

COP5615 – Project3

Neel Rami, UFID: 7712-3151

Ma Haodi, UFID:7719-2198

Email: nrami@ufl.edu , ma.haodi@ufl.edu

Oct 1, 2018

1. Introduction

The goal of this project is to implement in Elixir using the actor model the Chord protocol and a simple object access service to prove its usefulness.

In this project we implement the network join and routing as described in the Chord paper (Section 4) and encode the simple application that searches a key (same as the ids used in Chord). We implement it using a similar API to the one described in the paper. The result and findings will be shown in this report.

Assumptions made:

- We have built a simple application of doing a search for a key. This key is randomly generated, and we are not associating this key with a string just for simplicity.
- We are not storing keys in a node, but we assume that all keys from predecessor of a certain node to that node are stored in a node. For example, if there is a node with ID 85 and its predecessor is 51, then all keys from 52 to 85 are assumed to be in node with ID 85.
- In rare cases during network Join, the process of stabilization doesn't get completed due to some reasons which can lead the program to go into a infinite loop. So in those cases, please run the program again.

2. Implementation details

a) File explanation:

1) GenServerMethod.ex:

This file contains methods which are related to GenServer and it contains all methods which are essential for Chord Protocol.

2) proj3.ex:

This file serves as entry point of the project.

3) utilityFunctions.ex:

This file contains some utility functions which are used frequently.

b) Approach used:

1) Calculation of Node IDs:

- We have used the concept of consistent hashing to calculate the identifier for every node.
- We have used the `:crypto.hash(:sha "nodeID")` function to calculate SHA1.
- We calculate the hash of Node Index(i.e. 1,2,3...) to calculate the hash. Then we encode it to BASE 16 and truncate it to m bits. And then we convert the truncated result to integer and this integer serves as the node identifier for a node.

2) Calculation of Table size(m):

- We dynamically calculate the table size based on the number of nodes.
- We calculate $\log_2(\text{numNodes})$ to calculate the least number of bits required to represent a node and then we convert the logarithmic result to the nearest multiple of 4.

3) Network Join and Routing:

- We have implemented the functionality of Network Join by creating a Chord Ring of n-1 nodes and then join the remaining node.
- The functionalities of Network Join and Routing are implemented based on the explanation given in the research paper.

c) Instructions for running the code:

1) For Ubuntu based systems:

1. Go to the project directory
2. Type the command in the terminal: `mix escript.build` (Optional)
3. Type the command in the terminal: `./proj3 200 1`
4. Here the first command line argument is the number of nodes
5. Here the second command line argument is the number of requests
6. General command: `./projec3 <node-num> <numRequests>`

2) For Windows:

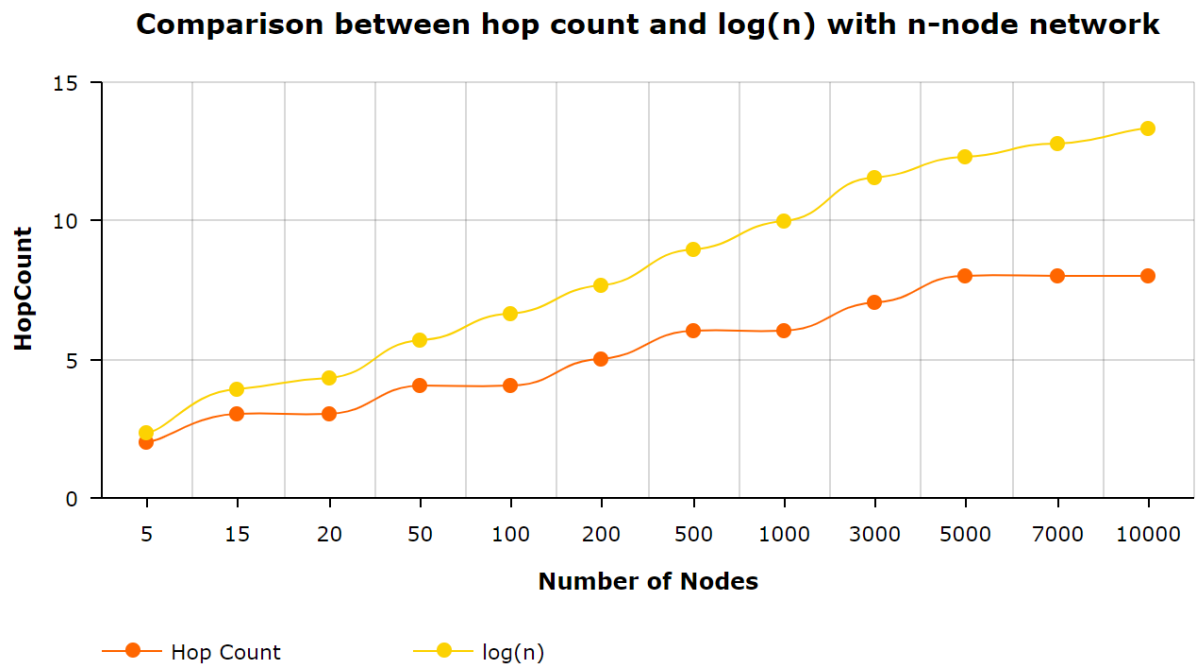
1. Go to the project directory
2. Type the command in the cmd: `mix escript.build` (Optional)
3. Type the command in the cmd: `escript .\proj3 200 1`
4. Here the first command line argument is the number of nodes
5. Here the second command line argument is the number of requests
6. General command: `escript .\projec3 <node-num> <numRequests>`

