DHT Routing

Presented by Emma Kilfoyle
October 24, 2013

DHT History/Background

- 1995 - Internet goes public
- Early 2000s - P2P file sharing, e.g. Napster (1999) and Gnutella (2000), gains popularity
- 2001 - researchers start developing fast, distributed lookup services (CAN, Chord, Tapestry, Pastry)
- Today - Cassandra (Facebook), Dynamo (Amazon), memcached (Twitter/Facebook), etc.

DHT History/Background

- 1995 - Internet goes public
- Early 2000s - P2P file sharing, e.g. Napster (1999) and Gnutella (2000), gains popularity
- 2001 - researchers start developing fast, distributed lookup services (CAN, **Chord**, Tapestry, Pastry)
- Today - Cassandra (Facebook), Dynamo (Amazon), memcached (Twitter/Facebook), etc.

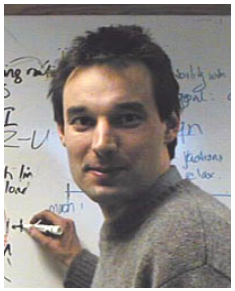
Chord: A P2P Lookup Service

- “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”
- MIT Laboratory for Computer Science and AI
- Presented at SIGCOMM 2001

Ion Stoica
UC Berkeley

Robert Morris
MIT, CSAIL

David Karger
MIT, CSAIL



M. F. Kaashoek
MIT, CSAIL

Hari Balakrishnan
MIT, CSAIL

Chord Goals

- Load balance
- Decentralization
- Scalability
- Availability
- Flexible naming

Chord Routing Basics

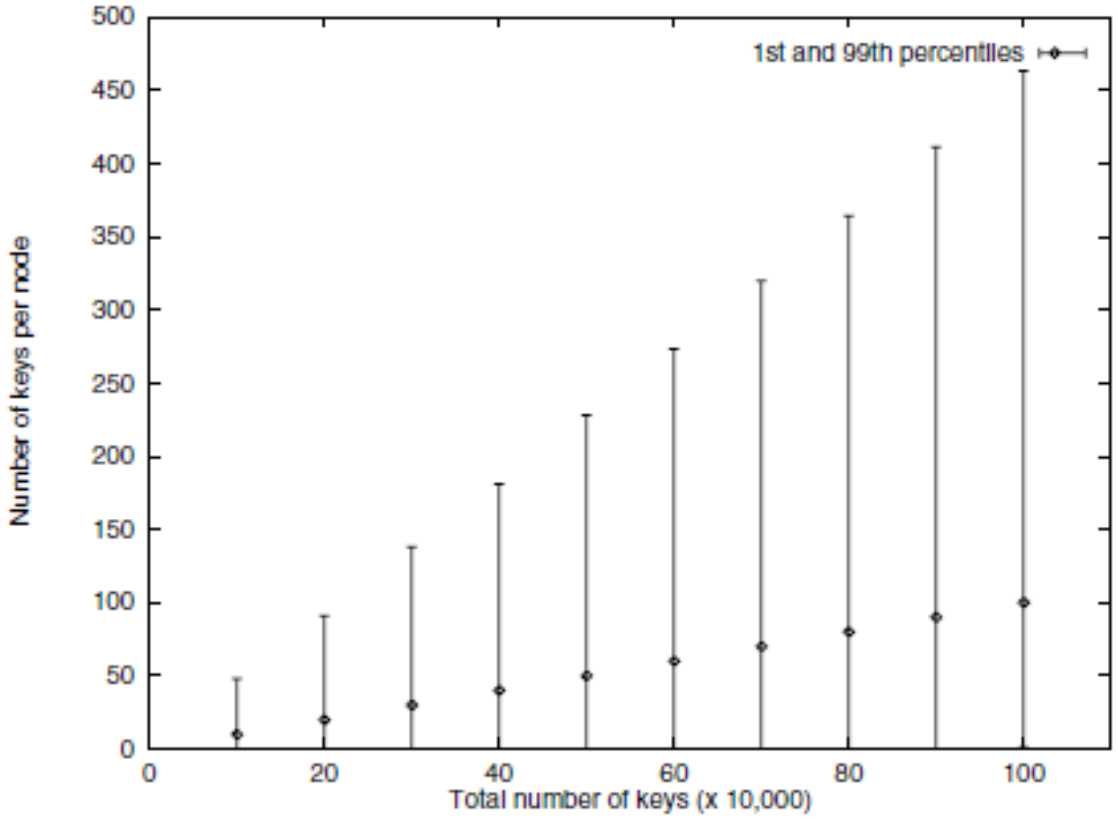
- Node - any machine running Chord software
- Successor - node with next largest ID
- Predecessor - node with next smallest ID
- Finger table - Chord routing table
  - Includes entries (“fingers”) for O(log N) other nodes
  - *k*th finger at node *n* contains the first node *s* that succeeds *n* by at least 2^{k-1}, i.e. successor($n + 2^{k-1}$)

