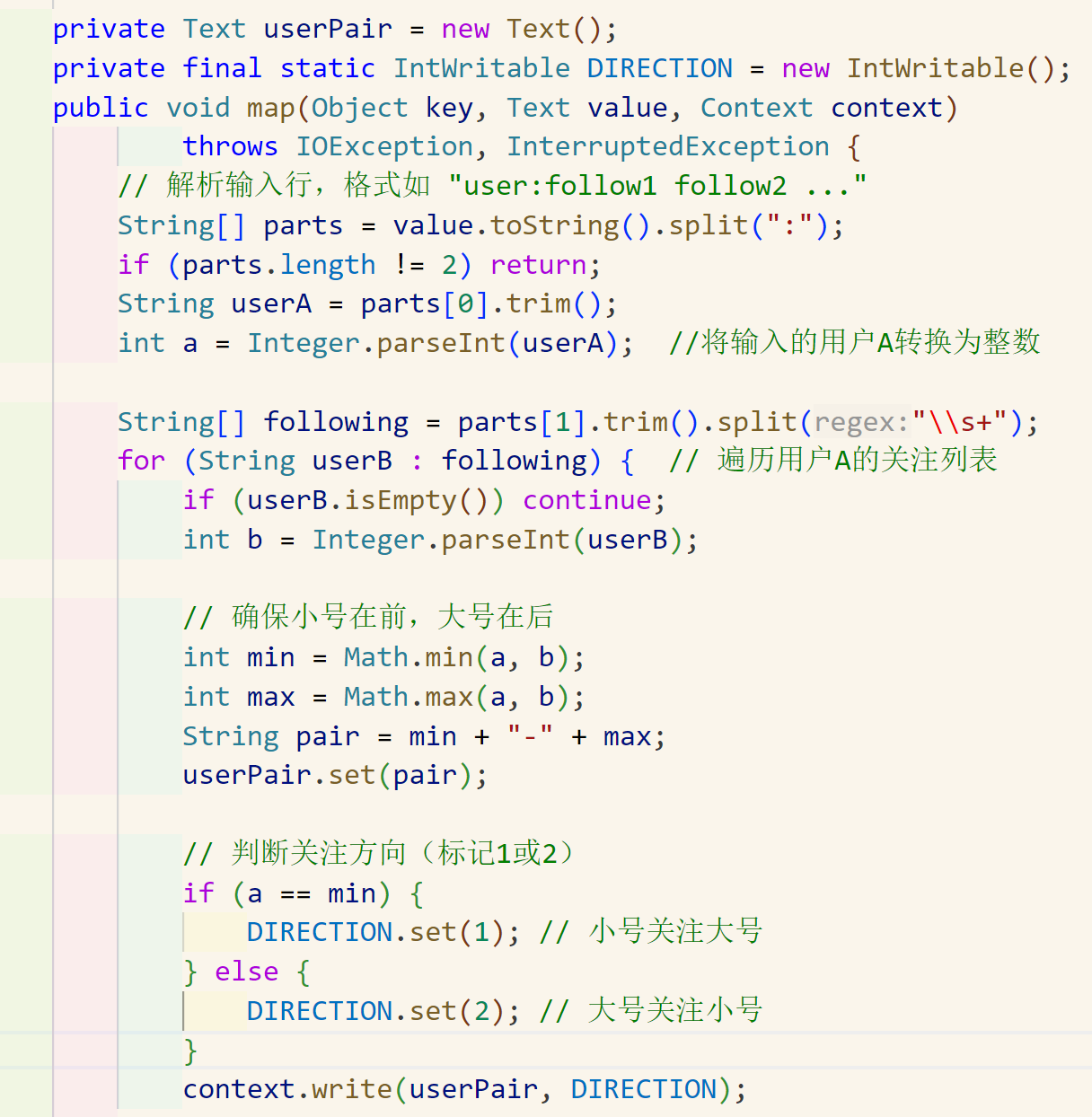
**实验 4：MapReduce高级编程技术**

陈德丹 221220159

**任务1：社交网络中的互相关注好友**

⚪**基本思路**：map阶段对于输入的每个用户的关注列表，各生成一个用户对，小号在前，大号在后，例如3-5，这样可以避免重复，同时打上标记，小号关注大号则标记1，大号关注小号则标记2，reduce阶段对于每个用户对，如果两种标记都存在，则说明互相关注并输出