3. Result

Number of nodes	Ceil(Hop Count)
5	2
15	3
20	3
50	4
100	4
200	5
500	6
1000	6
3000	7
5000	8
10000	8*

Table 1: Number of request = 1

*: Takes a lot of time

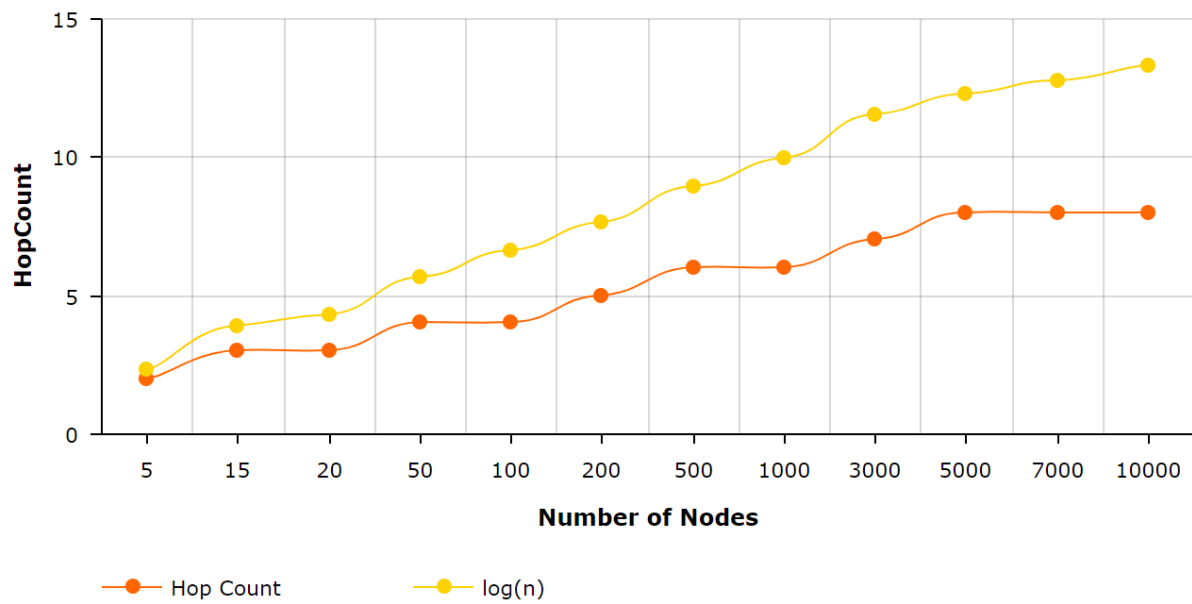


Number of nodes	Ceil(Hop Count)
5	2
15	3
20	3
50	4
100	5
200	5
500	6
1000	6
3000	7
5000	7
10000	8*

