Data Mining and Analysis

Algorithmic Toolbox

CSCE 676 :: Fall 2019
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee

This Week: Algorithmic Toolbox

Consistent Hashing

When a hash table is resized, only K/n keys need to be remapped on average, where K is the number of keys, and n is the number of slots.

Bloom Filters

Space-efficient probabilistic data structure that is used to test whether an element is a member of a set.

Consistent Hashing

History

1997: Original paper

1998: Akamai is founded based on consistent hashing (now serves 15-30% of all web traffic)

2001: Distributed hash tables core to peer-to-peer networks (e.g., CHORD); some aspects still in BitTorrent

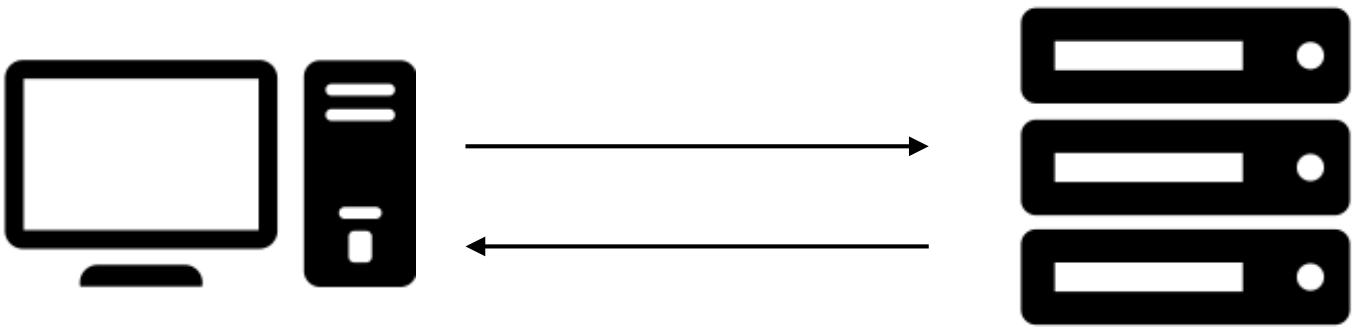
2006: Amazon Dynamo, highly-available key-value store

2010s Foundation of many “clouds” (in NoSQL databases, MapReduce/Spark, Docker, ...)

