

# 图文件系统-GBFS

丁峰、牛田、谢灵江、张立夫

# 目录

01 | 项目介绍

02 | 完成情况

03 | 可扩展性

01

# 项目介绍

01 | 立项依据

02 | 项目架构

# 1.1 立项依据

## 文件数量庞大

- 随着使用时间变长
- 电脑中个人文件数量过多

```
~/Projects/OSH/X-GBFS  P master  
pwd  
/Users/leafz/Projects/OSH/X-GBFS
```

## 文件间毫无关联

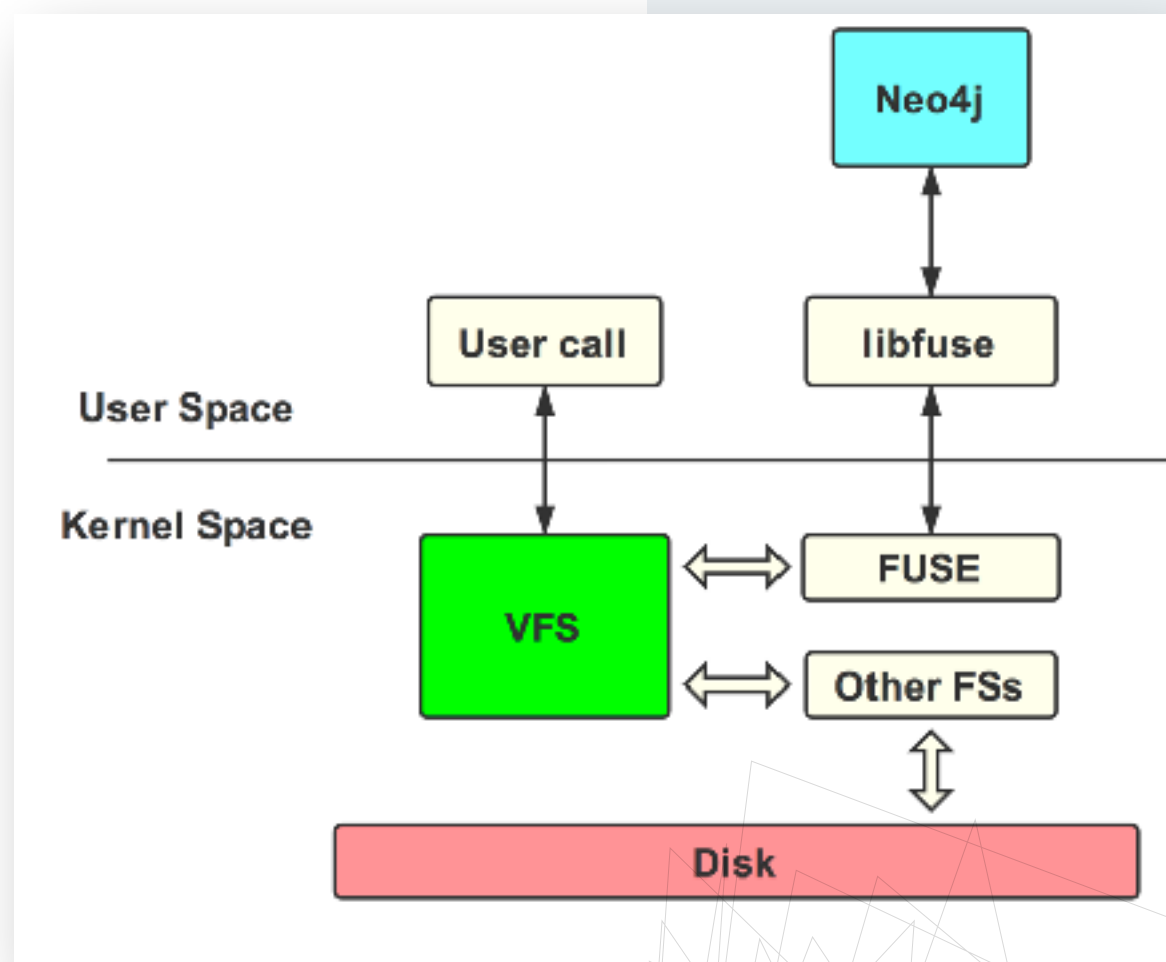
- 无法直接打开某一文件
- 相关联文件

