Peer-to-Peer and GRID Computing
ID2210

Seif Haridi (haridi@kth.se)
Fatemeh Rahimian (fatemeh@sics.se)
Amir H. Payberah (amir@sics.se)

Course Objective

Objectives

- Introduction to basic concepts and principles of large-scale dynamic decentralized distributed systems and distributed algorithms.
- Study of peer-to-peer overlays, DHTs, gossip based algorithms and content distribution networks.
- Implementation and evaluation of some of the peer-to-peer algorithms in a simulator environment.
- How to read, review and present a scientific paper.

Topics of Study

- Fundamental results in large-scale distributed algorithms.
- Overview of peer-to-peer systems, algorithms, and applications.
- Study of Distributed Hash Tables (DHTs), also called Structured Overlay Networks (SONs).
- Gossip and Epidemic Overlays.
- Content and Streaming Distribution Networks.

Non Objectives

- Learning about GRID concepts and applications.
- Learning how to program centralized services.
  - Web services technology

Material

- Mainly based on research papers.
- You will find all the material on the course webpage:
  http://www.ict.kth.se/courses/ID2210/

Examination

- Students will work in groups of 1 or 2.
- The course has four types of requirements:
- Reading assignment: 30 points
- Lab assignment: 40 points
- Group presentation: 15 points
- Quiz: 15 points

Reading Assignment

- Read, summarize and review three papers.
- For each paper studied, write a summary report
  - Identify and motivate the problem
  - Pinpoint the main contributions
  - Explain the solution(s)
  - Explain how it is evaluated
  - Identify positive and negative aspects of the solution/paper
  - Answer to a few given questions
- Your summary report is reviewed and graded by two other groups
  - The reviews affect the grade of the reviewer group only.

Lab Assignment

- You implement two peer-to-peer systems in Kompics.
- You evaluate in simulation, the performance or properties of the implemented systems.
  - For the evaluation part you will be given the source code.
- Report you results in a document.

Group Presentation and Quiz

- You give a 15 minutes talk on a scientific paper.
- The list of papers will be available in the course webpage.
- You are free to choose any other paper, but it should be confirmed by teacher assistants.
- In the quiz we will ask questions based on the lectures notes and their corresponding papers.

Final Grade

- Final grade is determined by the sum of the assignment grades.
  - A: 90 – 100
  - B: 80 – 89
  - C: 70 – 79
  - D: 60 – 69
  - E: 50 – 59
  - F: < 50

Discussion Forum

- Use the course discussion forum if you have any questions
  - http://www.ict.kth.se/courses/ID2210/
  - Please use your Firstname-Lastname for your login.

Teachers

- Course responsible
  - Seif Haridi (haridi@kth.se)
- Teaching assistant
  - Amir H. Payberah (amir@sics.se)
  - Fatemeh Rahimian (fatemeh@sics.se)
- Guest Lecturer
  - Sarunas Girdzijauskas (sarunas@sics.se)
  - Jim Dowling (jdowling@sics.se)

Course Overview

BitTorrent

napster

Spotify

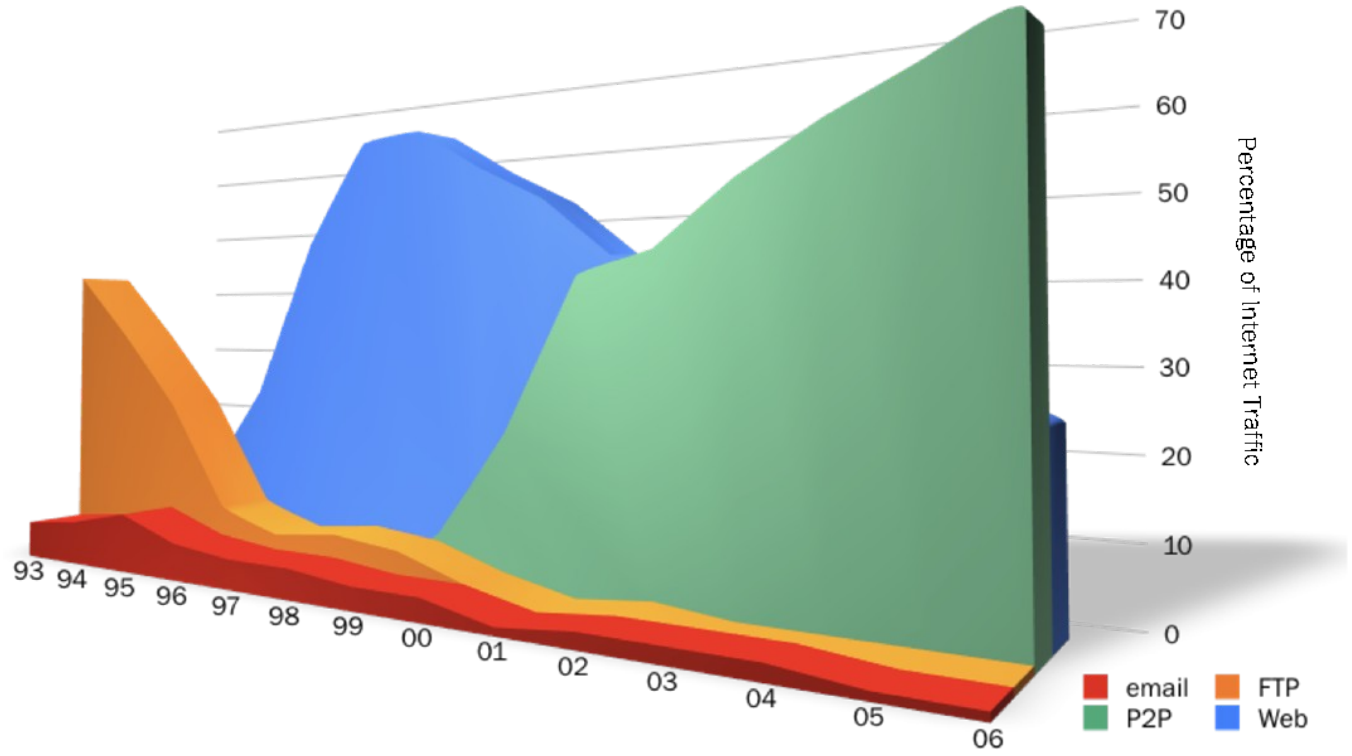
Lime wire

KaZaA

vcddler

P2P Why Should We Care?

CacheLogic Research
Internet Protocol Breakdown 1993 - 2006



Outline

- What is P2P?
- Evolution (Discovery Related)
  - 1st Generation: Centralized systems
    - Napster
  - 2nd Generation: Flooding-Based systems
    - Gnutella
  - 3rd Generation: Distributed Hash Tables (DHT)
    - Chord, Pastry, Kademlia, etc...

What is P2P Computing? (1/3)

- Oram (First book on P2P): P2P is a class of applications, that
  - Takes advantage of resources – (storage, cpu, etc,..) – available at the edges of the Internet.
  - Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers.

What is P2P Computing? (2/3)

- P2P Working Group (A Standardization Effort): P2P computing is
  - The sharing of computer resources and services by direct exchange between systems.
  - Peer-to-peer computing takes advantage of existing computing power and networking connectivity, allowing economical clients to leverage their collective power to benefit the entire enterprise.

What is P2P Computing? (3/3)

- Our view: P2P computing is distributed computing with the following desirable properties:
  - Resource Sharing
  - Dual client/server role
  - Decentralization/Autonomy
  - Scalability
  - Robustness/Self-Organization

P2P Research Issues

- Discovery:
  - Where are things?
- Content Distribution:
  - How fast can we get things?
- NAT/Firewalls
  - Jumping over them
- Security
- Anonymity
- ...