See more at https://web.stanford.edu/class/cs168/l/l1.pdf

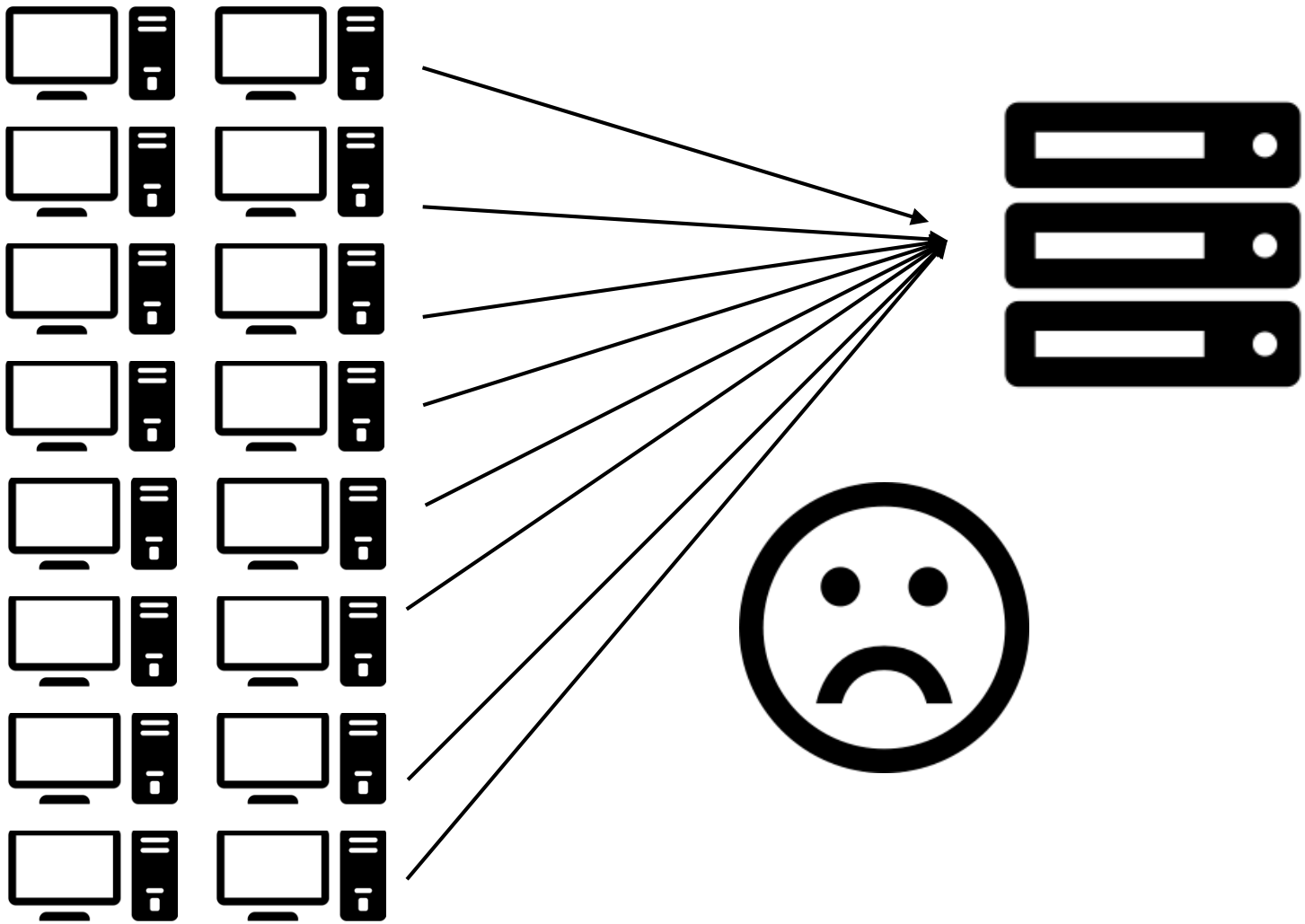
Initial Context: Web Caching

Suppose the new Star Wars trailer comes out …



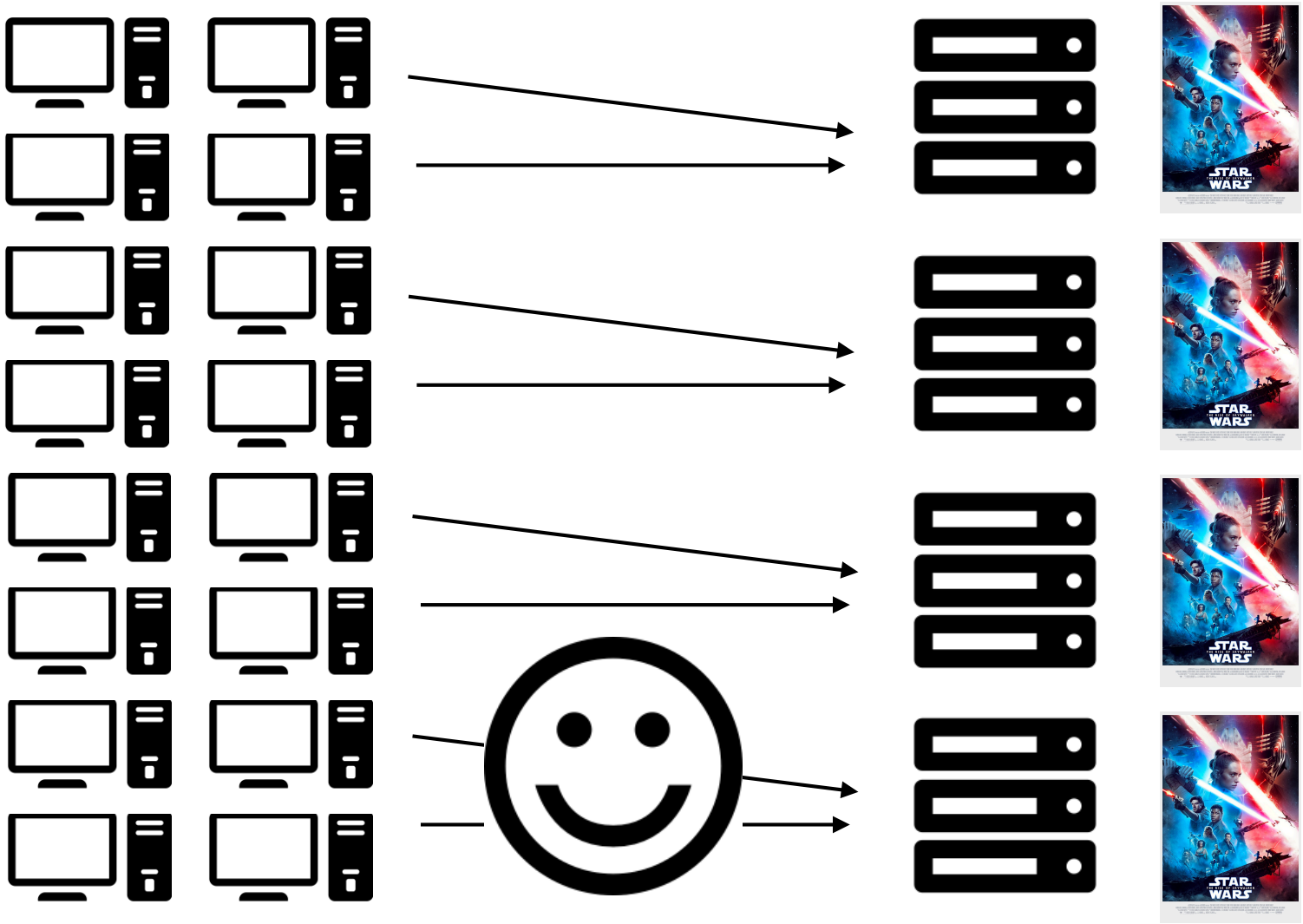
Initial Context: Web Caching

Suppose the new Star Wars trailer comes out …



Natural Idea: Cache Web Content

Suppose the new Star Wars trailer comes out …



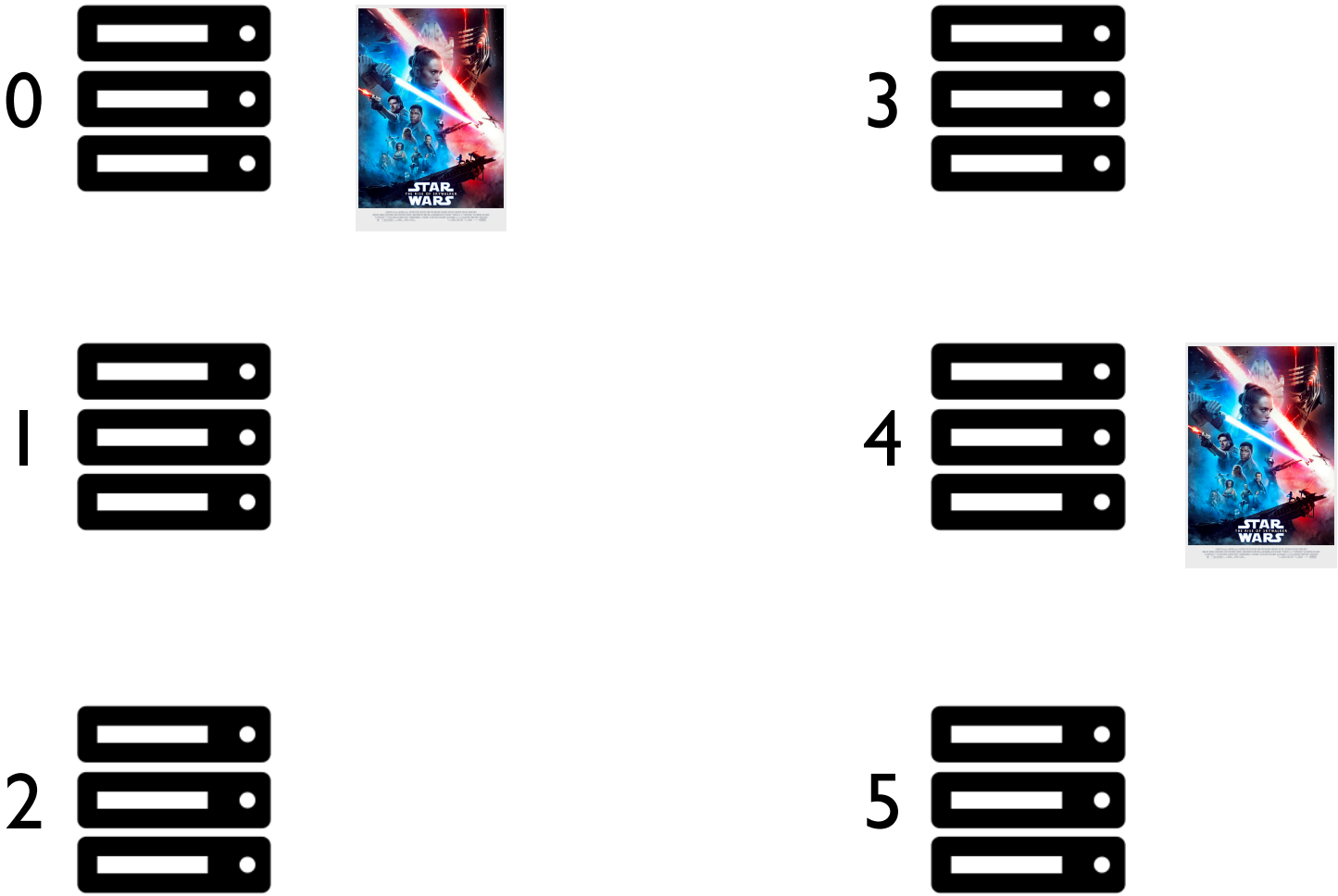
h(x) mod 6

hash(URL) mod n

Can use multiple hash functions to replicate content

hash(URL) mod n

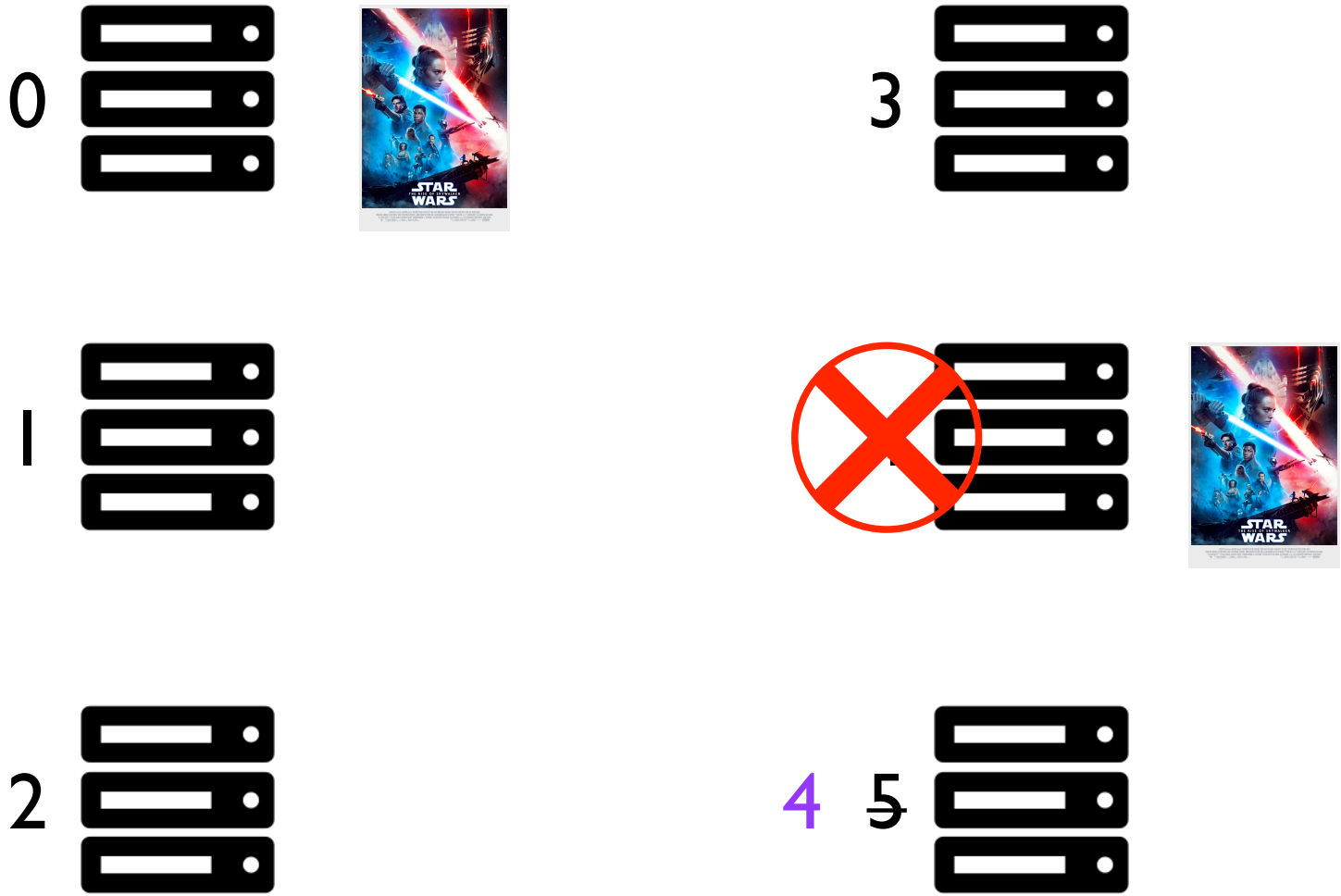
h(x) mod 6
g(x) mod 6



What if a cache dies?

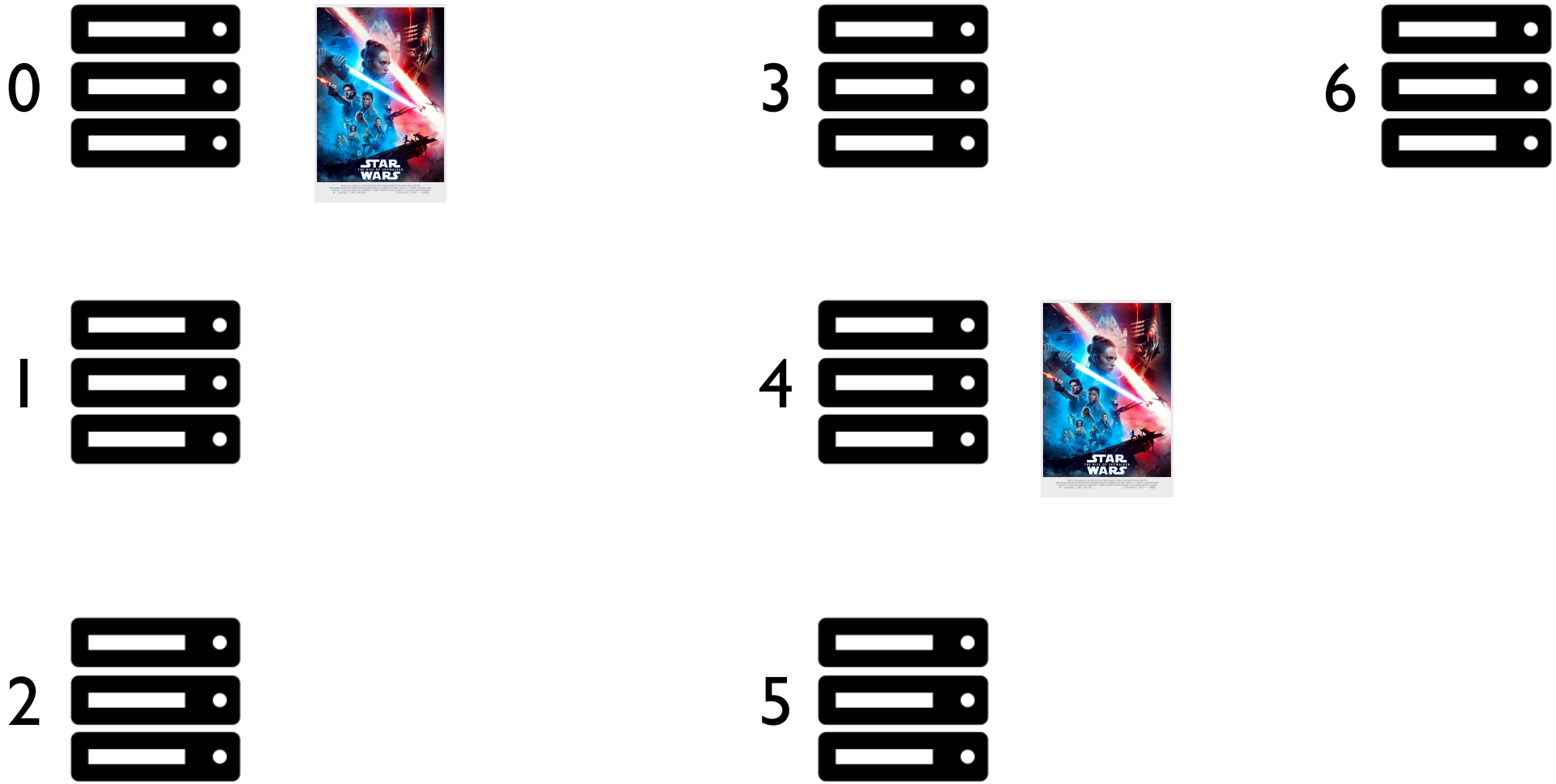
h(x) mod 5

g(x) mod 5



What if we add a cache?

h(x) mod7
g(x) mod 7



Big Issue

Any change to the number of caches (n), results in moving almost all content

Since we expect

$h(x) \bmod n \mathrel{!=} h(x) \bmod (n+1)$

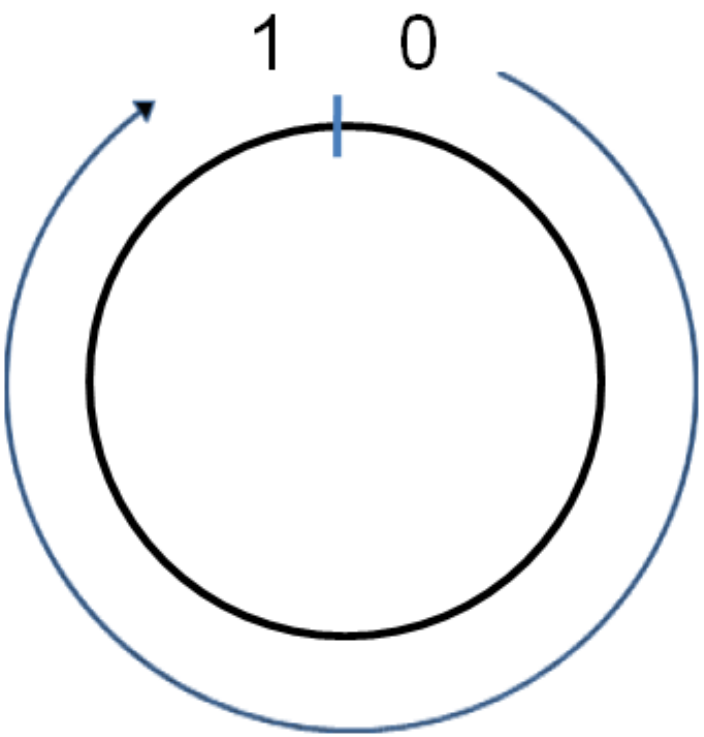
Consistent Hashing

Hash URLs (just like before), what we called x

But also hash the names of the cache servers too!

