

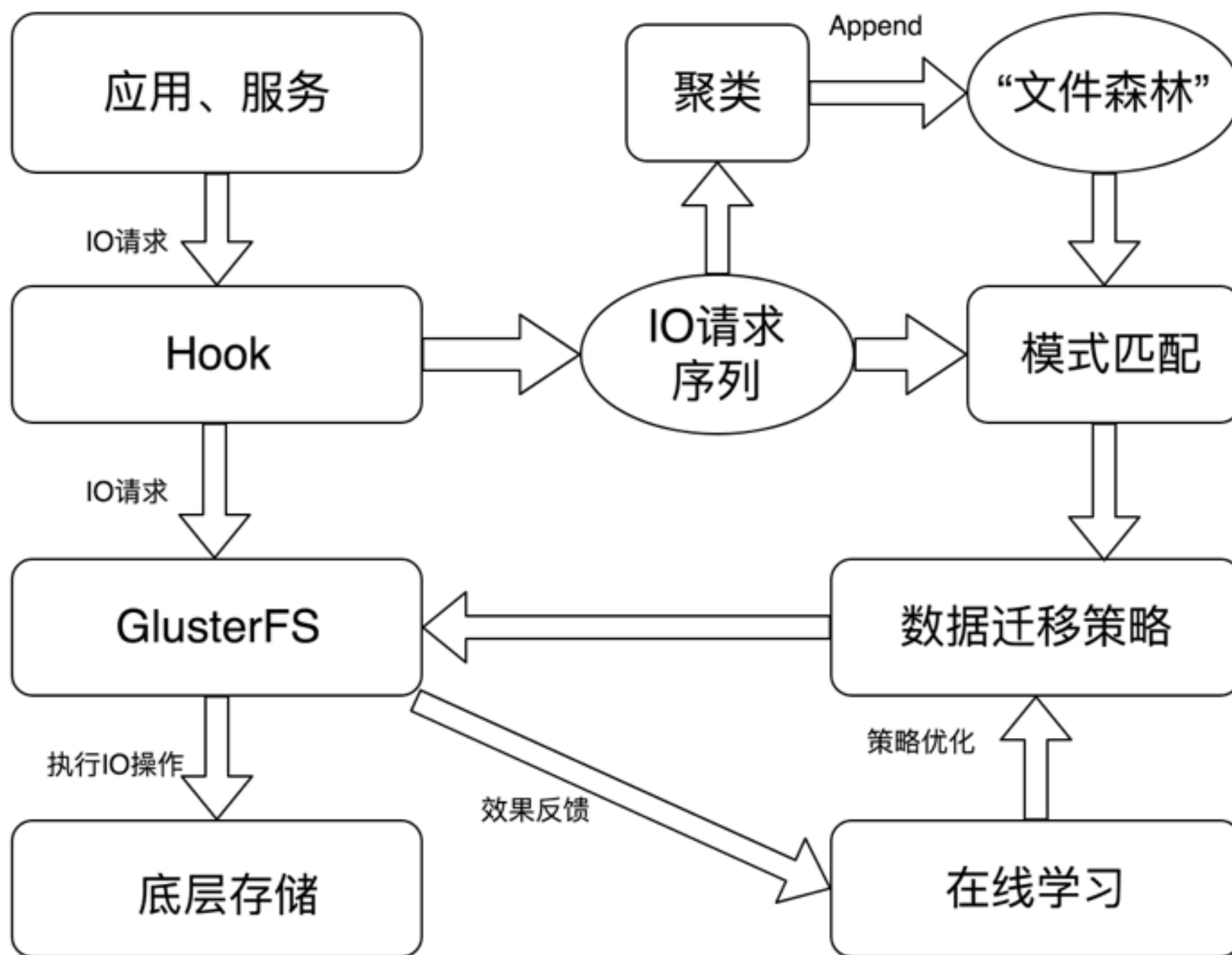
# 基于有向带权图的数据 迁移算法研究

陈辉

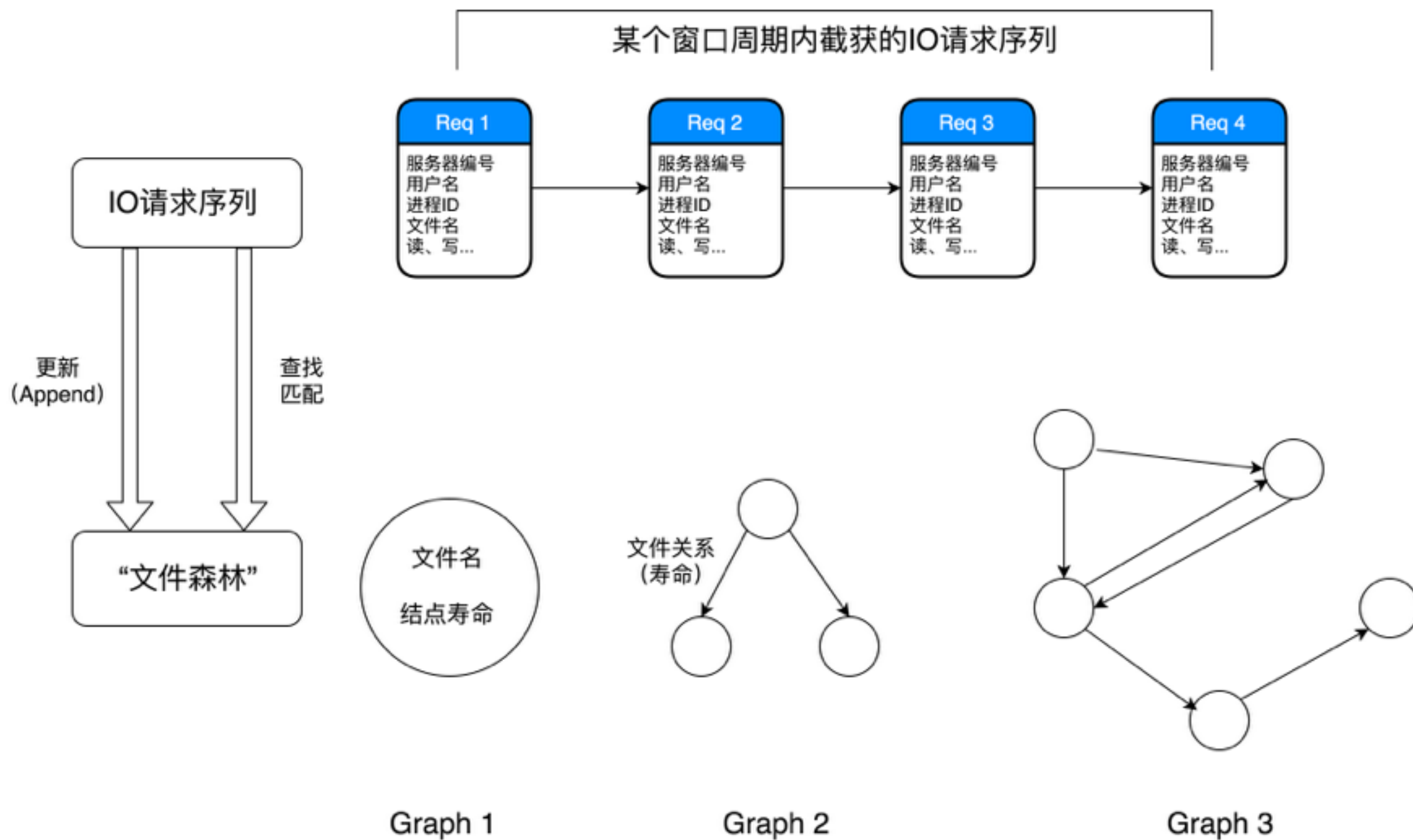
# 概述

- 本研究旨在优化分布式计算环境IO性能，主要思