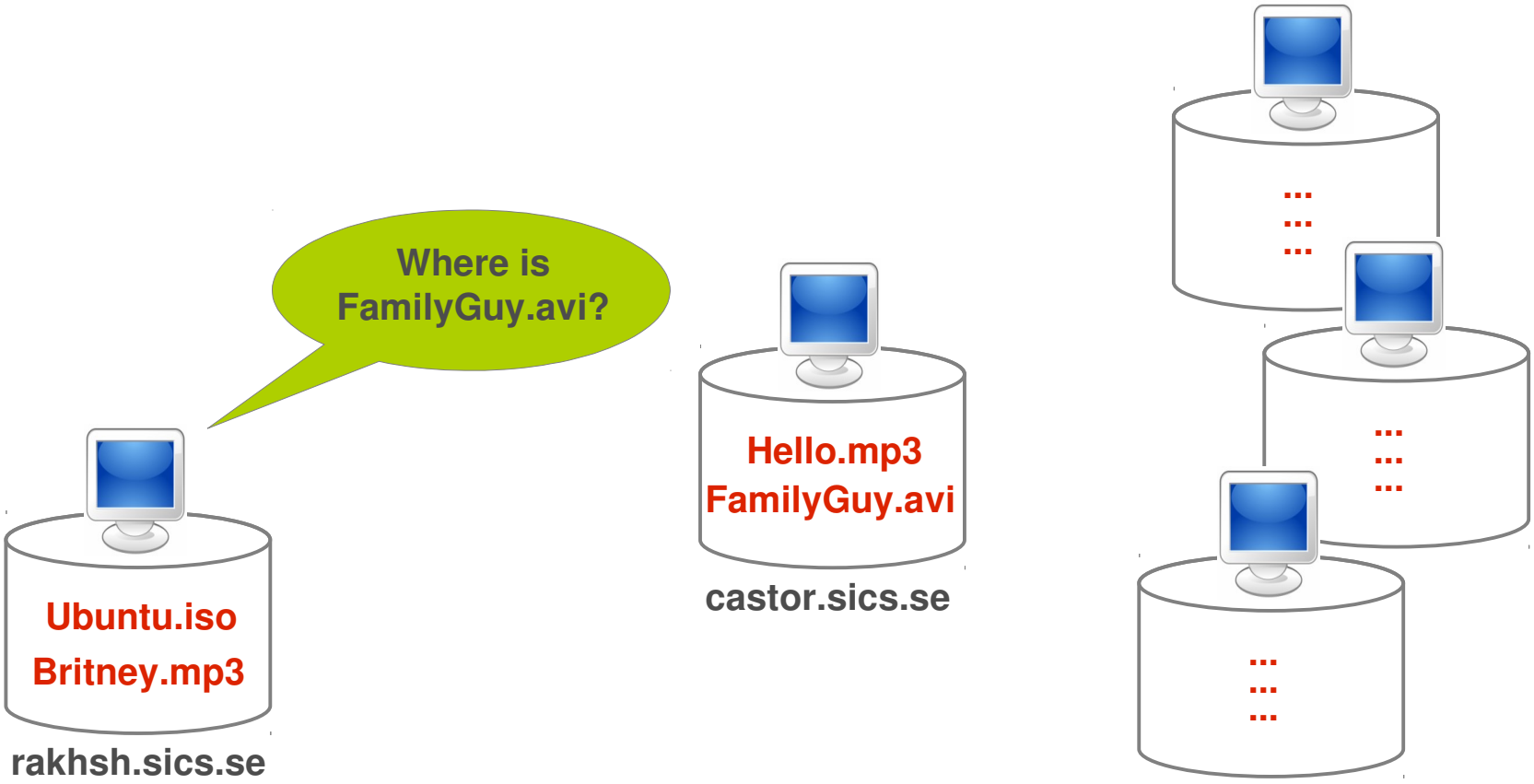
Let us see how did it all start ...

- Some users store data items on their machines.
- Other users are interested in this data.
- Problem: [d]

Let us see how did it all start ...

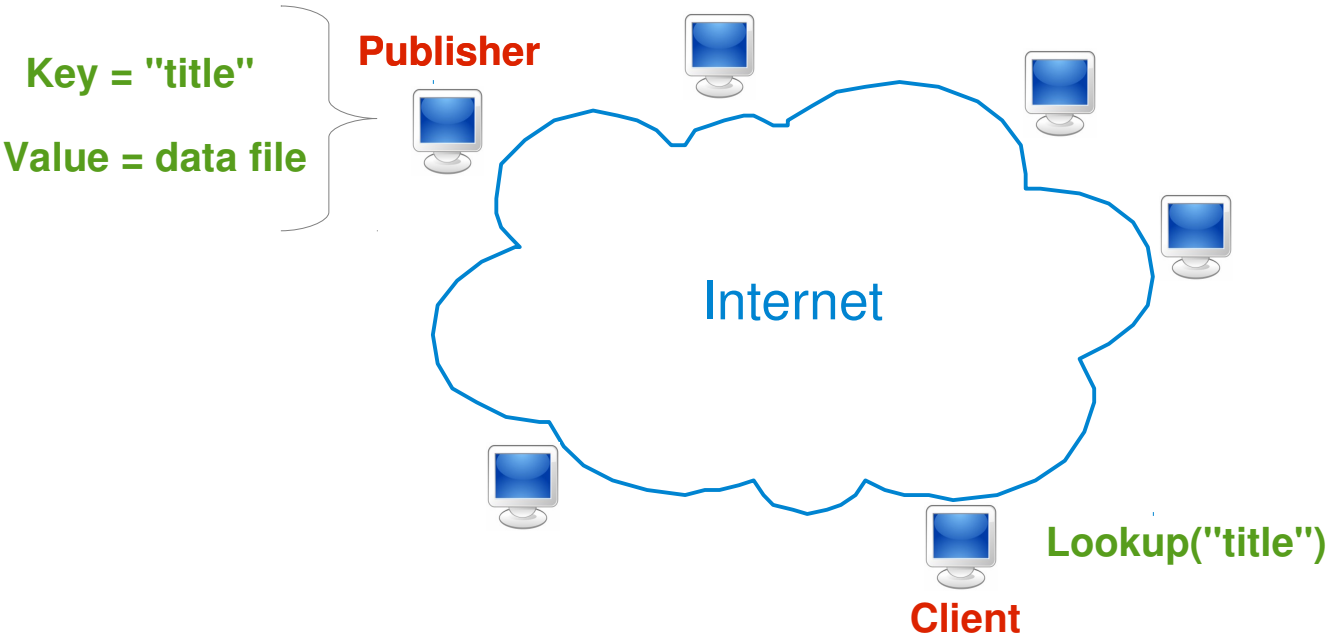
- Some users store data items on their machines.
- Other users are interested in this data.
- Problem: How does a user know which other user(s) in the world have the data item(s) that s/he desires?

Let us see how did it all start ...



Example P2P Problem: Lookup

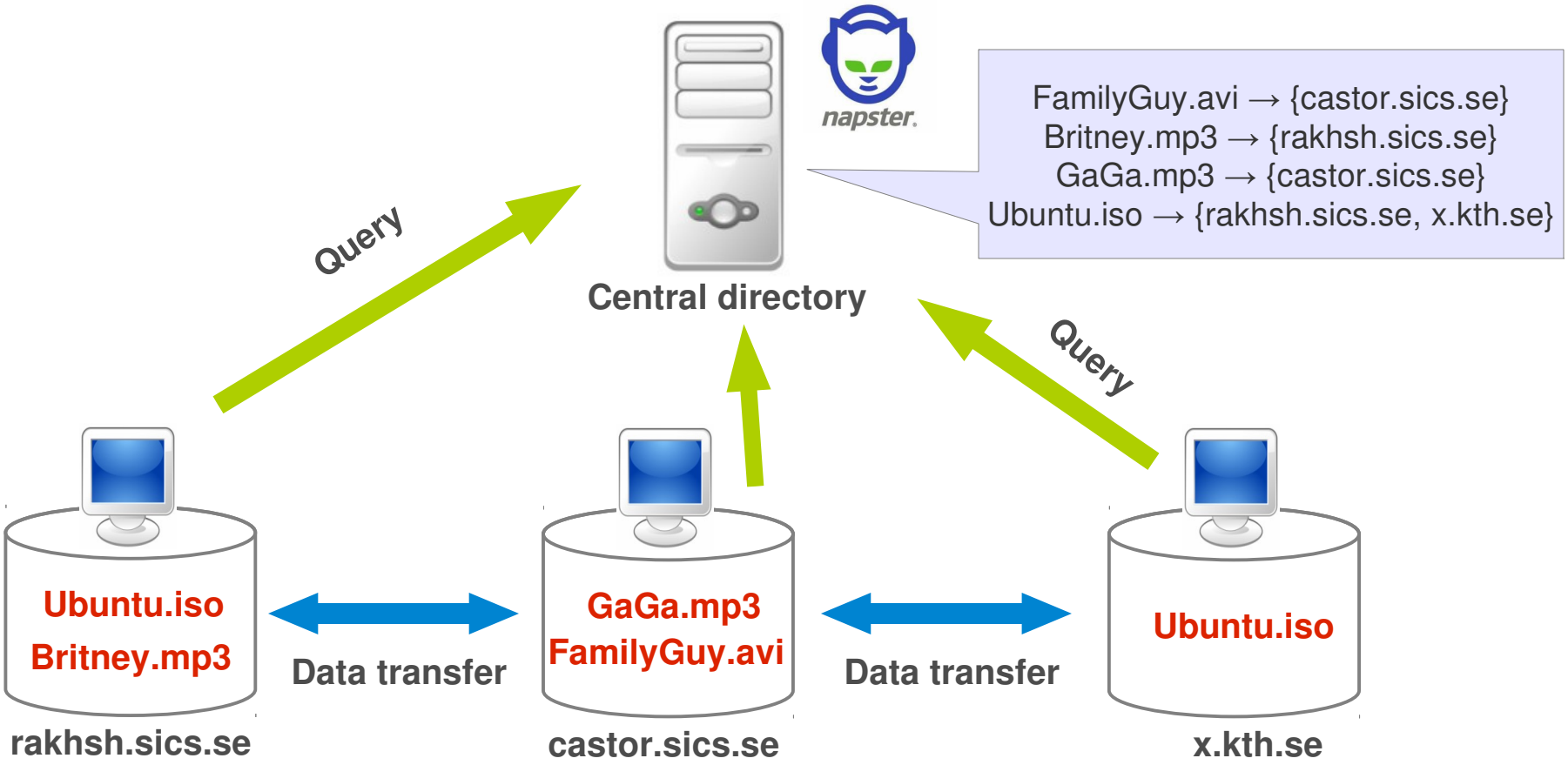
- At the heart of all P2P systems.