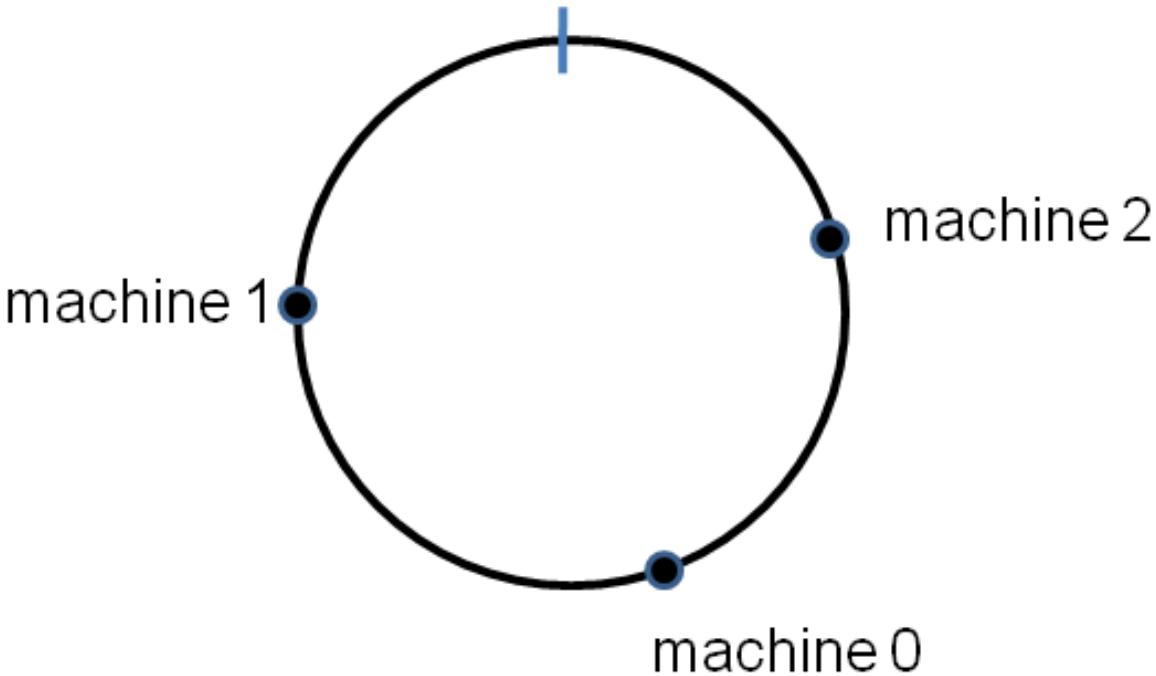
Assign x to “nearby” cache servers

Consistent Hashing

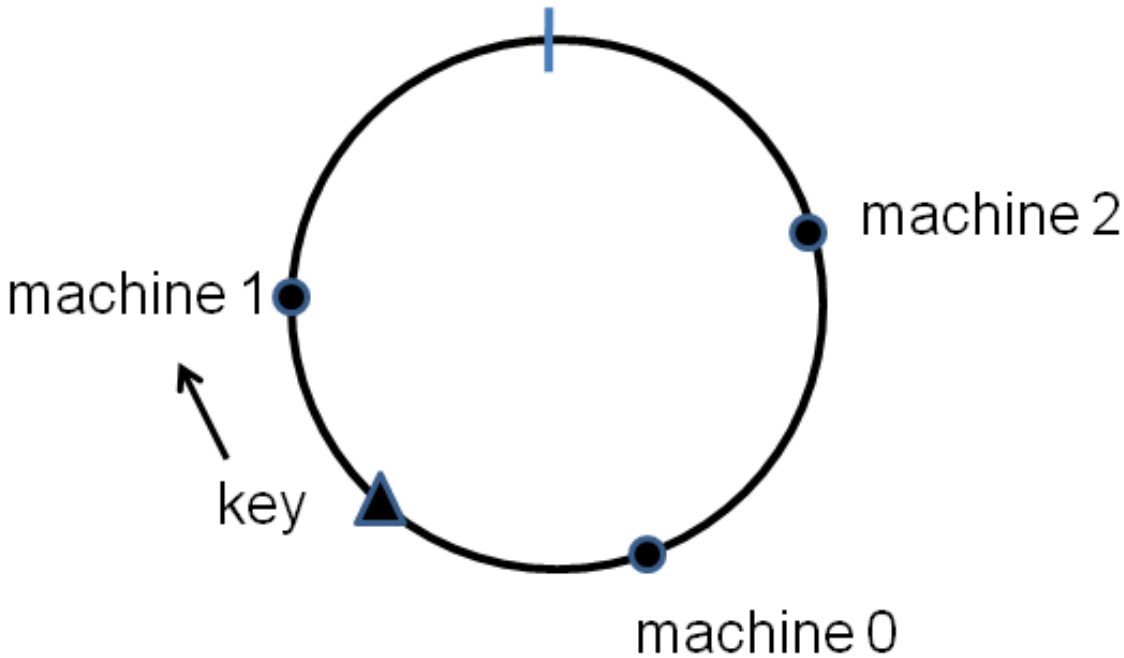


- Cache servers are numbered 0, …, n-1 (as before)
- Now we rescale the hash function to fit the unit circle!
- E.g., if hash function has range [0,R), then re-scale via hash(x) / R

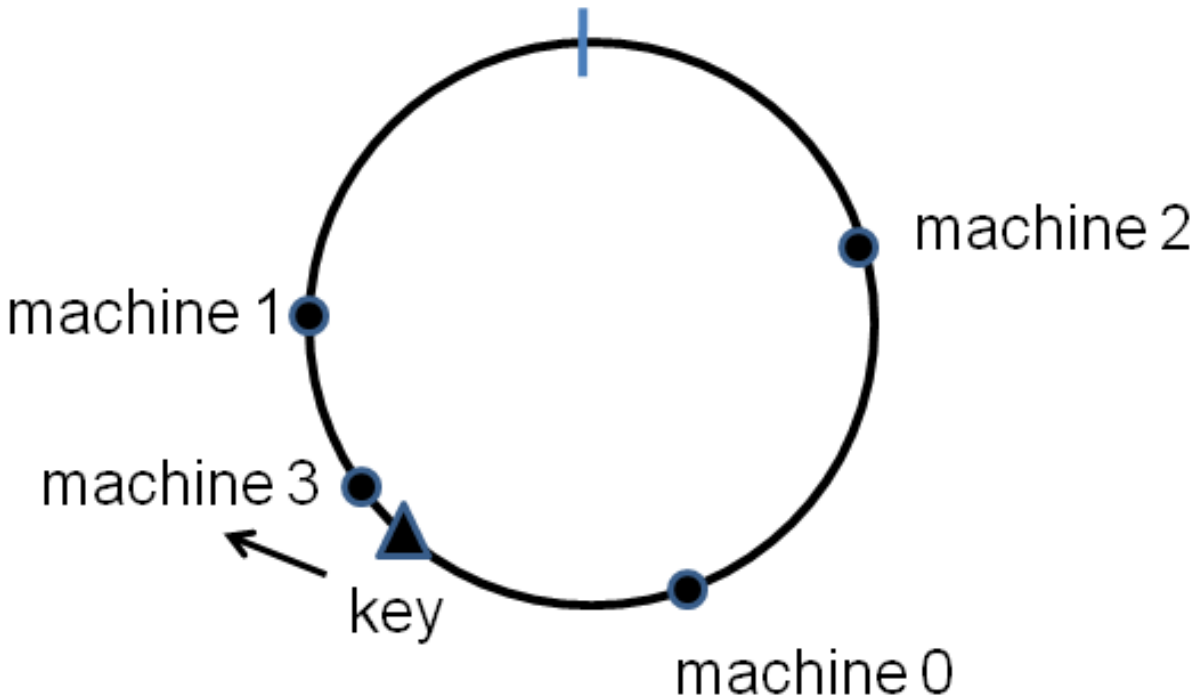
http://michaelnielsen.org/blog/consistent-hashing/



- Cache servers are hashed to the unit circle
- and so are keys!
  - e.g., "StarWars trailer"



- Keys are assigned to the cache server that comes first in a clockwise-fashion



- What if we add a new machine?
- Only a fraction of all key-value pairs have to move

Example

Distributed Hash Tables

Late 1990s / early 2000s

Rise (and fall) of peer-to-peer networks

- Napster, Gnutella, …
- “unstructured P2P networks”

Big interest in adopting ideas of consistent hashing into “structured P2P networks”

napster

Founded by Shawn Fanning in 1999 (later joined by Sean Parker)

http://en.wikipedia.org/wiki/Sean_Parker

later went on to be founding president of Facebook

History Lesson

1999: Started by 19-year-old college freshman (Northeastern University) Shawn Fanning

- Built only for sharing MP3 files Stirred up legal battles with $15B recording industry
- December: RIAA sues Napster, asking $100K per download

2000: 25% UWisconsin traffic is Napster, many universities ban it; 60M users

2001: 2.2M users/day, 28 TB data, 122 servers

- US Federal Appeals Court: users violating copyright laws, Napster is abetting this
- Napster decides to run paid service, pay % to songwriters and music companies

Napster v2.0 BETA 10.3

File View Actions Help

Shop for music at CDNOW

Home | Chat | My Files | Search | Hot List | Transfer | Discover | Help

Online
napster060620...
SpanishHay

Offline
jonknibbs
r_kaura

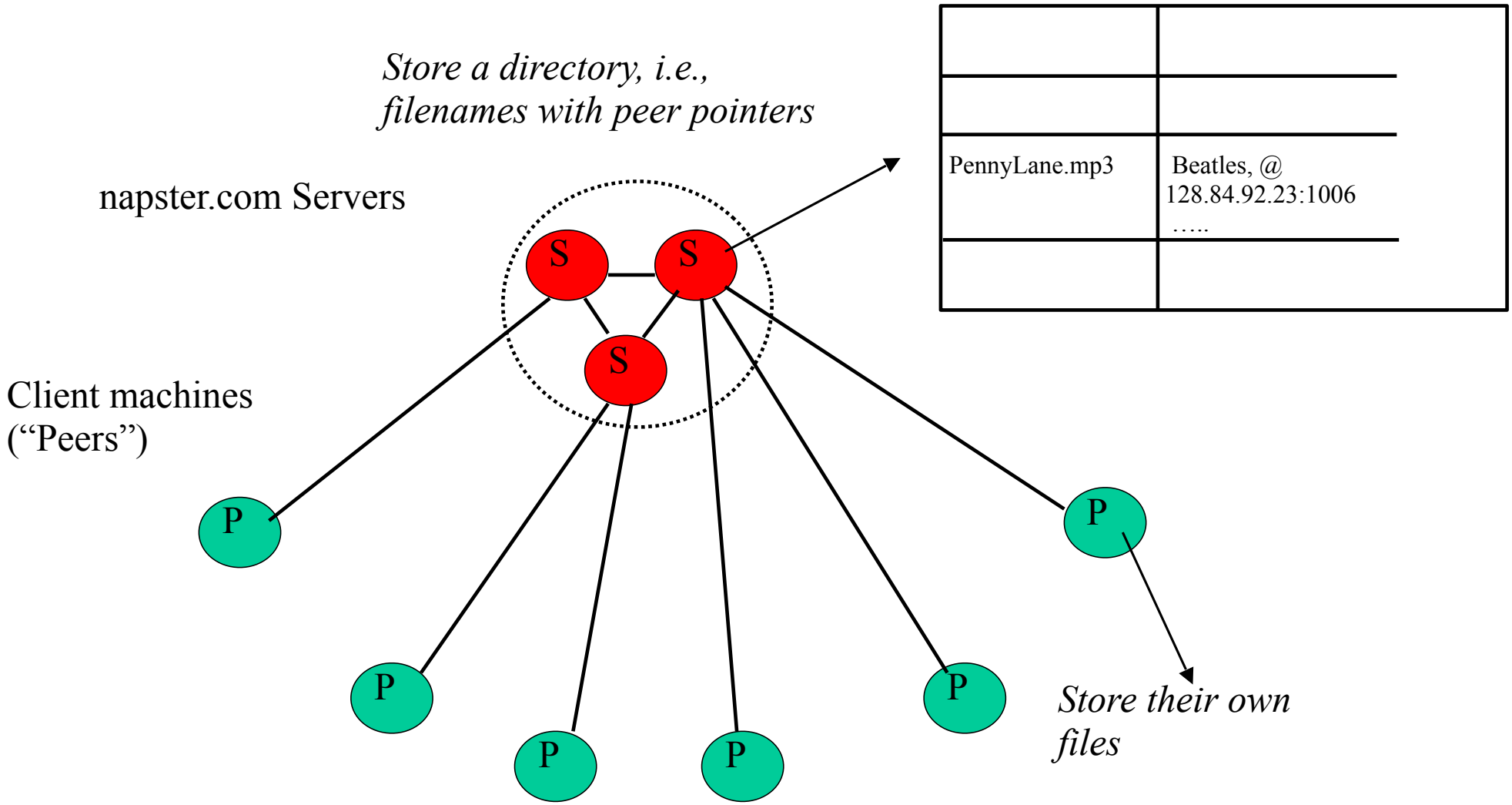
Filename	Filesize	Bitrate	Frequency	Length
Zion I - critical.mp3	3,798,959	128	44100	3:51
Yuji Oniki - paper tigers.mp3	3,505,970	128	44100	3:33
Young Scott Perrie - everytime.mp3	4,957,964	320	44100	2:01
Wonderlick - right crazy rough mix.mp3	4,572,891	128	44100	4:39
Willard Grant Conspiracy - christmas in nevada.mp3	3,579,530	128	44100	3:38
Unknown - coconuts.mp3	4,856,236	128	44100	4:56
Wellwater Conspiracy - of dreams.mp3	3,466,264	128	44100	3:31
Waco Brothers - fox river.mp3	3,903,738	128	44100	3:58
Vivendo de Pao - riso de gilberto.mp3	3,377,656	128	44100	3:26
Vivendo de Pao - jacare.mp3	3,166,169	128	44100	3:13
Virgil Shaw - water cooler.mp3	4,988,928	128	44100	5:04
Virgil Shaw - volvo ff.mp3	3,220,086	128	44100	3:16
Unknown - next degree.mp3	4,062,125	128	44100	4:07
Unknown - are we people.mp3	2,932,736	96	44100	3:58
Trail of Dead - mistakes and regrets merge.mp3	3,631,104	128	44100	3:41
Tom Heyman - bottle full of wishes.mp3	4,008,229	128	44100	4:04
Tom Armstrong - thinking of him.mp3	2,671,723	128	44100	2:43
Tom Armstrong - looking forward to looking back.mp3	3,475,877	128	44100	3:32
tobyslater - the next life.mp3	3,089,870	128	44100	3:08
Unknown - sky is not crying.mp3	6,891,439	128	44100	7:00
The Rosenbergs - sucking on a plum.mp3	3,120,193	128	44100	3:10
The Rosenbergs - soaked in polyester.mp3	4,795,374	128	44100	4:52
The Rosenbergs - secret.mp3	4,147,955	128	44100	4:13
The Rosenbergs - overboard.mp3	6,839,612	128	44100	6:57
The Rosenbergs - little lie.mp3	2,911,214	128	44100	2:57