Chord Routing Protocol

Example on blackboard!

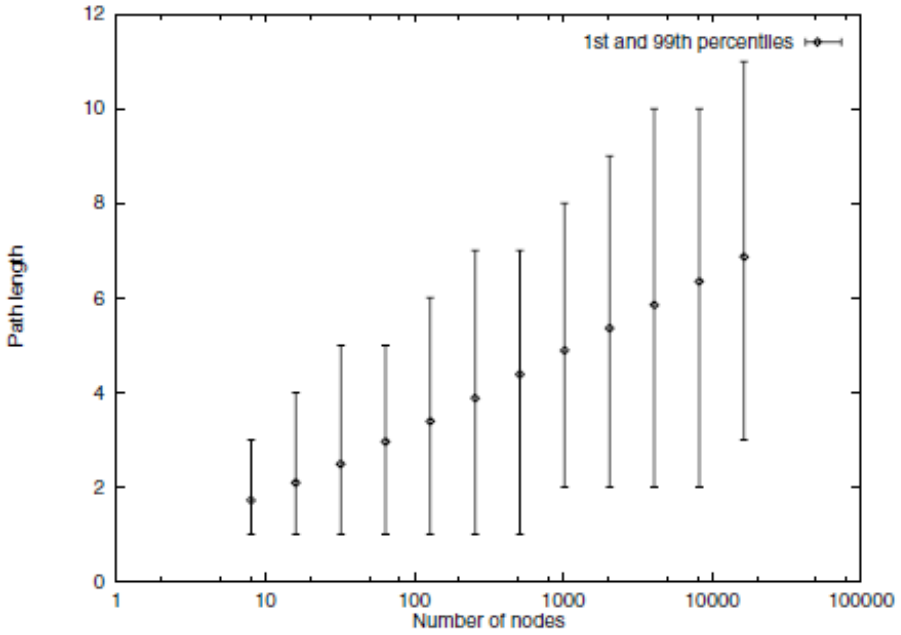
Chord Performance

Load balance in a 10^4 node network...

