chord-url-shortening

Building a Scalable and Fault-Tolerant URL Shortening Service

50.041 Distributed Systems & Computing

About TinyURL

1. A service that shortens long, unwieldy URLs into concise URLs
2. Shortened URLs redirect users to the original URL seamlessly

The Problem with a centralized TinyURL

Challenges

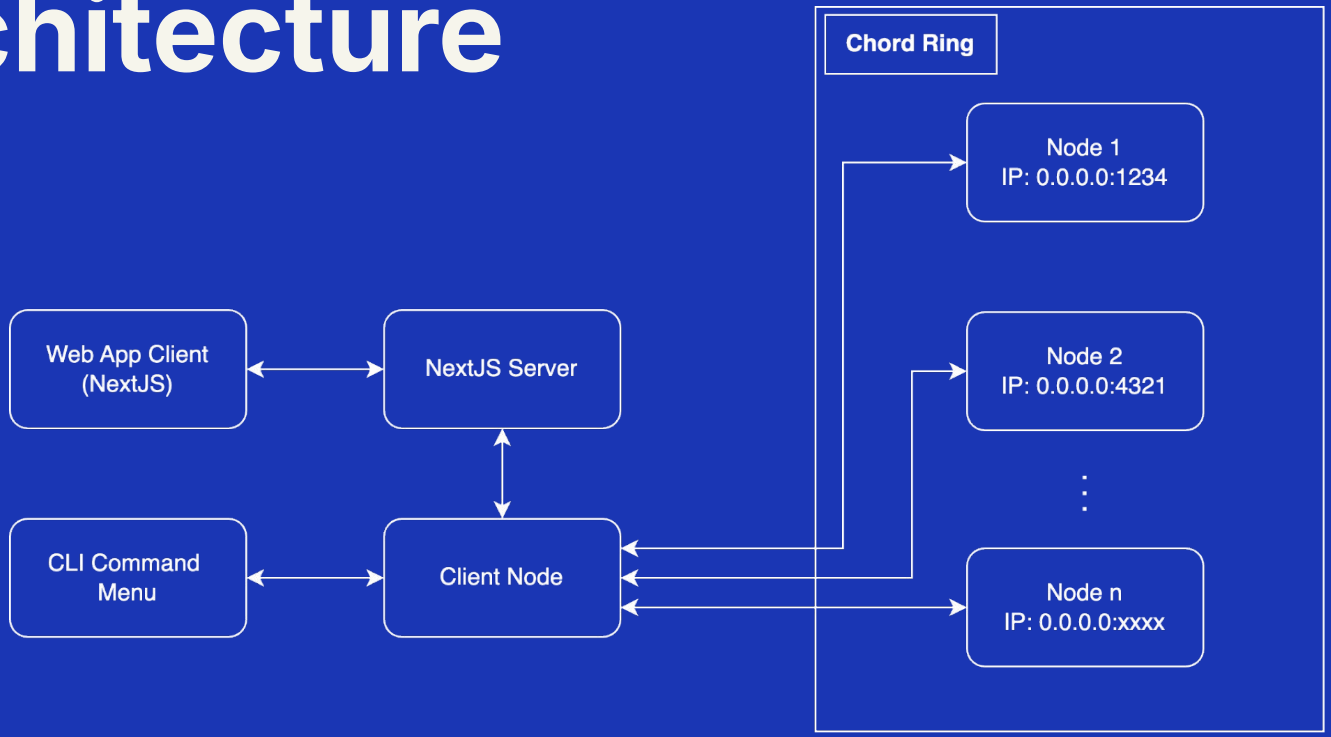
- Performance bottlenecks
- Scalability constraints
- Reliability risks

