# CLOUD COMPUTING CONCEPTS

with Indranil Gupta (Indy)

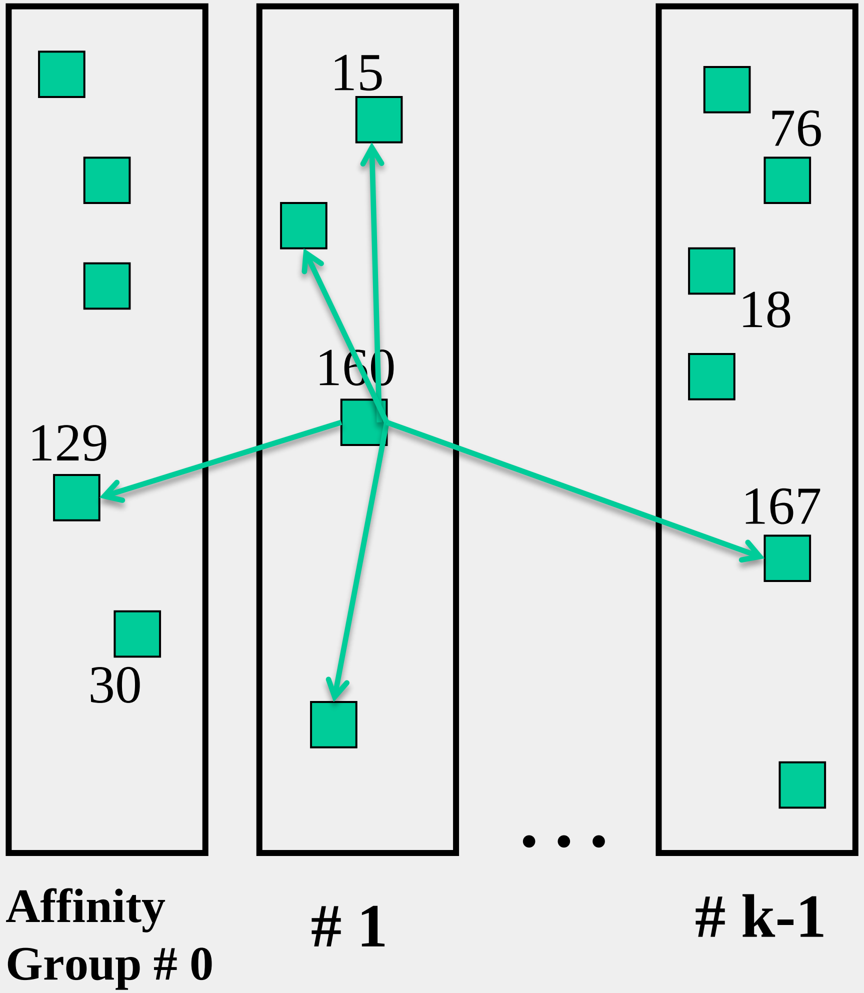
## P2P SYSTEMS

Lecture H

## KELIPS

# Kelips – A 1 hop Lookup DHT

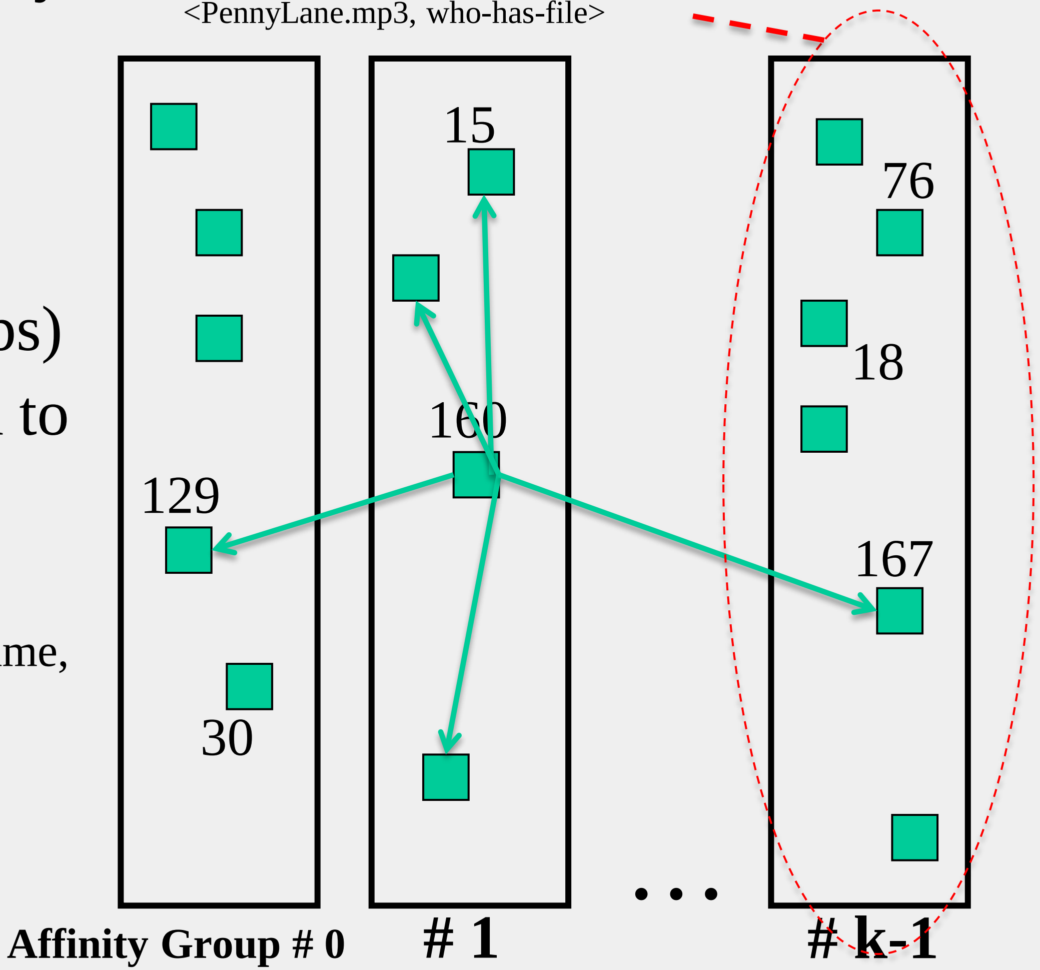
- k “affinity groups”
  - k ~ √ N
- Each node hashed to a group (hash mod k)
- Node’ s neighbors
  - (Almost) all other nodes in its own affinity group
  - One contact node per foreign affinity group



