

A close-up, shallow depth-of-field photograph of a computer keyboard. The keys are dark with white markings. A black rectangular overlay is positioned on the left side of the image, containing white text. The text is arranged vertically, starting with a title and followed by four names.

CHORD-BASED DHT

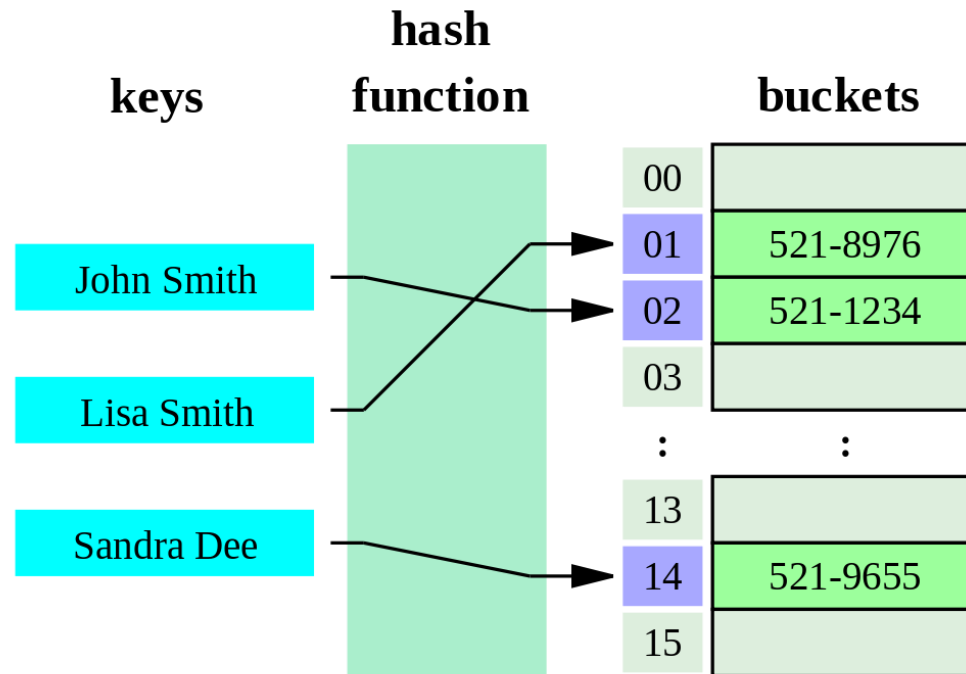
ÁLVARO ALBERO GRAN

TYLER JONES

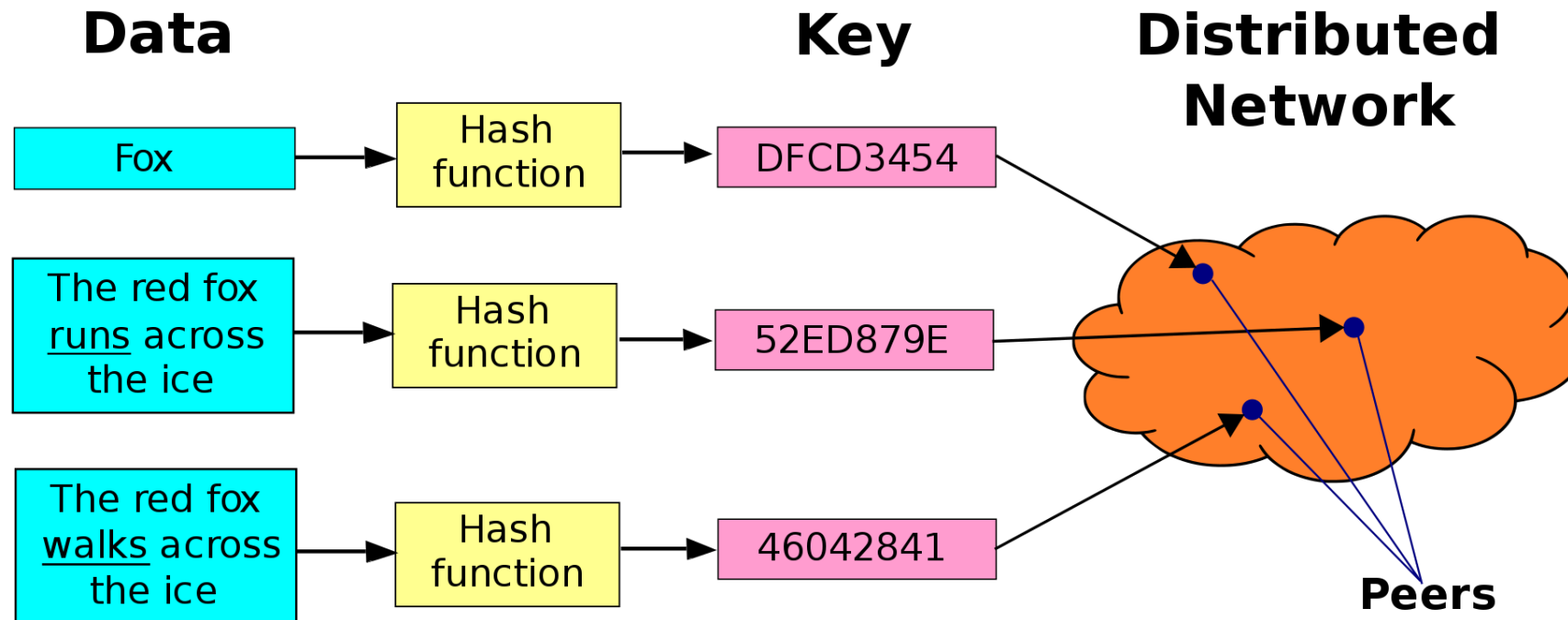
SEAN MCBRIDE

NAIM MERHEB

HASH TABLE



DISTRIBUTED HASH TABLE



CHORD OVERVIEW

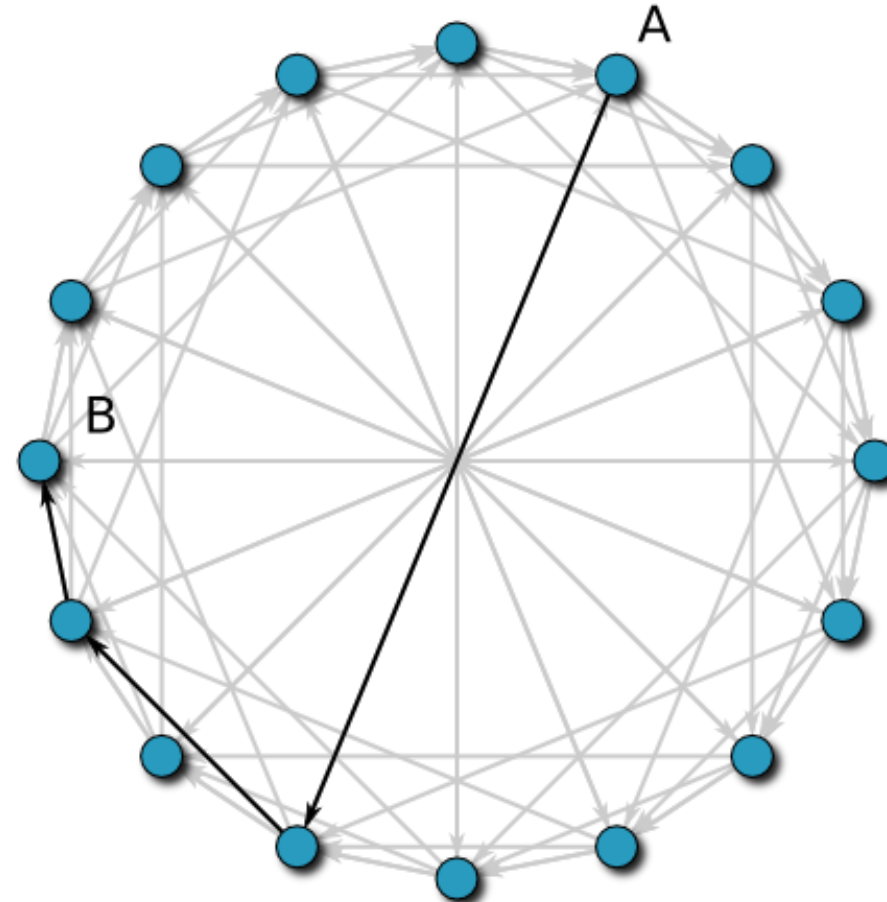
A peer-to-peer distributed hash table.

Developed by Ion Stoica, Robert Morris, David Karger, Kaashoek, and Hari Balakrishnan from MIT

Keys and IP addresses are hashed into a uniform identifier space (typically using SHA-1)

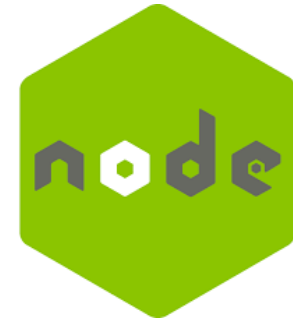
The keys of the distributed hash table are stored on their closest successor node.

Forms a logical ring with “finger tables” to reduce access time



MAJOR DESIGN DECISIONS

- Software Components
 - Node.js, our “least common denominator” language
 - gRPC, a modern and popular RPC framework
 - @grpc/proto-loader, a library capable of dynamically loading gRPC protobuf files
 - grpc-caller, a third-party gRPC library that simplified use of async/await syntax
 - vis-network, a data visualization library for graph data structures
- Collaboration Tools / Methodologies
 - Git and GitHub