Solution: chord-url-shortening

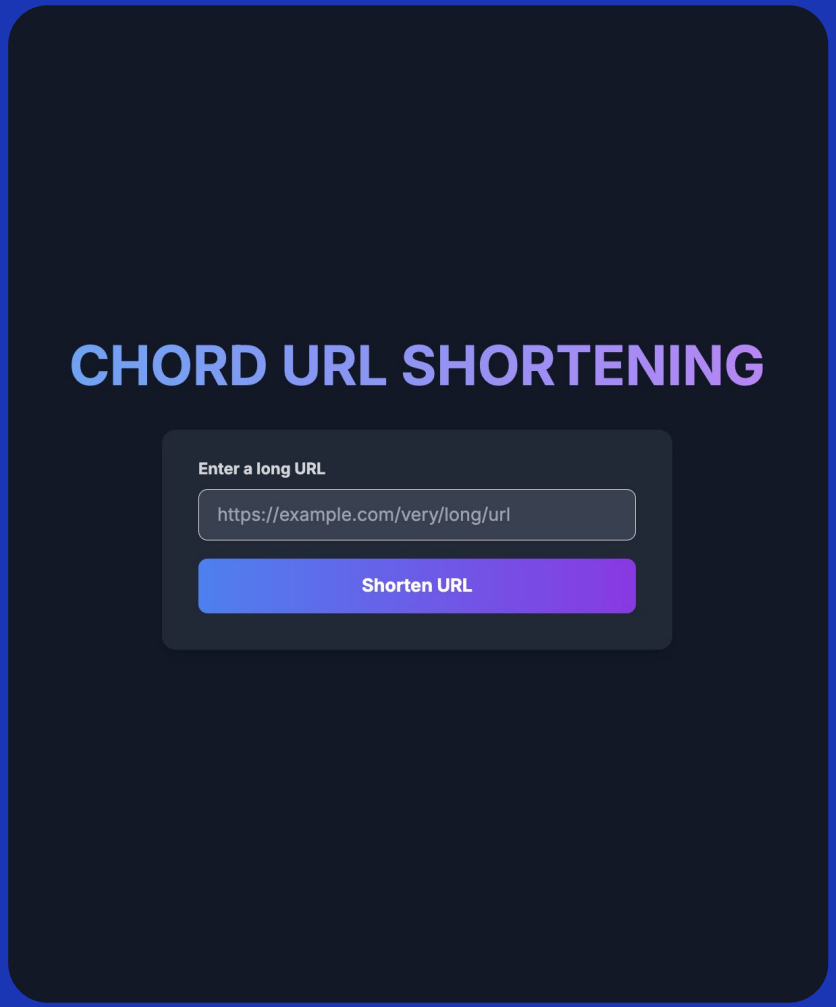
Chord-Inspired Distributed URL shortening

- Scalable lookup protocol using distributed hash tables (DHTs)
- Ensures even distribution of key-value pairs (short URL→long URL)
- Fault tolerance using successor lists

Architecture

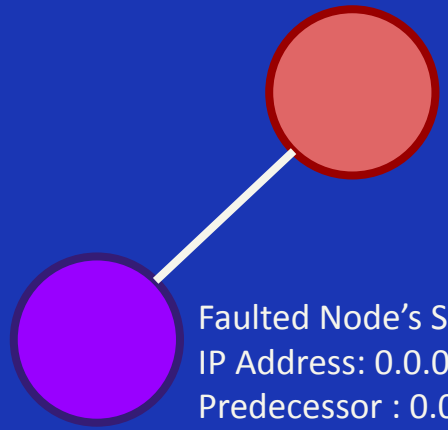


Demo



Permanent Faults

Before Successor Node Detects Failure:



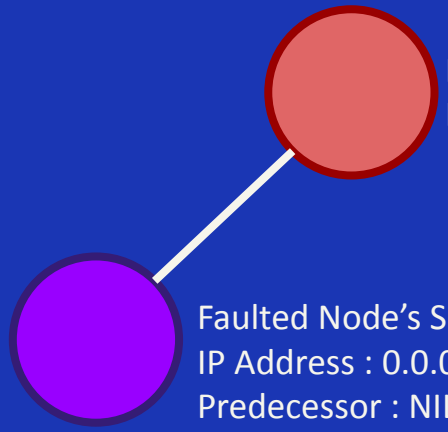
Faulted Node :
IP Address: 0.0.0.0:1115

Faulted Node's Successor :
IP Address: 0.0.0.0:1116
Predecessor : 0.0.0.0:1115

Current URLMap:

0.0.0.0:1115	longURL : "www.1115.com", shortURL : er315es315
0.0.0.0:1116	longURL : "www.1116.com", shortURL : fsroth2o43

After Successor Node Detects Failure:



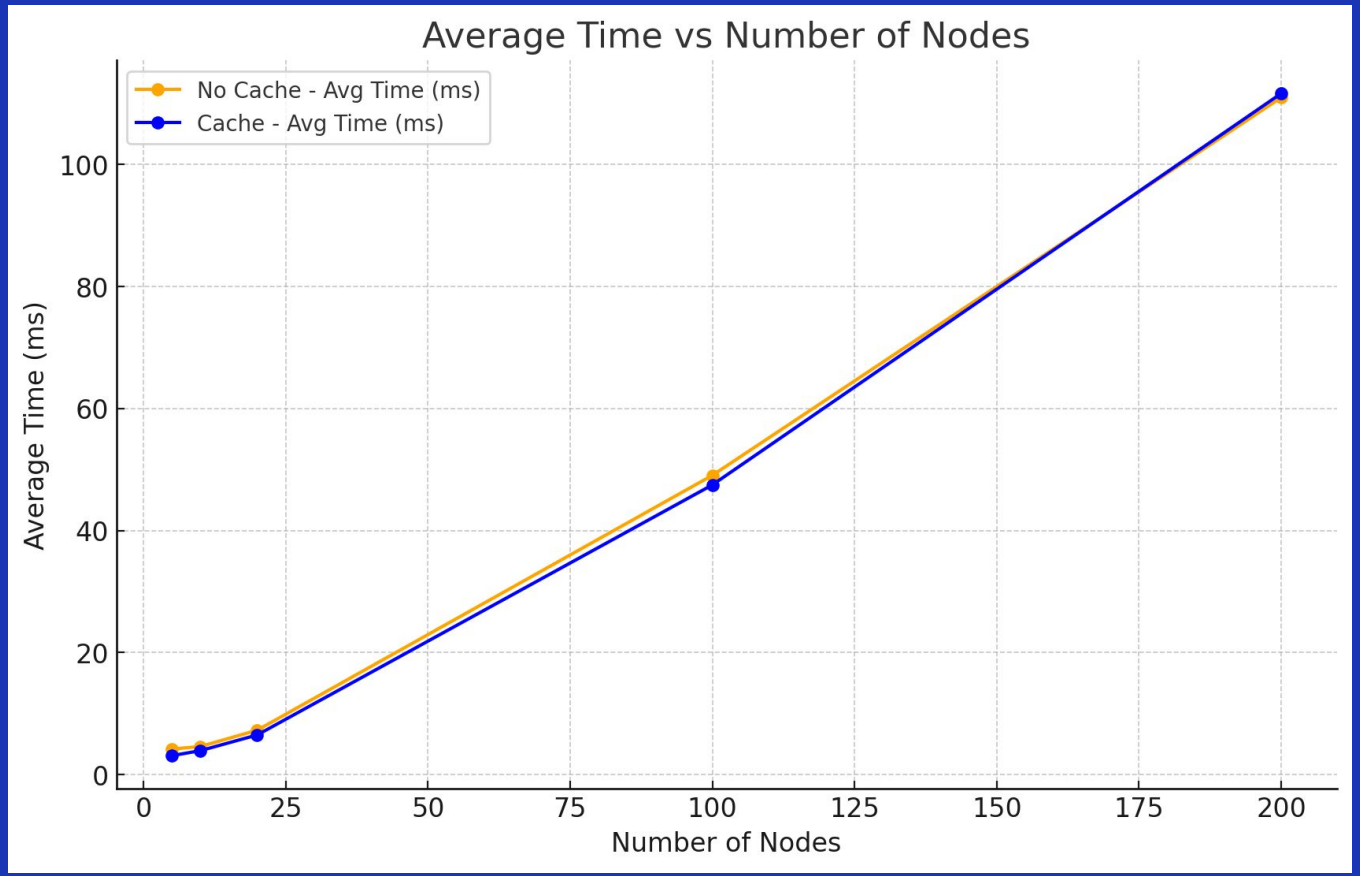
Faulted Node :
IP Address: 0.0.0.0:1115

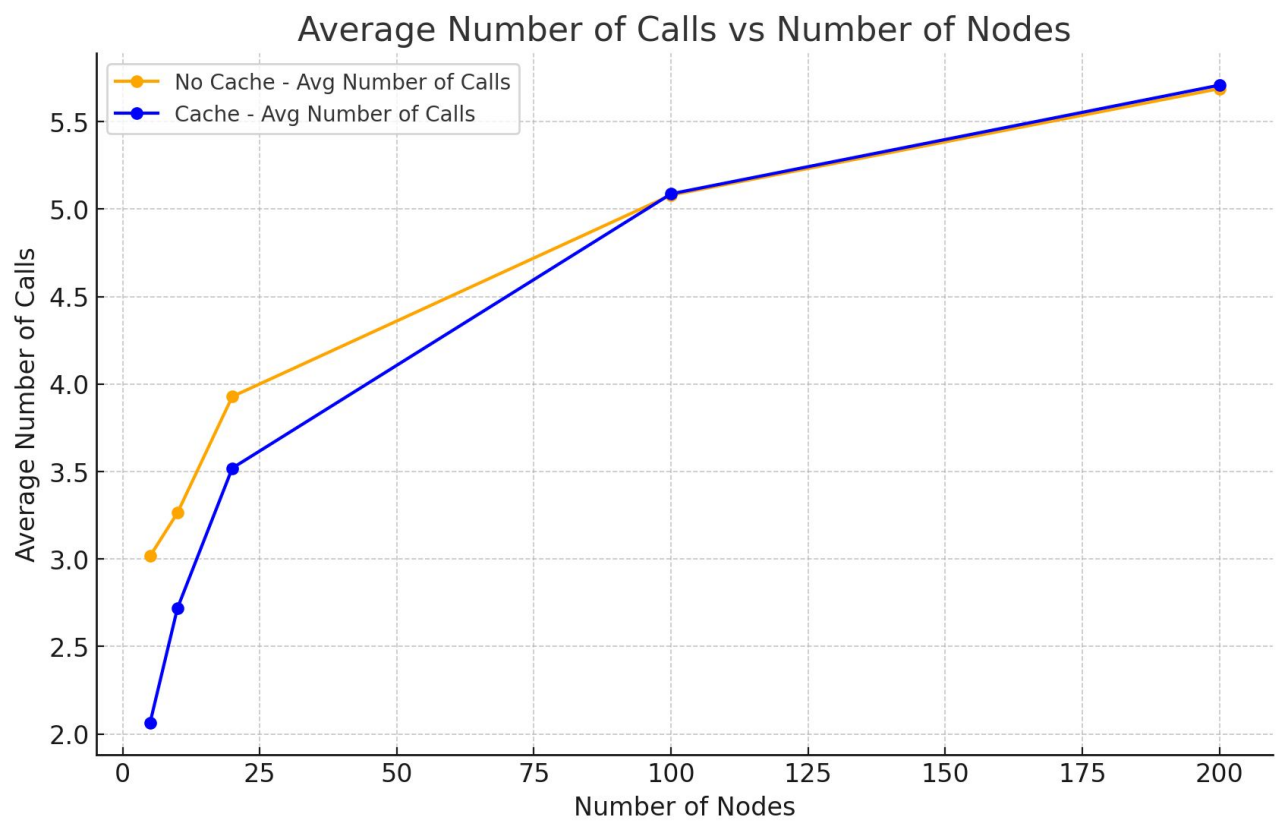
Faulted Node's Successor
IP Address : 0.0.0.0:1116
Predecessor : NIL