## 目录层次繁多

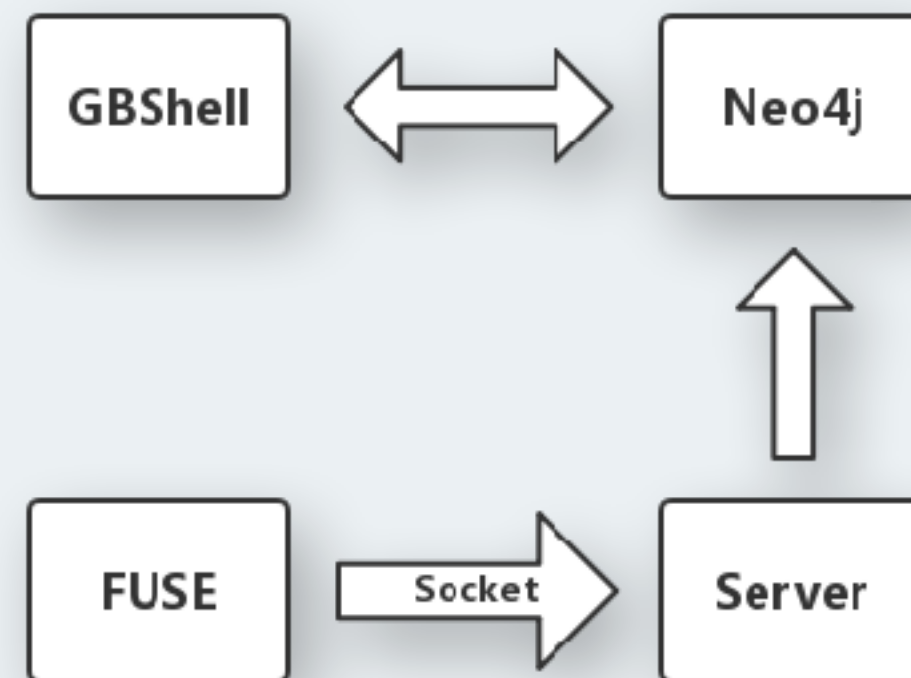
- 难以记忆文件具体位置
- 增加寻找特定文件难度

```
~  
ls -lR | grep "^_" | wc -l  
81550
```

# 1.2 项目架构



FUSE 流程



整体流程

FUSE 通过 Socket 向 Server 发送文件操作请求，通过 Server 实现对 Neo4j 数据库的修改与更新

# 02 | 完成情况

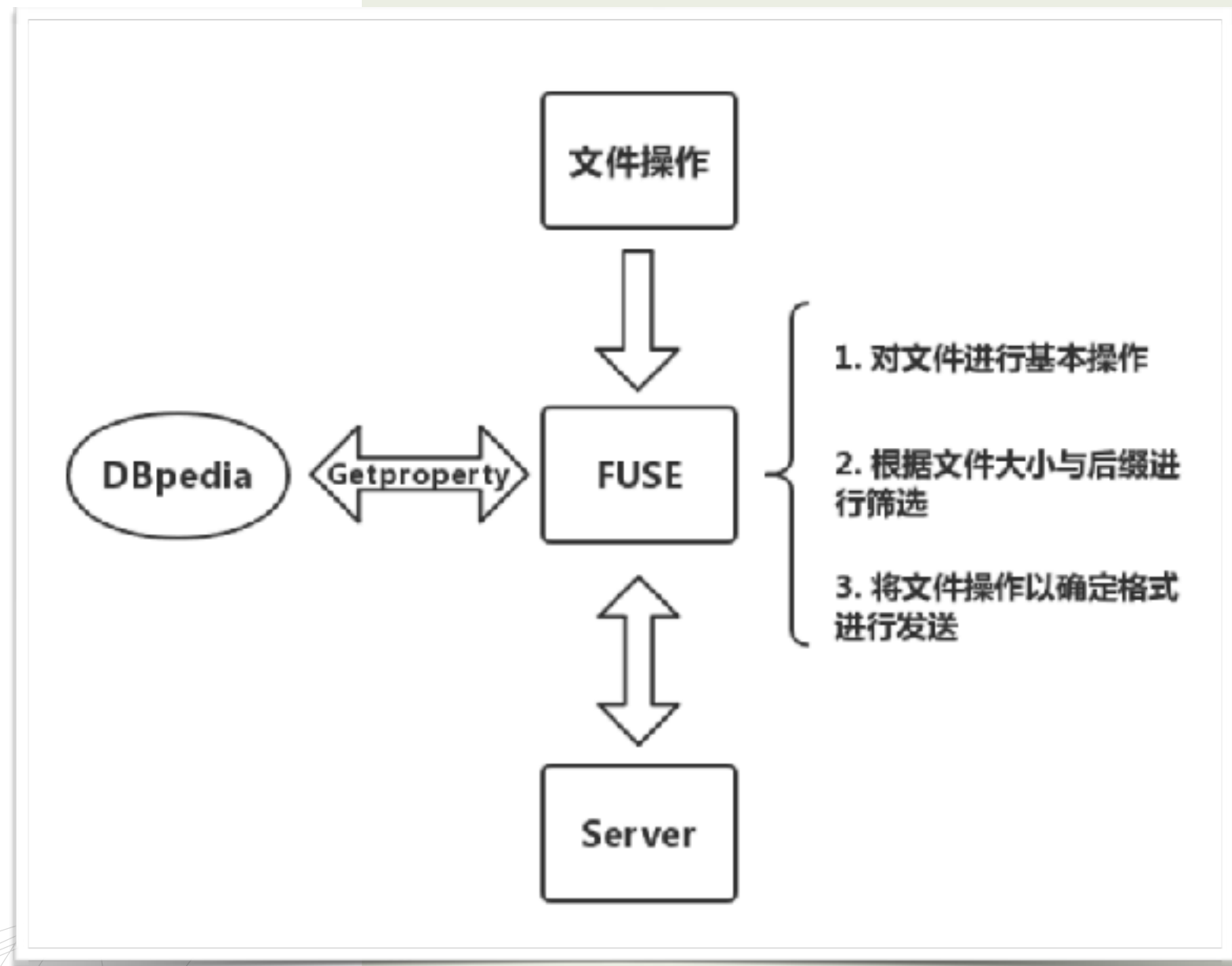
- 01 FUSE
- 02 Server
- 03 Recommend
- 04 GBShell



