Here is the list of the project topics which you can choose along with the expected team size. You do not have restrictions on the programming language or the framework for implementation. The description of the project is just for your reference. You can also choose different implementations and build a similar system. The allocation of the projects would be FCFS. Please fill this form.

Note: If you wish to work on your own distributed project idea, you need to make a project idea document with description, deliverables and team size for approval by 12th March 2022 by emailing me and ccing faculty.

Project List

1. GHS algorithm for minimum spanning tree
    - Implement GHS algorithm to calculate Minimum Spanning Tree using MPI.
    - Team size:- 3

2. Consider a large log file consisting of various exception errors thrown by the system. Using map reduce, implement a parser to count the number of exceptions of each type.
    - Team size:- 3

3. Graph Coloring
    - Implement Graph Coloring algorithm of Luby using MPI
    - Team size:- 3

4. Map-reduce framework.
    - Implement a basic distributed Map-reduce framework. Handle worker failure.
    - Team size:- 3

5. Distributed Password cracker.
    - A password cracker using brute force but takes advantage of multiple, networked computers
    - Team size:- 3

6. Make a distributed shared memory(DSM)
    - Processes running on different machines should be able to share an address space, plan to allow caching but consistency needs to be maintained.
    - Also make/find at least a program that takes advantage of this system to demonstrate the advantages
    - Team size:- 3

7. Distributed Mutual Exclusion
    - Implement the Ricart-Agarwala's mutual exclusion protocol that should handle
        - Handle node failures
        - Record state of different nodes
        - Add new node to system (BONUS)
    - Team size:- 3

8. Finding Large Primes
    - Implements a distributed system for finding large primes.
    - The system should have one server that is in control of the computation and a dynamic set of workers that are assigned numbers to test for primality.
    - Team size:- 4

9. Implement 2 phase commit protocol
    - Keep 3-4 servers either on local host, or on laptops of different team members and implement 2 phase commit protocol algorithm.
    - Consider a simple sql table to exist in all sites (servers) and where a simple query would be executed.
    - Based on the algorithm, the project will have a simulation of sites malfunctioning or have flag variables to not execute the query in order to test the algorithm.
    - Team size:- 4

10. Real Time Collaborative Text Editor
    - CRDTs + WebRTC
    - Maintaining Consistency among users, reducing latency
    - Team size:- 4

11. Watch Party Application
    - Multiple users can together watch on YouTube (or any other OTT platform of your choice)
    - Include functionalities like Group Çhat, etc.
    - Team size:- 4

12. Implement Raft, a replicated state machine protocol.
    - Leader Election: Implement Raft leader election and heartbeats
    - Log: Implement the leader and follower code to append new log entries
    - (Optional) If a Raft-based server reboots it should resume service where it left off. This requires that Raft keep a persistent state that survives a reboot.

- Team size:- 4

13. Domain Name System (DNS) using Distributed Hash Tables (DHT)
    - Prototyping an alternate architecture for DNS
    - Compare pros and cons
    - BONUS: caching among peers
    - Team size:- 4

14. Distributed Cache
    - CAP Theorem (CP, AP cache)
    - Location of Cache (each server has its own cache?), consensus
    - Adding of nodes, failure of nodes
    - Cache operations for CRUD operations in DB
    - Team Size:- 4

15. Implement Leader Election
    - Experiment with various algorithms for various topologies
    - Study which algorithms perform best for different topologies in terms of number of messages
    - Team Size:- 4

16. File Synchronizer
    - Implement a file synchronizer using client and server methodology
    - You can use rpc or pyro-5 for implementation in python.
    - Team size:- 4

17. Distributed Schema for Chat Application
    - Should be highly scalable and fault tolerant.
    - Team size:- 4

18. Implement Location Based replication of GFS
    - Sharding, Server Replication, Scalable, Load balancing
    - Team size:- 4

19. Sharded Key Value Storage
    - Build a sharded key value storage that partitions the keys over a set of replica with read write support.

- Team size:- 4

20. Distributed File Sharing System
    - Semantics of Dropbox for consistency
    - Team size:- 4

21. Distributed Web Crawler
    - Given an initial web page as input along with the depth n. Create a directed graph with urls and nodes, and a directed edge exists if the URL of the destination page is present in the source page.
    - Calculate page ranks of each node.
    - Team size:- 4