# Chord Protocol

## Team Members

| Name | UFID |
|---|---|
| Karthik Sudhakar Kashyap | 41120720 |
| Akshay Adithya | 76680818 |

## What is working?

We have implemented the peer-to-peer lookup service using the Chord protocol as described in the publication

### Network Construction

- We construct the network using the method described in the paper, where each node identifier is hashed and placed on the network.
- Each node has a finger-table that has a reference to its successor nodes, which is calculated using the formula:

$$fingerTable[k] = n + 2^{(k-1)}$$

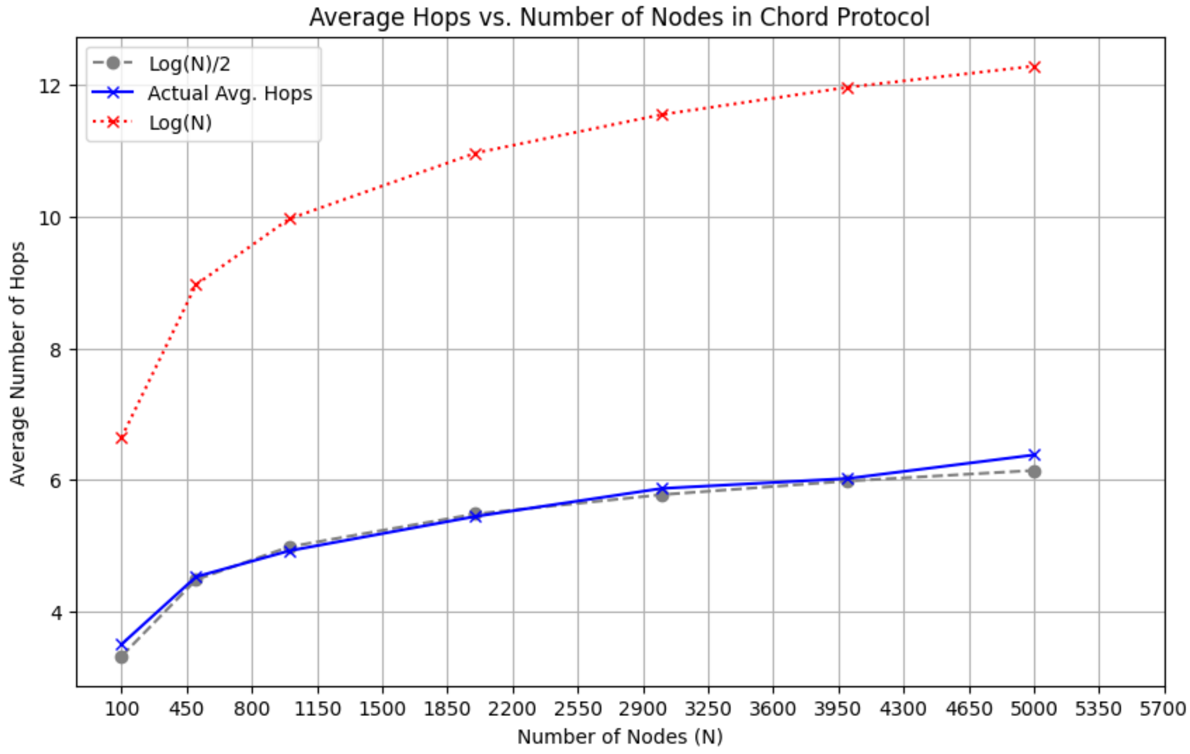
- Each node also has a reference to its predecessor node, which helps identify when the key is present with the current node
- Each node in the network is an independent actor (Pony) acting as a single unit of concurrency

## Key Lookup

- The program generates unique random keys based on user input every second using the Fisher-Yates algorithm
- Each node in the network is asked to find each of the keys generated
- Based on the Chord protocol, the node forwards the lookup or returns the key if present
- Upon successful lookup, the nodes notify the main actor with the number of hops.
- The main actor computes the average hops required to perform the lookup

## Average & Worst Case Hops

- According to the Section V of the paper, based on their experimental trials it is observed that the average lookup time is - $\frac{1}{2}\log(n)$
- The paper also talks about how the worst case lookup would not exceed - $log(n)$
- Our results aligned with the experimental trials conducted by the authors of the paper



## What is the largest network you managed to deal with?

- We were able to test on a network size of **80,000 nodes** for **10 unique message requests** and acheived average hops of **8.40202**
- This limit is purely a hardware bottleneck and the algorithm is capable of handling much larger requests.