First Generation

First Generation of P2P Systems

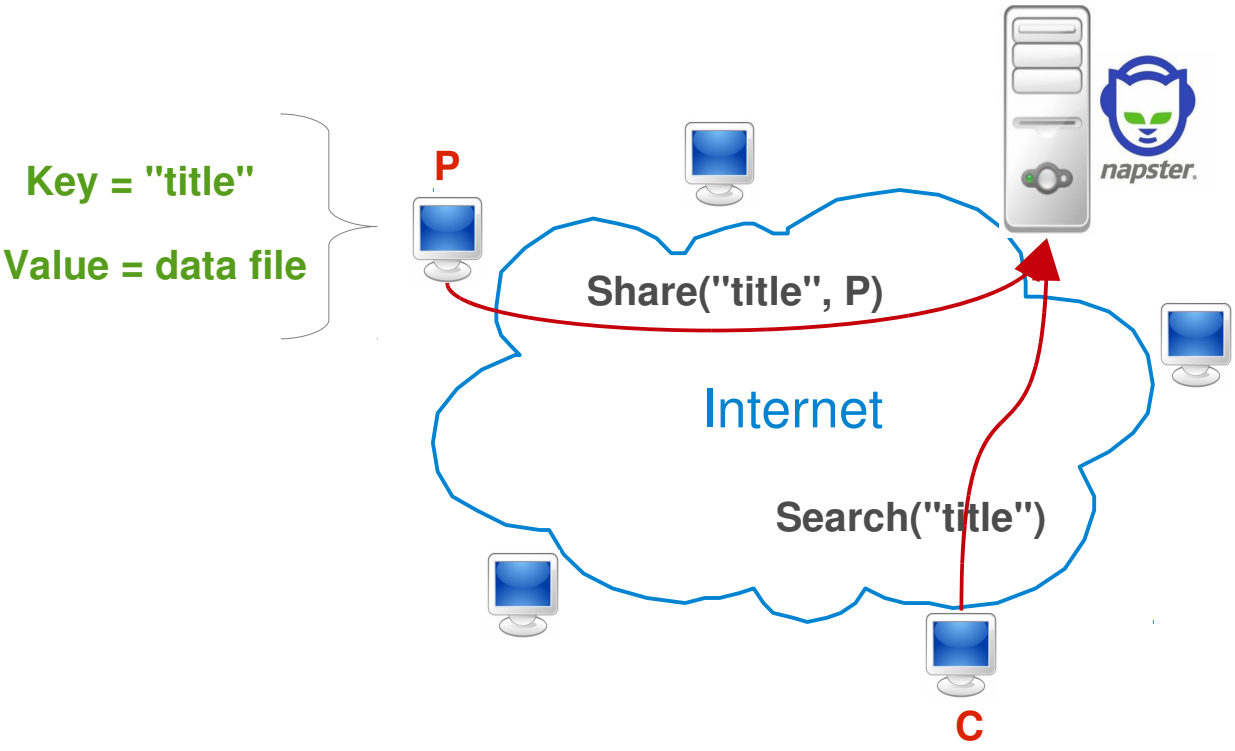
- Central Directory + Distributed Storage



Basic Operations in Napster

- Join
  - Connect to the central server (Napster)
- Share (Publish/Insert)
  - Inform the server about what you have
- Leave/Fail
  - Simply disconnect
  - Server detects failure, removes your data from the directory
- Search (Query)
  - Ask the central server and it returns a list of hits
- Download
  - Directly download from other nodes using the hits provided by the server

Centralized Lookup



The End of Napster

- Since users of Napster stored copyrighted material, the service was stopped for legal reasons.

Napster Advantage/Disadvantage?

- Advantage/Disadvantage [d]

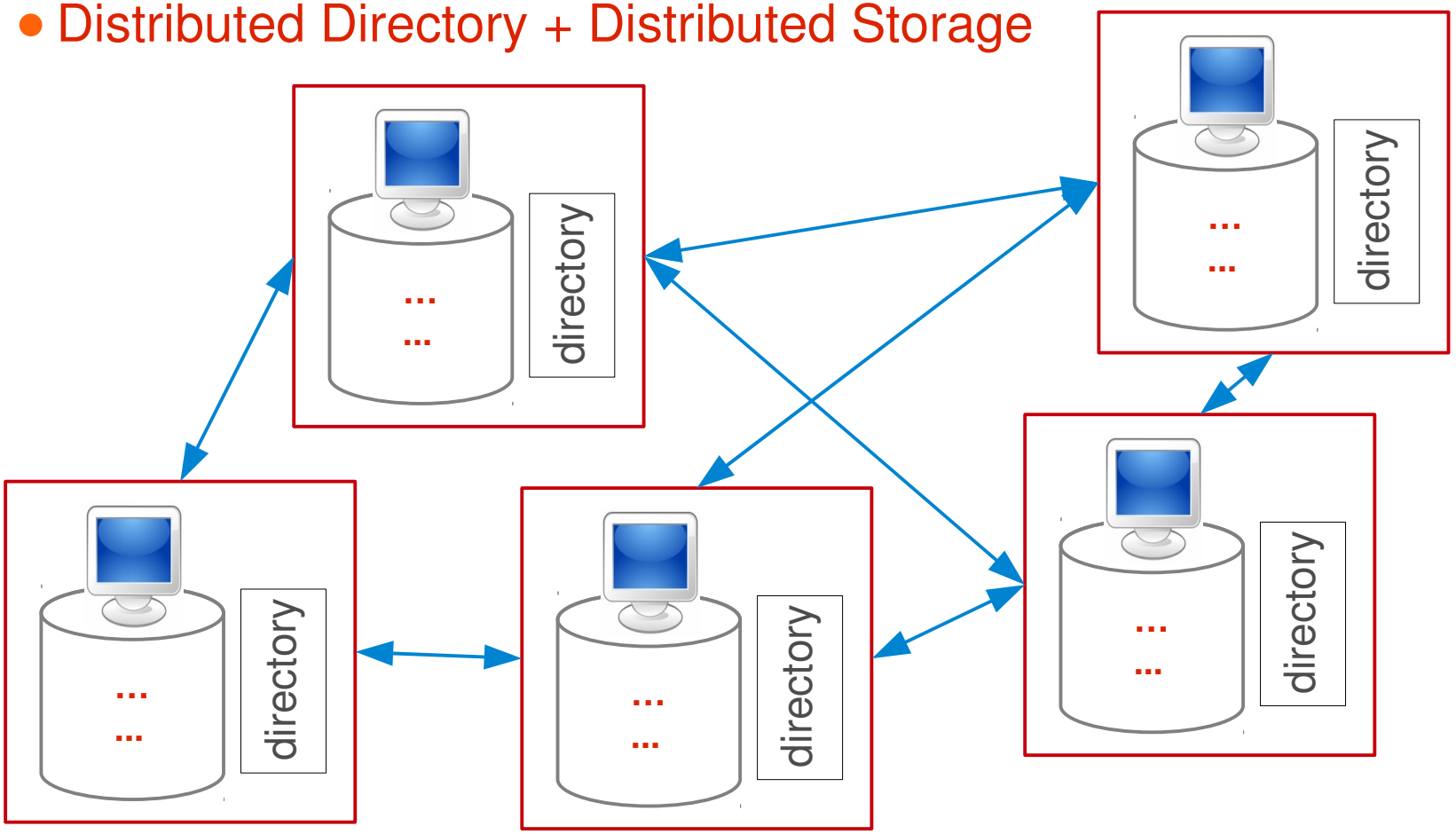
Napster Advantage/Disadvantage?