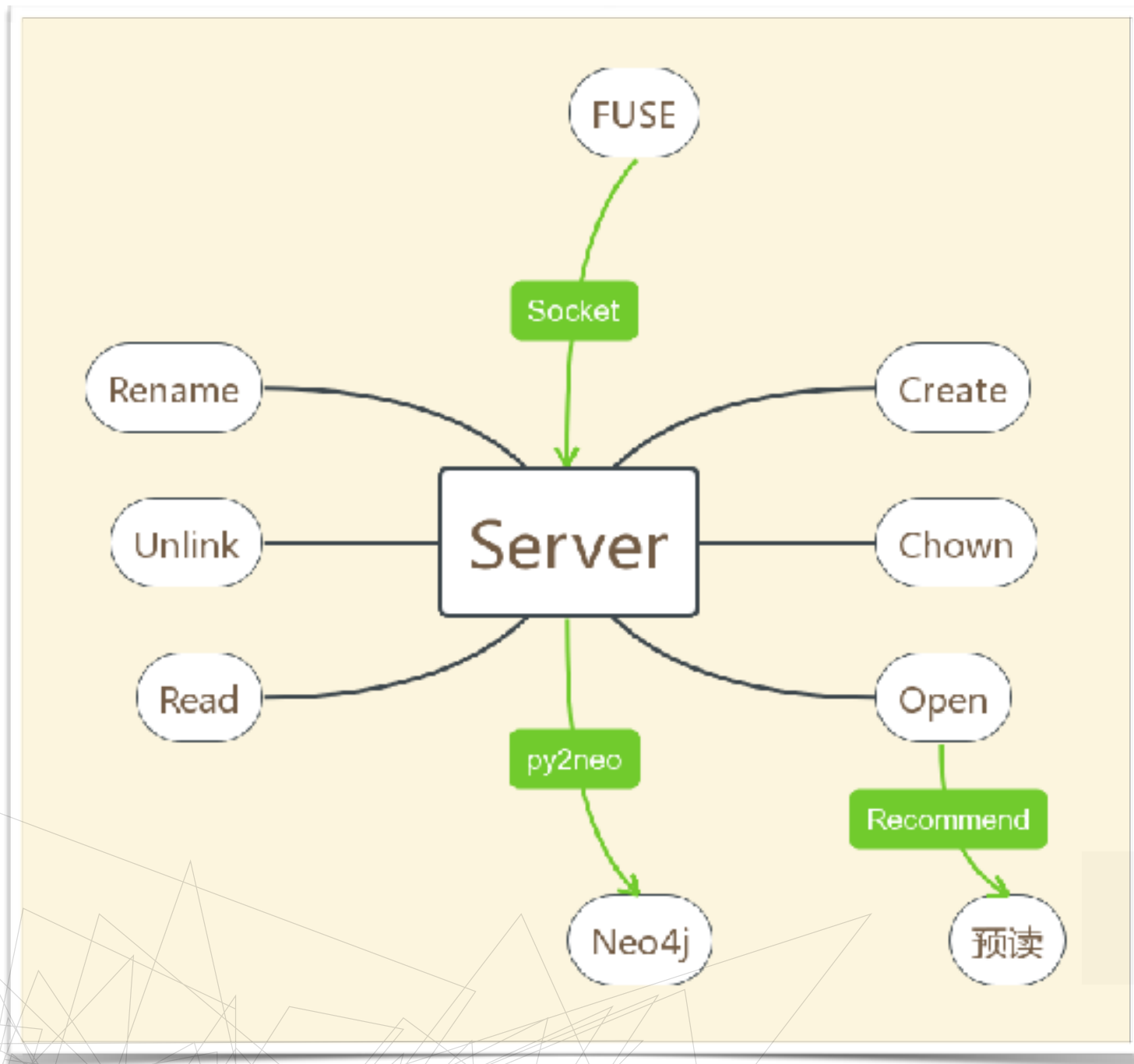
# 2.1 FUSE

## GetProperty:

- 调用 DBpedia 知识图谱接口，获取相关属性



## 2.2 Server