 - Screen sharing via Blackboard Collaborate
 - Mob Programming via Visual Studio Code “Live Share”



RESULTS

Complete

- Nodes able to join cluster
- Able to lookup successor
- Nodes run in actual processes and form a distributed system with a real-world RPC framework
- Downloaded and cleaned k-v store data from StackOverflow

Not yet complete

- Clean and refactor JavaScript source code.
- Implement SHA-1 hashing
- Troubleshoot and allow nodes to leave Chord cluster
- Seed Chord cluster with StackOverflow data
- Ensure data migrates as expected as nodes join or leave the cluster
- Build CRUD web app to view, modify, delete StackOverflow user data and validate system transparency
- Deploy to AWS or Azure
- Enhance Admin Interface to allow scaling of Chore nodes up and down

DEMO

-
1. START FIRST NODE
 2. GROW CLUSTER
 3. WATCH CHORD
RING GROW
-



NEXT STEPS!

- Clean and refactor JavaScript source code.
- Implement SHA-1 hashing
- Troubleshoot and allow nodes to leave Chord cluster
- Seed Chord cluster with StackOverflow data
- Ensure data migrates as expected as nodes join or leave the cluster
- Build CRUD web app to view, modify, delete StackOverflow user data and validate system transparency
- Deploy to AWS or Azure
- Enhance Admin Interface to allow scaling of Chore nodes up and down

- Stretch goals: Replication, TypeScript, Shift SHA-1 Hashing to Worker Thread