Table 2: Number of request = 2

*: Takes a lot of time

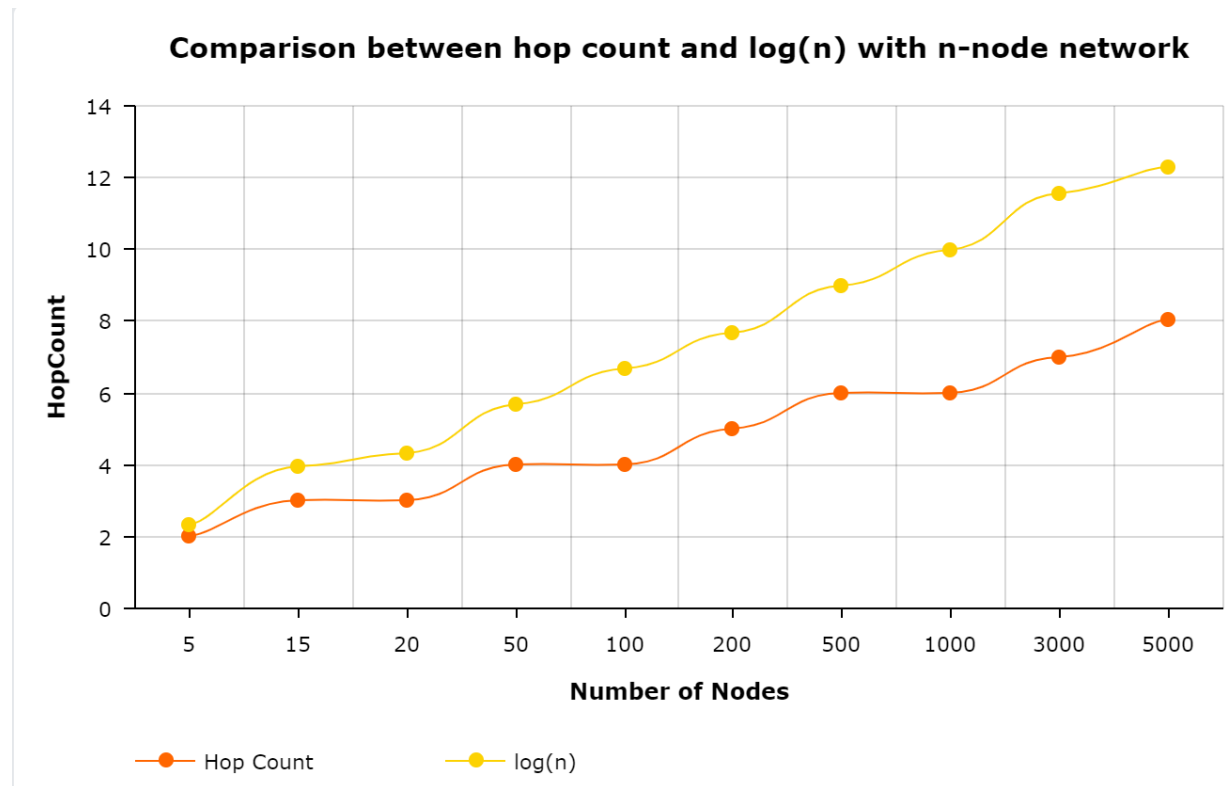
Comparison between hop count and $\log(n)$ with n-node network



Number of nodes	Ceil(Hop Count)
5	2
15	3
20	3
50	4
100	5
200	5
500	6
1000	6
3000	7
5000	7*

