Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications

Ion Stoica, Robert Morris, David Liben-Nowell,
David R. Karger, M. Frans Kaashoek,
Frank Dabek, Hari Balakrishnan

Presented by John Otto; 5 February 2008
Northwestern University – Winter 2008 – EECS345 Distributed Systems

Outline

- Overview
- DHT Comparison
- Goals and Applications
- Architecture and Protocol
- Evaluation
- Discussion and Questions

Overview

- Motivation: Distributed storage critical to P2P
- Provides simple key location service
- Slow, but correct function in face of failure
- Scalable

DHT Comparison

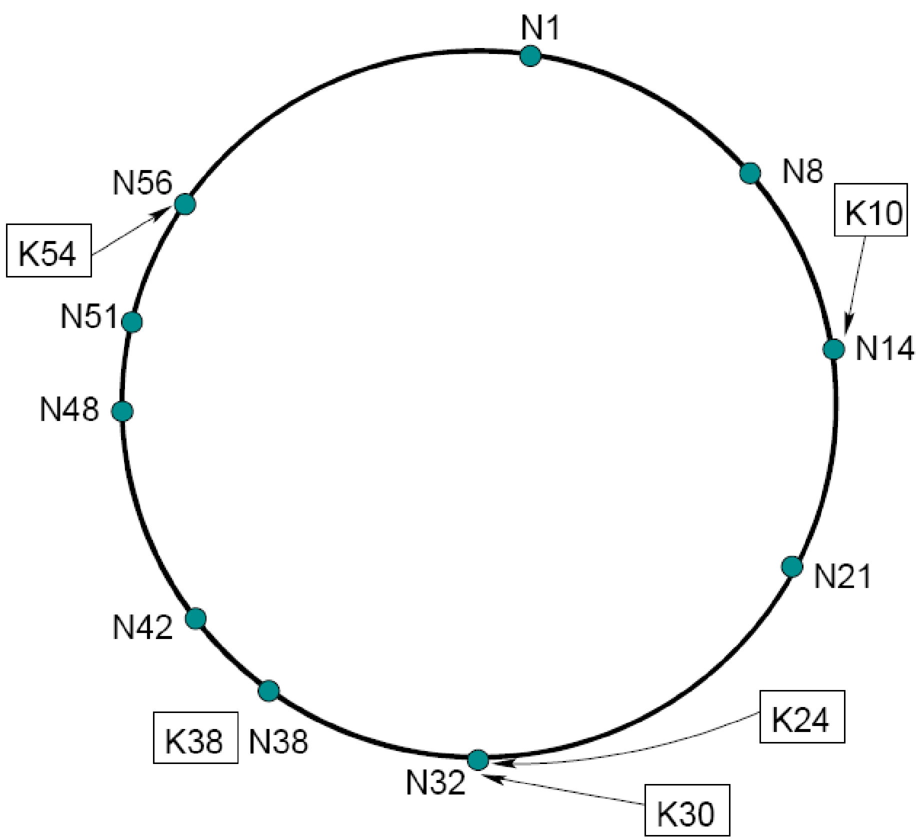
- DNS
  - Centralized: special servers, well-known addresses
  - Relies on administrative boundaries (domain names)
- Freenet
  - Decentralized, anonymous
  - Searches for cached copies
- Ohaha
  - Consistent hashing for fair loading
- Globe
  - Similar to DNS: static search tree
- Tapestry
  - Provides guarantees about distance query travels

Goals and Applications

- Load balancing
- Decentralization
- Scalability
- Availability
- Flexible naming