User: napster06062001 Total Files: 262 File Data: 1,014 megs Ping: N/A

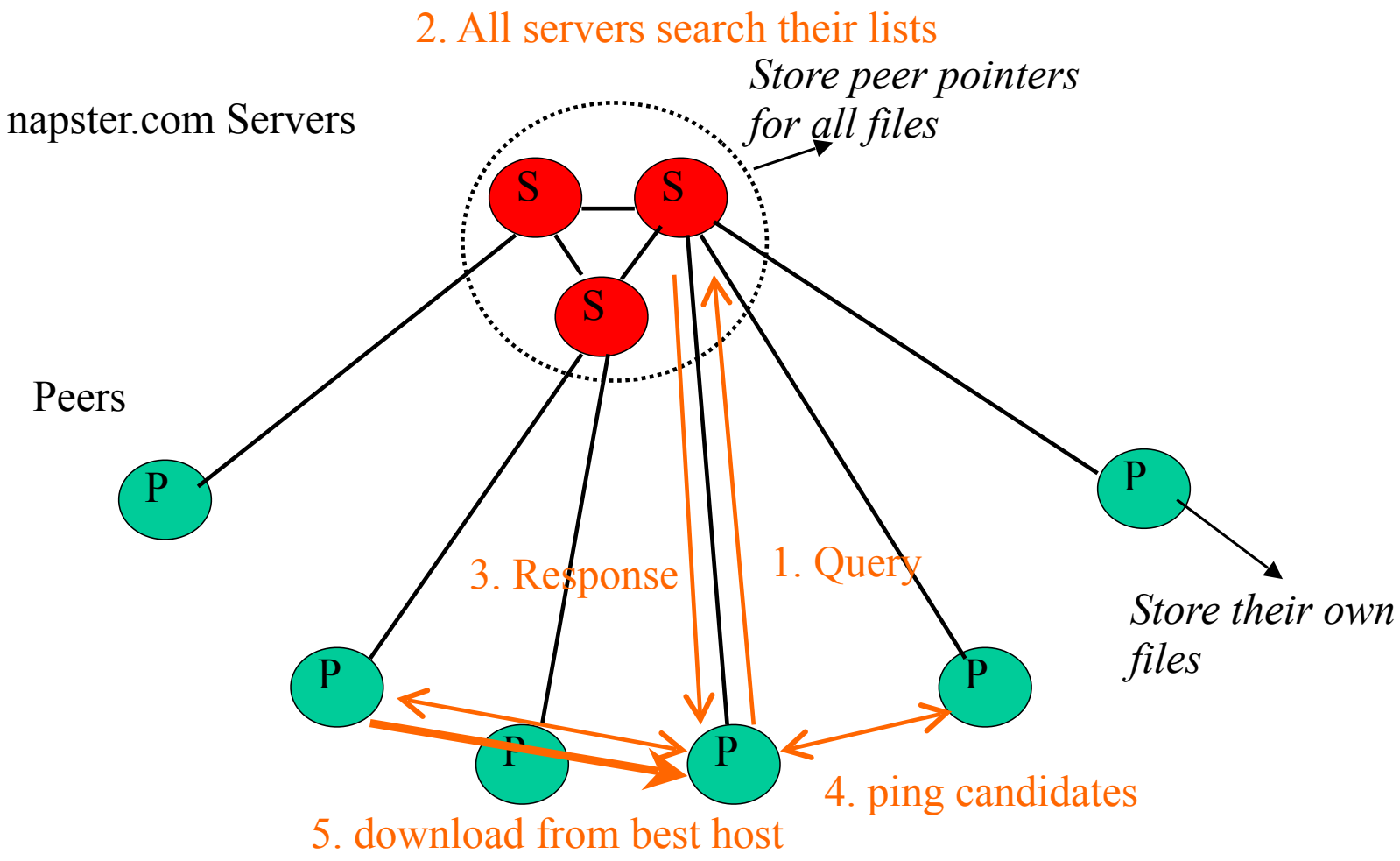
Get Selected Files | Add User to Hot List

Online (rishdeep): Sharing 0 files.

Currently 89,910 users sharing 138,637 files (549 gigs)


Store a directory, i.e.,
filenames with peer pointers
napster.com Servers
S
S
S
PennyLane.mp3
Beatles, @
128.84.92.23:1006
…..
Client machines
("Peers")
P
P
P
P
P
P
Store their own
files

Napster search



Problems?

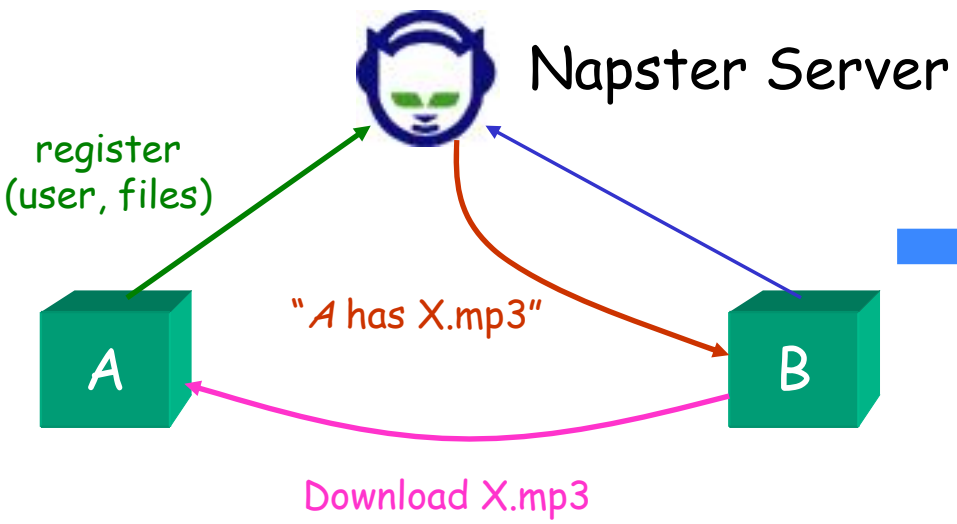
- Centralized server
  - a source of congestion
  - single point of failure
- No security
  - plaintext messages and passwords
- Courts declared napster.com responsible for users' copyright violation
  - "Indirect infringement"

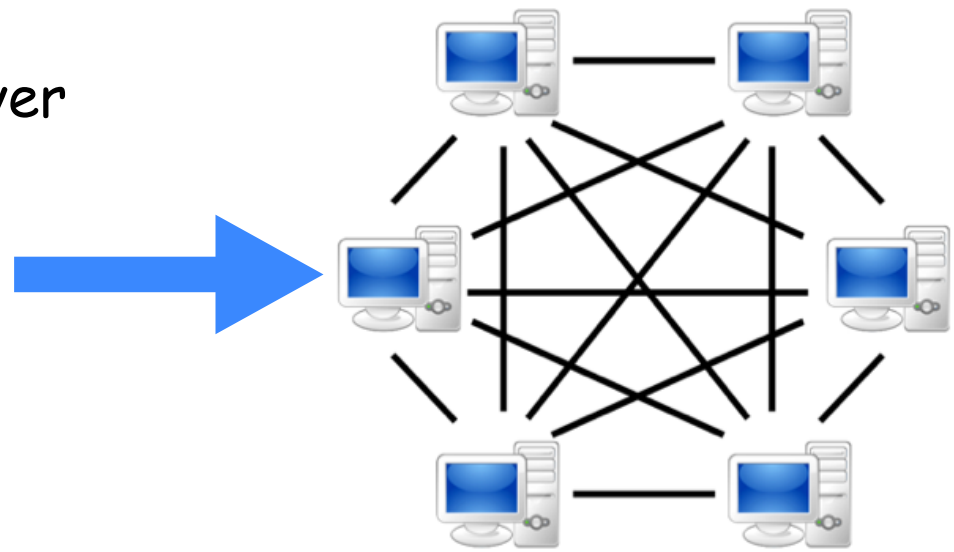
Can we do better than Napster?

Decentralize the network!

Truly create "the cloud" where the data would live

Even though Napster was going strong until Summer of 2001, the foundation was already being laid for the next generation of sharing technology

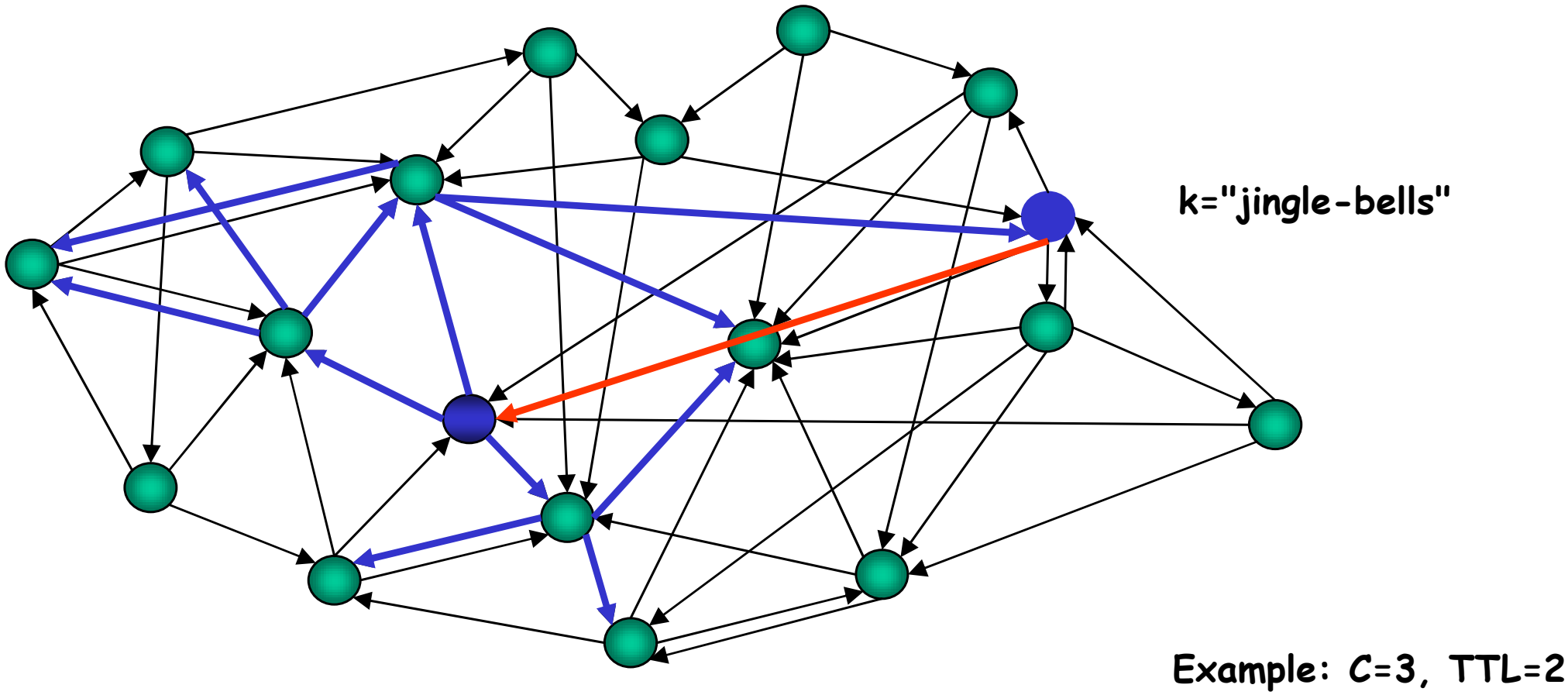




“Unstructured P2P” — Gnutella: Search by Message Flooding

- No index information is used
  - i.e. the information (key, peer) is only available directly from p
- Simplest approach: Message Flooding (Gossiping)
  - Send query message to C neighbors
  - Messages have limited time-to-live TTL
  - Messages have IDs to eliminate cycles

Gnutella: Search by Message Flooding



A REPORTER AT LARGE NOVEMBER 4, 2019 ISSUE

A CYBERSECURITY FIRM'S SHARP RISE AND STUNNING COLLAPSE

Tiversa dominated an emerging online market —before it was accused of fraud, extortion, and manipulating the federal government.

