

实验二 模拟DHT文件存取

实验目的

通过模拟实现Kademlia DHT中的文件读取，加深对IPFS的理解和认识。与此同时，增加大家对于算法的理解，同一个算法，既可以通过RPC调用，也可以通过调用相关对象的方法进行模拟，让大家对编程有更深入的了解。

实验前提：

1. 掌握了Kademlia DHT算法的基本原理和相关概念；
2. 学生应具备基本的编程知识和对Golang或TypeScript编程语言的了解；
3. 假设地址长度是20byte，每个桶中节点的数量为3个；
4. 大家已经完成实验一中的要求，每个程序启动了多个Peer，每个Peer中的桶不一样。

实验内容

假设大家已经实现了实验一中的内容，本次实验内容为：

- a. 每个节点中有100个Peer，每个Peer中有一个DHT结构体，这个DHT结构体中保存了至多160个桶，桶里面是其他节点的信息（Peer对象）。
- b. 为每个Peer增加两个函数：
 - i. SetValue(key, value []byte)，当一个Peer被调用这个函数时：
 1. 判断Key是否是Value的hash，如果不是，返回false；
 2. 判断当前Peer是否已经保存了这个键值对，如果已经保存，则返回true，否则保存这个键值对，并执行第3步；
 3. 判断Key距离自己的PeerID的距离，算出这个节点对应的桶，然后从对应的桶里面选择2个距离Key最近的节点，再调用SetValue(Key, Value)函数，然后返回true。
 - ii. GetValue(Key []byte) []byte：当一个Peer被调用这个函数时：
 1. 判断当前的Key自己这个Peer是否已经存储对应的value，如果自己这个Peer中有，则返回对应的value；
 2. 如果自己没有存储当前Key，则对当前的Key执行一次FindNode操作，找到距离当前Key最近的2个Peer，然后对这两个Peer执行GetValue操作，一旦有一个节点返回value，则返回校验成功之后的value，否则返回nil。

功能测试

1. 在main函数中，先完成对100个节点的初始化，即每个节点在什么桶里；
2. 随机生成200个字符串，长度内容都是随机，然后计算出他们的Hash之后，随机从100个节点中选出一个执行SetValue(Key, Value)操作，并找一个数组记录这200个字符串的Key；
3. 从200个Key中随机选择100个，然后每个Key再去随机找一个节点调用GetValue操作。

实验报告

1. 需要撰写实验报告，记录实现过程中的思考、遇到的问题和解决方案；
2. 实验报告可以附上核心代码，完整代码放在github自己的仓库中；
3. 报告中还应包含运行截图和分析；
4. 报告通过PDF的方式发送给学委，学委汇总后发到我的邮箱806275025@qq.com。