



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

Egemen Berk Galatali  
[egemen.galatali@ceng.metu.edu.tr](mailto:egemen.galatali@ceng.metu.edu.tr)

Department of Computer Engineering  
Middle East Technical University  
Ankara Turkey

March 26, 2024

# Outline of the Presentation

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
  - Model, Definitions
  - Background, Previous Works
- 5 Contribution
  - Main Point 1
  - Main Point 2
  - Main Point 3
- 6 Experimental results/Proofs
  - Main Result 1
  - Main Result 2
  - Main Result 3
- 7 Conclusions

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Chord

## Distributed Key LookUp

### Distributed LookUp

A fundamental problem that confronts peer-to-peer applications is to efficiently locate the node that stores a particular data item. This paper presents Chord, a distributed lookup protocol that addresses this problem. Chord provides support for just one operation: given a key, it maps the key onto a node

# Agenda

- 1 The Problem
- 2 The Contribution**
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Chord: A Scalable Peer-to-peer Lookup Service

## Contributions and Key Ideas of the Chord Lookup Algorithm

Our contribution in this presentation is the exploration and explanation of the efficient and robust Chord Lookup Algorithm, a fundamental building block for distributed applications in dynamic peer-to-peer networks. Listed below are the key points we focus on:

- The architecture of Chord algorithm, its working principles and mechanisms.
- Analysis of why Chord offers a **practical, scalable, and robust** way of service discovery and how it addresses load balancing and node failures.
- The application of Chord in real-world scenarios like file sharing systems, distributed databases, and content delivery networks.

The main takeaway from this presentation is the understanding of how Chord efficiently solves the problem of searching for a value associated with a key in a distributed network, in  $O(\log N)$  time.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance**
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Motivation Behind the Chord Algorithm

The driving forces and importance of the Chord Lookup Algorithm

The Chord Lookup Algorithm arose from the necessity to handle complex issues faced by distributed, peer-to-peer systems. The algorithm establishes a performance-effective method to handle service location in dynamic network environments.



# Motivation Behind the Chord Algorithm

The driving forces and importance of the Chord Lookup Algorithm

- **Scalability:** Large peer-to-peer networks face scalability issues. Chord, with its  $O(\log N)$  lookup time, offers a scalable solution.
- **Robustness:** Chord robustly handles node failures and sudden departures, preserving the integrity of the network despite such events.
- **Distributed Hashing:** Chord offers a distributed hashing method that evenly distributes load across nodes.
- **Real-world Applications:** Chord has broad applicability, including file sharing systems, distributed databases, and content delivery networks.

Operating in the larger picture of improving efficiency and robustness in peer-to-peer network systems, Chord thus helps address the everyday pain points in managing and navigating massive distributed systems.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works**
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions

# Model, Definitions

Formal definition of the Chord Lookup Algorithm and its Components

- **Chord:** A protocol that maps keys to nodes in a distributed network.
- **Node:** A machine or process participating in the Chord network.
- **Key:** An entity in the network that is assigned to a node.
- **Successor:** The node that follows a given node in the Chord ring, responsible for storing keys that hash to values up to its own hash value.
- **Predecessor:** The node that directly precedes a given node in the Chord ring.
- **Finger Table:** A routing table stored at each node that points to various nodes in the Chord ring for expedited lookups.

# Background - Previous Works

Several research works and protocols have explored distributed, peer-to-peer lookup services before the advent of Chord. Here are brief summaries about these distinct protocols:

- **CAN:** A scalable, distributed infrastructure that provides hash table-like functionality on Internet-like scales.
- **Tapestry:** A peer-to-peer overlay routing infrastructure offering efficient, scalable, location-independent routing of messages directly to nearby copies of an object or service.
- **Pastry:** A scalable, decentralized object location and routing substrate for wide-area peer-to-peer applications, with a strong emphasis on fault tolerance.

Each protocol offered potential solutions, with their own strengths and weaknesses.

# Background - Chord Protocol

The Chord algorithm distinguishes itself from its predecessors with its simplicity, provable correctness, and provable performance.

- It offers a solution for efficient key lookups in a distributed network with  $O(\log N)$  time complexity.
- Each node in the Chord protocol maintains minimal state information - only  $O(\log N)$  pointers.
- The protocol exhibits resilience even while dealing with node failures or network changes.

This makes the Chord algorithm a significant departure from previous works, offering substantial potential in the field of large, dynamic peer-to-peer networks.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution**
- 6 Experimental results/Proofs
- 7 Conclusions

# Main Point 1: Chord Lookup Algorithm

## Handling Key Lookup in a Distributed Network

The Chord Lookup Algorithm offers several key contributions in managing large-scale distributed networks:

- **Simplicity:** Chord simplifies resource location in p2p networks using identifiers for nodes and keys.
- **Scalability:** The algorithm scales logarithmically with nodes, enabling efficient routing.
- **Robustness:** Chord maintains effective key lookups even under node join and leave operations.
- **Fault Tolerance:** Chord operates efficiently even when certain nodes fail, thanks to successor lists.

