目录

[1. 绪论 3](#_Toc476693561)

[1.1. 研究背景 3](#_Toc476693562)

[1.2. 论文工作 3](#_Toc476693563)

[1.3. 论文组织 3](#_Toc476693564)

[2. 图相关工作 3](#_Toc476693565)

[2.1. 问题定义 3](#_Toc476693566)

[2.2. 关键技术 3](#_Toc476693567)

[2.3. 常见框架 3](#_Toc476693568)

[2.4. 存在问题 3](#_Toc476693569)

[2.5. 本章小结 3](#_Toc476693570)

[3. 基于状态更新的流式图计算模型 3](#_Toc476693571)

[3.1. 图算法特征分析 3](#_Toc476693572)

[3.2. 模型定义 3](#_Toc476693573)

[3.3. 状态存储和更新 3](#_Toc476693574)

[3.4. 模型应用举例 3](#_Toc476693575)

[3.5. 本章小结 3](#_Toc476693576)

[4. GraphFlow系统的设计与实现 4](#_Toc476693577)

[4.1. 系统架构 4](#_Toc476693578)

[4.2. 模型实现 4](#_Toc476693579)

[4.3. 算法实现 4](#_Toc476693580)

[4.4. 本章小结 4](#_Toc476693581)

[5. 实验与分析 4](#_Toc476693582)

[5.1. 实验环境 4](#_Toc476693583)

[5.2. 实验结果 4](#_Toc476693584)

[5.3. 本章小结 4](#_Toc476693585)

[6. 结束语 4](#_Toc476693586)

[6.1. 工作总结 4](#_Toc476693587)

[6.2. 下一步工作 4](#_Toc476693588)

[参考文献 4](#_Toc476693589)

[发表文章目录 4](#_Toc476693590)

[作者简历 4](#_Toc476693591)

[7. 致谢 5](#_Toc476693592)

# 绪论

## 研究背景

图是计算机科学中常用的一类数据结构，它很好的表达了数据之间的关联性。现实世界中有很多数据都可以抽象成图数据，例如Web网页之间的链接、社交人物之间的互动以及买卖双方的交易都可以抽象成彼此关联而形成的图。而随着互联网的快速发展，图数据的总量也在急剧增加。如截至2014年第一季度Facebook包含了12.3亿个活跃用户，每个用户平均好友130个；web链接图顶点数达到Ｔ级，边的个数达到Ｐ级[1]。

因为图数据能够很好的表达数据之间的关联性和聚集情况，因此针对图数据表达的关联关系可以挖掘出很多有用信息。比如，通过为购物者之间的关系建模，就能很快找到口味相似的用户，并为之推荐商品；在社交网络中，通过传播关系发现意见领袖。图算法及相关的处理框架已经广泛运用在社交分析、商品推荐、舆论监测、欺诈检测等各个领域。

处理这些海量动态的图数据也对现有的图计算模型提出了挑战。一方面，这种超大规模的图数据很难一次性的全部导入内存中进行处理，即使能够借助外存一批一批的处理图数据，也使得计算延迟显著增加；另一方面，这些数据又是动态变化，实时更新的，现有的图计算模型要能够在这种动态的数据集上进行增量计算。

现有的成熟的图计算系统如Google Pregel[2], Spark GraphX[3], 这些图计算框架都采用了分布式的集群和BSP（Bulk Synchronized Parallel）[4]消息同步机制来处理图数据。然而这些系统都是在静态的图数据结构上进行的离线批量处理,即每次针对整体的图进行计算，当图动态变化时，需要在变化后的整个图上重新计算一遍。这使得用户等待周期长，无法满足实时计算的要求，也浪费了系统资源。[5]

因此针对图数据不断变化的情况，提出了很多在动态图上直接进行计算的方法。针对这类动态图计算的问题，大致可以分为两类：估计计算和准确计算。对于估计计算，大部分的算法是希望通过采样的方式来降低时间和空间开销，并通过特定的采样方法来减少真实值和实测值之间的差距。 如文献[6][7][8]研究了在流图上如何通过设计采样规则来估计图中三角形数目；文献[9][10][11] 通过将原始的图信息转化为简单的数据结构来保存图中元素，降低了内存消耗。虽然估计算法能够在一定程度上节约了内存和计算开销，但其估计的误差在实际的生产环境中往往变得不可控制，文献[12]指出，针对大体量的无法全部载入内存的图数据，近似算法的错误率在95%-133%之间。对于准确计算，现有的KineoGraph[13]和IncGraph[5]提出采用增量计算的模型进行实时计算，然而这种增量式的更新是串行执行的，实时性有限。SpecGraph[14]虽然在上述两个增量模型的基础上有所改进，提出了基于推测机制的并发更新模型，然而该模型中假设顶点的状态只依赖于顶点当前接收的信息，而与顶点之前的旧状态无关。这种假设使得系统的适用性差，很多算法不仅跟顶点接收消息有关，还跟顶点的旧状态有关，因此模型的表达能力有限。

## 论文工作

针对动态图计算的实时性和准确性要求，本文在上述已有的研究基础之上，提出了**基于状态更新的动态图计算模型**，能够在原有图状态上并发的计算增量信息对状态的影响，而无需在整个图上重新计算，同时通过控制更新影响范围来提高并发性，实现状态的并发更新。本文的工作主要有以下3点：

（1）分析现有的图计算的特点，抽象出在流式场景下图算法的典型特征。从影响范围和计算次数两个维度分析了4个典型的图算法：节点度分布（DD, Degree Distribution），三角形数目（TC, Triangle Count），单源点最短路径（SSSP, Single Source Shortest Path），和PageRank（PR）。

（2）根据这些流式图算法的特点，建立基于状态更新的动态图计算模型，该模型能够允许用户自定义状态，并且采用并发更新的方式来快速计算结果。

（3）在上述建立的基于状态更新的动态图计算模型基础上实现了典型的动态图算法：TC和连通子图（CC, Connected Components），并且从正确性和实时性对算法进行评估，结果表明：基于状态更新的动态图计算模型构建的算法能够得到较为正确的计算结果，计算偏差在1%以内；而且算法能够在5ms内返回增量计算的结果，符合实时性要求。

## 论文组织

# 图相关工作

## 问题定义

## 关键技术

## 常见框架

## 存在问题

## 本章小结

# 基于状态更新的流式图计算模型

## 图算法特征分析

## 模型定义

## 状态存储和更新

## 模型应用举例

## 本章小结

# GraphFlow系统的设计与实现

## 系统架构

## 模型实现

## 算法实现

## 本章小结

# 实验与分析

## 实验环境

## 实验结果

## 本章小结

# 结束语

## 工作总结

## 下一步工作

# 参考文献

# 发表文章目录

# 作者简历

# 致谢