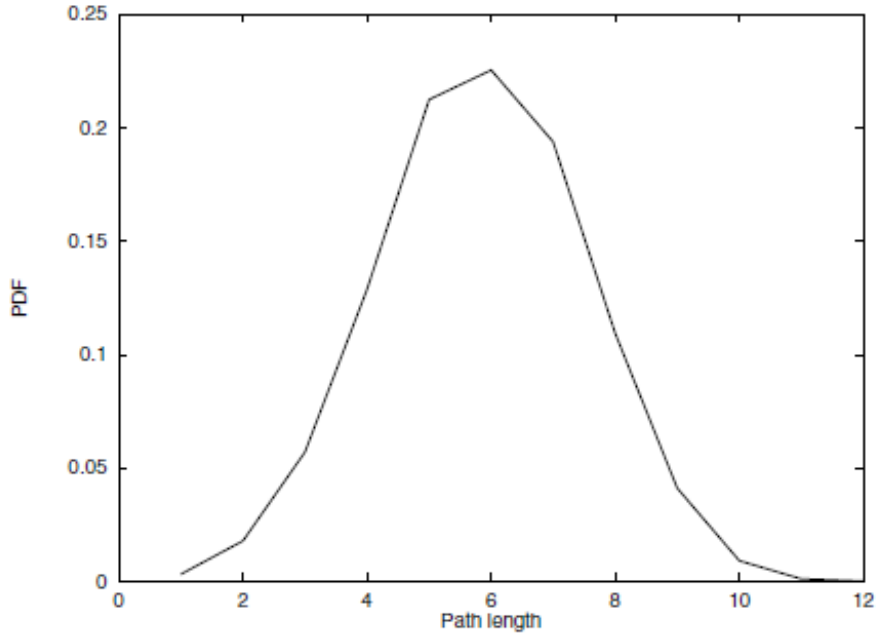


Chord Performance

Path length as a function of network size...



Path length PDF in a 2^{12} node network...



Chord Extras

What we didn’t talk about...

- Virtual nodes
- Stabilization processes
- Concurrent node joins/departures/failures

Discussion

1. How well did Chord address its 5 goals?
   - load balance
   - decentralization
   - scalability
   - availability
   - flexible naming
2. Are provably short *path lengths* enough to ensure fast routing in a WAN?

Impact of DHT Routing Geometry

- “The Impact of DHT Routing Geometry on Resilience and Proximity”
- Presented at SIGCOMM 2003

- K. Gummadi
- R. Gummadi
- S. Gribble
- S. Ratnasamy
- S. Shenkar
- I. Stoica

Routing Geometries

- Ring, e.g. Chord
- Tree, e.g. Tapestry
- Hypercube, e.g. CAN
- Butterfly, e.g. Viceroy
- XOR, e.g. Kademlia
- Hybrid (Ring+Tree), e.g. Pastry