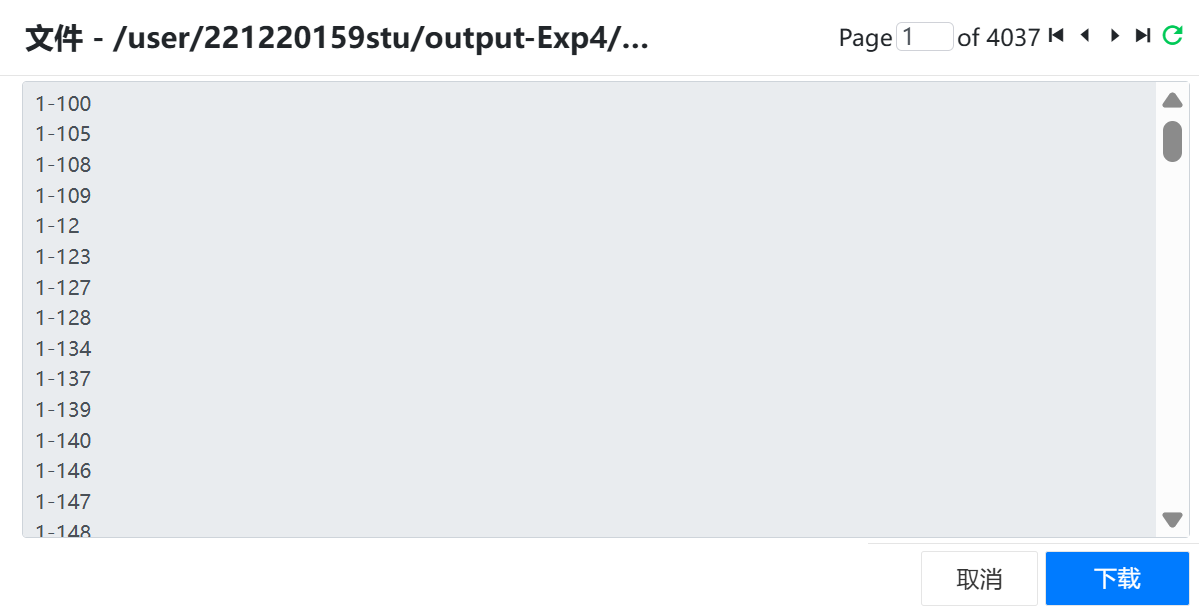
⚪**mapper**输出的key的类型为Text，value为IntWritable



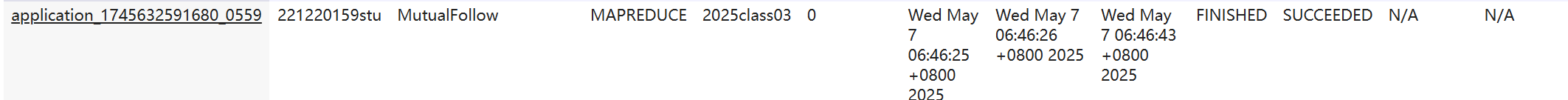
⚪**reducer**输出的key的类型为Text，value为NullWritable