路是将IO请求序列转化为有向带权图（集）以分析和预测将要读写的文件，将其预先加载至SSD，从而降低IO延迟。

# 整体工作流程

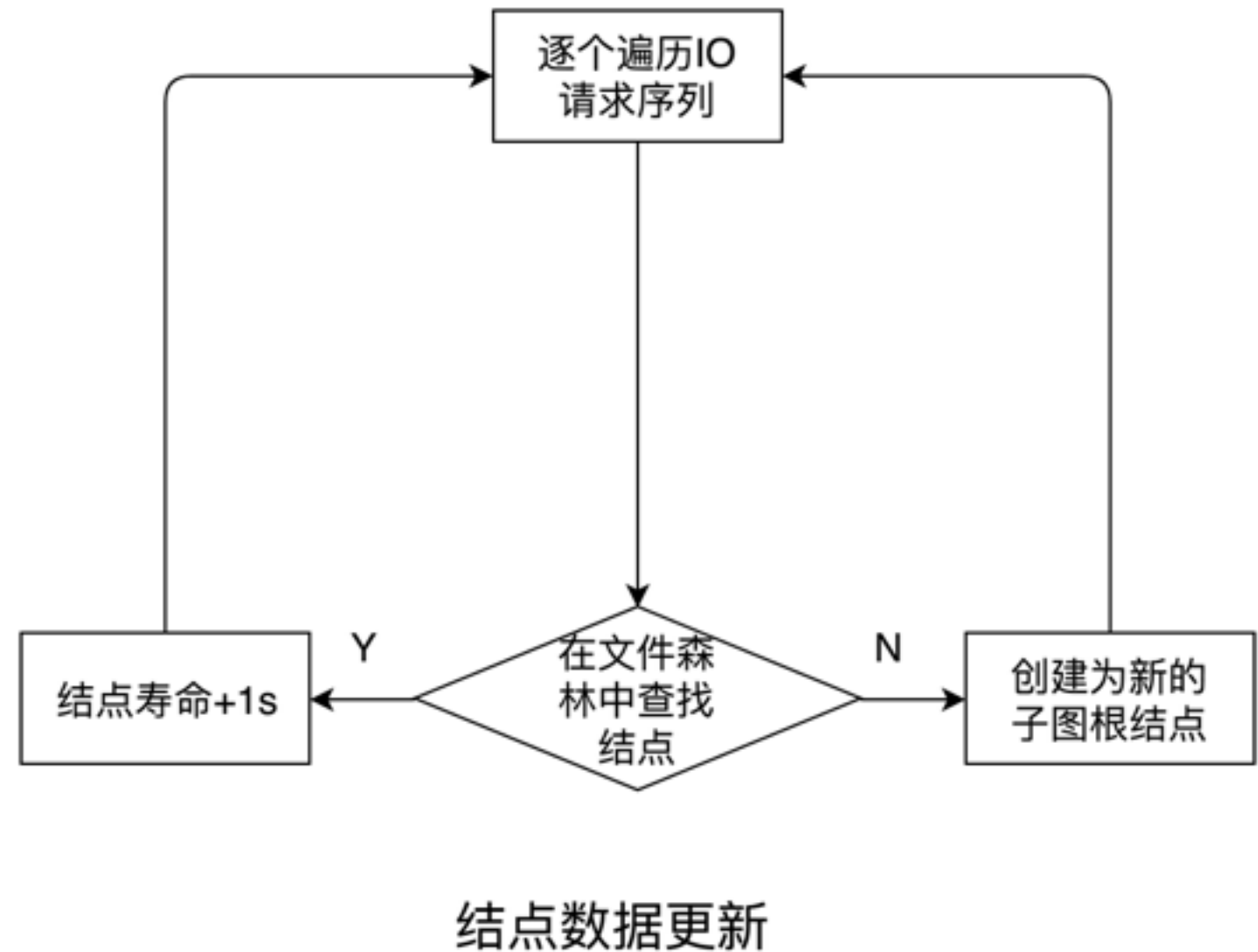


# 主要数据结构