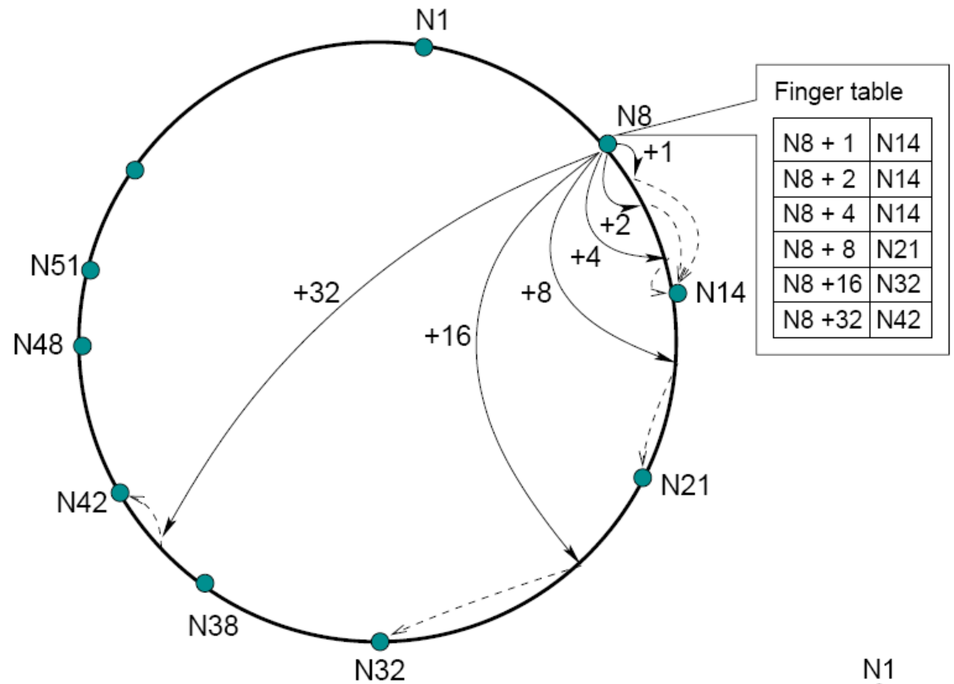
- Cooperative mirroring
- Time-shared storage
- Distributed indexes
- Large-scale combinatorial search