By Raffi Khatchadourian October 28, 2019

https://www.newyorker.com/magazine/2019/11/04/a-cybersecurity-firms-sharp-rise-and-stunning-collapse

So far …

Unstructured P2P networks

- simple
- robust! (almost impossible to kill)

Performance

- update and maintenance cost low

Drawbacks

- tremendous bandwidth consumption for search
- free riding

Can we do better?

Rise of "Structured P2P" Approaches

Different strategies

- P-Grid: distributing a binary search tree
- **Chord: constructing a distributed hash table**
- CAN: Routing in a d-dimensional space
- FreeNet: caching index information along search paths

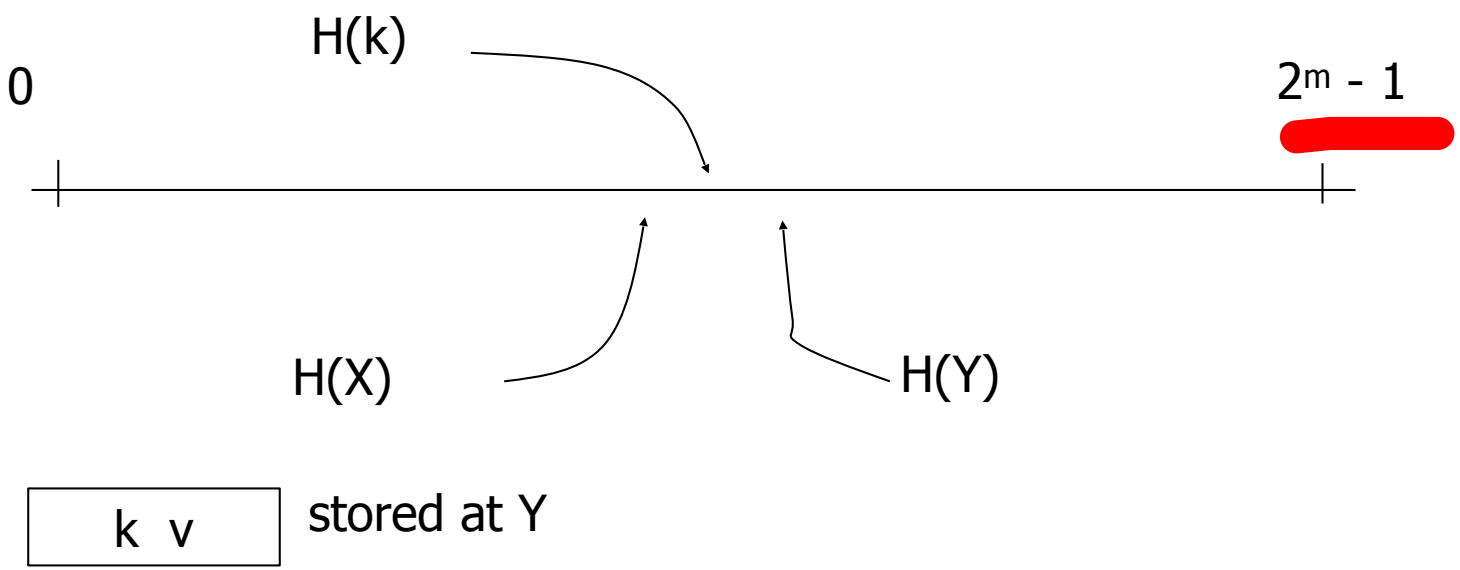
Commonalities

- each peer maintains a small part of the index information (routing table)
- searches performed by directed message forwarding

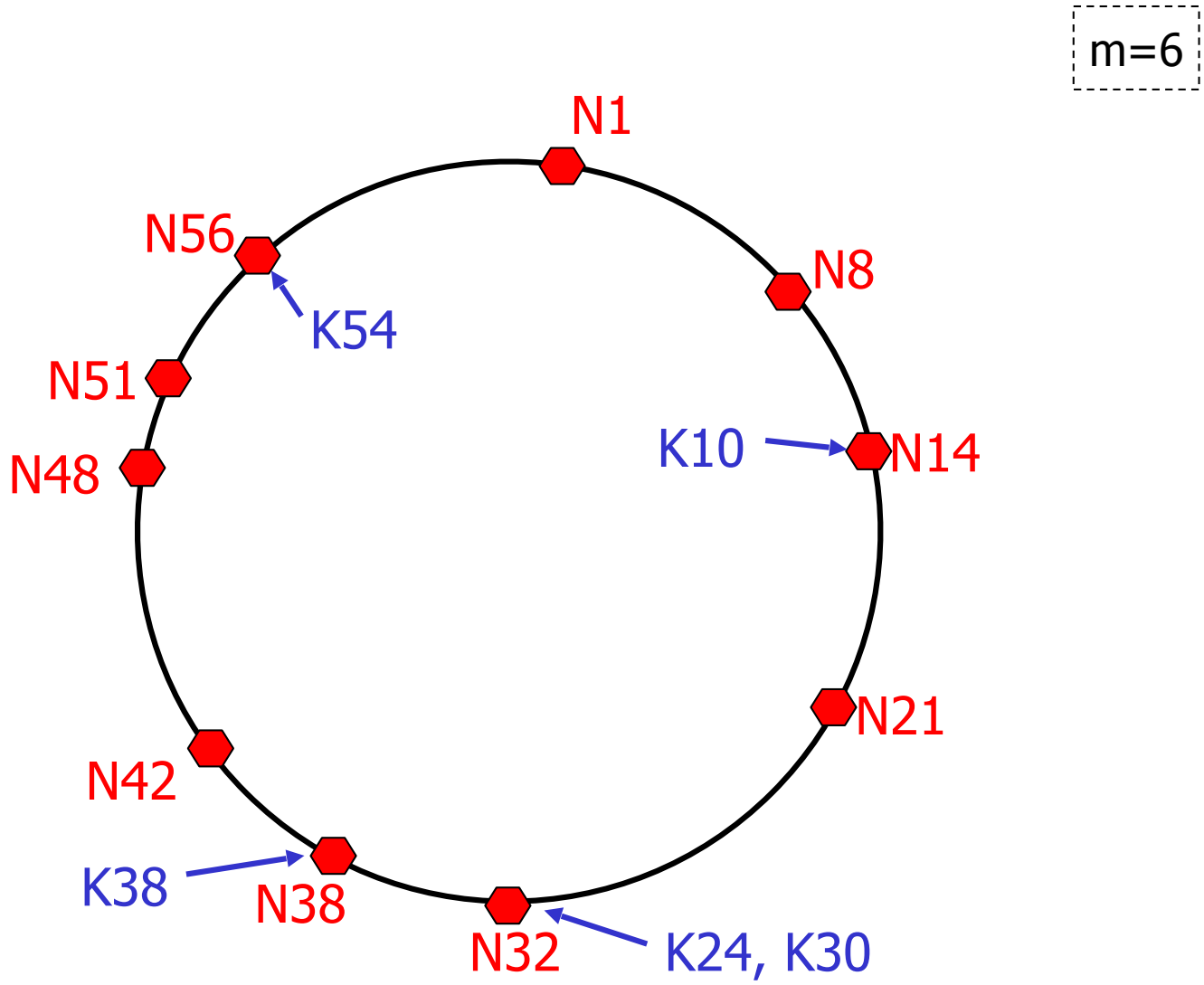
CHORD: Using Consistent Hashing for Structured P2P

Hashing Values and Nodes

- H(v) is m-bit number (v is value)
- H(X) is m-bit number (X is node id)
- Hash function is "good"

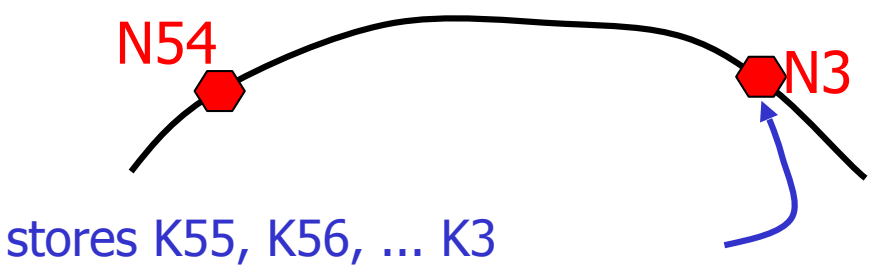


Chord Circle



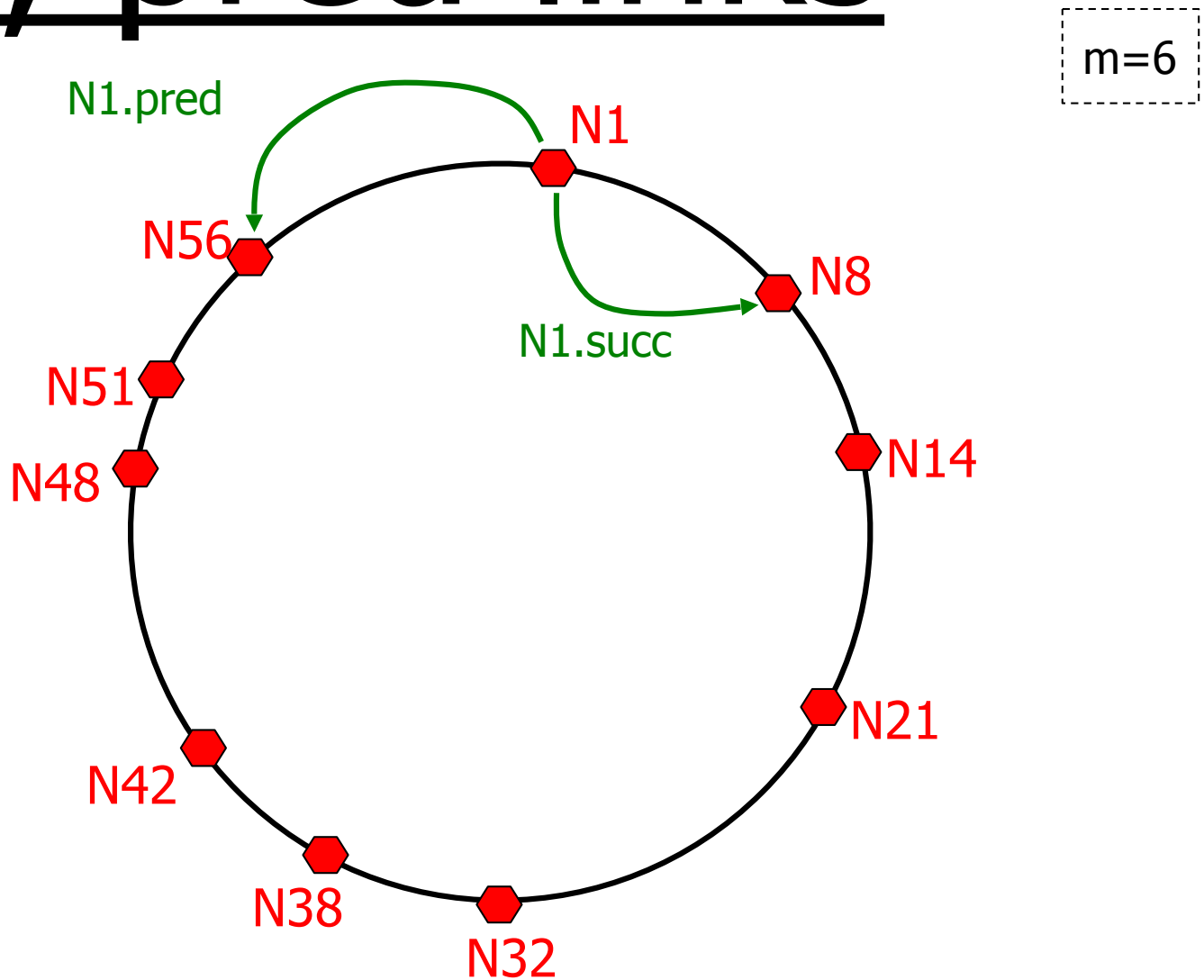
Rule

- Consider nodes X, Y such that Y follows X clockwise
- Node Y is responsible for keys k such that H(k) in (H(X), H(Y)]