Table 3: Number of request = 5

*: Takes a lot of time



4. Output

1. Output for numNodes=20 and numRequests=1

```
neel@nrami: ~/Desktop/DOS/Projects/Project3/proj3
"Key 243 Found."
"Key 58 Found."
"Key 120 Found."
"Key 213 Found."
"Key 147 Found."
"Key 7 Found."
"Key 240 Found."
"Key 21 Found."
"Key 178 Found."
"Key 97 Found."
"Hopcount for Key 243 is 1."
"Hopcount for Key 58 is 1."
"Hopcount for Key 213 is 2."
"Hopcount for Key 120 is 2."
"Hopcount for Key 7 is 3."
"Hopcount for Key 21 is 2."
"Hopcount for Key 240 is 2."
"Hopcount for Key 147 is 2."
"Hopcount for Key 27 is 3."
"Hopcount for Key 178 is 3."
"Hopcount for Key 97 is 1."
"Key 89 Found."
"Key 207 Found."
"Key 122 Found."
"Key 18 Found."
"Key 223 Found."
"Key 75 Found."
"Key 188 Found."
"Hopcount for Key 89 is 2."
"Hopcount for Key 207 is 1."
"Hopcount for Key 122 is 2."
"Hopcount for Key 18 is 2."
"Hopcount for Key 223 is 1."
"Hopcount for Key 188 is 3."
"Hopcount for Key 75 is 1."
"Key 118 Found."
"Hopcount for Key 118 is 1."
"Key 107 Found."
"Hopcount for Key 107 is 2."
"----- AVERAGE HOP COUNTS -----"
3
neel@nrami:~/Desktop/DOS/Projects/Project3/proj3$
```

2. Output for numNodes=50 and numRequests=1

```
neel@nrami: ~/Desktop/DOS/Projects/Project3/proj3
"Hopcount for Key 3446 is 3."
"Hopcount for Key 3343 is 5."
"Hopcount for Key 2411 is 4."
"Hopcount for Key 2596 is 2."
"Hopcount for Key 3785 is 4."
"Hopcount for Key 1030 is 2."
"Hopcount for Key 1702 is 3."
"Key 770 Found."
"Key 3891 Found."
"Hopcount for Key 2929 is 3."
"Hopcount for Key 3226 is 4."
"Hopcount for Key 1940 is 4."
"Key 3462 Found."
"Key 3309 Found."
"Key 2349 Found."
"Key 2755 Found."
"Key 3617 Found."
"Key 998 Found."
"Key 1681 Found."
"Hopcount for Key 770 is 3."
"Key 2041 Found."
"Hopcount for Key 3891 is 4."
"Hopcount for Key 3462 is 3."
"Hopcount for Key 3309 is 5."
"Hopcount for Key 2349 is 5."
"Hopcount for Key 2755 is 4."
"Hopcount for Key 3617 is 2."
"Hopcount for Key 998 is 1."
"Hopcount for Key 1681 is 2."
"Hopcount for Key 2041 is 3."
"Key 3271 Found."
"Key 2700 Found."
"Key 1075 Found."
"Key 1851 Found."
"Hopcount for Key 3271 is 4."
"Hopcount for Key 2700 is 4."
"Hopcount for Key 1851 is 3."
"Hopcount for Key 1075 is 2."
"Key 3395 Found."
"Hopcount for Key 3395 is 5."
"----- AVERAGE HOP COUNTS -----"
4
neel@nrami:~/Desktop/DOS/Projects/Project3/proj3$
```

3. Output for numNodes=500 and numRequests=1

```
neel@nrami: ~/Desktop/DOS/Projects/Project3/proj3
"Hopcount for Key 613182 is 6."
"Hopcount for Key 203787 is 7."
"Hopcount for Key 191258 is 5."
"Hopcount for Key 311336 is 4."
"Hopcount for Key 620668 is 5."
"Key 1027741 Found."
"Hopcount for Key 932380 is 5."
"Hopcount for Key 701634 is 4."
"Key 152093 Found."
"Key 157979 Found."
"Key 366544 Found."
"Key 1034177 Found."
"Hopcount for Key 758645 is 5."
"Key 633906 Found."
"Key 425756 Found."
"Key 207661 Found."
"Hopcount for Key 1027741 is 7."
"Hopcount for Key 152093 is 5."
"Key 699890 Found."
"Key 928756 Found."
"Hopcount for Key 157979 is 5."
"Hopcount for Key 366544 is 4."
"Hopcount for Key 1034177 is 7."
"Hopcount for Key 633906 is 5."
"Hopcount for Key 425756 is 5."
"Hopcount for Key 207661 is 6."
"Key 1029308 Found."
"Hopcount for Key 699890 is 5."
"Hopcount for Key 928756 is 6."
"Key 635016 Found."
"Key 423164 Found."
"Hopcount for Key 1029308 is 6."
"Key 932242 Found."
"Hopcount for Key 635016 is 5."
"Hopcount for Key 423164 is 4."
"Hopcount for Key 932242 is 6."
"Key 424464 Found."
"Key 938903 Found."
"Hopcount for Key 424464 is 6."
"Hopcount for Key 938903 is 3."
"----- AVERAGE HOP COUNTS -----"
6
neel@nrami:~/Desktop/DOS/Projects/Project3/proj3$
```