- Advantage/Disadvantage [d]
- Advantage
  - Simple
- Disadvantage
  - O(N) state in server
  - Single point of failure

Second Generation

Second Generation of P2P Systems

- Distributed Directory + Distributed Storage

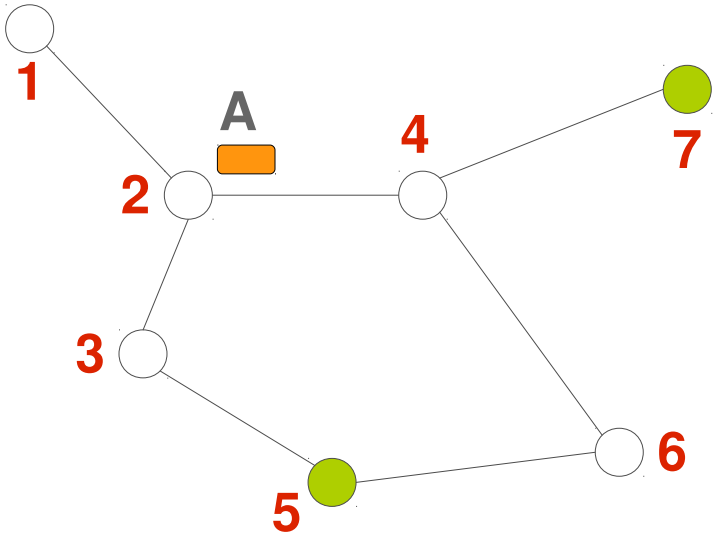


Gnutella Protocol Messages

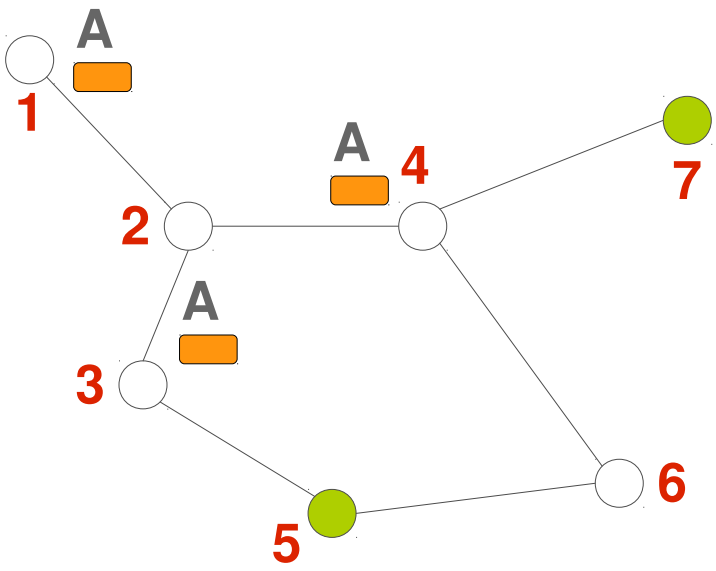
- Broadcast Messages
  - Ping: initiating message ("I'm here") for overlay maintenance
  - Query: search pattern and TTL (time-to-live)
- Back-Propagated Messages
  - Pong: reply to a ping, contains information about the peer
  - Query Hit: contains information about the computer that has the requested file
- Node-to-Node Messages
  - GET: return the requested file
  - PUSH: push the file to the requester node

Gnutella Search Mechanism

- Node 2 initiates search for file A

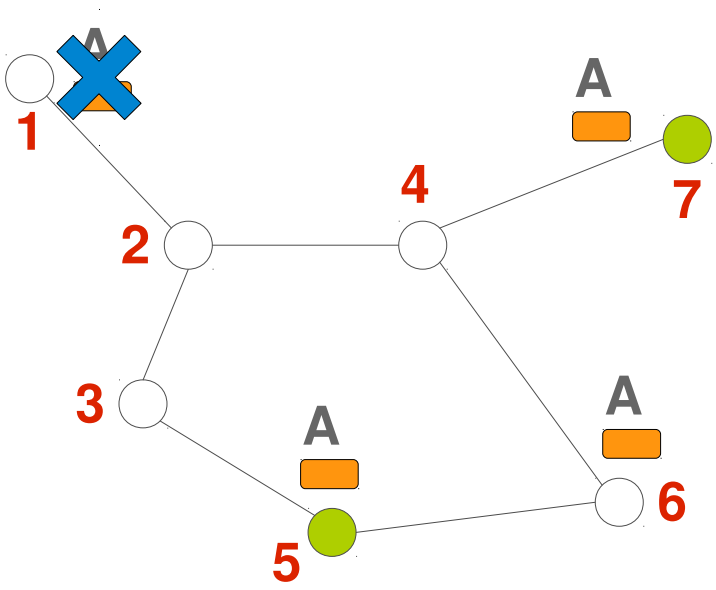


Gnutella Search Mechanism



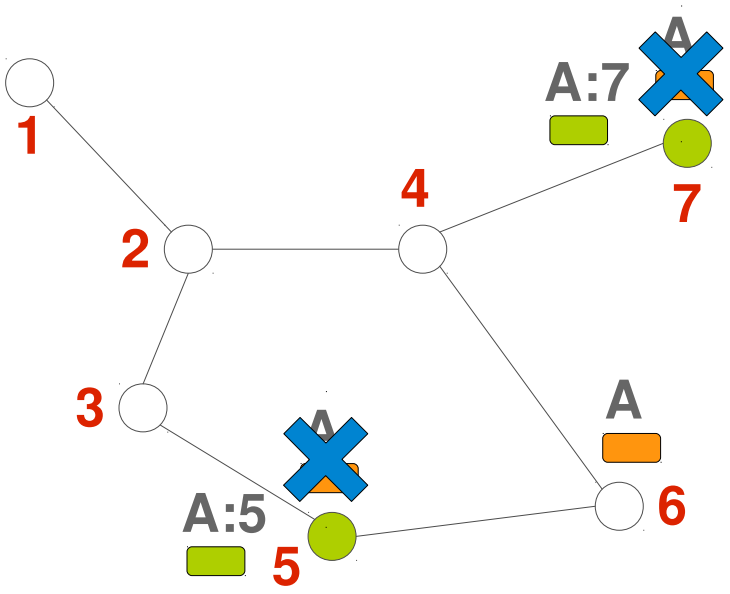
- Node 2 initiates search for file A
- Sends message to all neighbours

Gnutella Search Mechanism



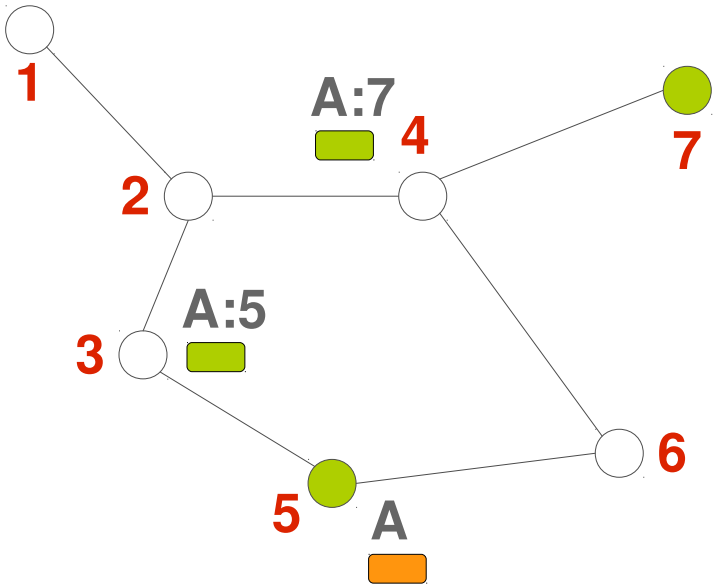
- Node 2 initiates search for file A
- Sends message to all neighbours
- Neighbours forward message