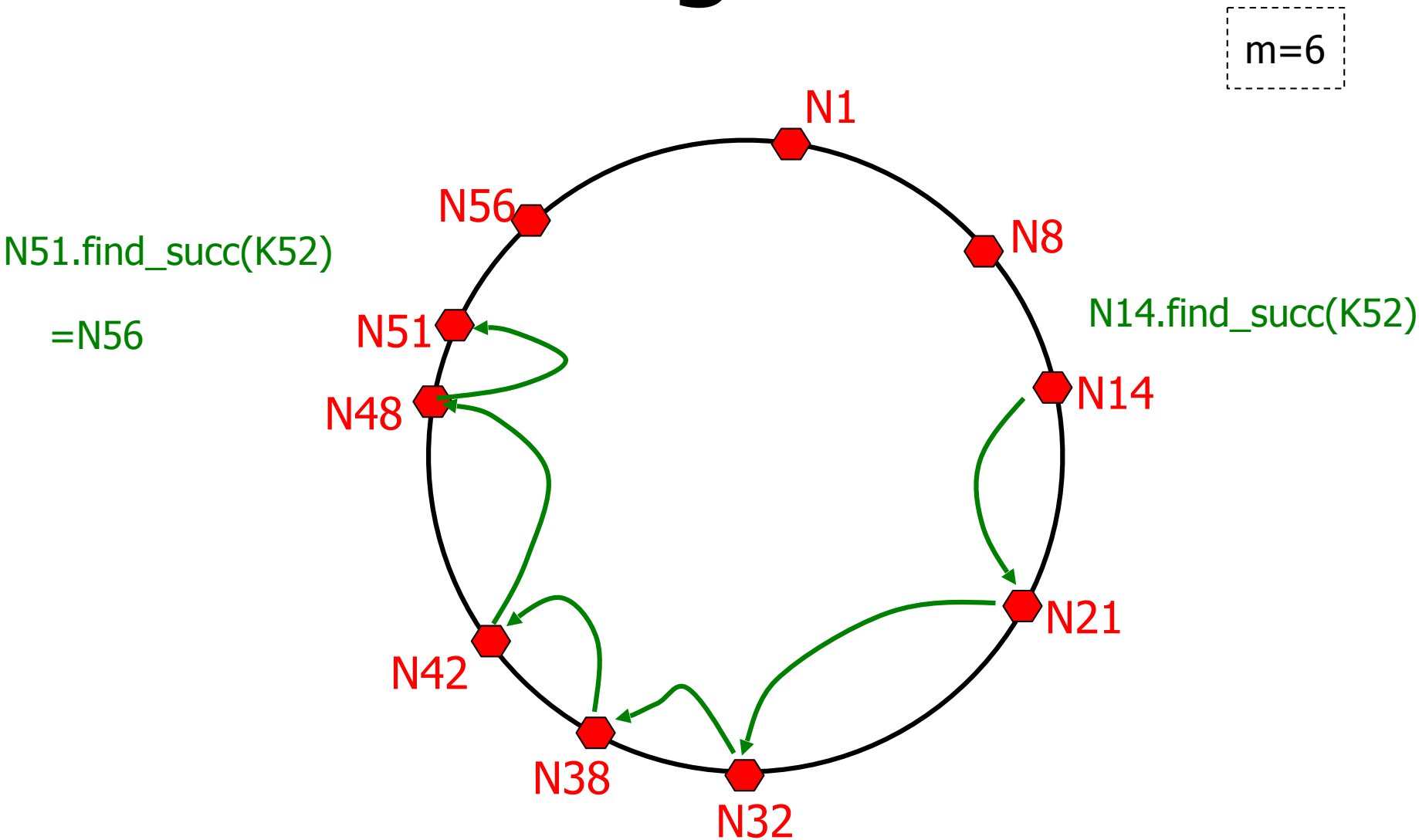


use hashed values...
e.g., N54 is node whose id hashes to 54.

Succ, pred links



Search using succ links



Finger Table

m=6

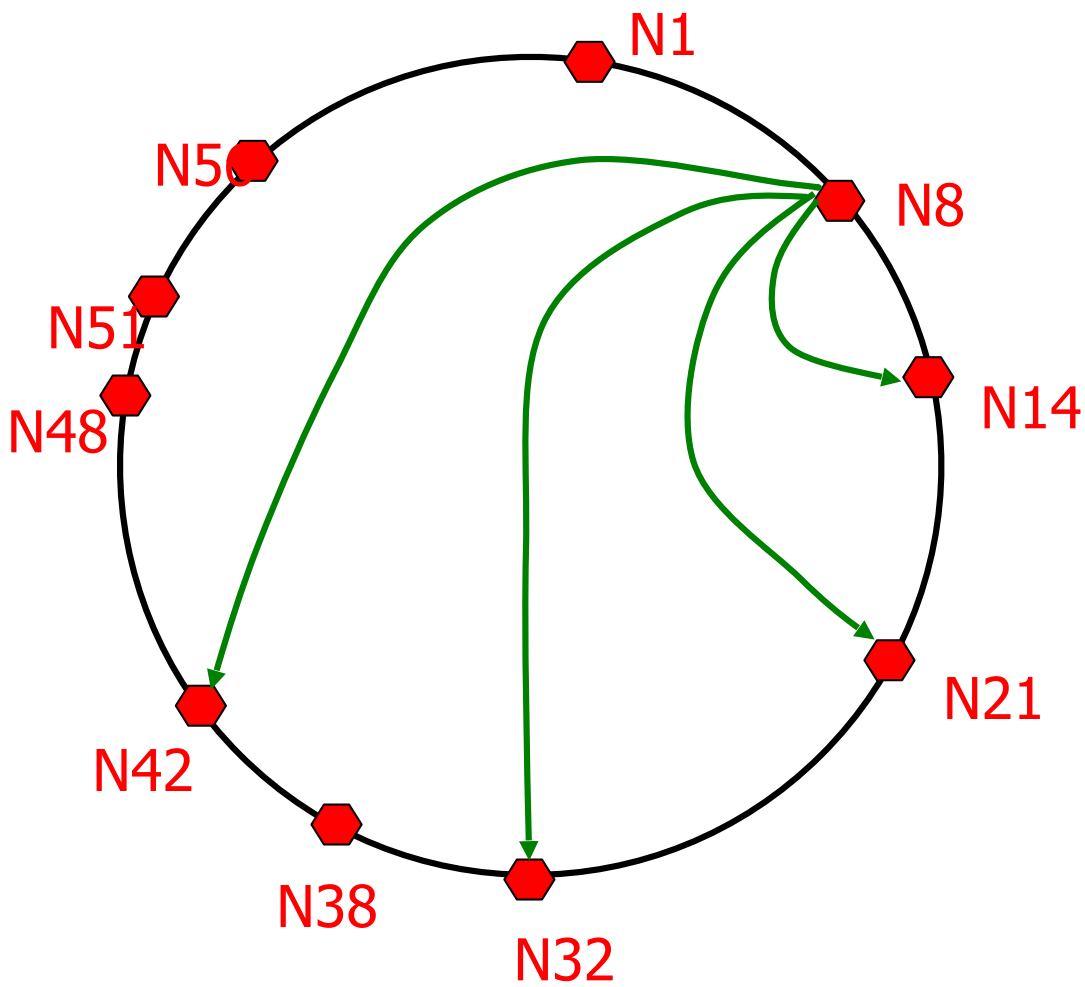
finger table for N8

N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

- The i-th entry of node n contains the address of successor:
  - $(n + 2^{i-1}) \bmod 2^m$