# Kelips Files and Metadata

- File can be stored at any (few) node(s)
- Decouple file replication/location (outside Kelips) from file querying (in Kelips)
- Each filename hashed to a group
  - All nodes in the group replicate pointer information, i.e., <filename, file location>
  - Affinity group does not store files

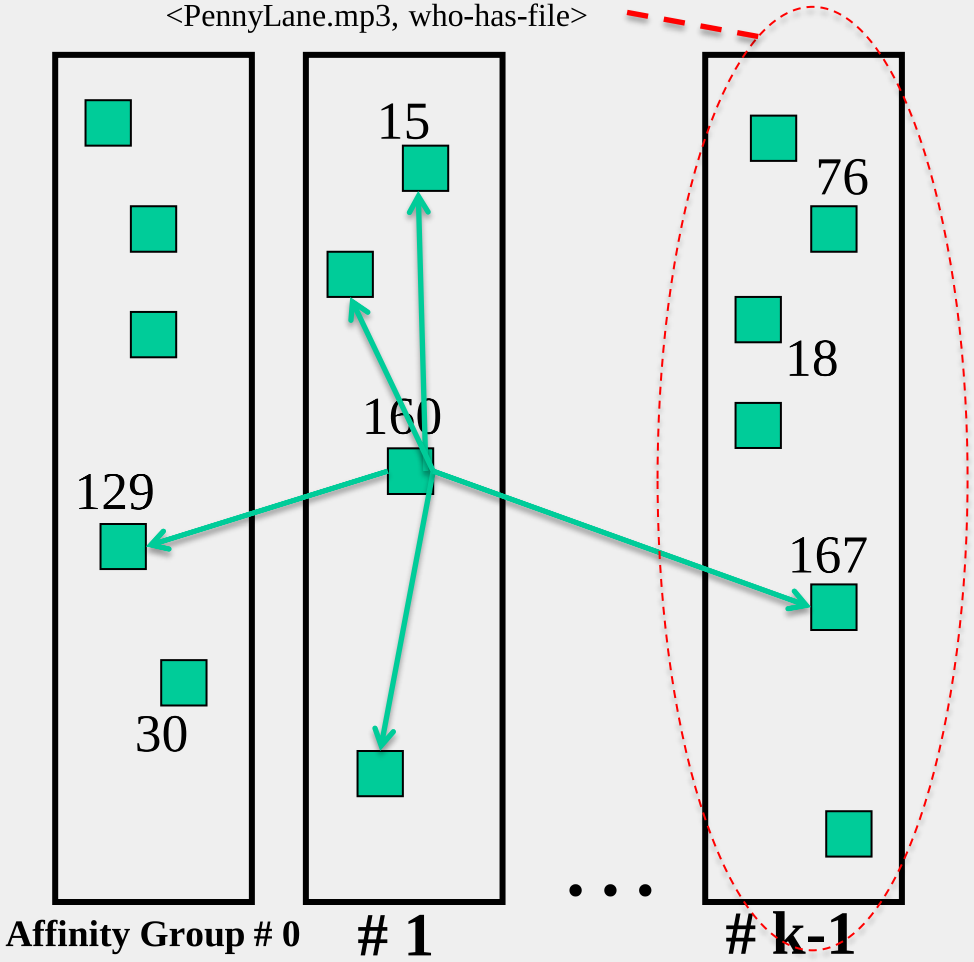
- PennyLane.mp3 hashes to k-1
- Everyone in this group stores <PennyLane.mp3, who-has-file>



