后期: 2022.9-2023.4

项目进度 一 前期: 2021.11-2022.1

中期: 2022.1-2022.9

1. 数据清洗

结合教务的教师信息,小组分工,手工收集了dblp 学者数据库内每位教师对应的 URL。期间遇到了学者重名和英文名命名不规范的问题,小组采用人工消歧和制定英文名标准的方法解决。

2. 数据库搭建

小组讨论 MongoDB 数据库结构,提出"关系型模型"和"一对象一表的面相对象模型"两种方案。通过与指导老师探讨,考虑到今后数据的扩展和非结构化特点,采用了"面相对象的模型",但却为日后的性能优化造成难题。

3. 数据入库

通过 dblp 提供的 API,将每位学者的个人信息 - 和合作关系写入数据库。

期间遇到了 API 数据格式变更导致数据库数据错乱,通过删库并重新适配新 API 解决。

4. 前端页面制作

本项目前端设计主要以 VUE 为载体,并使用 Echarts 表格进行辅助呈现。

本项目目前已经实现了三张表格及相关功能。首 先对于中间的学者合作关系网图,此图展示了以 某一学者为核心的合作关系网。用户可以通过将 鼠标放置于点和线上来查询学者的基本信息与学 者间的合作关系的详细信息。本项目还实现了用 户点击节点切换中心查询节点的功能,用户可以 点击节点来切换查询对象来查看其他学者的详细 合作关系网。

对于左右两张表格,它们代表了中心节点学者的 统计信息,包含中心节点学者与某一学院的合作 总人数与中心节点学者与某一学院全部学者的总 合作篇数。

在实现前端页面的过程中也遇到了一些问题,其大致包含: Echarts 配置文件的编写与改进,数据传递内容的统一等等。这些问题最终通过本项目组的一直努力都被成功解决。

5. 突发事件: 数据库被勒索

2022.7.15, 位于阿里云的 MongoDB 服务器由于未设置密码遭遇勒索攻击,要求支付 BTC 以解除加密。

转机:由于 MongoDB 采用 Docker 的部署方式,通过自动快照策略恢复了容器。

6. 业务逻辑实现

主要包含 DAO 层,并将渲染合作关系图所需数据封装成 json 文件。

由于采用"一对象一表"的数据库结构,查找学者 信息需要遍历库中所有表,等待时间长,用户体 验不佳。

目前通过对每张表建立索引,查找性能得到提升,但仍需探索针对多表查询的优化。

7. 合作关系图渲染

通过引入极坐标,基于合作者之间的的亲密程度, 计算图中各节点的大小、相对位置和连线粗细。

8. 后端服务器部署

后端选用 Flask 最为服务器,响应 URL 并返回前端所需的 json。

期间解决了开发环境下"非本机访问"和" CORS 跨域访问"。

9. 测试调优

MongoDB 数据库,Web 服务器和后端服务器 分别部署于三台不同网络的物理主机,主机间通 讯均采用 HTTPS 加密协议以确保安全。 修正相关 bug 后,测试连接无误,可以正常演示 功常。

_ 至此,计划内的中期任务全部完成。与原计划相 比,推迟了1个月。