**分布式全网存储和检索——kad**

**Kademlia简述**

Kad属于一种典型的结构化P2P覆盖网络(Structured P2P Overlay Network)，以分布式的应用层全网方式来进行信息的存储和检索是其尝试解决的主要问题。

**Kad网络中究竟存储了哪些信息**

所有信息均以的哈希表条目形式加以存储，这些条目被分散地存储在各个节点上，从而以全网方式构成一张巨大的分布式哈希表

网络存储、维护两张分布式哈希表：关键词字典、文件索引字典。

* + **关键词字典**：关键词→其所对应的文件名称及相关信息，key=关键词字符串的160比特SHA1散列，value为一个三元组列表 (文件名，文件长度，文件的SHA1校验值) 。
  + **文件索引字典**：文件信息→文件的拥有者(下载服务提供者)，key=文件的SHA1校验值，value也是一个三元组列表 (拥有者IP，下载侦听端口，拥有者节点ID)。

**节点的ID和节点之间的距离**

每一个节点有一个专属ID（一个160bit的整数），由节点自己随机生成的。距离为两个ID的二进制异或值。 避免物理相近的节点同时宕机情况。

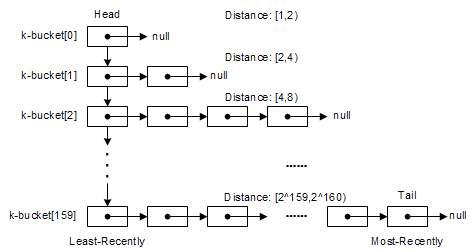
**Kad存储**

Kad网络规定：条目依据其key值被复制到目标节点ID距离最近的k个节点中。

k取值准则：“任意选择至少k个节点，它们在任意时刻同时不在线的几率几乎为0”。

k的典型取值为20。

**Kad网络节点维护的状态信息**



List（k-桶）的更新原则：

1）目标节点信息已经在list中，将其移至队尾；

2）list未满，且目标节点不在其中，其信息将直接添入list队尾；

3）list已满，先检查队首节点是否仍有响应，如果有，则队首节点被移至队尾，目标节点被抛弃；如果没有，则抛弃队首节点，将最新访问的节点信息插入队尾。

**Kad寻找节点**

查找与目标节点网络距离最近的k个节点所对应的网络信息(NodeID，IP地址，UDP端口)。

1）发起者从自己的k-桶中选出若干距离目标ID最近的节点，并向它们同时发送异步查询请求；

2）被查询节点收到请求后，从自己的k-桶中找出自己所知的目标ID的若干近邻返回给发起者；

3）发起者收到返回信息后，再次从当前已知的近邻节点中选出若干未被请求的，并重复步骤1。

上述过程直至无法获得k近邻的更新时停止。

在查询过程中没有及时响应的节点将立即被排除。

**Kad条目搜索**

搜索发起方以迭代方式不断查询距离key较近的节点

↓

直至查询路径中的任一节点返回所需查找的value。

**Kad新节点加入**

1. 与节点I通信，获取节点I信息
2. 对节点I查询自己的ID，获得与自己距离相近的节点
3. 刷新K-桶