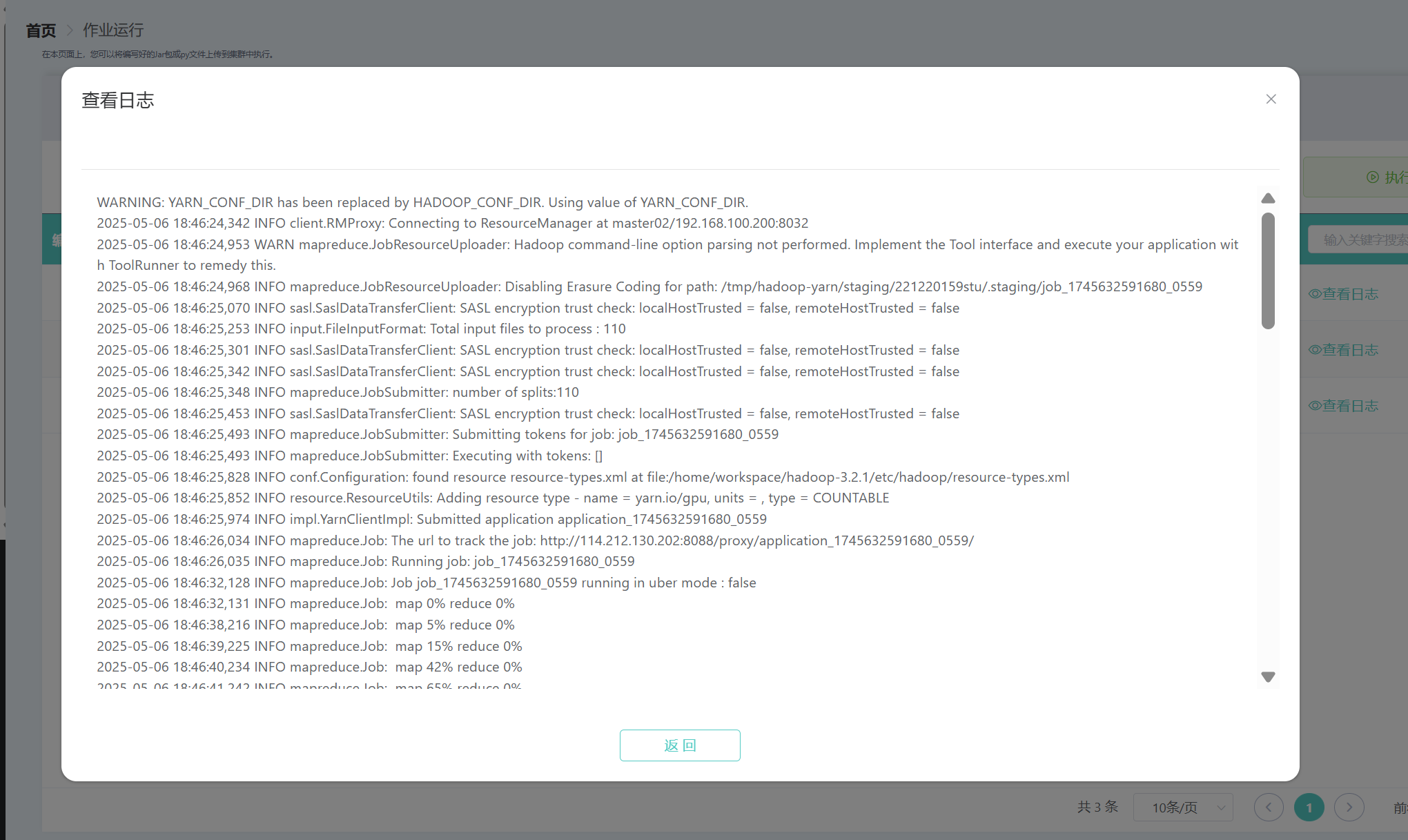


⚪平台输出结果（路径为/user/221220159stu/output-Exp4/1）



⚪平台运行过程





**任务2：社交网络寻找共同关注**

⚪**基本思路**： 根据任务1的输出结果，首先建立 每个用户 与 参与的所有互相关注对 的映射，map阶段对于输入的用户，把它参与的互相关注对分别作为键，它的关注列表作为值输出，这样在reduce阶段，对于一个互相关注用户对（如1-2），能接收到1的关注列表和2的关注列表，从而计算它们的共同关注并输出