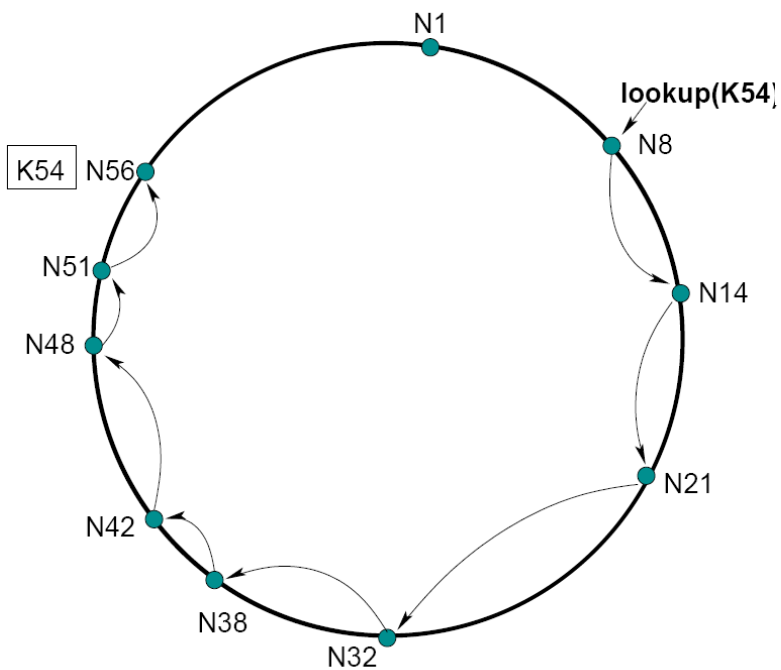
Architecture and Protocol

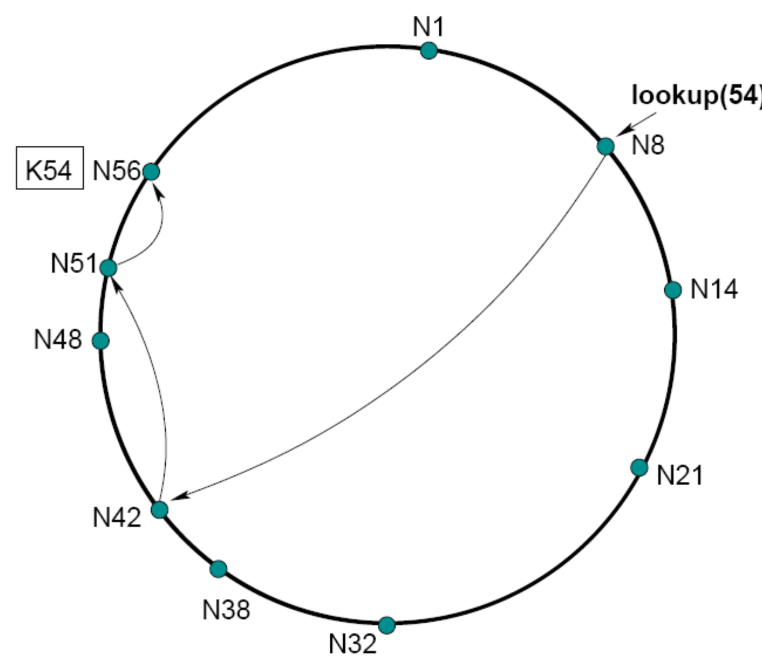


- Node, key hashing
  - Assumptions
- Scalability
  - Load balancing
- Stabilization
  - Keeps finger tables, successor, predecessor information up to date
- Resiliency
  - List of r successors

Benefit of Finger Table







Join Operation

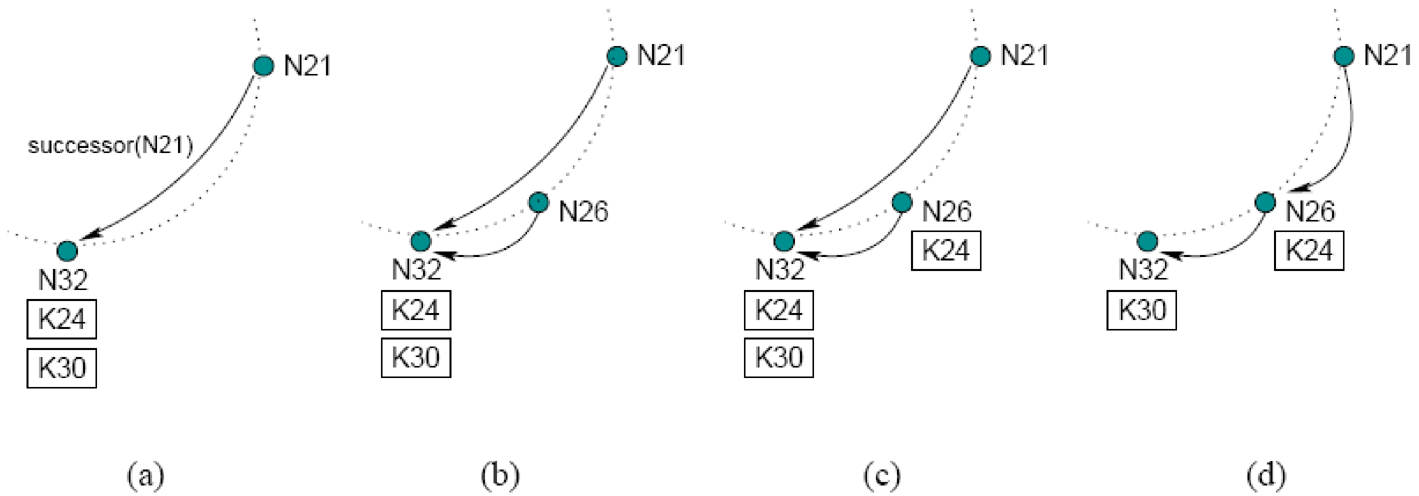


Fig. 7. Example illustrating the join operation. Node 26 joins the system between nodes 21 and 32. The arcs represent the successor relationship. (a) Initial state: node 21 points to node 32; (b) node 26 fi nds its successor (i.e., node 32) and points to it; (c) node 26 copies all keys less than 26 from node 32; (d) the stabilize procedure updates the successor of node 21 to node 26.