Gnutella Search Mechanism



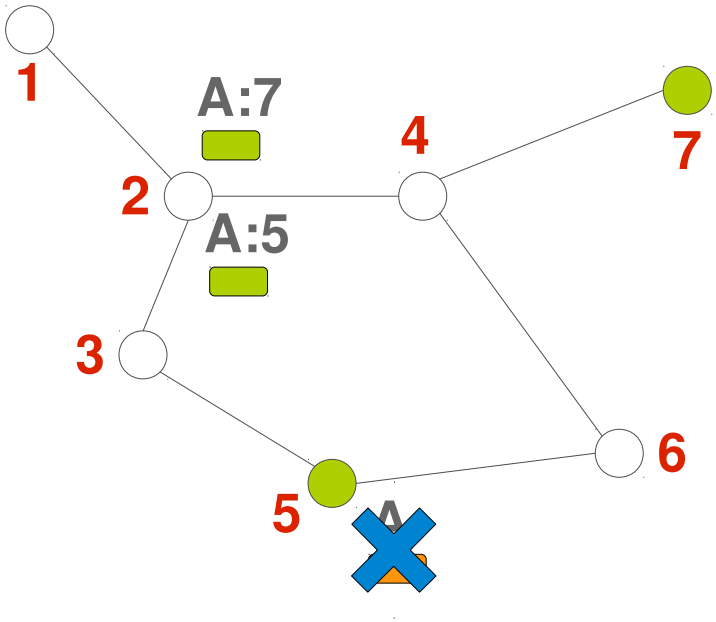
- Node 2 initiates search for file A
- Sends message to all neighbours
- Neighbours forward message
- Nodes that have file A initiate a reply message

Gnutella Search Mechanism



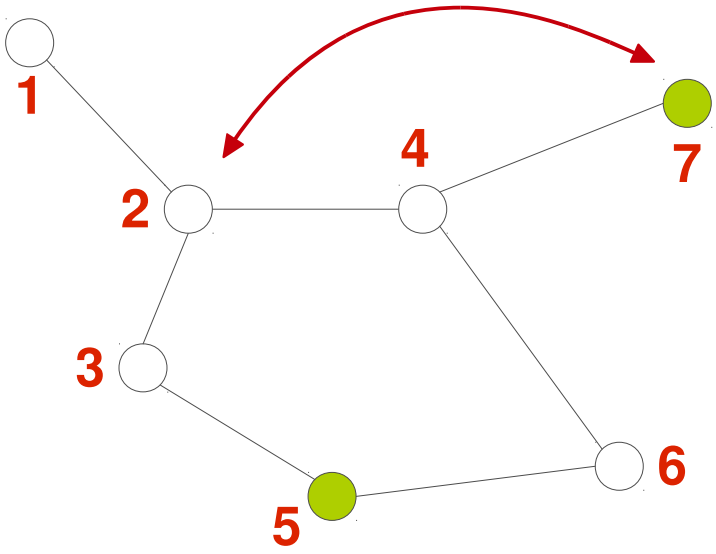
- Node 2 initiates search for file A
- Sends message to all neighbours
- Neighbours forward message
- Nodes that have file A initiate a reply message
- Query reply message is back propagated

Gnutella Search Mechanism



- Node 2 initiates search for file A
- Sends message to all neighbours
- Neighbours forward message
- Nodes that have file A initiate a reply message
- Query reply message is back propagated

Gnutella Search Mechanism



- Node 2 initiates search for file A
- Sends message to all neighbours
- Neighbours forward message
- Nodes that have file A initiate a reply message
- Query reply message is back propagated
- Nodes 2 directly connects to node 7 and downloads file A

Gnutella Advantage/Disadvantage?

- Advantage/Disadvantage [d]

Gnutella Advantage/Disadvantage?

- Advantage/Disadvantage [d]
- Advantage
  - Robust
- Disadvantage
  - Worst case O(N) message per lookup
  - No guarantees to find data item
    - Because of TTL

Third Generation

Distributed Hash Tables (DHT)

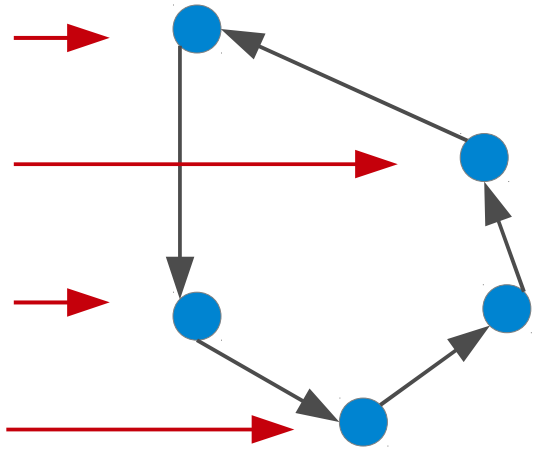
- An ordinary hashtable, which is ...

Key	Value
Fatemeh	Stockholm
Ali	California
Tallat	Islamabad
Cosmin	Bucharest
Seif	Stockholm
Amir	Tehran

Distributed Hash Tables (DHT)

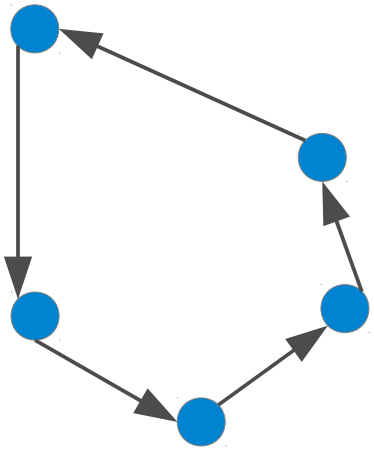
- An ordinary hashtable, which is distributed.

Key	Value
Fatemeh	Stockholm
Ali	California
Tallat	Islamabad
Cosmin	Bucharest
Seif	Stockholm
Amir	Tehran



Distributed Hash Tables (DHT)

- put(key,value), get(key) interface.
- The neighbours of a node are well-defined and not randomly chosen.
- Values are no longer stored at their owners, instead the network chooses at which node a data item will be stored.
- Every node provides a lookup operation.
- Nodes keep routing pointers
  - If item not found, route to another node



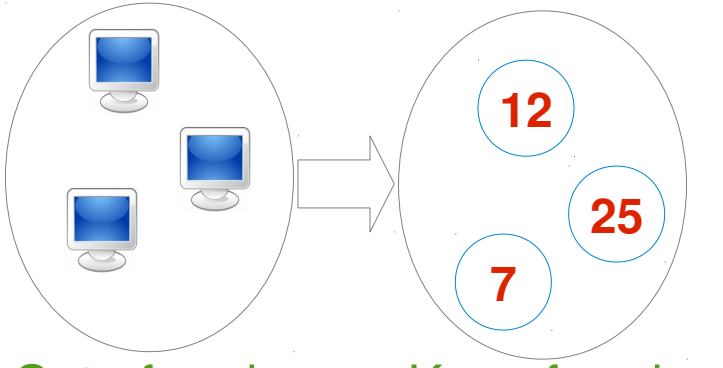
The Key Idea in DHTs

- 1st and 2nd Generation:
  - Each data item is stored in the machine of its creator/downloader.

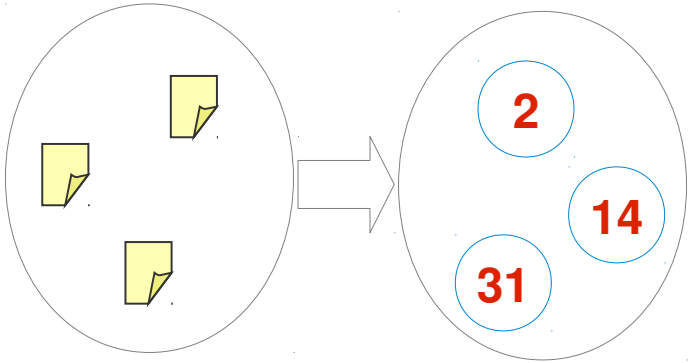
- 3rd Generation (DHTs):
  - The ID of a data item determines the machine on which it is going to be stored.