⚪**输入**依次为：①.原始关注数据（user:follow1 follow2...）②任务1生成的互相关注用户对（如1-2）③x

**输出**：

共同关注数 ≤x 的结果到 lessX 文件（如less15）

共同关注数 >x 的结果到 greaterX 文件（如greater15）

⚪**Mapper**需要加载互相关注用户对数据，并为每个用户对关联其关注列表

首先在setup阶段，将每个用户相关联的所有互相关注用户对存到一个Map中，例如用户1对应1-2，1-3



map阶段，读入每个用户的关注列表，遍历它参与的所有互相关注用户对，将关注列表作为值传给reducer（map输出的键和值类型均为Text）



⚪**Reducer**的setup阶段，首先根据传入的x，动态生成输出文件名

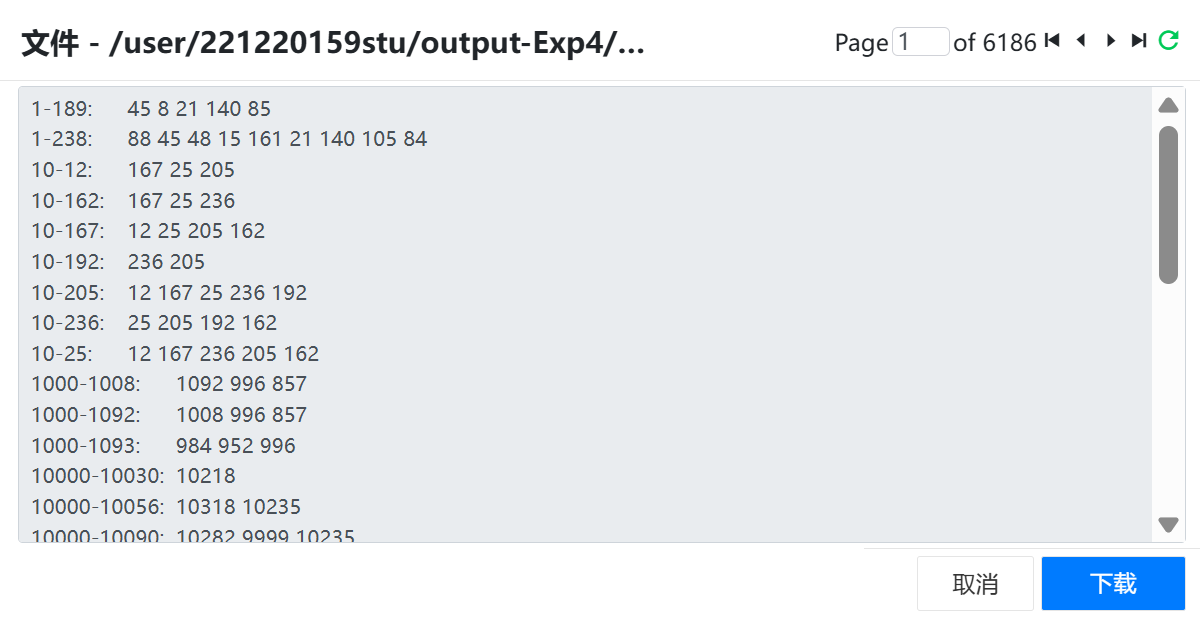


reduce函数中，分别存储用户对中两个用户的关注列表，通过集合的retainAll方法筛选出共同关注，并根据共同关注数量选择输出路径（输出的键和值类型均为Text）