Virtual Nodes for Fair Key Distribution

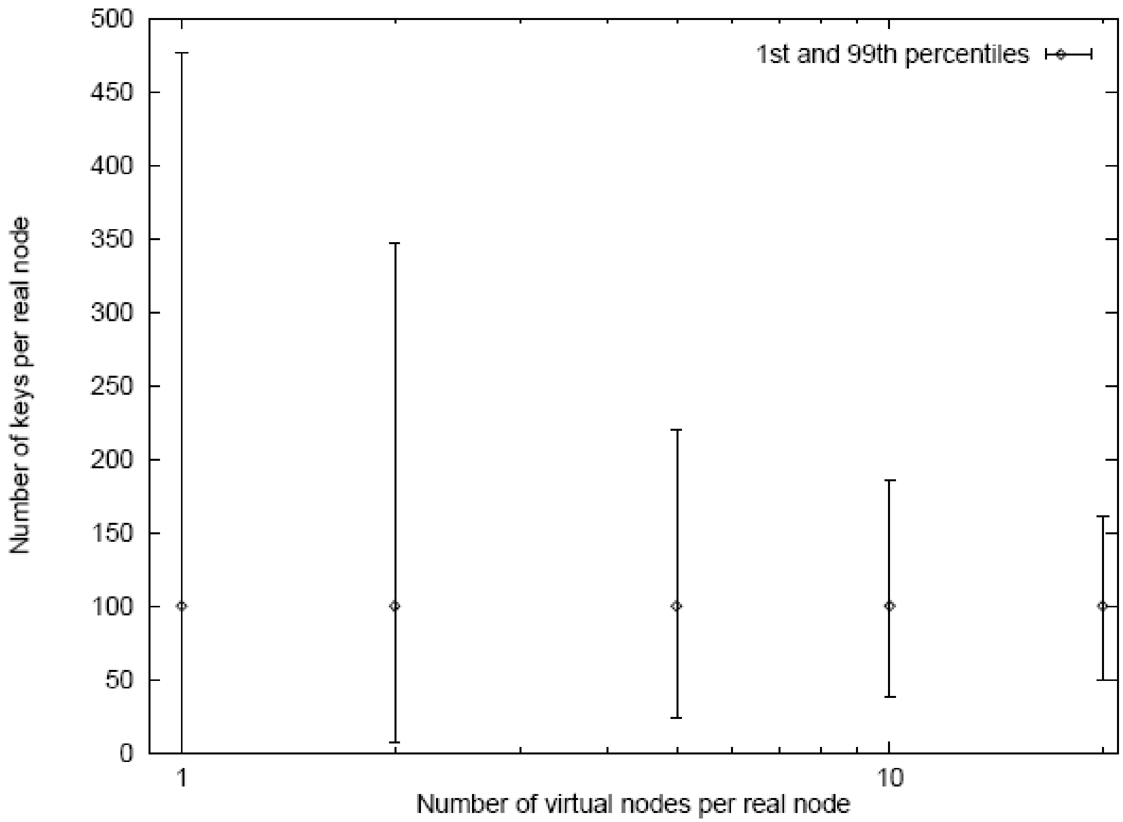
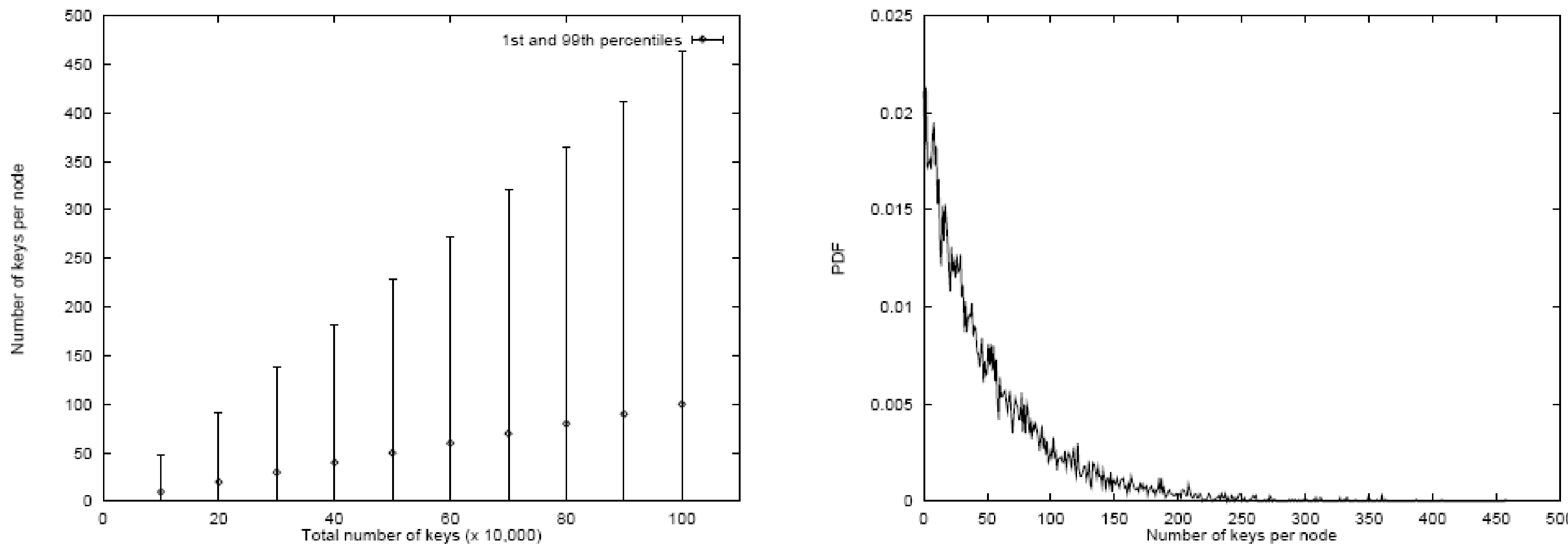


