Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems
Assignment 10

1 Distributed Storage

Quiz

1.1 Hypercubic Networks

Draw the following hypercubic networks:

a) $M(3,1)$

b) $M(3,2)$

c) $SE(2)$

d) $M(2,4)$

Basic

1.2 Iterative vs. Recursive Lookup

There are two fundamental ways to perform a lookup in an overlay network: recursive and iterative lookup.

Assume node n_0 is attempting to look up an object in a DHT. In the recursive lookup n_0 selects a node n_1 which is closest according to the DHT metric and sends a request to it. Upon receiving the request n_1 selects its closest known neighbor n_2 and forwards the request to it and so on. The request either ends up at the node storing the object, returning the object along the same path, or it ends at a node that does not store the object and does not have a closer neighbor.

In the iterative case n_0 looks up the closest neighbor n_1 and sends it the request. Upon receiving the request n_1 is either the node storing the object and it returns the object, or it knows a closer node n_2 and returns n_2 to the n_0. If n_0 receives a node n_2 it will add it to its neighbor set and sends a new request to n_2 which is now its closest neighbor. The lookup terminates either when n_0 sends a request to the node storing the object, or no closer node can be found.

a) What are the advantages of recursive lookups over the iterative lookups?

b) Most systems that are in use today use the iterative lookup, and not the recursive lookup, why?

1.3 Building a set of Hash functions

Consistent hashing relies on having k hashing functions $\{h_0, \ldots, h_{k-1}\}$ that map object ids to hashes. There are several constructions for these hash functions, the most common being iterative hashing and salted hashing. In iterative hashing we use a hash function h and apply it iteratively so that the hashes of an object id o are defined as

$$h_i(o) = \begin{cases} h(o) & \text{if} \quad i = 0 \\ h(h_{i-1}(o)) & \text{otherwise.} \end{cases}$$

With salted hashing the object id is concatenated with the hash function index i resulting in the following definition

$$h_i(o) = h(o|i).$$

Which hashing function derivation is better and why?

Advanced

1.4 Multiple Skiplists

In the lecture we have seen the simple skip list in which at each level nodes have probability $1/2$ of being promoted to the next level. We have also discussed a variation known as a skip graph. For yet another option, we once again redefine the promotion so that a node is promoted to a list s if s is a suffix of the binary representation of the node's id. At each level l we now have 2^l lists (some empty), each defined by a string of bits s of length l. In particular, the root level $l = 0$ is constructed with s being the empty string. The second level has one list for each $s \in \{0, 1\}$, the third level one list for each $s \in \{00, 01, 10, 11\}$, and so on. We call the resulting network a multi-skiplist. For the purposes of this question, assume that all lists are circular.

a) Assuming we have an 8 node network, with ids $\{000, \ldots, 111\}$, draw the multi-skiplist graph.

b) What is the minimum degree of a node in the multi-skiplist if we have d levels?

c) What is the maximum number of hops a lookup has to perform?