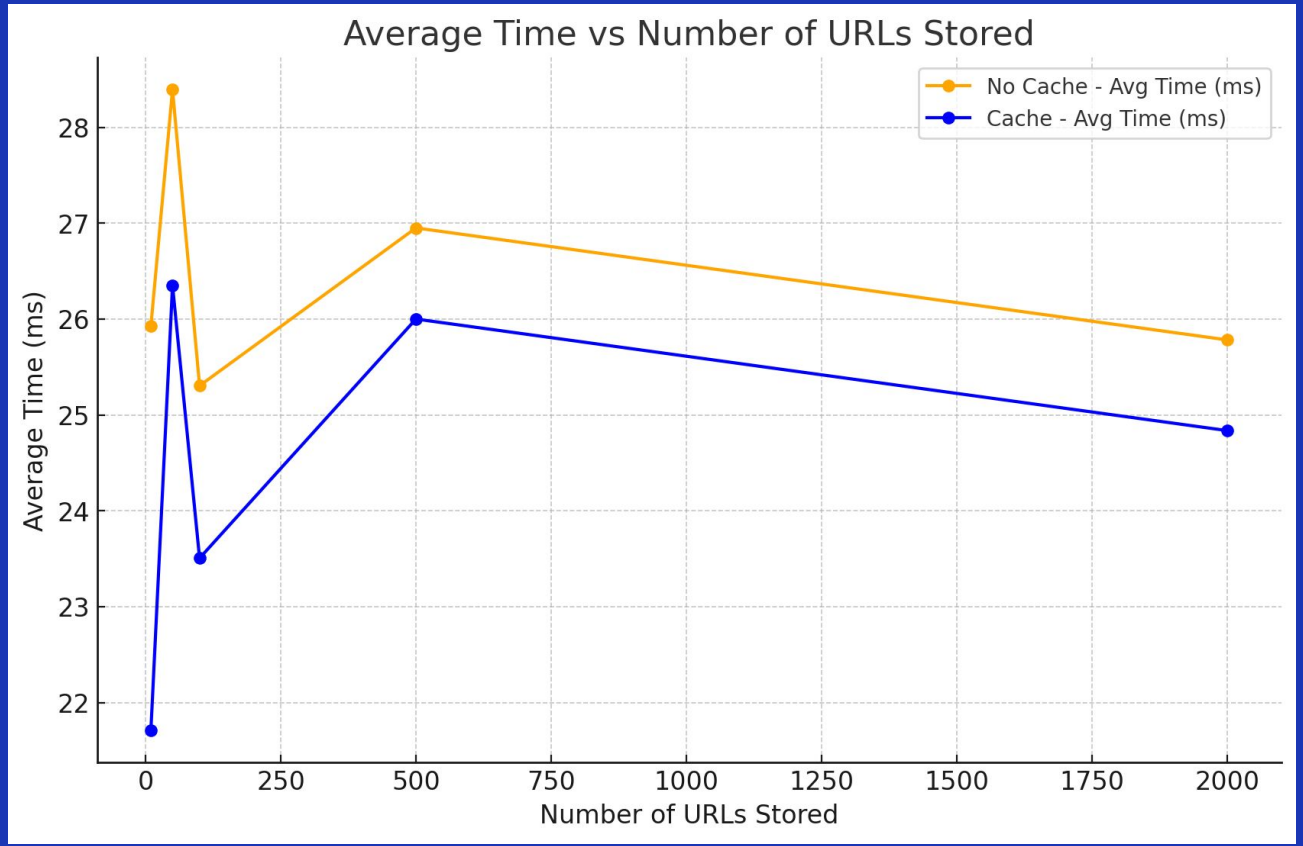
Current URLMap:

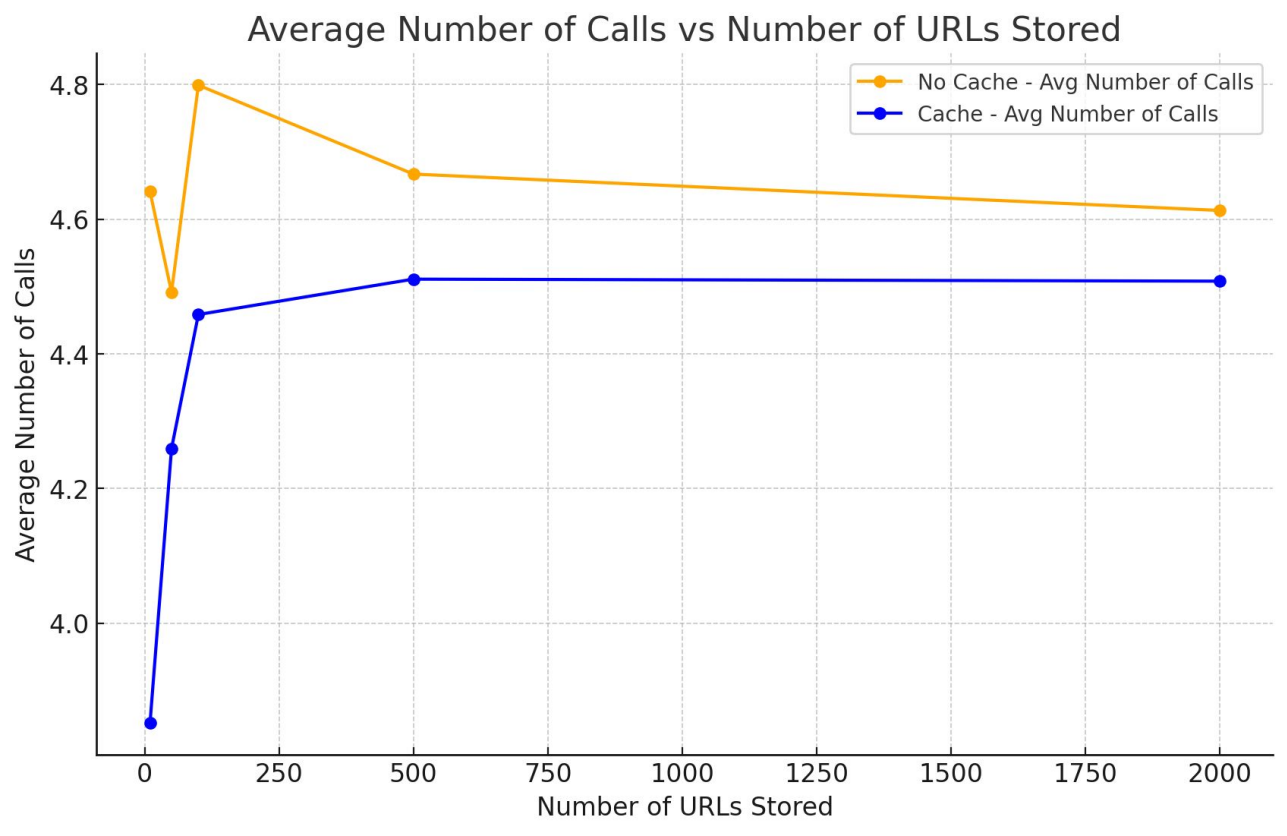
0.0.0.0:1116	longURL : "www.1115.com", shortURL : er315es315
	longURL : "www.1116.com", shortURL : fsroth2o43

Experiments









Thanks