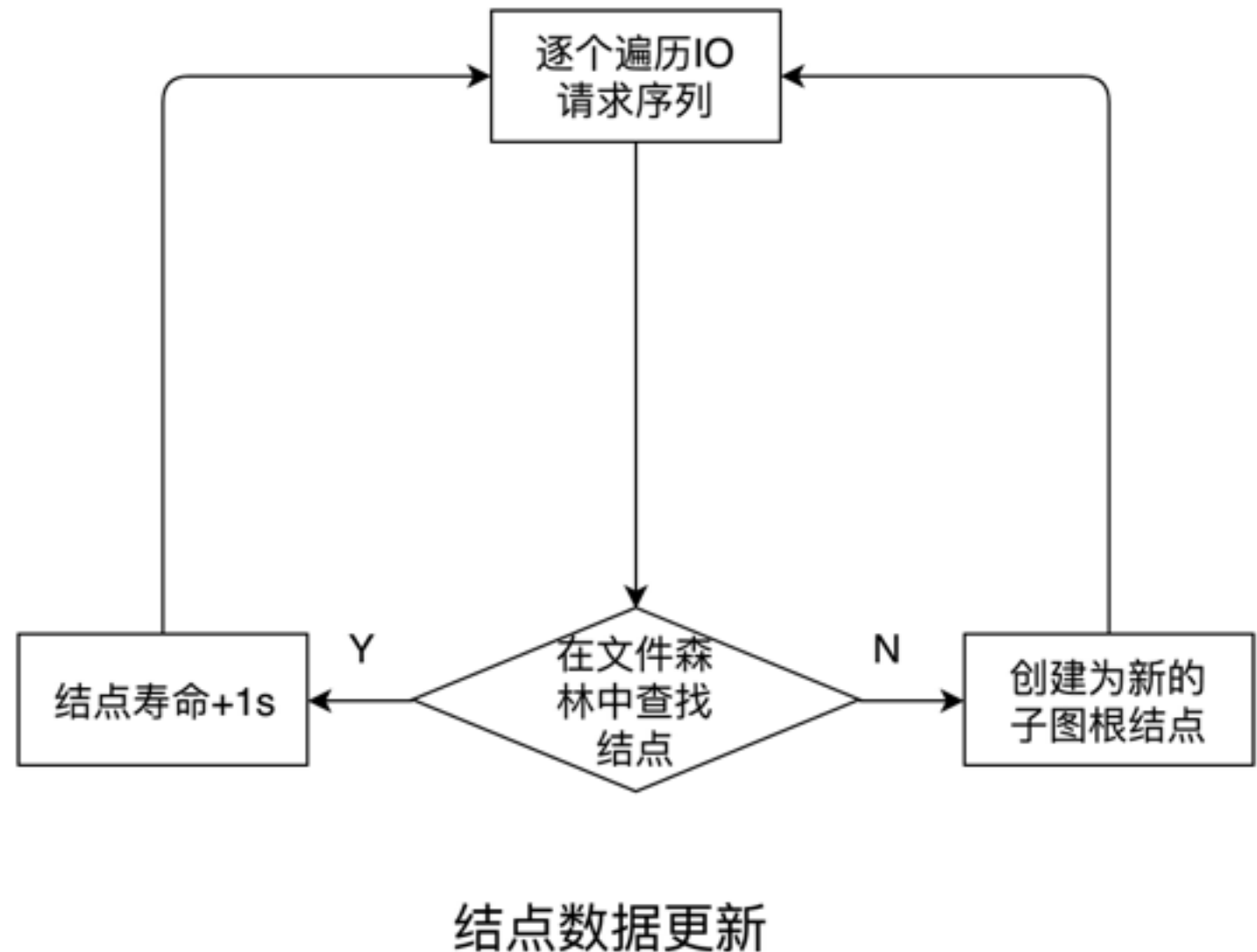
# “文件森林”的生成与更新

- 为什么要定义结点寿命？
- 1. 结点寿命能确切地量化该结点文件在系统运行过程中的热度。
- 2. 结点寿命会随着时间推移衰减，衰减至0时从文件森林中删除。