Distributed Hash Tables (DHT)

1. Decides on common key space for nodes and values

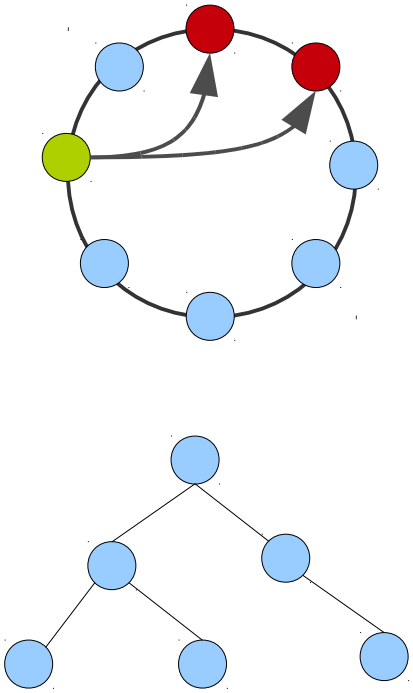


Set of nodes Key of nodes

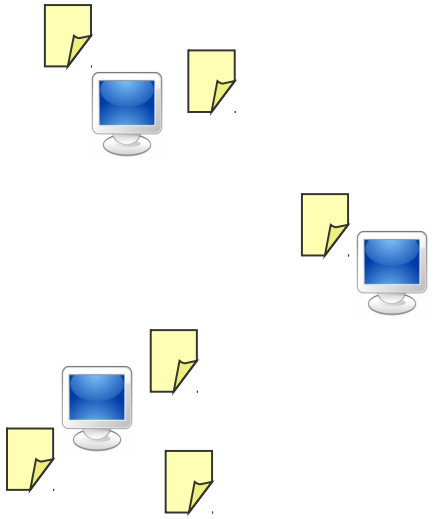


Set of items Key of items

2. Connects the nodes smartly



3. Make a strategy for assigning items to nodes



Consistent Hashing using a Ring (1/6)

- Identifier space of size 16, [0, 15].

rakhsh.sics.se

H(rakhsh.sics.se)=12

castor.sics.se

H(castor.sics.se)=3

x.kth.se

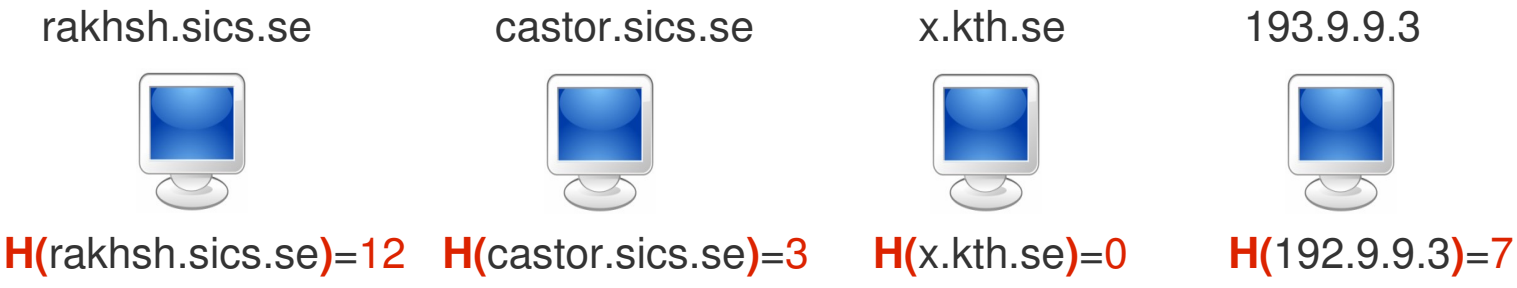
H(x.kth.se)=0

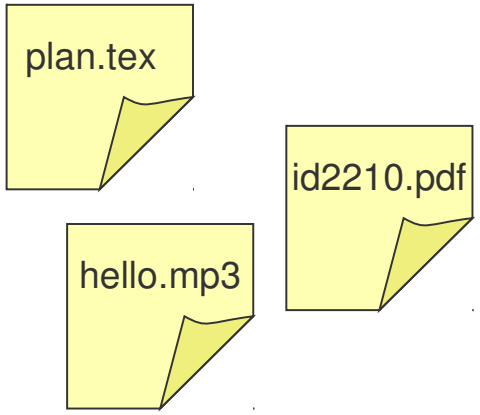
193.9.9.3

H(192.9.9.3)=7

Consistent Hashing using a Ring (2/6)

- Identifier space of size 16, [0, 15].





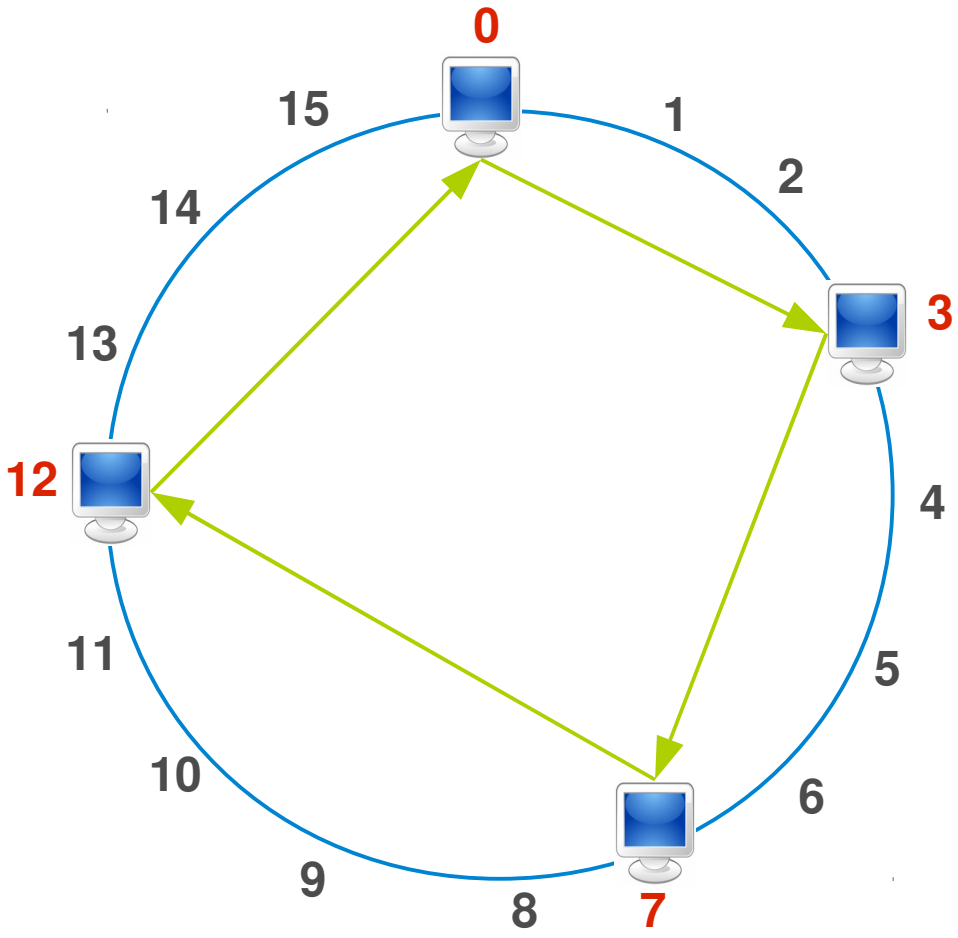
H(plan.tex)=2

H(id2210.pdf)=12

H(hello.mp3)=14

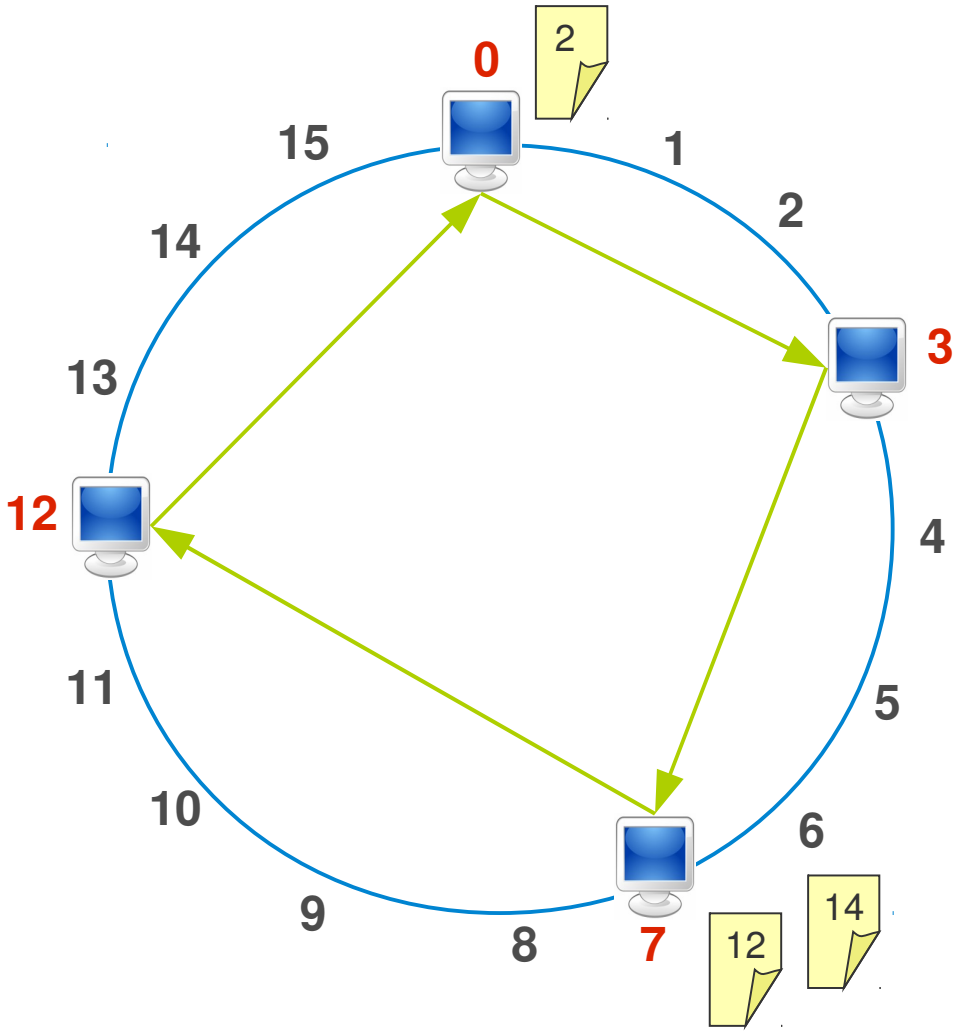
Consistent Hashing using a Ring (3/6)