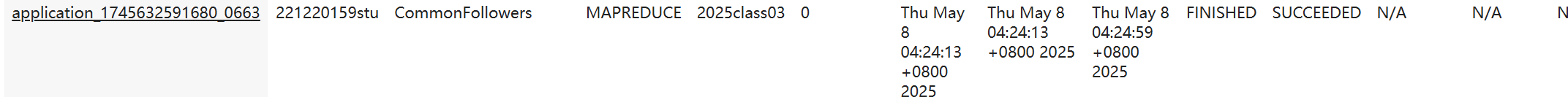


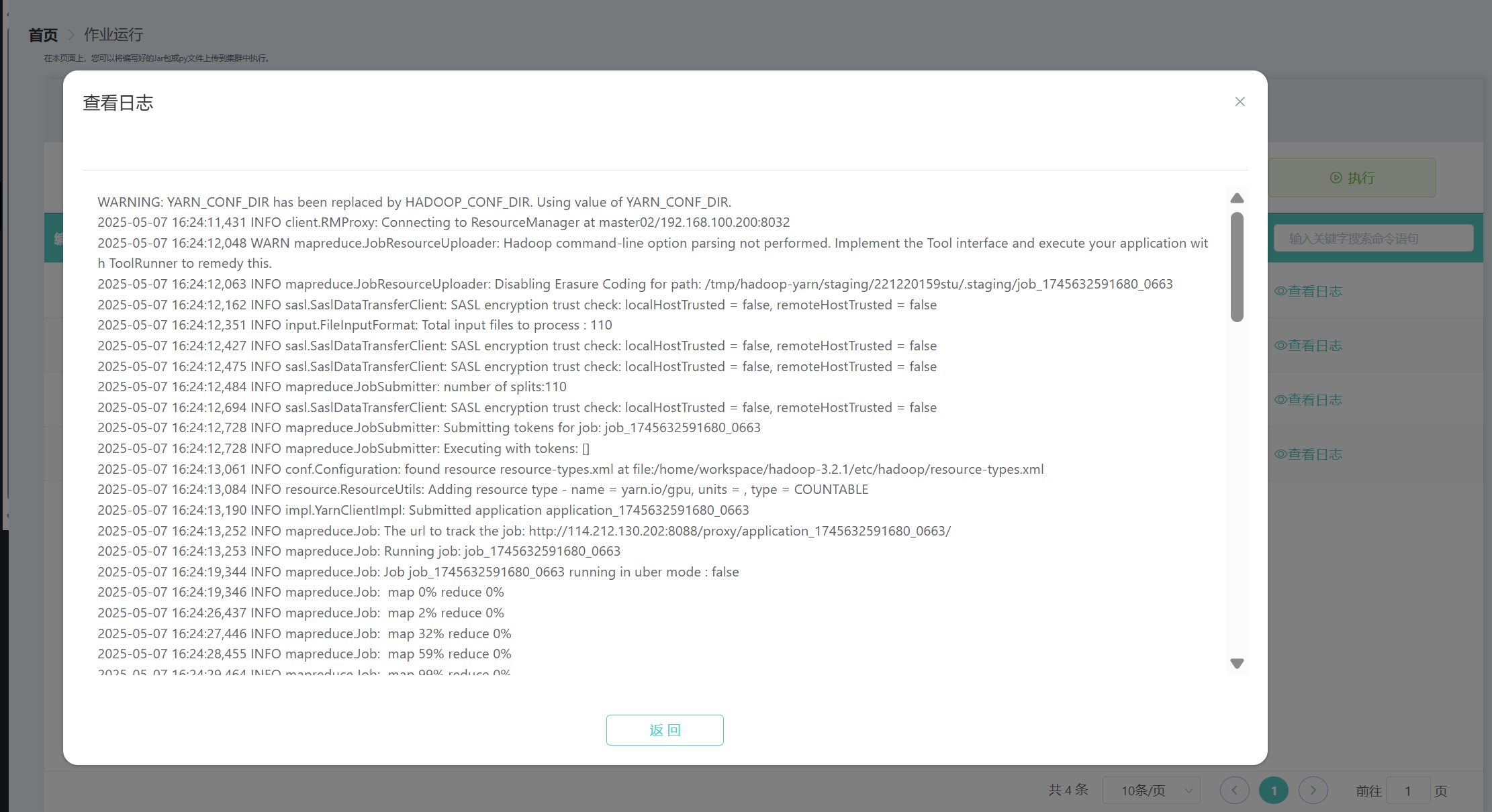
⚪平台输出结果（路径为/user/221220159stu/output-Exp4/2）（x=15）





⚪平台运行过程





**任务3：好友推荐**

⚪**基本思路**：观察示例可知，如果用户B和用户A互关，用户A和用户C互关，但B没有关注C，则向B推荐C。因此可以根据任务1的输出结果建立每个用户互相关注的用户集合，对于每个用户，遍历它互相关注的互相关注，如果它没有关注则推荐，再将计数前5的结果输出即可。

⚪**Mapper**的setup阶段，从互相关注用户对中建立每个用户互相关注的用户集合



map函数，首先存储输入用户的关注列表，再遍历它互相关注的互相关注，如果没有关注则输出（map的key和value的类型均为Text）



⚪**Reducer**对于每个用户的候选被推荐用户，用一个Map记录推荐次数，排序后输出推荐次数前5的用户（键和值类型均为Text）



⚪平台输出结果（路径为/user/221220159stu/output-Exp4/3）



⚪平台运行过程