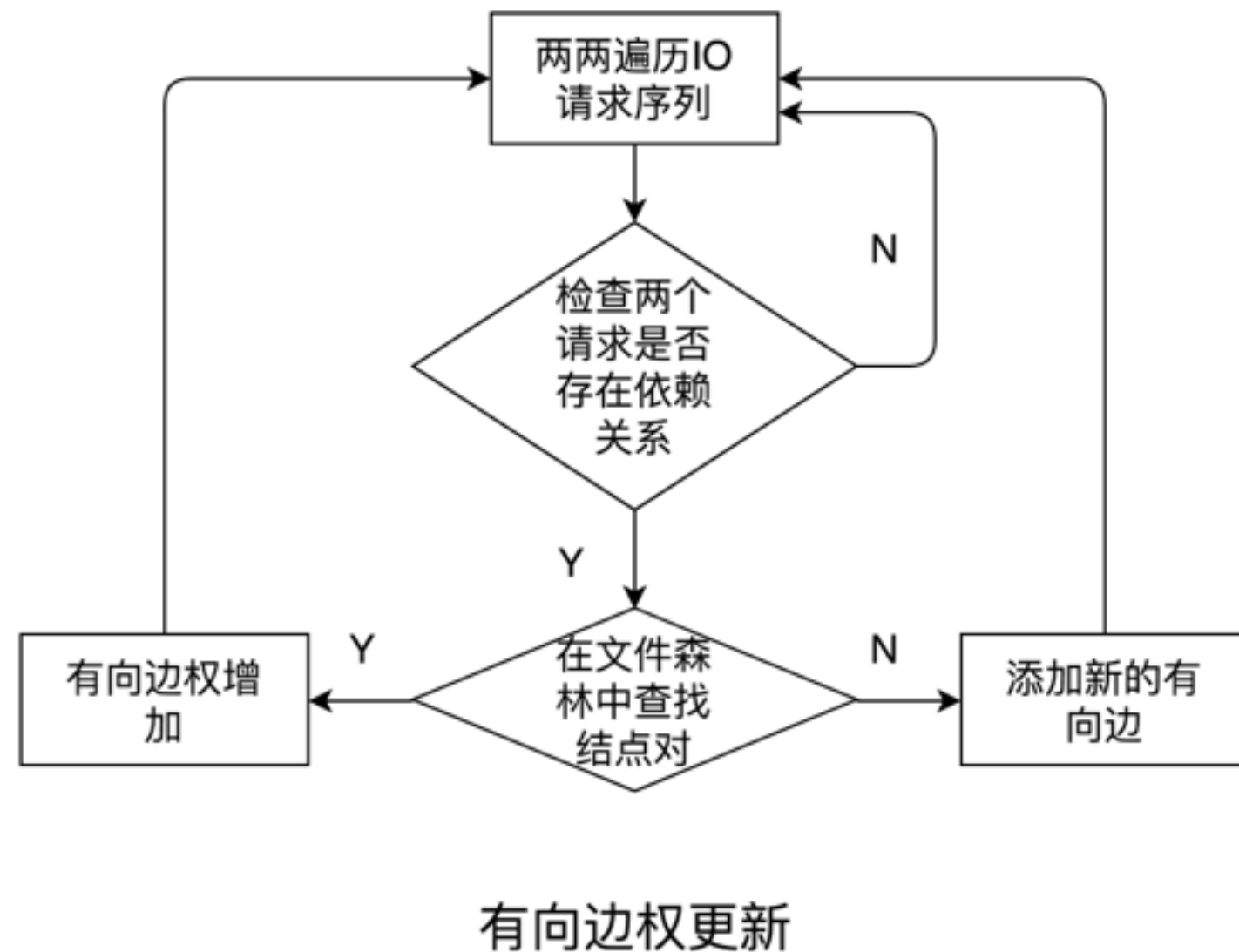
# “文件森林”的生成与更新

- 结点寿命衰减的原因？
  1. 长时间未被读写的文件热度会自然下降。
  2. 任何文件在系统运行过程中都可能被重命名、删除，完全掌握这些过程开销太大且不必要，随时间自然消除效果更好。