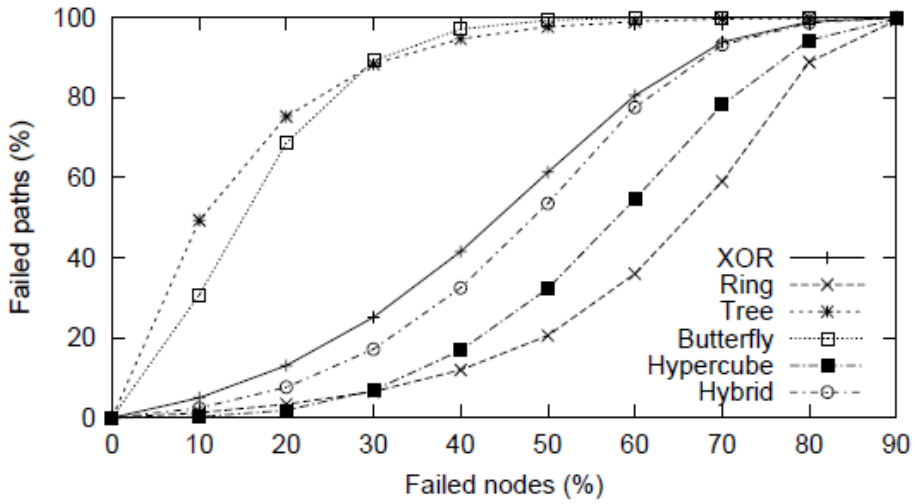
Flexibility, Resilience & Proximity

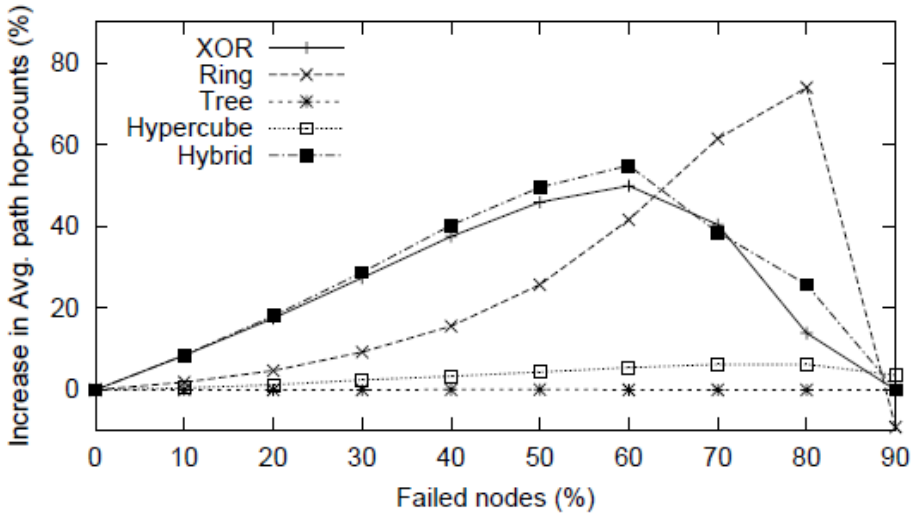
- *Flexibility* - how many different ways to route a request
  - Neighbor selection
  - Route selection
- *Resilience* - keep routing requests after nodes fail/depart
- *Proximity* - route requests through nodes that are “close together” w.r.t. some metric, e.g. network latency

Flexibility, Resilience & Proximity

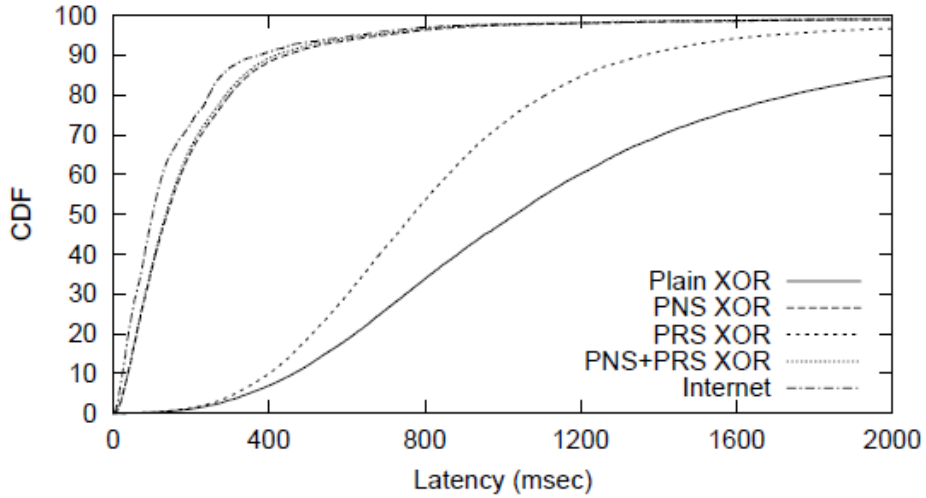
- *Flexibility* - how many different ways to route a request
  - Neighbor selection
  - Route selection
- *Resilience* - keep routing requests after nodes fail/depart
- *Proximity* - route requests through nodes that are “close together” w.r.t. some metric, e.g. network latency
- **<u>Hypothesis</u>: Greater flexibility leads to DHTs with higher resilience and better proximity of routes**

