# P2-基于Flink Gelly的反欺诈场景的实现和调优

## 项目背景

[这段重点描述整个项目的意义，为什么要在Flink上做，Flink Gelly的简介等，注意行文流畅]

## 预期成果

1. Flink Gelly内核解析文档
2. 实时流式图计算引擎可行性研究
3. 基于Gelly的反欺诈场景的算法选型和算法实现
4. 在公开数据集和华为公司提供的数据集做验证

## 研究方案

1. 分析批处理图计算模型（包括BSP、GSA等）
2. 分析Flink Gelly内核实现
3. 分析反欺诈场景和社区发现的主流算法
4. 在批处理模型（基于Gelly）上实现这些算法
5. 研究如何在流式图上应用这些算法
6. 抽象出流式图计算模型

## 研发计划

[以下这段是摘抄自google doc文档，只是总结性的概述整个研发进展，如有需要请详细撰写]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **时间段** | **类别** | **研究问题** | **系统实现** | **新发现或结论** |
| 2016.10-2016.11 | Flink框架学习 | 学习Flink的框架及API的使用 | 完成了《Learn Flink》手册，介绍Flink的架构和API使用方法 |  |
| 2016.11-2016.12 | Flink Gelly学习 | 学习Gelly的的架构及图相关的算法 | 完成了《Flink Gelly》手册，介绍了Gelly的框架和部分图算法的实现 | Flink Gelly是基于BSP模型实现的运行在批处理上的图计算框架。 |
| 2016.12-2017.01 | Flink Gelly测试 | 测试Flink Gelly中的TC算法 | 完成了论文的实验部分 | Flink Gelly是建立在批处理上的，图计算的相关性，导致当数据量增大时，节点之间的通信显著增加，吞吐率下降 |
| 2017.01-2017.02 | GraphFlow系统设计和实现 | 设计并实现了基于Flink Streaming的图计算框架 | 完成了原型系统GraphFlow，完成了DD和TC算法。 | 流处理充分考虑计算的局部性，采用分流的方式减少节点之间通信，因此当数据量增大时，相比批处理，吞吐率变化不太明显，更为稳定。 |
| 2017.02 | HzGraphFlow系统的设计与实现 | 设计并实现了基于Hazelcast的图计算框架 | 完成了原型系统HzGraphFlow,实现了DD,TC,SSSP和PR算法。 | 根据图算法的特点，利用分布式数据结构存储图状态信息。 |

## 目前进展

[结合预期成果、研究方案和研发计划，详细具体阐述每一步的工作重点和工作结果，注意，你这段时间研究的K-cores算法要写进去]