Finger Table

m=6

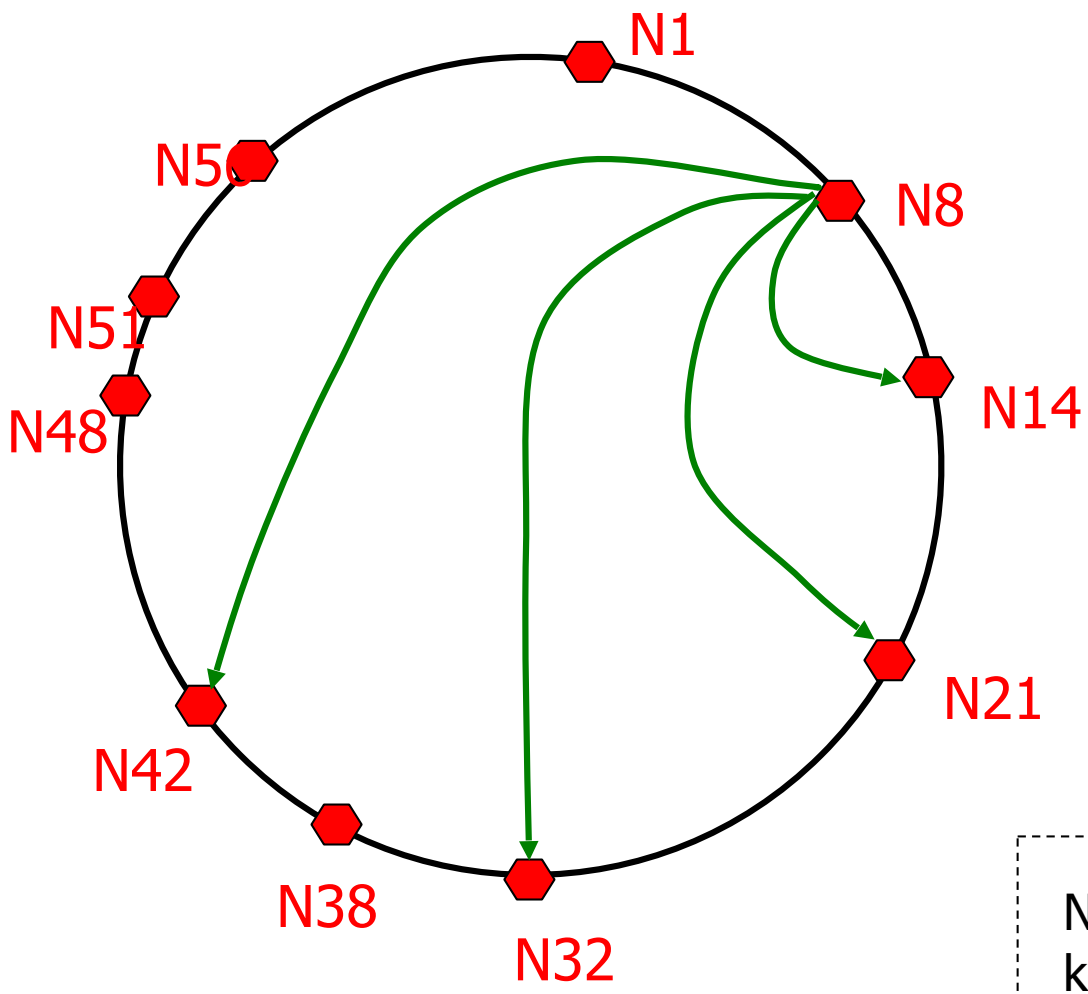


finger table for N8

N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

Finger Table

m=6

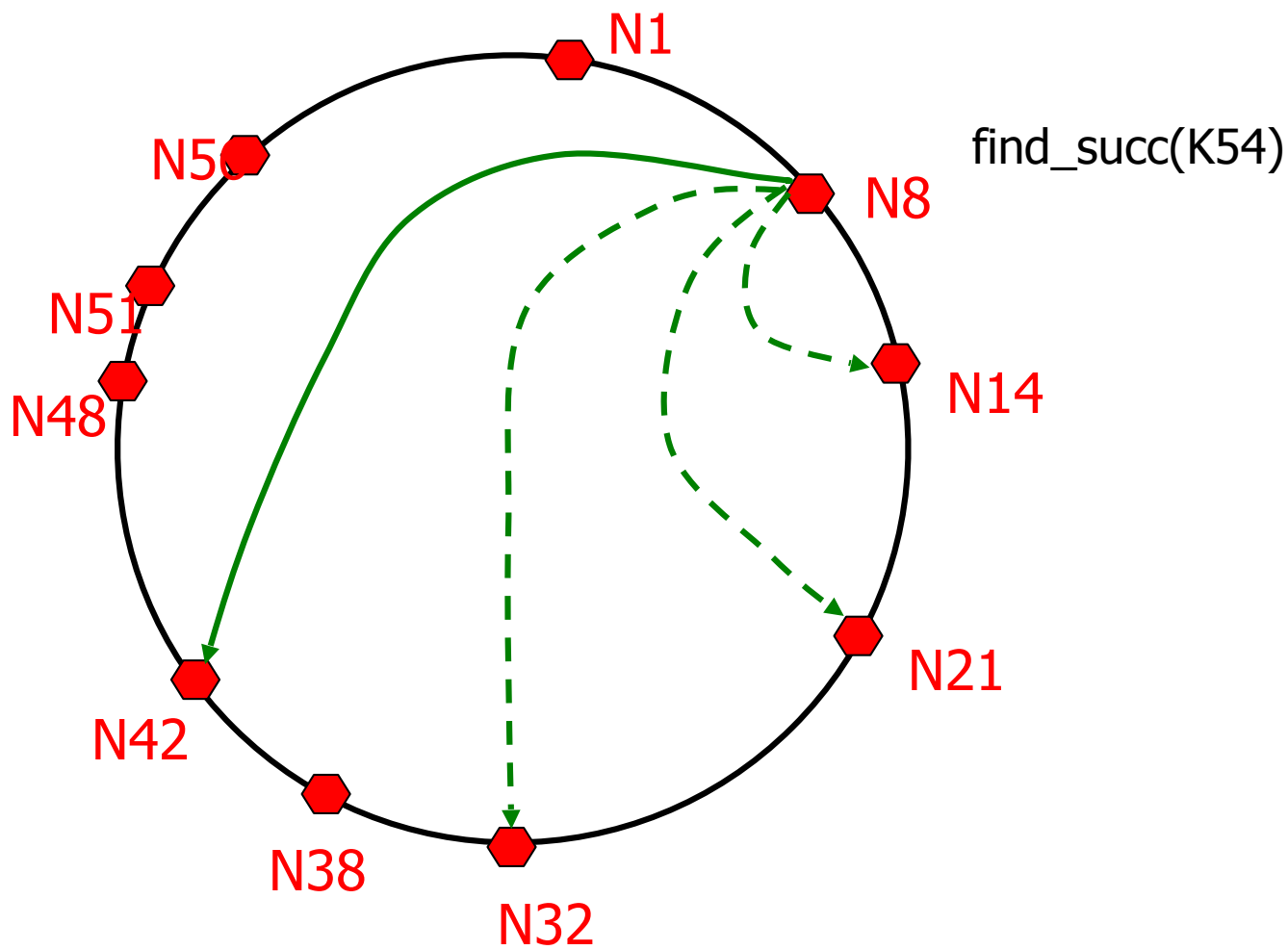


finger table for N8

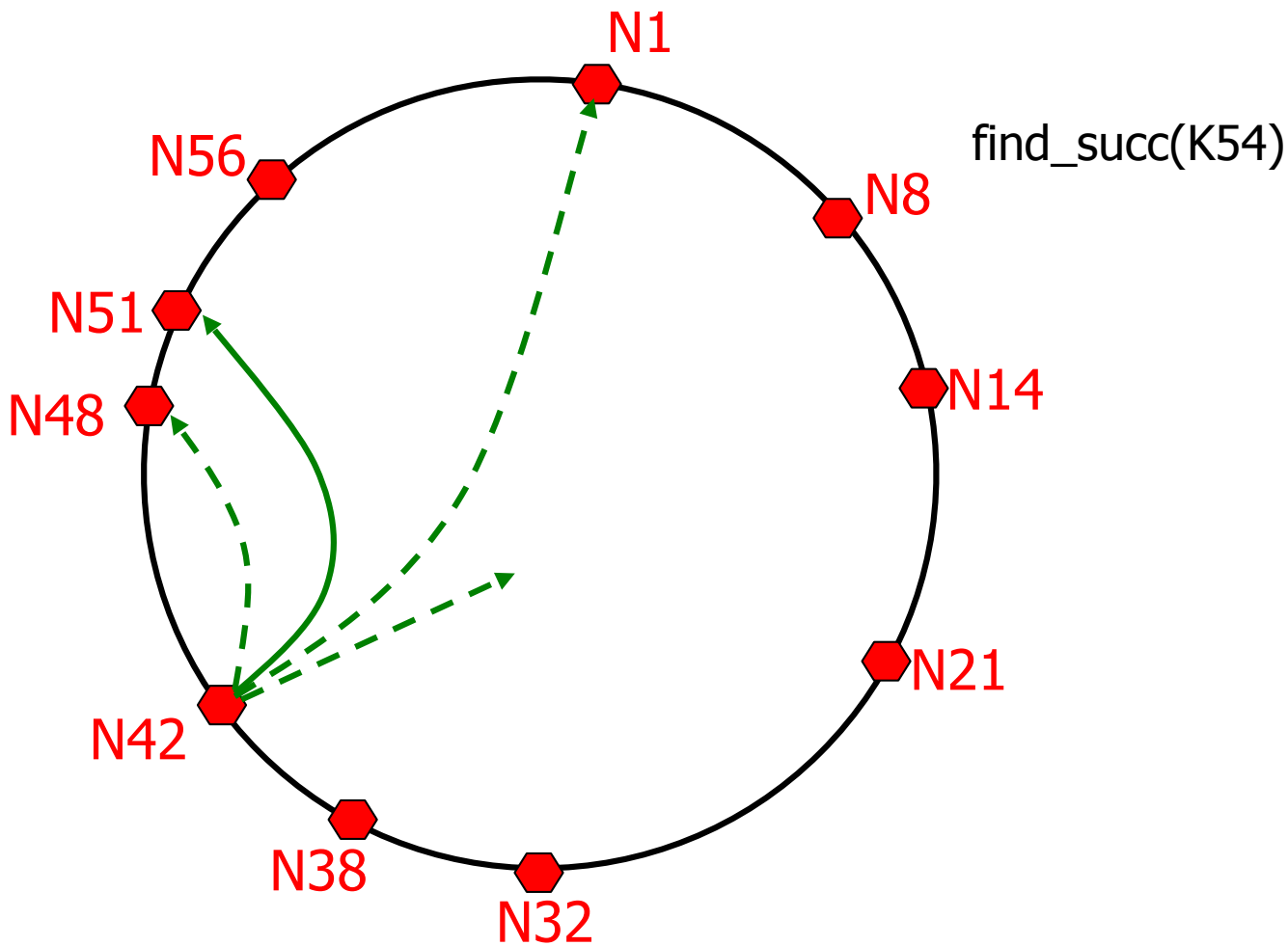
N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

