。除此之外，由于AAP模型动态调整计算节点间的相对进程，因此其减少了模型间切换所带来的而外开销。

## 研究目标和内容

在已有的图计算模型中，同步模型存在“木桶效应”问题；异步模型存在大量的冗余计算；混合计算模型为实现同步与异步间自由切换产生了额外的内存与预测开销；而自适应模型中，每个计算节点通过引入限定值动态调整节点间相对进度，从而解决了同步、异步、混合模型下的问题，本文在已有自适应计算模型的原型下，提出了新的基于运行时间预测的自适应图计算模型，该模型可在慢机的情况下规避上述模型的弊端，加速并行图计算。本文的主要研究内容如下：

1. 提出基于运行时间预测的自适应并行图计算模型，以下简称 MPAP模型（Message Predict Async Parallel），并将对消息量的预测转化为对运行时间及消息到达速率的预测。

2. 我们将对算法运行时间的预测问题定义为机器学习中的回归问题，使用均方相对误差（MSRE）做为损失函数，最终选取机器学习中适合图计算领域的回归模型作为候选，进行训练预测，从而分析各模型在分布式图计算时间预测问题上的优缺点。

3. 我们选取包含单源最短路径（SSSP）、深度优先搜索（BFS）、网页排序（PageRank）、节点结构分析（HITS）、图采样（Graph Sampling）、图连通性（WCC WCC-HashMin）、标签传播（Label Propagation）、社区发现算法（Louvain）等9种学术届与工业界常用的图计算应用算法，并尝试将算法进行分类，针对每类算法，给出不同的特征提取方案，并展示预测效果。

4. 我们在已有的自适应图计算系统原型下，开发完成基于消息量预测的自适应并行图计算系统，并与同步、异步、及自适应原型系统进行对比，评估效果。

## 论文组织结构

本文主要分为七个部分。主要的组织结构如下：

# 

# 相关理论与技术研究

## 图模型与图应用算法

### 图模型

图作为计算机领域重要的数据结构之一，其语义与结构的表达都要比线性表和数结构更加复杂。最基础的图模型被称为简单图，这类图中，仅有顶点与连接顶点的边组成，根据边是否具有指向性又分为有向图与无向图。在图计算领域中，除了需要关注由顶点与边组成的图数据结构外，还关注顶点与边上附加的属性信息，在多数的图应用背景下，这种附加的属性信息更为重要，我们将在下一小节详细介绍本文用到的经典图应用算法，这里给出本文所讨论的简单图模型的基本定义。

给图，其中代表图中点的有限集合，代表图中边的集合，代表图中顶点或边上的附加属性信息，即，其中表示图中的顶点；，其中表示图中的边，且，表示是连接顶点与的边；表示顶点与边上属性的集合。根据边是否具有指向性，图又可以分为有向图与无向图，其中无向图中若存在顶点指向的边，那么一定也存在顶点指向的边，而在有向图中则不成立，图2.1所示为简单的有向图模型示例：

图

图2.1(a)包含n个顶点与m条边，顶点的编号为从1开始的整数序列，图中顶点没有附加属性信息，边上附有单一的属性信息且为整数 。

现实世界中，多数图数据的分布极其不均匀，即少数顶点会与周围大量的邻居相连，而多数顶点则与周围邻居交互较少。这类图数据反映到图结构即为少数顶点度数较高，而多数顶点度数较低，造成这一现象的的原因也十分直观，比如社交网络中少数明星会与大量的粉丝产生交互行为，如评论留言；淘宝订单数据中，少数用户的年均订单量明显大于其余用户，以此来鉴别是否用户具有刷单行为。这类分布不均匀的现象对图计算造成极大的挑战，容易出现负载不均衡现象，造成计算效率低甚至由于内存限制导致无法完成图计算。下图为真实的图数据中顶点度数的分布情况，我们可以从图看出，这个分布不均匀现象十分明显。

图（横坐标是点数，纵坐标是度数）

### 图应用算法

图数据处理与图数据的挖掘得到了学者们的广泛研究，不同的实际问题会抽象出不同的图应用算法[]，除了最短路径、图着色、最小生成树、连通性、可达性等基础图论算法外，还包括社区发现、标签传播、页面排序、子图匹配、图中元素分类与聚类等多种复杂的算法，本小节给出工业界与学术界常用的图应用算法。