Fig. 9. The 1st and the 99th percentiles of the number of keys per node as a function of virtual nodes mapped to a real node. The network has 10^4 real nodes and stores 10^6 keys.

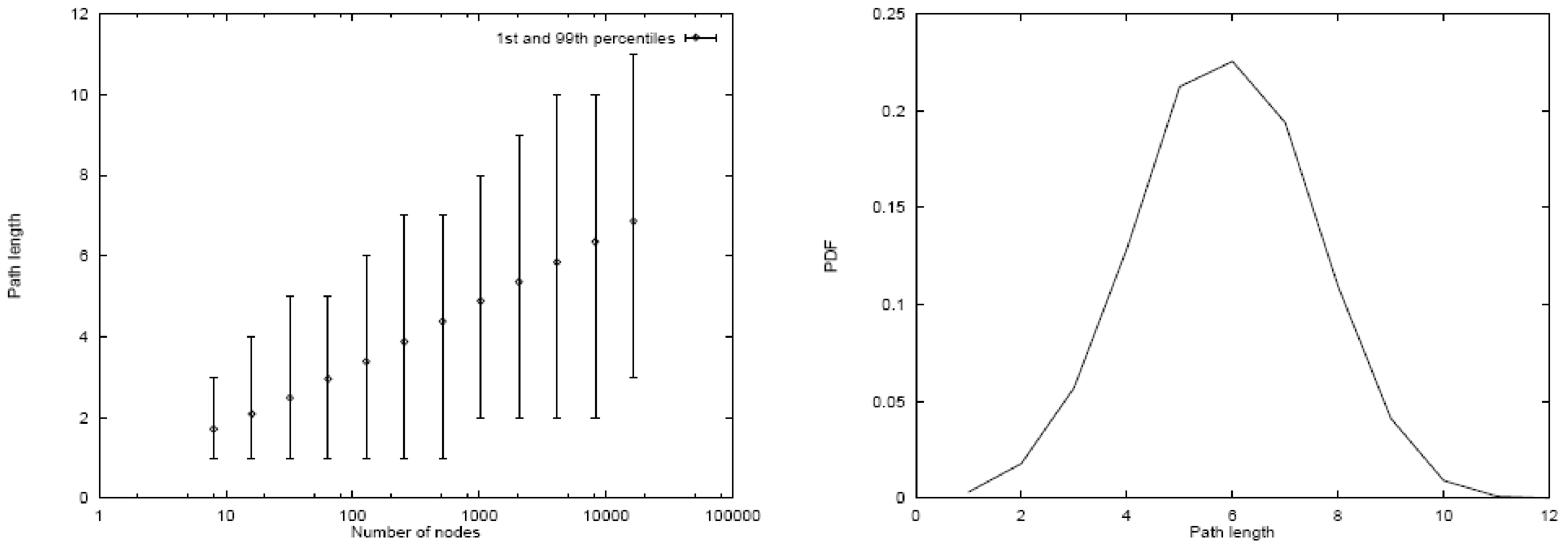
Evaluation: Load Sharing



(a) (b)

Fig. 8. (a) The mean and 1st and 99th percentiles of the number of keys stored per node in a 10^4 node network. (b) The probability density function (PDF) of the number of keys per node. The total number of keys is 5×10^5.

Path Length and Node Failures



(a) (b)

Fig. 10. (a) The path length as a function of network size. (b) The PDF of the path length in the case of a 2^{12} node network.

Fraction of failed nodes	**Mean path length (1st, 99th percentiles)**	**Mean num. of timeouts (1st, 99th percentiles)**
0	3.84 (2, 5)	0.0 (0, 0)
0.1	4.03 (2, 6)	0.60 (0, 2)
0.2	4.22 (2, 6)	1.17 (0, 3)
0.3	4.44 (2, 6)	2.02 (0, 5)
0.4	4.69 (2, 7)	3.23 (0, 8)
0.5	5.09 (3, 8)	5.10 (0, 11)

TABLE II

The path length and the number of timeouts experienced by a lookup as function of the fraction of nodes that fail simultaneously. The 1st and the 99th percentiles are in parenthesis. Initially, the network has 1,000 nodes.

Failure Rates under Churn

Node join/leave rate (per second/per stab. period)	Mean path length (1st, 99th percentiles)	Mean num. of timeouts (1st, 99th percentiles)	Lookup failures (per 10,000 lookups)
0.05 / 1.5	3.90 (1, 9)	0.05 (0, 2)	0
0.10 / 3	3.83 (1, 9)	0.11 (0, 2)	0
0.15 / 4.5	3.84 (1, 9)	0.16 (0, 2)	2
0.20 / 6	3.81 (1, 9)	0.23 (0, 3)	5
0.25 / 7.5	3.83 (1, 9)	0.30 (0, 3)	6
0.30 / 9	3.91 (1, 9)	0.34 (0, 4)	8
0.35 / 10.5	3.94 (1, 10)	0.42 (0, 4)	16
0.40 / 12	4.06 (1, 10)	0.46 (0, 5)	15

TABLE III

The path length and the number of timeouts experienced by a lookup as function of node join and leave rates. The 1st and the 99th percentiles are in parentheses. The network has roughly 1,000 nodes.

Discussion and Questions

- Replication of data: spreading around owner?
- Weakness against adversary?
- Hybrid system architecture:
  - centralized and DHT?
  - DHT and random graph?