# **CS 6381: Distributed Systems Principles: Spring 2023**

**Programming Assignment 2: Chord DHT-based Discovery, Multi-Dissemination Strategy Publish-Subscribe**

**Handed out:** 02/08/2023; **Due dates**: Milestone 1: 02/18/23 11:59 pm; Milestone 2: 02/25/23 11:59 pm; Milestone 3: 03/04/23 11:59 pm

**Note 1:** Programming Assignments are to be done individually. Discussions on Slack are highly encouraged. But no direct help in writing code is permitted. All work for the specifics of the assignment must be an individual effort.

**Note 2:** Please read the writeup carefully and in its entirety, and ask clarifications, if any. Please do not wait till the last day.

**Note 3:** Use the progassign2 channel in Slack for discussions on this Assignment. Also, please try your level best to check if your question was already asked and answered. Please monitor the different threads.

# **Overview**

Since all the subsequent assignments have the same objective, we do not repeat the description but focus only on the changes. In Assignment 1, we focused on centralized discovery and a multi-dissemination approach. In this assignment, we will not change the dissemination approach. So, you should be able to reuse all of your code for that part from Assignment #1.

**Key changes from Assignment #1:**

1. We will completely change our discovery approach by moving away from a single, centralized registry to multiple discovery instances organized in a logical DHT ring that execute the Chord Distributed Hash Table (DHT) algorithm.
2. You will have to write the logic to build the finger tables per DHT node and implement the search algorithm that we studied in class.
3. Since the DHT nodes will be talking to each other, essentially, DHT nodes are peers (as in the P2P paradigm), i.e., each DHT node is both a client and server. Consequently, you will need to maintain additional ZMQ sockets at the middleware object level of the Discovery service to be used in the DHT node to node communication.
4. Since the scale of the experiments in this assignment is likely to be larger than Assignment #1 due to the presence of several DHT nodes, experiments will have to be conducted on Mininet (and if the Cloud environment is ready, then we will also deploy in Kubernetes pods but this is still under construction). Local execution may be hard to do with so many entities in the system.

# **Specification**

The specification comprises two independent parts as before:

1. **Discovery/Lookup strategy:** In this assignment, we will maintain a Chord DHT-based discovery service. These DHT nodes will store and lookup the information that they are asked to. A publisher, subscriber and broker entity will need to know any one of the DHT nodes and make invocations on it. This node need not be the exact node where the information is located. Thus, the Chord algorithm will propagate the queries in the logical ring. Such information is always in the form of a <key, value>. Very likely there is not going to be any change in the data structure that you are using in Assignment #1 to store the information arriving on the register messages. However, here we will be passing the key through a hash function that then determines which DHT node stores that info.
2. **Information Dissemination strategy:** No change in the dissemination strategies in this assignment. Thus, all the code from previous assignment should work as is for dissemination.

Ensure that your code is designed with extensibility in mind because subsequent assignments are going to enhance this design further by adding various other criteria and other technologies

**Experiments:** Since we already conducted the latency experiments for dissemination in Assignment #1, in this assignment, we will focus on the latency for registration and lookups, i.e., we will evaluate the performance of the Discovery service in this assignment. To get an accurate picture, it may be best to test the register/lookup on different DHT nodes as the starting point and then getting an idea of average, std dev, percentiles (90, 95). Moreover, such a performance under large loads (i.e., many publishers and subscribers making register or is\_ready or lookup) requests will be an important insight to obtain.

# **Assumptions Made**

In this assignment we make the following assumptions:

* There are multiple Discovery instances organized on the Chord DHT ring; a publisher, subscriber and broker can reach any of the DHT node, and the Chord DHT algorithm will take care of setting/getting the info requested on the appropriate DHT node.
* The list of topics used in the system are known ahead of time i.e., we don’t support newly defined topics for this assignment. Thus, we can reuse the topic selector helper from our starter code.
* The number of publishers and subscribers is also fixed ahead of time thereby following the same assumptions from Assignment #1. Thus, we will continue to use the same API of “register”, “is\_ready” and “lookup pub by topic” that we used in Assignment #1.

# **API Specification**

The API does not really change much from what you had in Assignment #1. If you need additional capabilities, feel free to add them to your design.

# **Starter Code**

Since most of you have designed your assignment based on the starter code, continue to extend the design. I will provide additional code with the hash function and possibly some initial idea about generating the command lines to deploy the different entities in the Mininet environment.

# **Milestones and Rubrics**

The assignment will be developed in three phases (called milestones). The three milestones and minimum expected requirements that serve as the rubrics are as follows:

## **Milestone1: Ungraded but still students are expected to work and report status.**

* Setting up of the Chord DHT nodes
* Finger tables created at each DHT node using the Chord’s approach
* Update the status in Brightspace; no video etc needed.

## **Milestone 2: Ungraded but still students are expected to work and report status.**

* Chord’s algorithm to store/query information
* Update the status in Brightspace; no video etc needed.

## **Milestone 3: Points 100 (which includes successful completion of all milestones)**

* Once the DHT nodes are all working correctly, then the Discovery process is considered as successful. Thereafter, verify that both the Direct and ViaBroker dissemination approaches continue to work as in Assignment #1.
* Evaluation in Mininet (and possibly Cloud)
* Experimental results
* Status update to Brightspace including full video available for peer grader.

# **Code Availability and Grading**

As in previous assignment, it will be good to have a Github private repo where you are maintaining your code. For the online section, you may need to invite the grading peer in case the video is insufficient. For on-ground, the same holds if our TA Ziran is unable to grade the assignment based just on the video.

Please refer to the generic programming assignment grading guidelines for the grading approach and rubrics.

**Feedback to Instructor**

The peer grader in the online section should send a detailed report to the instructor on how the submitted assignment met the grading requirements and grade received by the submitting peer along the specified rubrics that the grader used. Please CC the student whose assignment you have graded. The TA will also provide me an overall idea of how students did in the on-ground class.