# Kelips Lookup

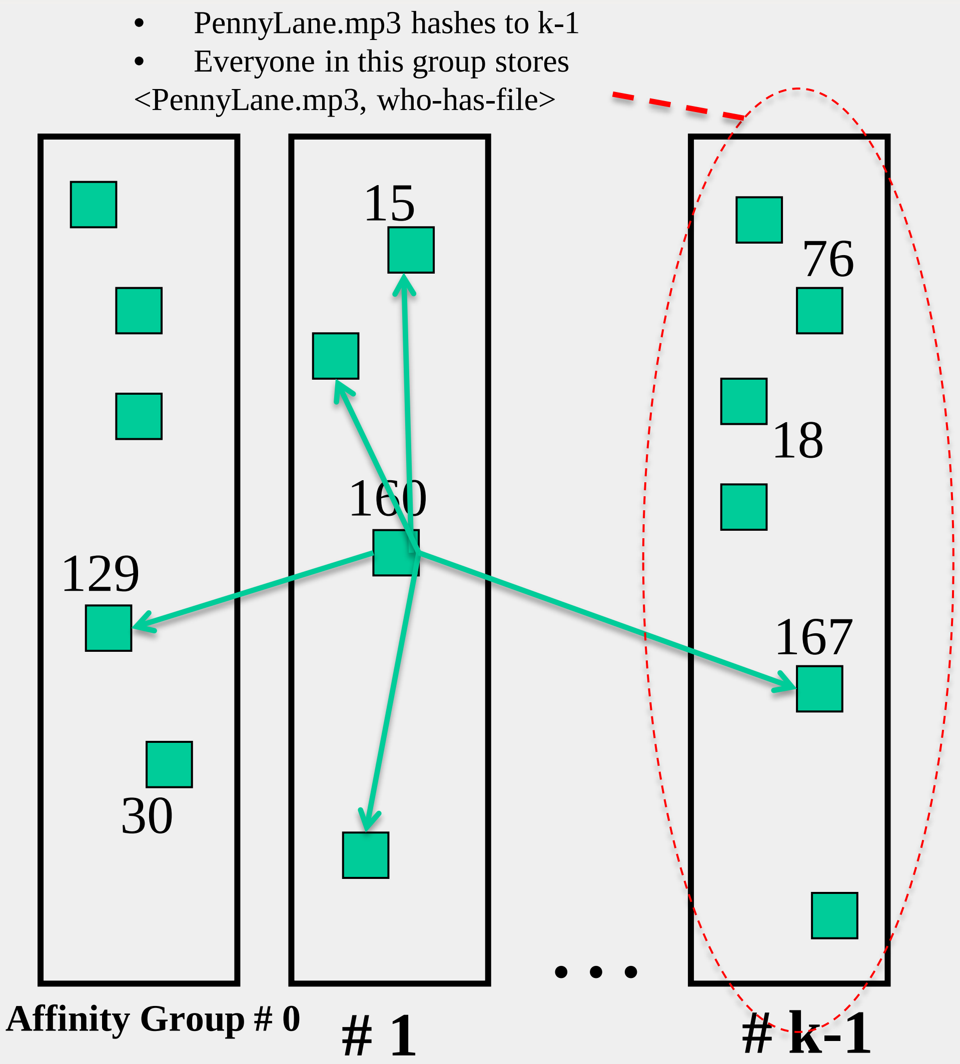
- Lookup
  - Find file affinity group
  - Go to your contact for the file affinity group
  - Failing that try another of your neighbors to find a contact
- Lookup = 1 hop (or a few)
  - Memory cost O(√ N)
  - 1.93 MB for 100K nodes, 10M files
  - Fits in RAM of most workstations/laptops today (COTS machines)

- PennyLane.mp3 hashes to k-1
- Everyone in this group stores <PennyLane.mp3, who-has-file>



# Kelips Soft State

- Membership lists
  - Gossip-based membership
  - Within each affinity group
  - And also across affinity groups
  - *O(log(N))* dissemination time
- File metadata
  - Needs to be periodically refreshed from source node
  - Times out



- PennyLane.mp3 hashes to k-1
- Everyone in this group stores <PennyLane.mp3, who-has-file>

# Chord vs. Pastry vs. Kelips

- Range of tradeoffs available
    - Memory vs. lookup cost vs. background bandwidth (to keep neighbors fresh)

# What We Have Studied

- Widely-deployed P2P systems
  1. Napster
  2. Gnutella
  3. Fasttrack (Kazaa, Kazaalite, Grokster)
  4. BitTorrent
- P2P systems with provable properties
  1. Chord
  2. Pastry
  3. Kelips