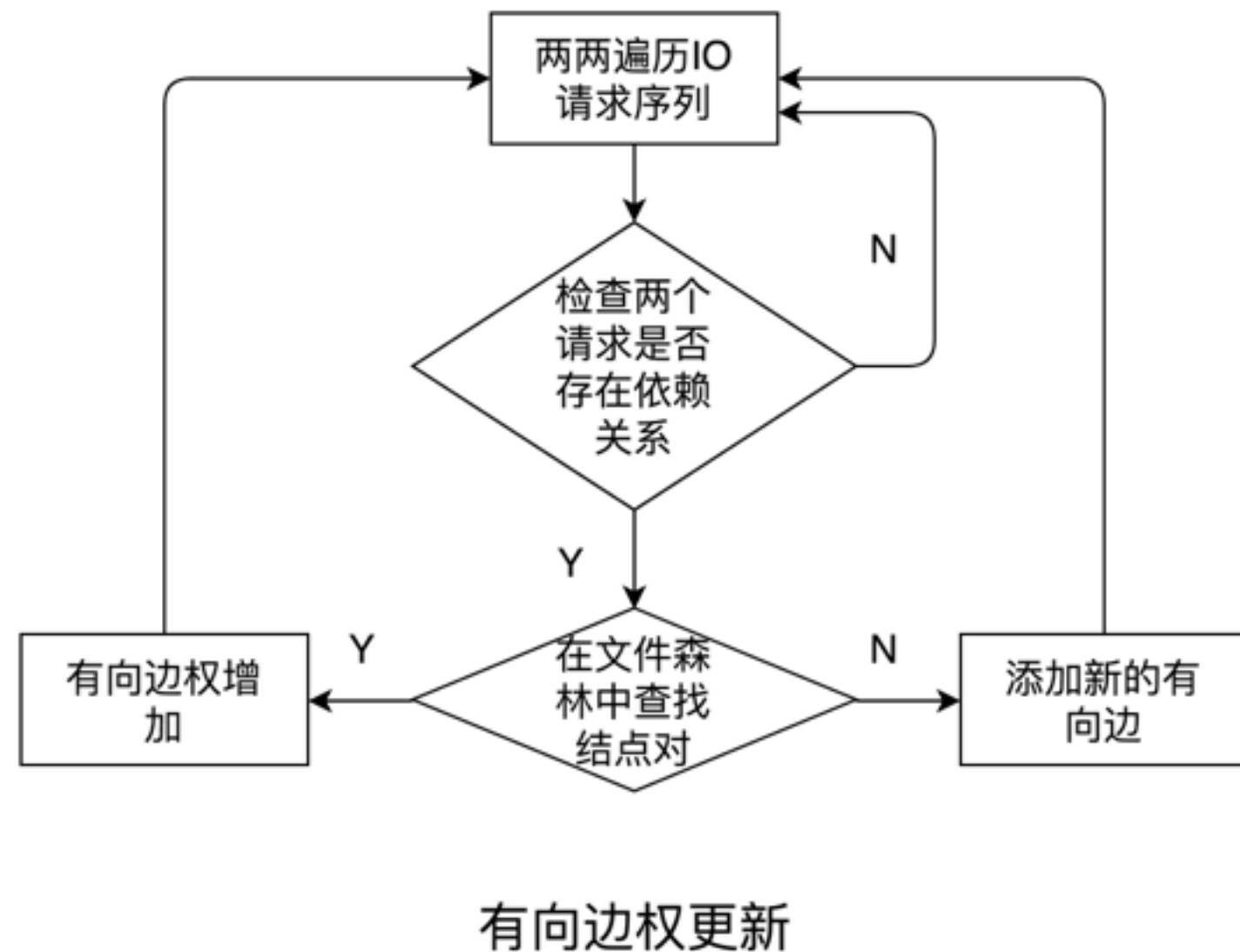
# “文件森林”的生成与更新

- 有向边包含的信息？
  1. 方向表示两个文件（也可以是自身，如先读后写）的时序关系。
  2. 权重代表依赖关系的频率、可信度等。