4. Output for numNodes=500 and numRequests=5

```
neel@nrami: ~/Desktop/DOS/Projects/Project3/proj3
"Key 295897 Found."
"Hopcount for Key 638301 is 5."
"Key 148794 Found."
"Hopcount for Key 698935 is 6."
"Key 742883 Found."
"Hopcount for Key 148794 is 6."
"Key 638889 Found."
"Hopcount for Key 295897 is 6."
"Key 699112 Found."
"Key 297371 Found."
"Hopcount for Key 742883 is 7."
"Hopcount for Key 638889 is 7."
"Hopcount for Key 297371 is 5."
"Hopcount for Key 699112 is 6."
"Key 297171 Found."
"Key 744671 Found."
"Key 701005 Found."
"Hopcount for Key 297171 is 5."
"Key 639508 Found."
"Key 299569 Found."
"Hopcount for Key 701005 is 5."
"Hopcount for Key 744671 is 5."
"Key 699719 Found."
"Hopcount for Key 639508 is 6."
"Hopcount for Key 299569 is 5."
"Key 636170 Found."
"Key 299840 Found."
"Hopcount for Key 699719 is 8."
"Hopcount for Key 299840 is 5."
"Hopcount for Key 636170 is 5."
"Key 291939 Found."
"Key 633769 Found."
"Hopcount for Key 291939 is 5."
"Hopcount for Key 633769 is 5."
"Key 637096 Found."
"Hopcount for Key 637096 is 6."
"Key 640853 Found."
"Hopcount for Key 640853 is 5."
"Key 633804 Found."
"Hopcount for Key 633804 is 6."
"----- AVERAGE HOP COUNTS -----"
6
neel@nrami:~/Desktop/DOS/Projects/Project3/proj3$
```

5. Output for numNodes=10000 and numRequests=1

```
neel@nrani: ~/Desktop/DOS/Projects/Project3/proj3
"Key 268201659 Found."
"Hopcount for Key 268149351 is 8."
"Key 875039 Found."
"Hopcount for Key 268367958 is 6."
"Hopcount for Key 268201659 is 5."
"Key 219328 Found."
"Hopcount for Key 875039 is 5."
"Hopcount for Key 219328 is 7."
"Key 141131081 Found."
"Key 246852933 Found."
"Key 55949164 Found."
"Hopcount for Key 141131081 is 6."
"Hopcount for Key 246852933 is 10."
"Hopcount for Key 55949164 is 7."
"Key 45057144 Found."
"Key 98735071 Found."
"Hopcount for Key 45057144 is 8."
"Hopcount for Key 98735071 is 10."
"Key 128799324 Found."
"Hopcount for Key 128799324 is 7."
"Key 268280676 Found."
"Key 134252781 Found."
"Key 1267803 Found."
"Hopcount for Key 268280676 is 5."
"Hopcount for Key 134252781 is 3."
"Hopcount for Key 1267803 is 7."
"Key 14911846 Found."
"Key 268346567 Found."
"Hopcount for Key 14911846 is 7."
"Hopcount for Key 268346567 is 6."
"Key 591102 Found."
"Key 680 Found."
"Hopcount for Key 591102 is 4."
"Key 1546847 Found."
"Key 6125137 Found."
"Key 198113065 Found."
"Hopcount for Key 680 is 3."
"Hopcount for Key 1546847 is 8."
"Hopcount for Key 6125137 is 8."
"Hopcount for Key 198113065 is 8."
----- AVERAGE HOP COUNTS -----
8
neel@nrani:~/Desktop/DOS/Projects/Project3/proj3$
```

5. Largest Network

The largest network on which our program ran successfully was 10000 nodes.