

CS-438

# Decentralized Systems Engineering

Week 5

# Search & routing

Unstructured - last week  $\approx O(n) \dots O(\sqrt{n})$  (flooding ...)

Structured - this week  $\rightarrow O(\log n)$

## Distributed Hash Tables (DHTs)

- Functional API: stores key/value pairs

- GET (key)  $\rightarrow$  value/error

- PUT (key, value)  $\rightarrow$  success/error

- Goal: distribute state & work across network

Some interesting DHT algorithms:

- Chord, Pastry, Kademlia, ...

# Chord

uses hashes, e.g. SHA256 - for both node IDs & keys

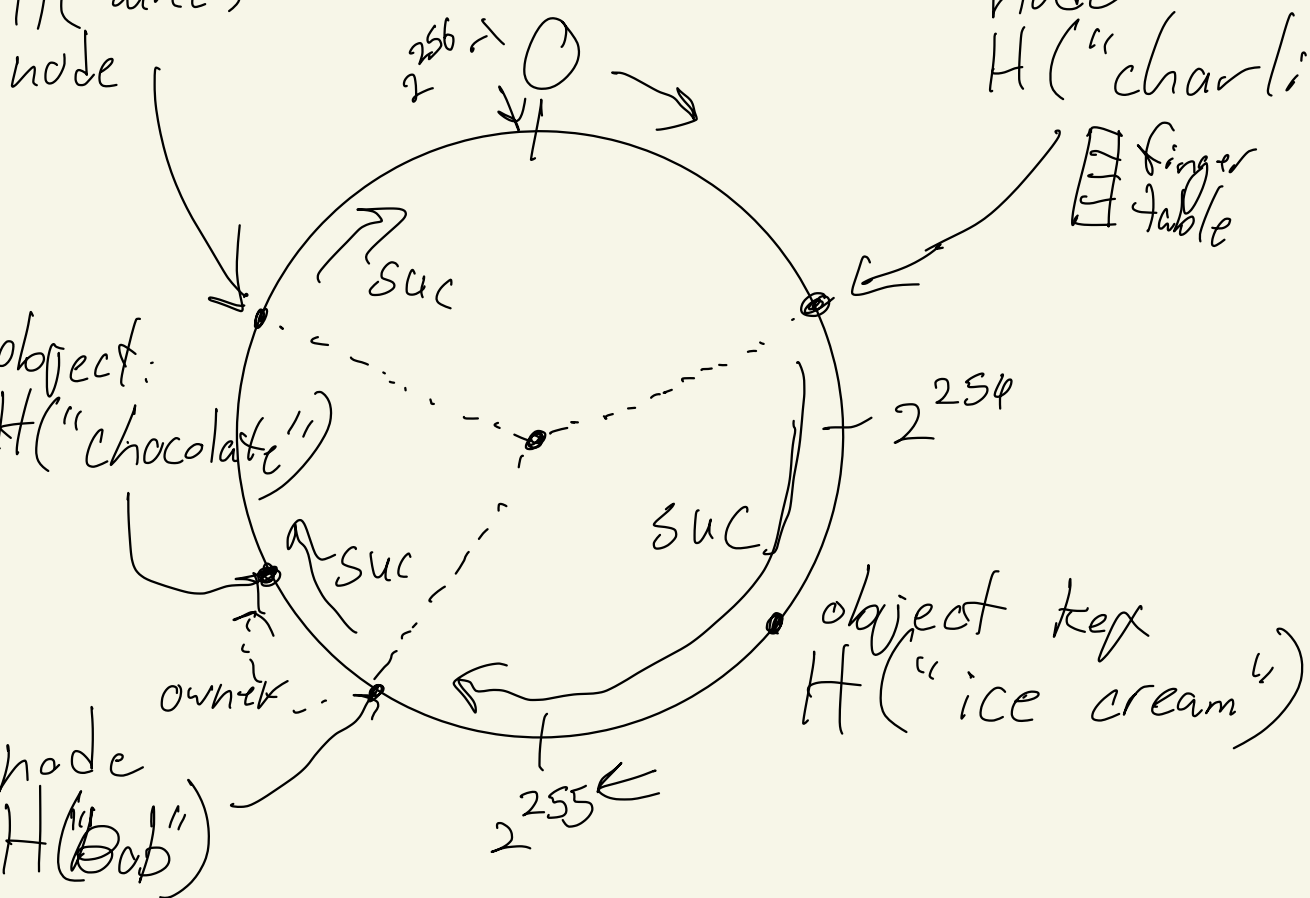
$H(\text{"alice"})$

node

node  
 $H(\text{"charlie"})$

100 objects  
20 nodes

$\rightarrow 5$  objs/node



each node remembers  
its key/value pairs

balanced:  
perfectly  $O(\frac{n_o}{n_n})$   
statistically  $\max(\frac{n_o}{n_n} \log n_n)$

How do we find a node or object?

- follow successor pointers until found