**(1) 单源最短路径**

单源最短路径问题中，给定一个带权有向图 ，其中代表每条边上的权重且，给定某点做为源点，现要计算从源点到图中其它各顶点的最短路径长度，该长度即为到目标顶点的路径中所走边的权重之和。Dijkstra算法作为解决单源最短路径问题经典的算法之一，其按各顶点与源点间的路径长度的递增次序，生成源点到各顶点的[最短路径](https://baike.baidu.com/item/%E6%9C%80%E7%9F%AD%E8%B7%AF%E5%BE%84/6334920)的算法，既先求出长度最短的一条最短路径，再参照它求出长度次短的一条最短路径，依次类推，直到从源点到其它各顶点的最短路径全部求出为止。

**(2) 图遍历**

**(3) 图联通性**

**(4) 页面排序**

页面排序对于从超大量的页面文档中筛选出有价值的信息十分重要。PageRank的提出开启了链接分析研究的热潮，基于这种链接分析方法，可以客观的衡量网页质量，并且能够做到独立于内容，不需要人工干预就可以自发的挖掘互联网上有价值的资源。PageRank算法作为最早提出的链接分析算法之一，并在Google公司获得了巨大的商业成功，包含了两个基本假设：一是超文本链接包含用户对一个页面或网站的判断信息；二是某个页面被其他页面链接的数量越多，该网页的重要性越高，越能提供有价值的信息。其基本思想是如果网页存在指向网页的链接，表明网页的所有者任务网页重要，从而把网页的一部分重要性传递给网页，该部分重要的计算由网页的PageRank值与该网页的出链接数共同决定，即，而网页的PageRank值计算公式如下：

其中表示阻尼系数，其意义为用户到达某一页面后，继续沿该页面向后浏览的概率；代表所有链接到网页的页面。上述公式经过多轮迭代，系统的PageRank值逐渐收敛。

**(5) 社区发现**

**(6) 标签传播**

**(7) 图采用**

## 分布式图计算相关技术

## 运行时间预测

## GRAPE+系统

## 本章小结

# 邮件重要性特征分析

由于之前的相关工作并没有验证哪些特征对于邮件重要性评价是有效的。

## 邮件特征

## 特征有效性实验

### 数据集及标注

### 分类器选择

# 邮件会话网络的构建

## 邮件会话网络的改进

## 邮件会话网络的生成

。

## 邮件会话网络的存储

## 实验评估

### 实验方案

# 基于会话网络的邮件重要性评价方法

本文的研究

## PageRank算法

PageRaEmailRank算法

在网页实验评估

### 实验方案

我们将第三本章小结

# EmailRank算法在实际系统中的应用

通过以上章

## 应用于邮件客户端

现有的邮件客户端如Gmail、Outook、Foxmail W3C会员影响力评价系统实现

### 需求分析

W3C是一个制定网络标准的全球会员组织总体架构设计

W3C会员影响力评价系统主要包括四大模块，系统模块实现

## 本章小结

本章我们介绍了提出的邮件重要性评价方法在实际系统的作用。

总结与展望

## 论文总结

在社交媒体和即时通讯飞速发展的如今，。

## 未来工作展望

本文主要研究了邮件重要性评价问题。

参考文献

1. The Radicati Group. Email-Market-2018-2022 [EB/OL]. https://www.radicati.com/wp /wp-content/uploads/2018/05/Email-Market-2018-2022-Executive-Summary.pdf, 2018-06-11/2018-11-20

攻读硕士学位期间取得的学术成果

**硕士期间发表的论文**

1. **Jiang Kun**, Hu chunming, Sun Jie, Shen Qi. Email importance evaluation in mailing list discussions [C]. International Conference on Web Information Systems Engineering. 2018. (CCF推荐会议列表C类，workshop，12页)
2. **Jiang Kun**, Hu chunming.：Characterizing and predicting important emails in mailing list discussions [C]. 北京航空航天大学第十五届研究生学术论坛. 2018.

**硕士期间申请的专利**

[1] 胡春明, **蒋坤**, 何潇安. 一种基于会话网络的邮件重要性评价方法[P]. CN：201811441956.X，2018-11-29.(专利学生第一发明人)

致 谢

在北京

。