# “文件森林”的生成与更新

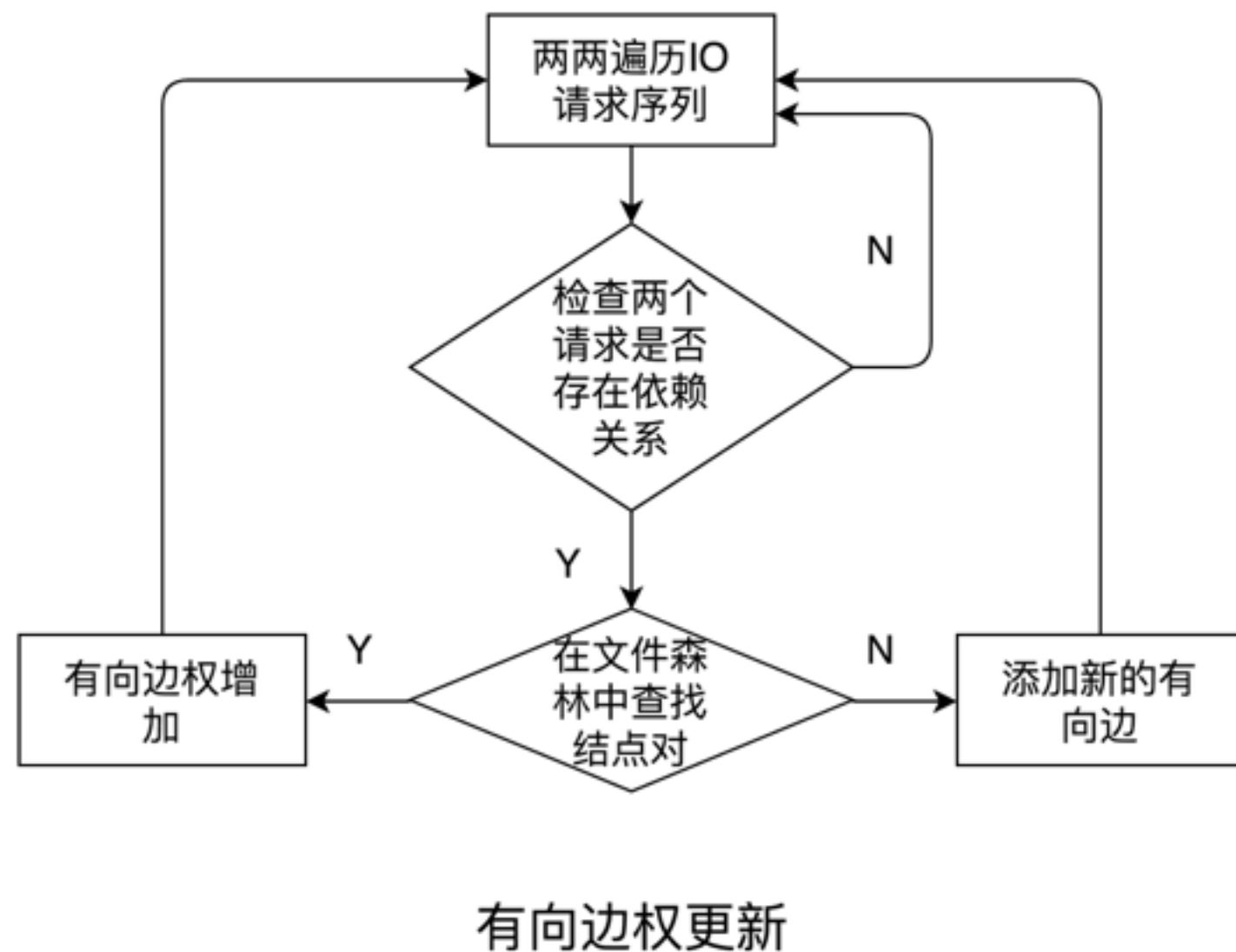
- 边权如何计算？
- 原始的IO请求序列包含用户名、进程（线程）ID、文件名等标识信息，当至少有一项相同时，说明存在依赖关系，不同的组合代表不同形式的依赖。