- Assume the ID space is [0, 15], i.e. a maximum of 16 nodes.
- We treat this range as a circular id space.
- succ(x): is the first node on the ring with id greater than or equal x, where x is the id of a document or node.
- The successor of node i is succ(i+1).
- Thus, the nodes are forming a ring.



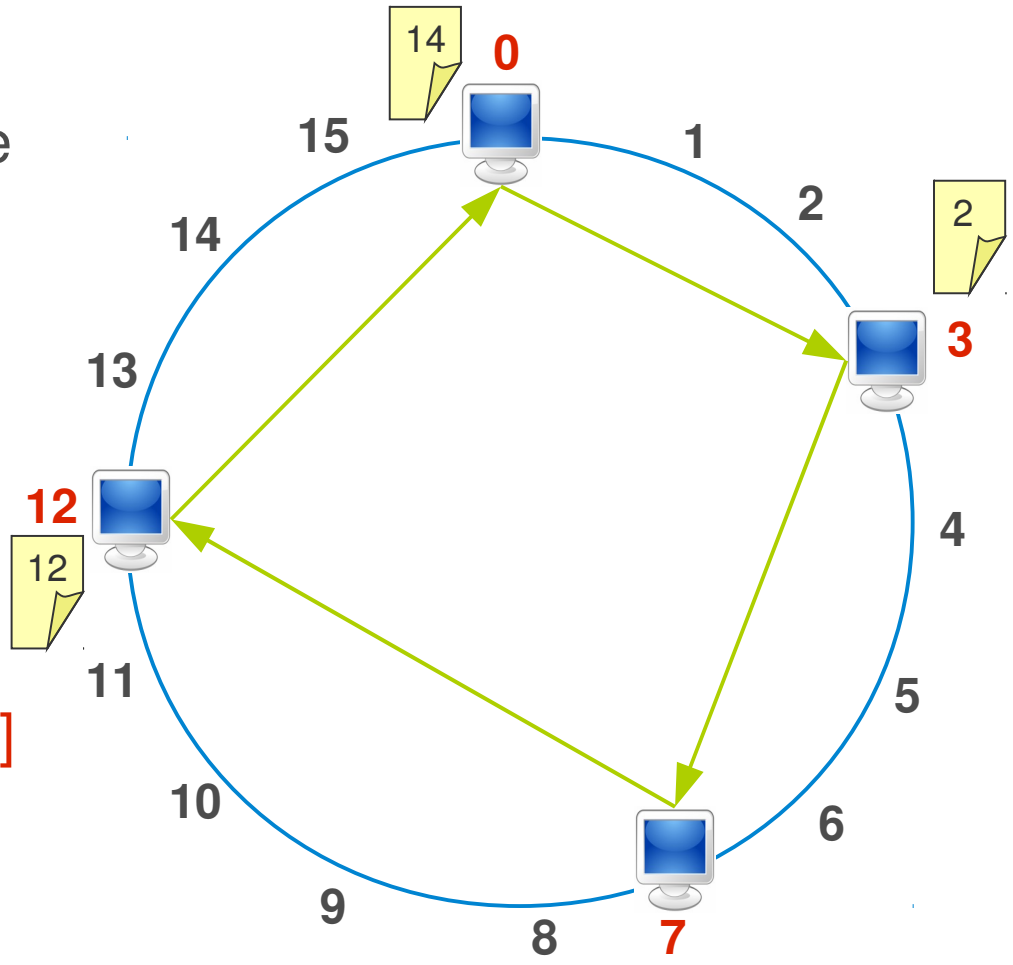
Consistent Hashing using a Ring (4/6)

- Using this ring, we can decide which item is stored at which node.
- Initially, node 0 stored item 2 and node 7 stored items 12 and 14.
- The policy is: An item with ID x, would be stored at the node with id succ(x).



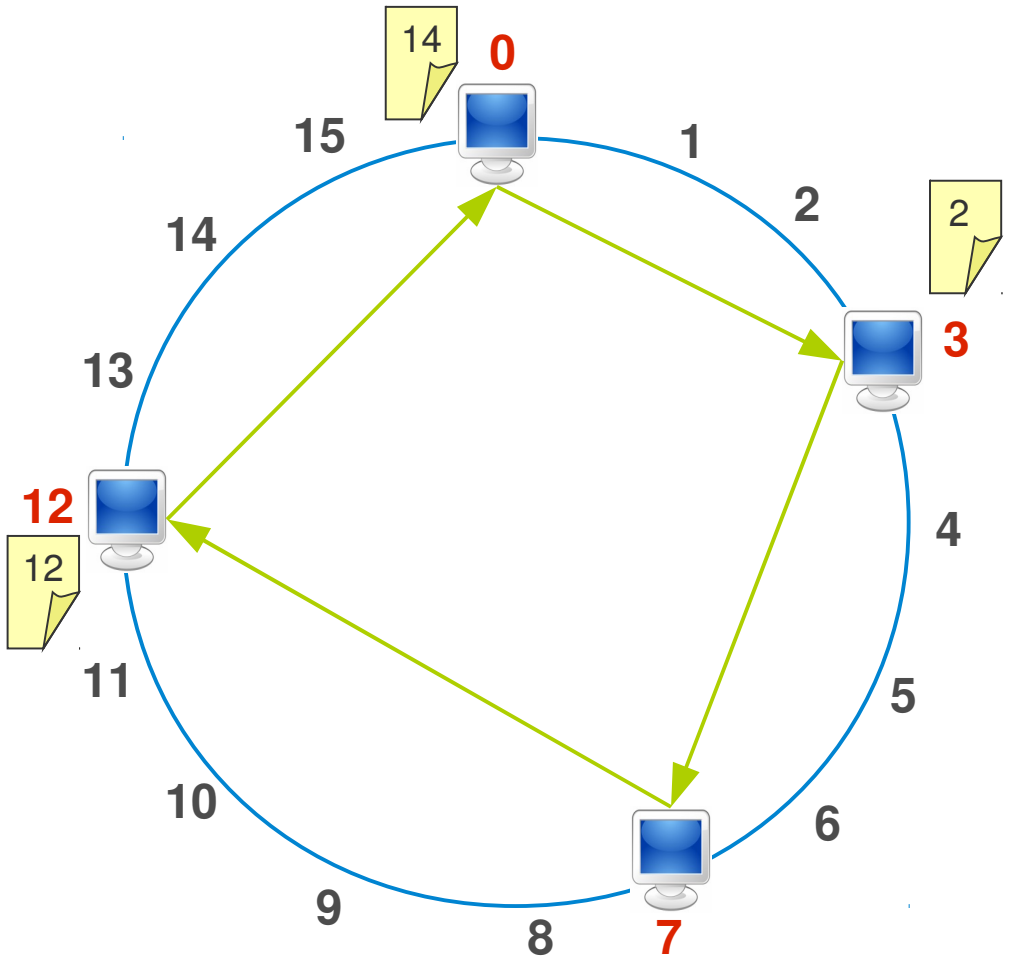
Consistent Hashing using a Ring (5/6)

- The policy is: An item with ID x, would be stored at the node with id succ(x).
- So, node 0 gets to store item 14, node 3 to store item, and node 12 to store item 12.
- But how can we do this? [d]