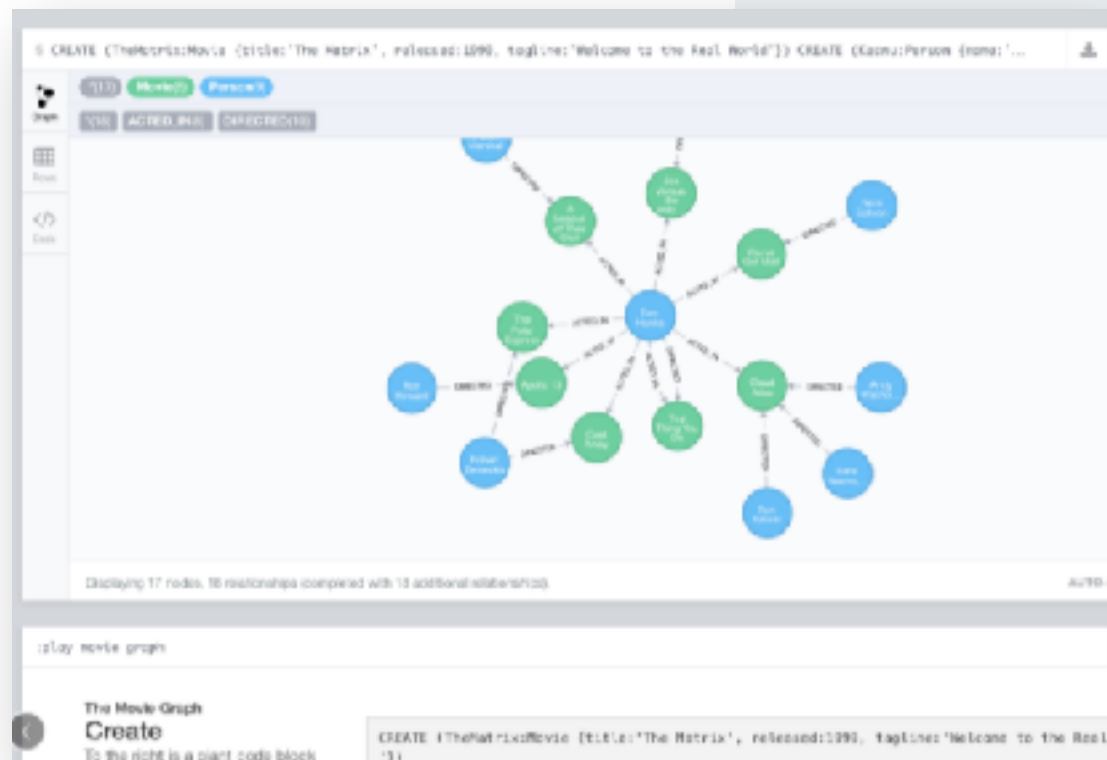


### Server:

- 从 Socket 中获取操作指令
- 在 Open 过程中通过由 Recommend 模块得到权值相近文件进行预读
- 将对应的文件节点属性通过 py2neo 进行数据库更新