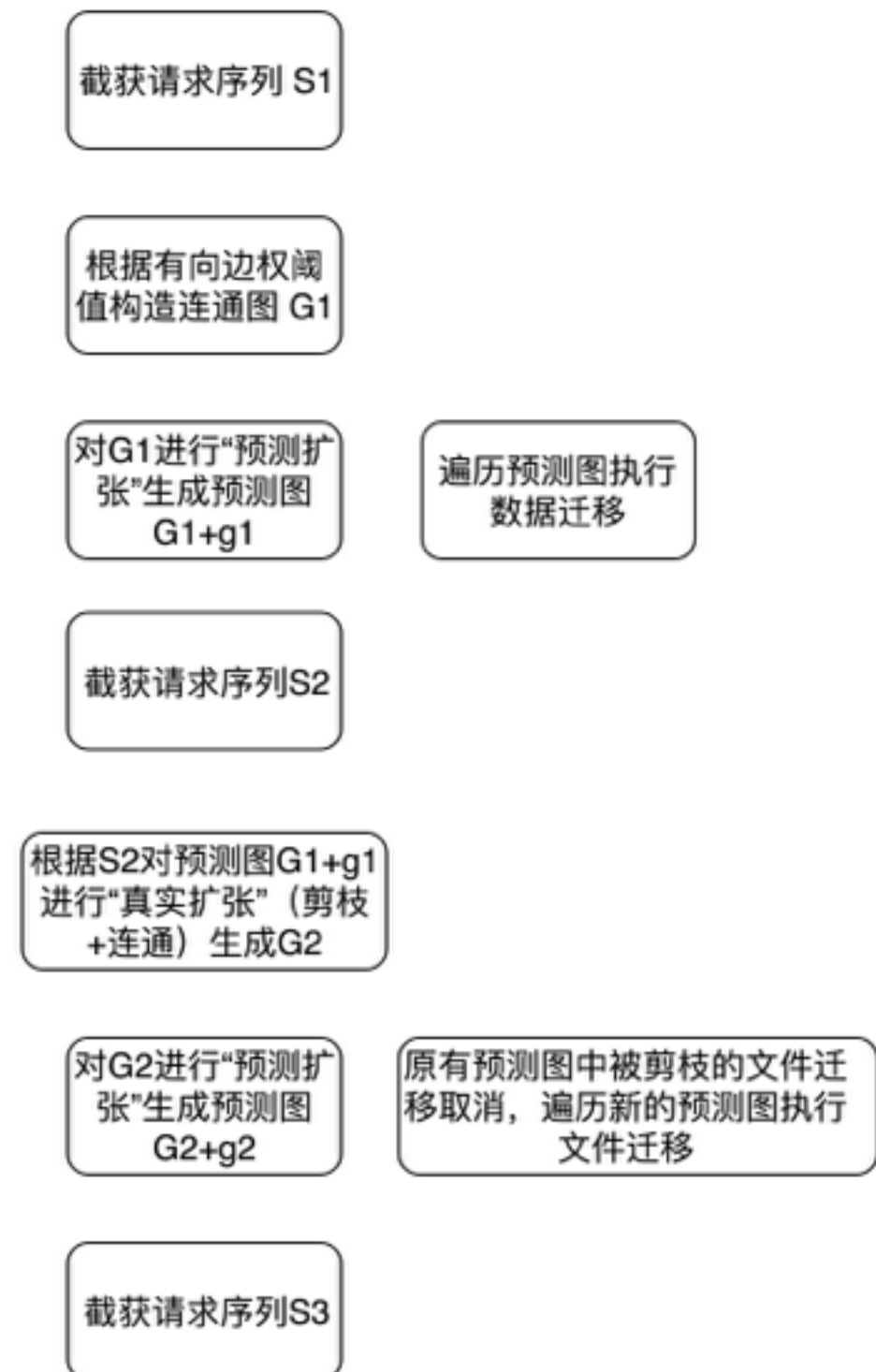
# “文件森林”的生成与更新

- 边权和结点寿命一样也会衰减，且衰减速度应相对更高。  
理由：
- 1. 两个文件先后读写未必真的具有逻辑上的依赖关系，需要较高的边权、频繁地出现才有一定的置信度。
- 2. 图算法中边对复杂度的影响相比结点更显著，过多无用的边会严重影响性能