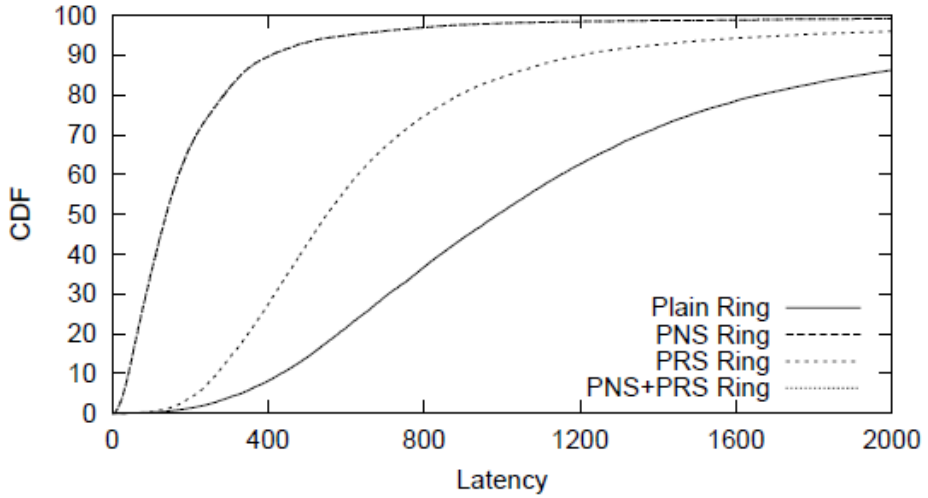
Results: Resilience



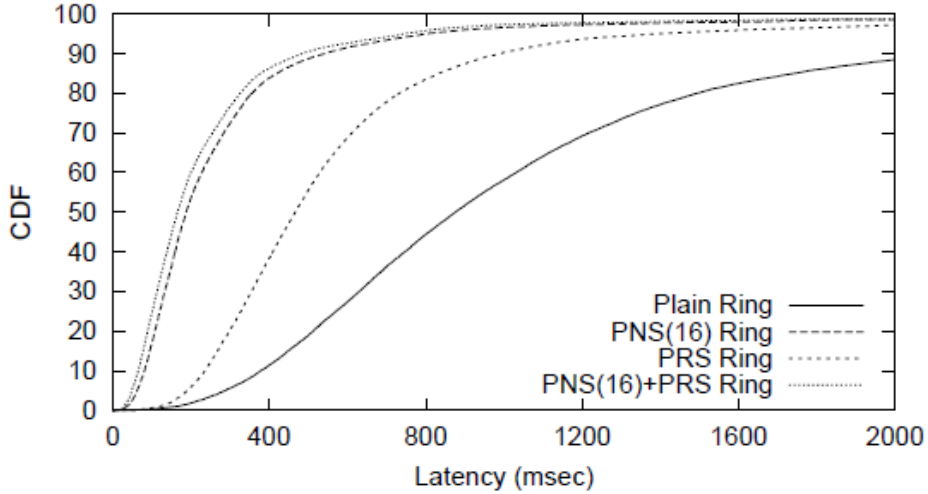


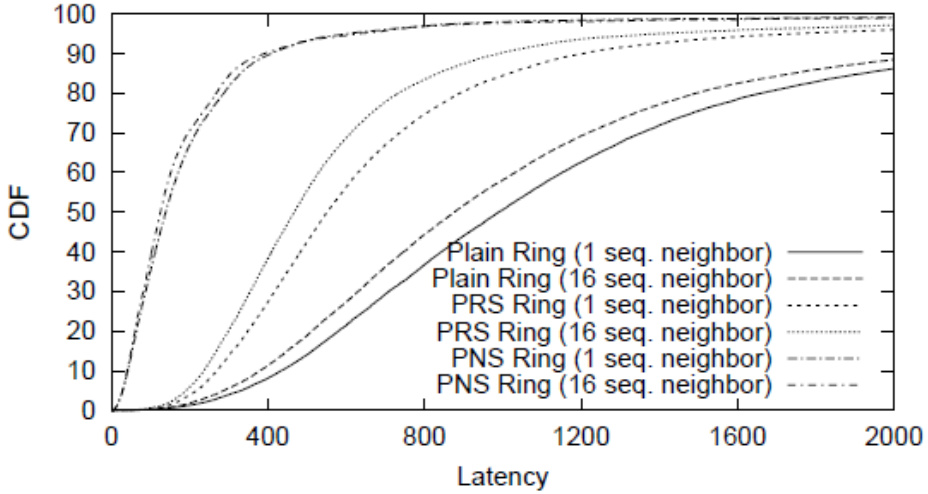
Results: Proximity





Results: Proximity cont.





Takeaways

- DHT routing geometry matters!
- Flexibility in neighbor selection is important
- Simple Ring geometry works surprisingly well