# Chord Lookup Algorithm

## Efficient Key Lookup in Peer-to-Peer Networks

The Chord algorithm for efficient key lookup in peer-to-peer networks is shown in Algorithm 1.

**Implements:** ChordLookup **Instance:** cf

**Uses:** DistributedOverlayNetwork **Instance:** don

**Events:** Join, Stabilize, Lookup

**Needs:**

**OnJoin:** (node  $n$ ) do

1: "Find node's position in the network and transfer keys."

**OnStabilize:** () do

2: "Periodically update finger tables and successor lists."

**OnLookup:** ( key  $k$  ) do

3: "Find node responsible for key."

**Algorithm 1:** ChordLookup algorithm



# Main Point 2

## Explain the Significance of the Results

Pause, and explain the relationships between the formal theorems that you have just presented and the informal description that you gave in the Introduction. Make it clear to the audience that the results do live up to the advance publicity. If the statements of the theorems are very technical then this may take some time. It is time well-spent.

# Main Point 3

## Sketch a Proof of the Crucial Results

The emphasis is on the word “sketch”. Give a very high-level description of the proofs, emphasizing the proof structure and the proof techniques used. If the proofs have no structure (in which case it may be assumed that you are not the author of the paper), then you must impose one on them. Gloss over the technical details. It is a good idea to point them out but not to explore them.

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs**
- 7 Conclusions

# Main Result 1

Choose **just the key results**. They should be important, non-trivial, should give the flavour of the rest of the technical details and should be presentable in a relatively short period of time. Use figures instead of tables instead of text.

Better to present 10% the entire audience gets than 90% nobody gets

# Main Result 2

Try a subtitle

- Make sure your notation is clear and consistent throughout the talk. Prepare a slide that explains the notation in detail, in case that is needed or if somebody asks.
- Always label all of your axes on graphs; use short but helpful captions on figures and tables. It is also very useful to have an arrow on the side which clearly shows which direction is considered better (e.g., "up is better").
- If you have experimental results, make sure you clearly present the experimental paradigm you used, and the details of your methods, including the number of trials, the specific analysis tools you applied, significance testing, etc.
- The talk should contain at least a brief discussion of the limitations and weaknesses of the presented approach or results, in addition to their strengths. This, however, should be done in an objective manner – don't enthusiastically put down your own work.

# Main Result 3

- If time allows, the results should be compared to the most related work in the field. You should at least prepare one slide with a summary of the related work, even if you do not get a chance to discuss it. This will be helpful if someone asks about it, and will demonstrate your mastery of the material.
- Spell check again.
- Give for each of the x-axis, y-axis, and z-axis
- Label, unit, scale (if log scale)
- Give the legend
- Explain all symbols
- Take an example to illustrate a specific point in the figure

# Agenda

- 1 The Problem
- 2 The Contribution
- 3 Motivation/Importance
- 4 Background/Model/Definitions/Previous Works
- 5 Contribution
- 6 Experimental results/Proofs
- 7 Conclusions**

# Conclusions

Hindsight is Clearer than Foresight

Advices come from [1].

- You can now make observations that would have been confusing if they were introduced earlier. Use this opportunity to refer to statements that you have made in the previous three sections and weave them into a coherent synopsis. You will regain the attention of the non- experts, who probably didn't follow all of the Technicalities section. Leave them feeling that they have learned something nonetheless.
- Give Open Problems It is traditional to end with a list of open problems that arise from your paper. Mention weaknesses of your paper, possible generalizations, and indications of whether they will be fruitful or not. This way you may defuse antagonistic questions during question time.
- Indicate that your Talk is Over An acceptable way to do this is to say "Thank-you. Are there any questions?"[2]



# References

- [1] B. Spillman and I. Parberry, "How to present a paper: A speaker's guide," *North American Membrane Society*, 2000.
- [2] A. Einstein, "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]," *Annalen der Physik*, vol. 322, no. 10, pp. 891–921, 1905.

# How to prepare the talk?

Please read <http://larc.unt.edu/ian/pubs/speaker.pdf>

- The Introduction: Define the Problem, Motivate the Audience, Introduce Terminology, Discuss Earlier Work, Emphasize the Contributions of your Paper, Provide a Road-map.
- The Body: Abstract the Major Results, Explain the Significance of the Results, Sketch a Proof of the Crucial Results
- Technicalities: Present a Key Lemma, Present it Carefully
- The Conclusion: Hindsight is Clearer than Foresight, Give Open Problems, Indicate that your Talk is Over

# Questions

THANK YOU

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

presented by Egemen Berk Galatali  
[egemen.galatali@ceng.metu.edu.tr](mailto:egemen.galatali@ceng.metu.edu.tr)



March 26, 2024