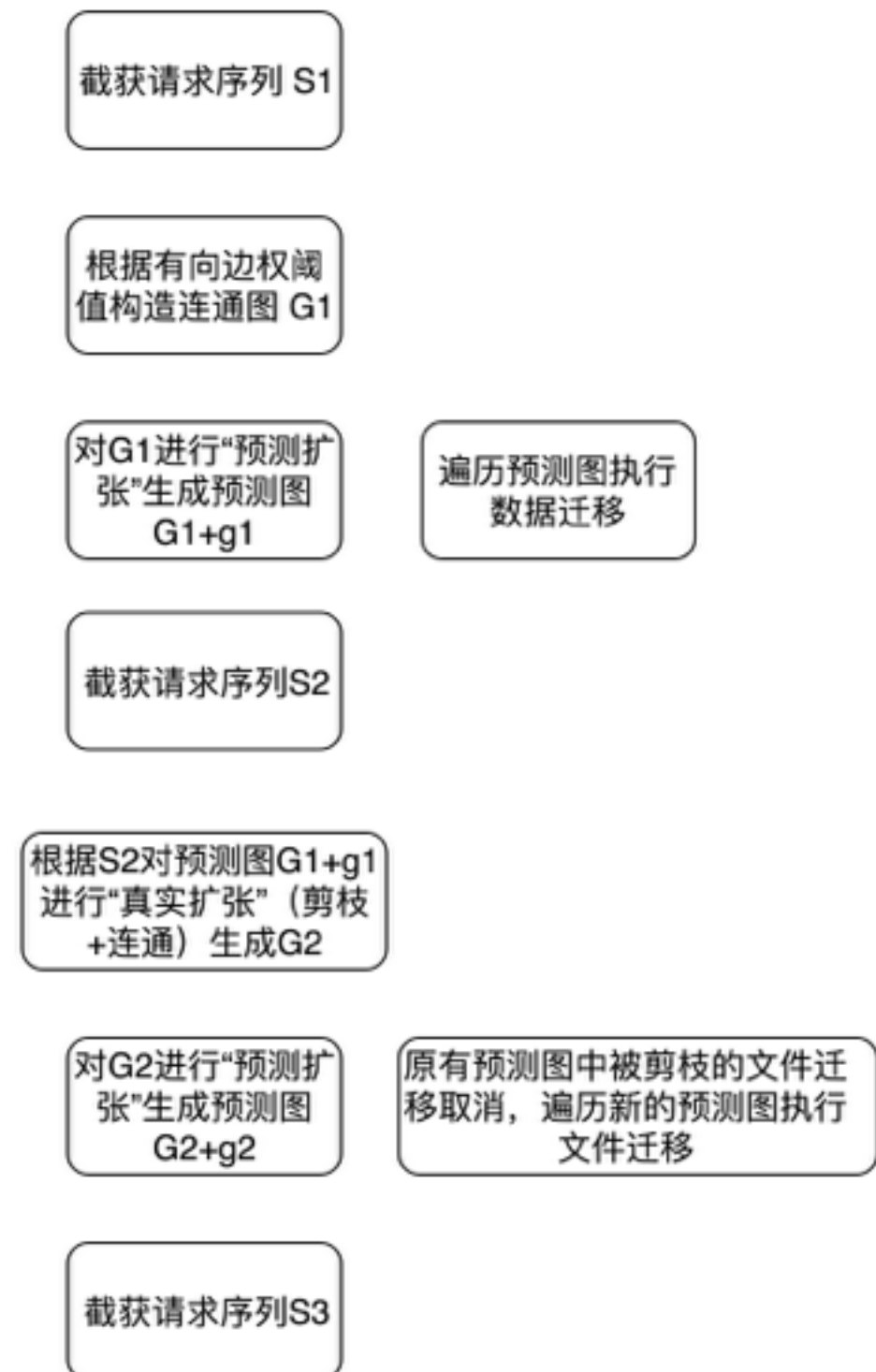
# 数据迁移策略

- 数据迁移的过程主要包括以下操作：
- 1. 截获IO请求序列。
- 2. 将IO请求序列在“文件森林”中生成若干个子连通图。
- 3. 在连通图的基础上向外遍历，以预测接下来要读写的文件。
- 4. 截获新的IO请求序列，对原来的子连通图进行修正。



# 数据迁移策略

- 连通图操作的原则：
  1. 边权大于设定阈值才视为连通。
  2. 将IO请求序列在“文件森林”中生成若干个子连通图。
  3. 在连通图的基础上向外遍历，以预测接下来要读写的文件。
  4. 截获新的IO请求序列，对原来预测的子连通图进行修正（剪枝或生成新的子图）。错误的预测分支不会马上整体剪枝，而是在生命周期内由外至内衰减。
  5. 真实图遍历完毕后，会逐渐消亡（退化为预测图）以腾出SSD空间。若后续新的IO请求在图内部，则保持该局部结点附近的“真实性”。



# 测试环境

- 前期以分布式环境下大型工程编译作为工作负载进行测试