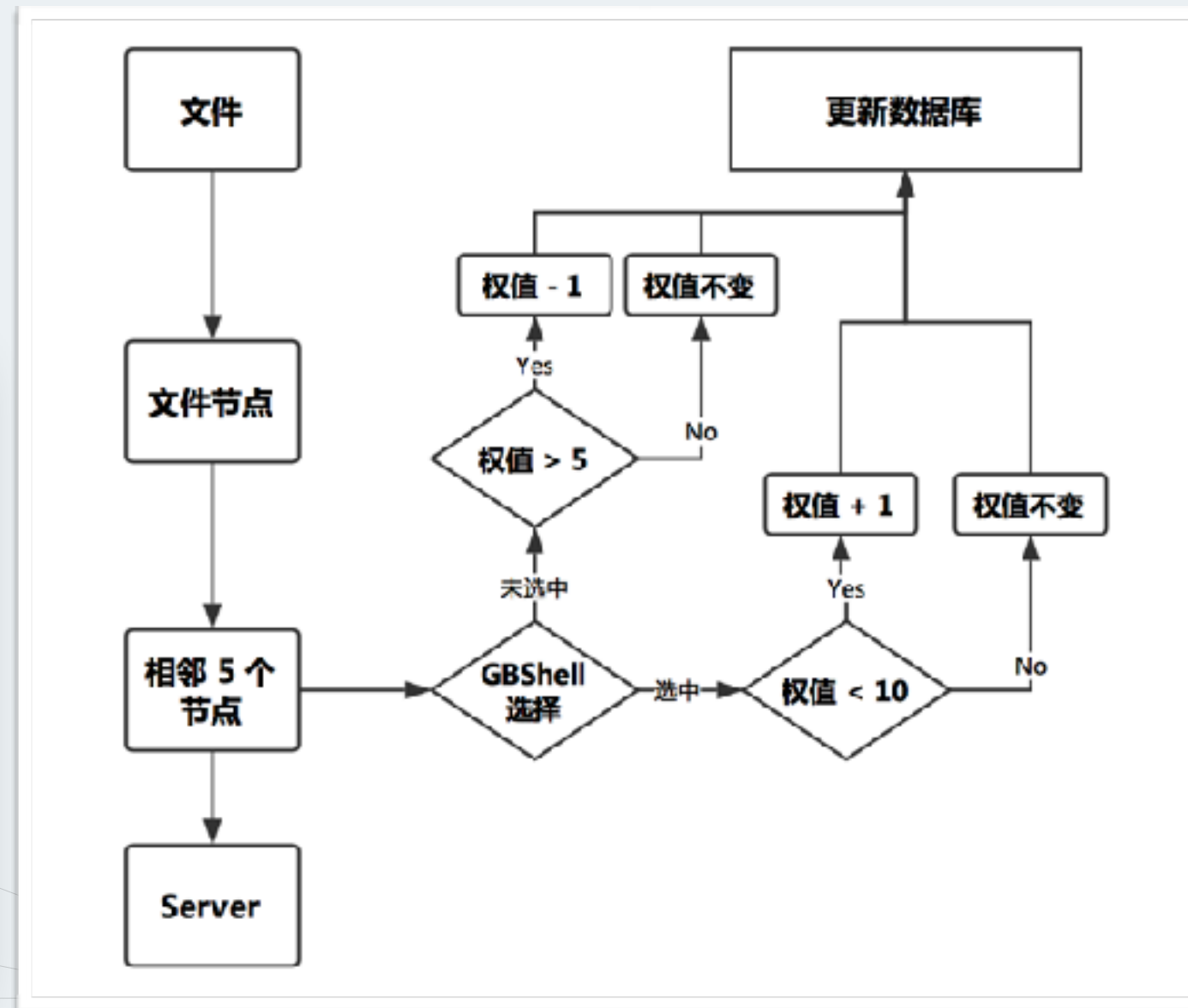


## 2.3 Recommend



### 预读测试效果

通过获取相邻文件进行预读  
二次打开速度对比

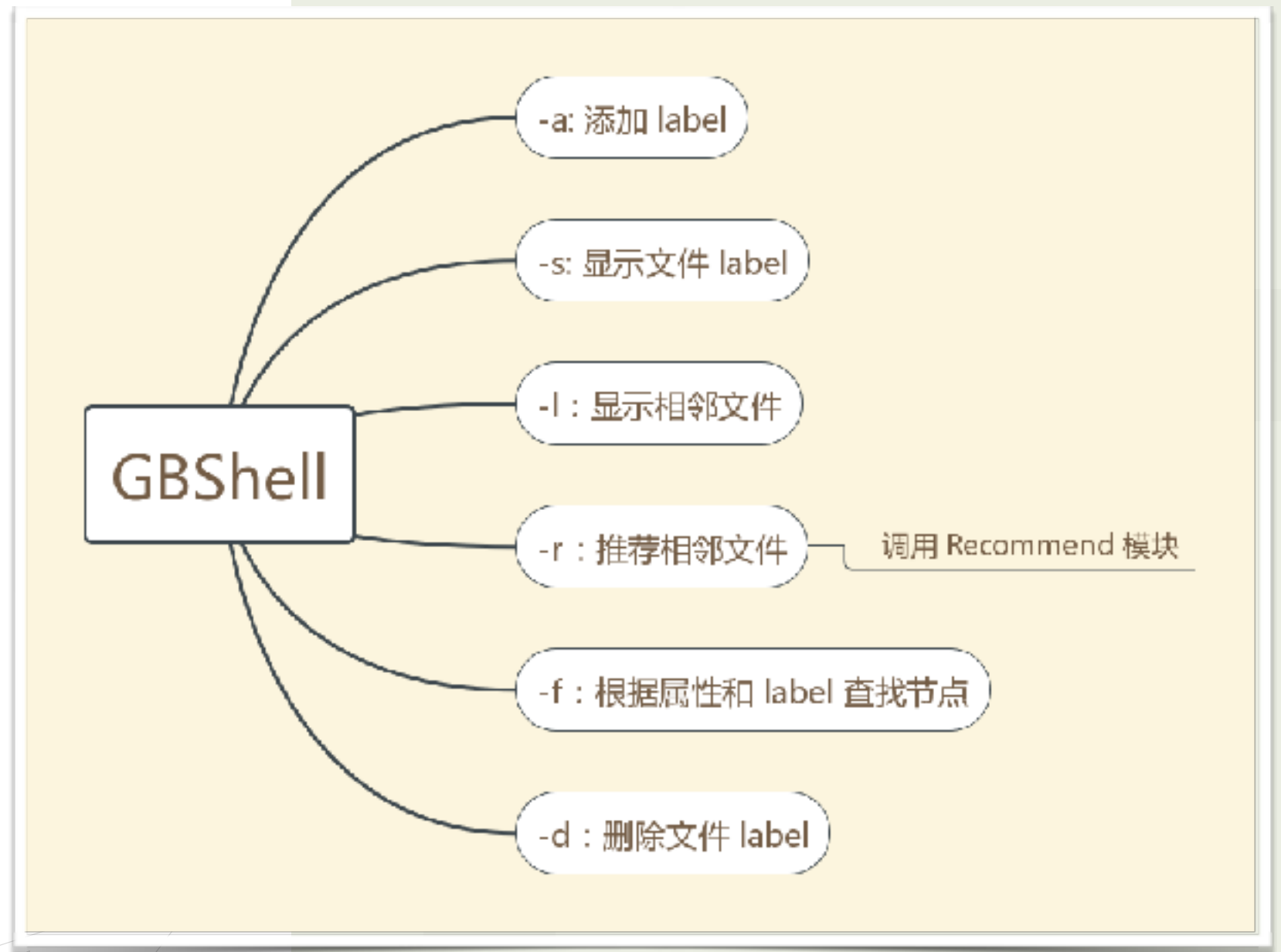


### 权值更新过程

获取相邻节点  
对权值进行更新

## 2.4 GBShell

- GBShell: 辅助操作命令行
- 根据传入参数选择相关操作
- 通过 py2neo 对 Neo4j 进行操作



# 03

## 可扩展性

01

分布式

02

预读优化

03

推荐算法优化

04

知识图谱

# 3 可扩展性

## 分布式

通过 Socket 进行通信  
方便修改为分布式图文件系统

## 推荐算法优化

采用更合适推荐算法  
通过知识图谱避免冷启动

## 预读优化

从底层实现预读操作

## 知识图谱

根据需要采用  
更合适的知识图谱

# End | Thank you!

GBFS 小组成员：

- 01 丁 峰：PB16110386
- 02 谢灵江：PB16111096
- 03 牛 田：PB15010419
- 04 张立夫：PB15020718