Node responsible for key 8+32 = 40

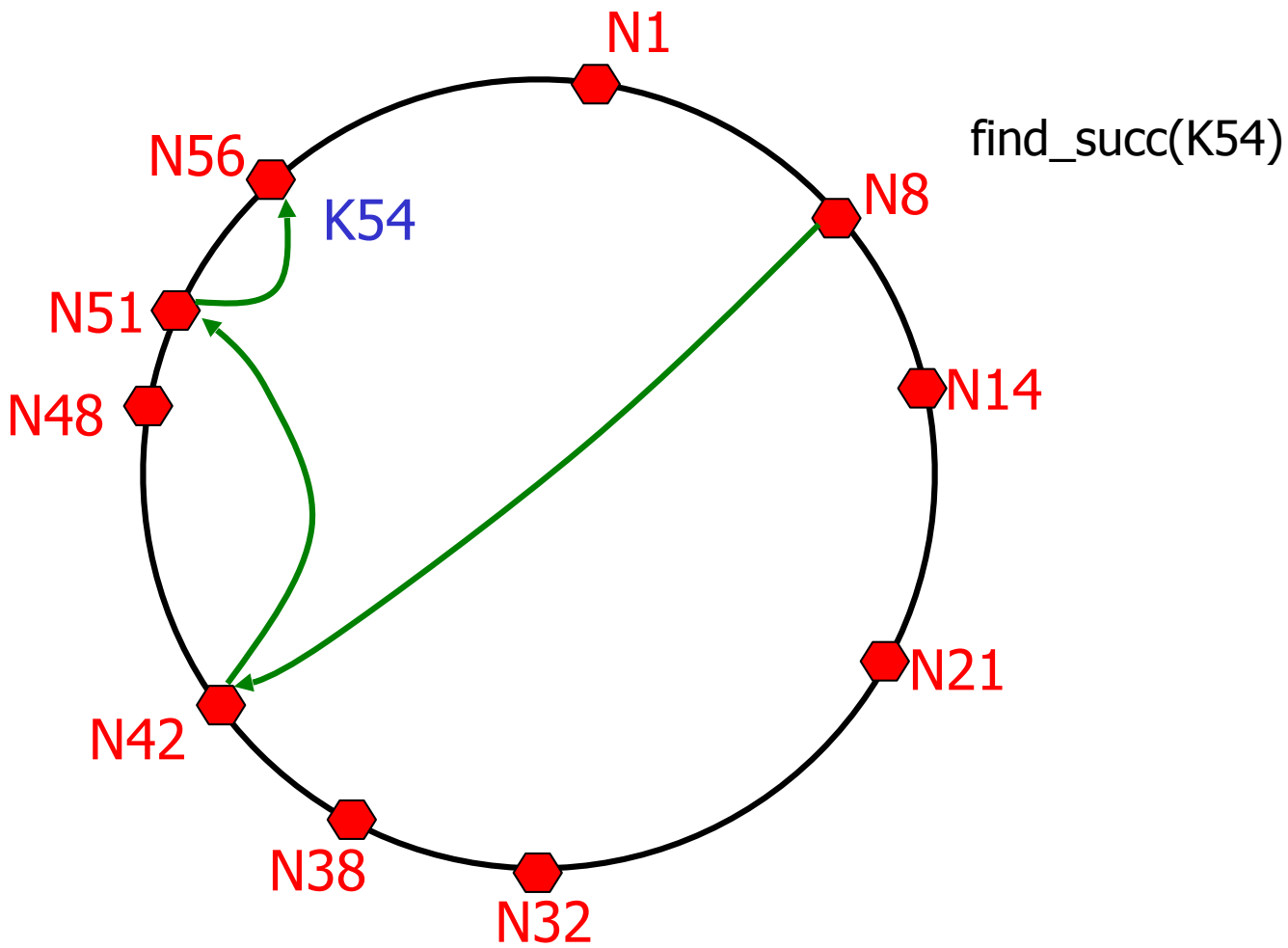
Example



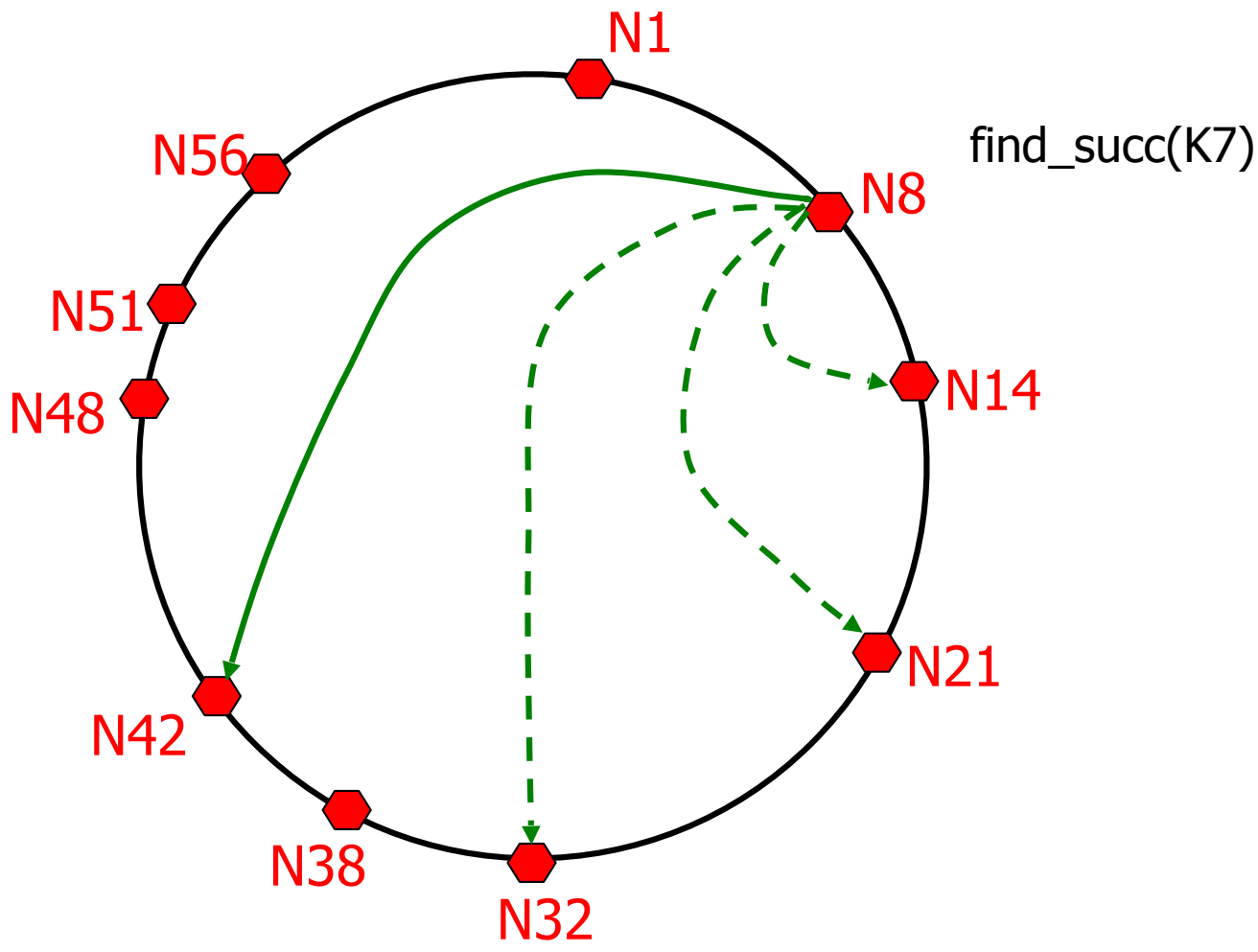
Example



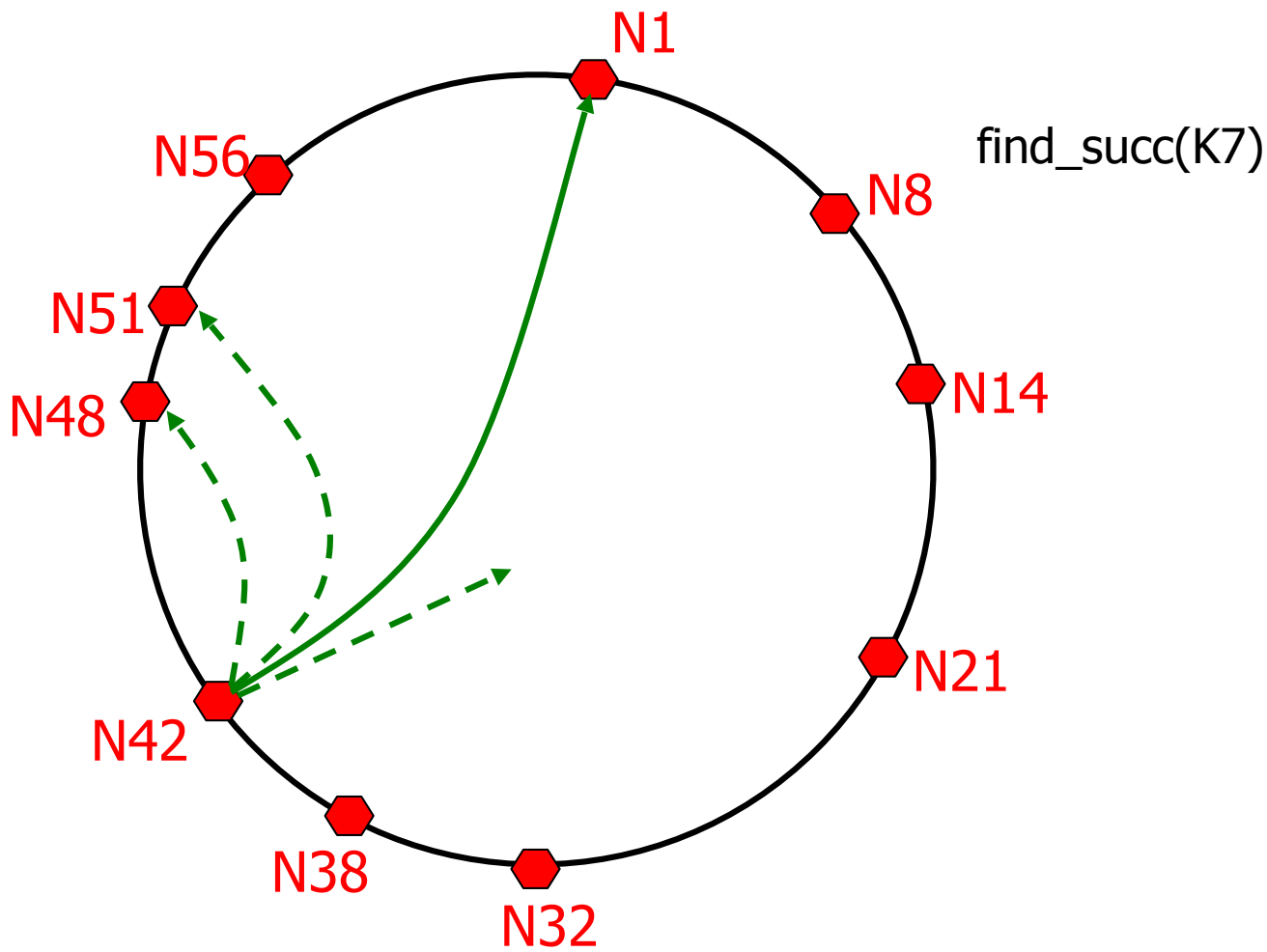
Example



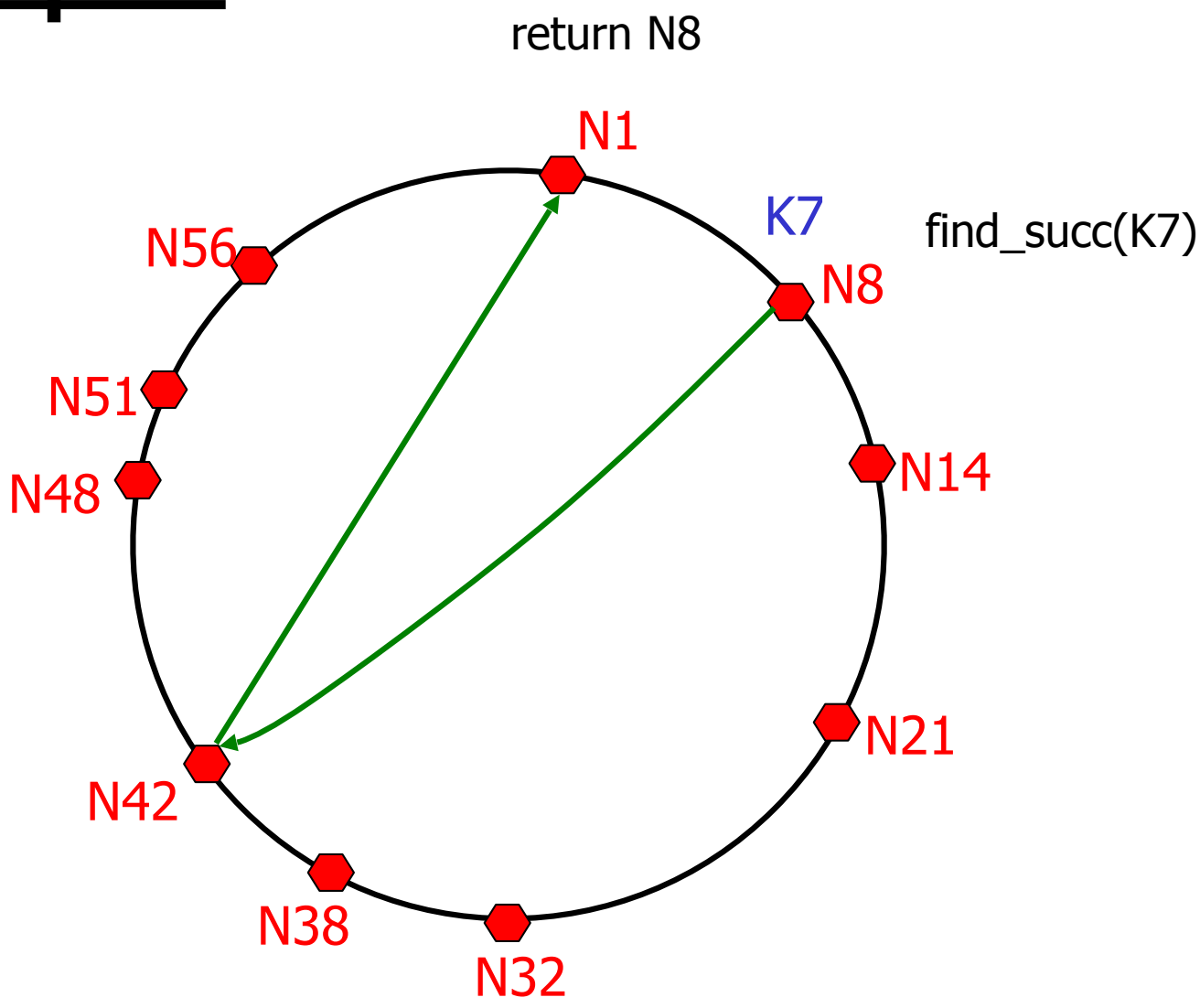
Another Example



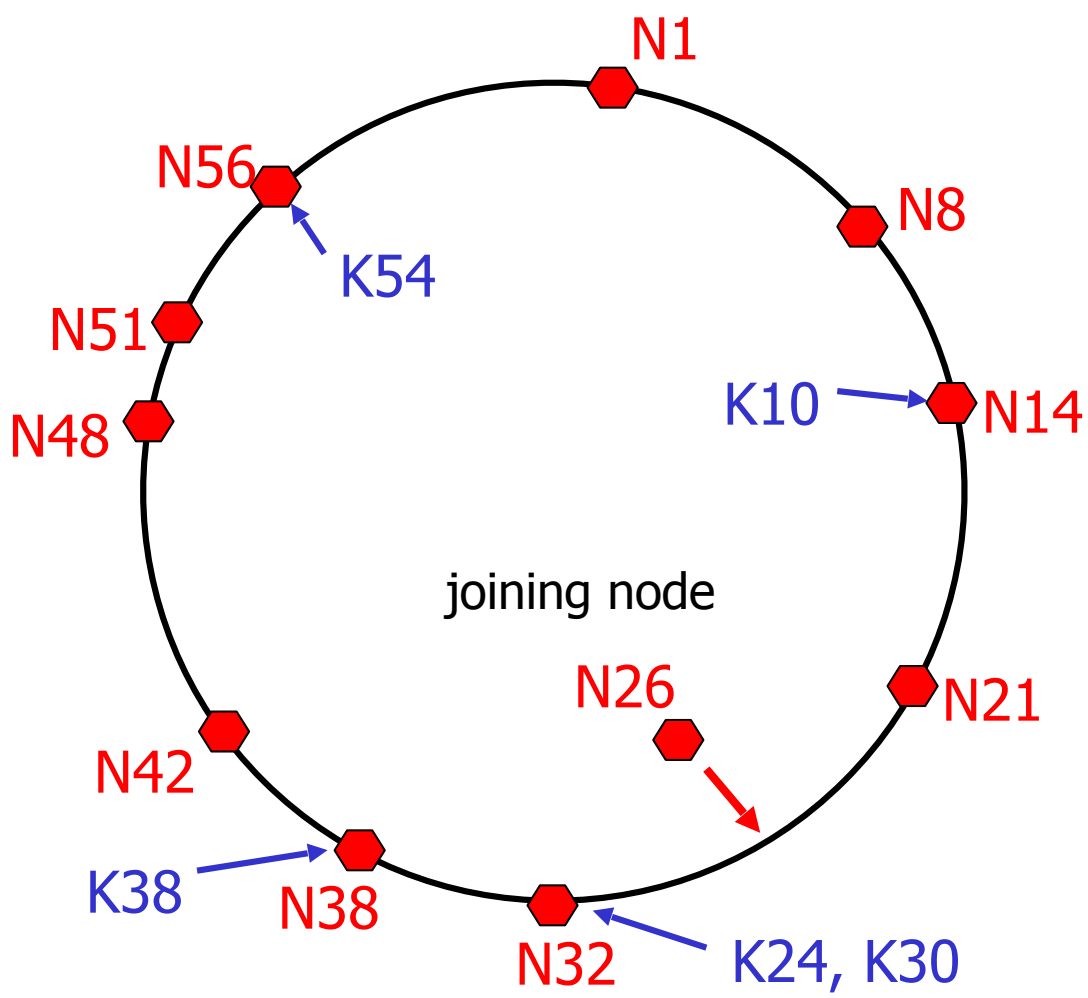
Example



Example



Adding Nodes to Circle



need to
1. update links
2. move data

for now, assume nodes never die

Chord Discussion

Performance

- Search is O(log n)
- Resilience to failures: replication to successor nodes

Qualitative Criteria

- Search predicates: equality of keys only
- Global knowledge: key hashing, network origin
- Peer autonomy: nodes have by virtue of their address a specific role in the network

In terms of practical use, CHORD and similar structured P2P networks did not catch on for consumer-oriented file sharing, but the ideas are embedded in virtually all cloud and distributed applications today