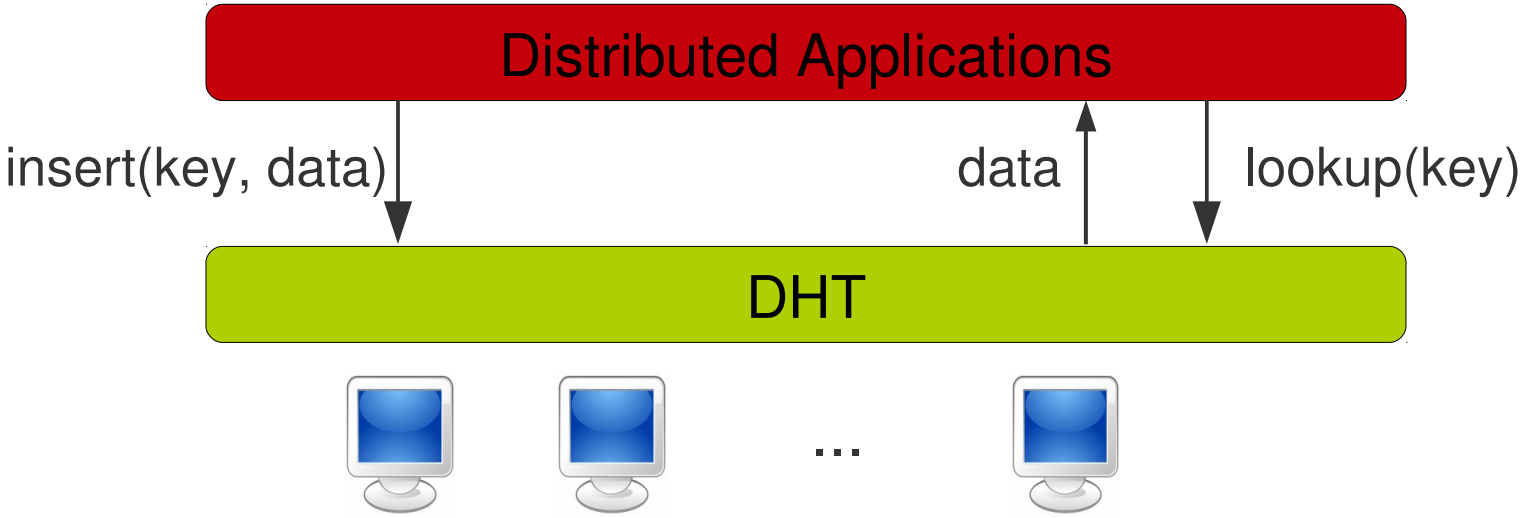
Consistent Hashing using a Ring (6/6)

- But how can we do this? [d]
- If the successor pointers are already there, the two operations, get and put would be simply done by following them sequentially.
- From any node, you can do: put(hash(item), item)
- From any node, you can do: get(hash(item))



Distributed Hash Tables (DHT)

- Nodes are the hash buckets
- Key identifies data uniquely
- DHT balances keys and data across nodes
- DHT replicates, caches, routes lookups, etc.



Why DHTs Now?

- Demand pulls
  - Growing need for security and robustness.
  - Large-scale distributed applications are difficult to build.
  - Many applications use location-independent data.
- Technology pushes
  - Faster, and better computers: every PC can be a server.
  - Scalable lookup algorithms.
  - Trustworthy systems from untrusted components.

DHT is a Good Interface

- Supports a wide range of applications, because
  - Keys have no semantic meaning
  - Values are application dependent

- Minimal interface

DHT	UDP/IP
lookup(key) → data insert(key, data)	send(IP addr, data) recv(IP addr) → data

DHT is a Good Shared Infrastructure

- Applications inherit some security and robustness from DHT
  - DHT replicates data
  - Resistant to malicious participants
- Low-cost deployment
  - Self-organizing across administrative domains
  - Allows to be shared among applications
- Supports large scale workloads

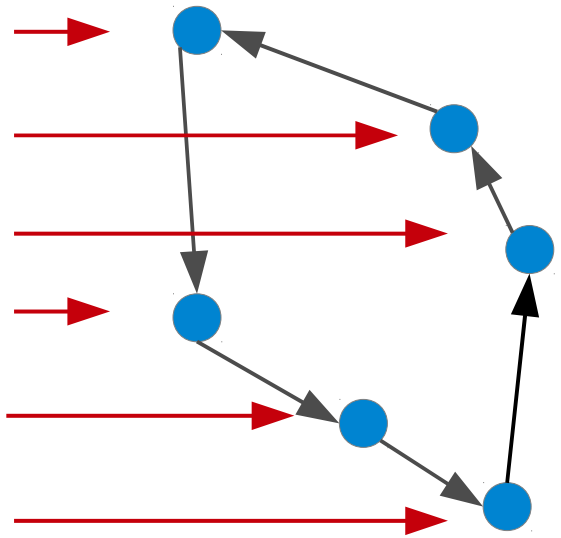
DHT Applications

- Distributed File Systems [CFS, OceanStore, PAST, Arla/DKS]
- Web cache/archives [Squirrel]
- Censor-resistant stores [Eternity, FreeNet]
- Event notification [Scribe, DKS]
- Naming systems [ChordDNS, INS]
- Query and indexing [Kademlia]
- Communication primitives [I3]
- Backup store [HiveNet]
- Distributed Authorizations Delegation

Name-based Communication

- Map names to locations

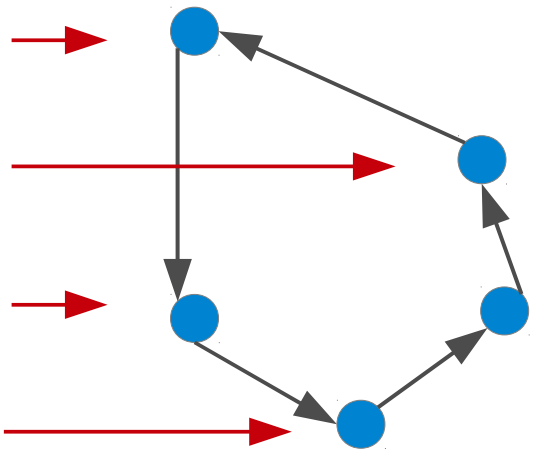
Key	Value
Fatemeh	130.237.32.10
Ali	192.9.12.5
Amir	18.7.23.5
Cosmin	10.10.95.4
Seif	127.5.220.12
Tallat	12.110.210.2



Distributed Backup

- Clients install the backup tool
- Decide on amount of space to share
- Choose files for backup
- Data is encrypted
- Stored in the directory

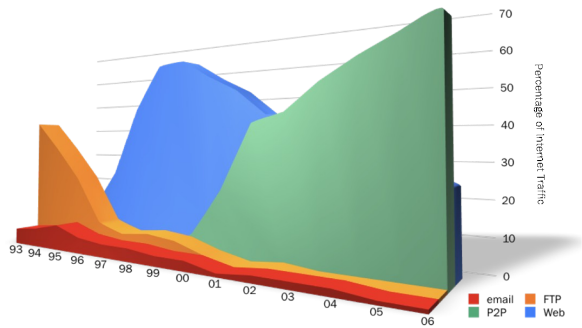
Key	Value
Hi.mp3	2343
2210.txt	2511
Bye.avi	4539
...	...
...	...
...	...



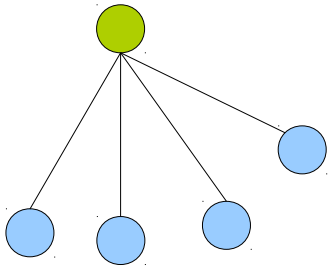
A Page to Remember

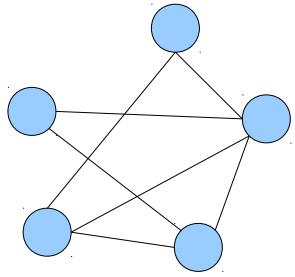
A Page to Remember

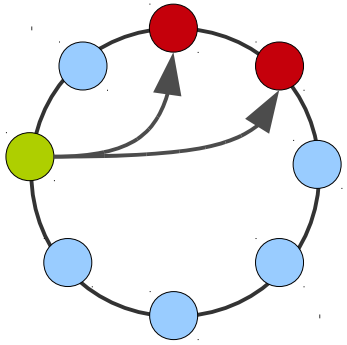
- P2P computing
  - Resource Sharing
  - Dual client/server role
  - Decentralization/Autonomy
  - Scalability
  - Robustness/Self-Organization



- Three generations:
  - 1st Generation: Centralized systems
    - Napster
  - 2nd Generation: Flooding-Based systems
    - Gnutella
  - 3rd Generation: Distributed Hash Tables (DHT)
    - Chord, Pastry, Kademlia, etc...







Question?