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.

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 3rd March 2023 by emailing me and ccing faculty

There are two components as part of the project submission:

Project proposal. The proposal is a short (maximum of one page) proposal for what your project will be. It should state what problem you are solving, why you are solving it, what software you will write, and what the expected results will be. You won't be judged on your proposal, it is there for the TA's to understand what you are trying to build and provide feedback based on it. Your project proposal will also be used as the baseline to judge your other deliverables.

Project Demo The final project eval conducted by the TA's. You'll be required to live demo your project as part of this evaluation. Your demo should cover the following: problem description and motivation, a description of the design of your solution, a description of your implementation, and an evaluation (both quantitative and qualitative) of how well your system solved the original problem.

Project List

Project ideas for team with 2 members:

1. GHS algorithm for minimum spanning tree
    a. Implement GHS algorithm to calculate Minimum Spanning Tree using MPI.
    b. Team size:- 2

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

3. Distributed Stochastic Gradient Descent
    a. Implement a parallelized Stochastic Gradient Descent using Map Reduce
    b. Team size:- 2

4. Suzuki-Kasami algorithm
    a. Implement the Suzuki Kasami algorithm using MPI to handle:

        i. Addition of new node
        ii. Record state of nodes
        iii. (Optional) Node failure
    b. Team size:- 2

5. Distributed Password cracker.
    a. A password cracker using brute force but takes advantage of multiple, networked computers
    b. Team size:- 2

6. Graph Coloring
    a. Implement Graph Coloring algorithm of Luby using MPI
    b. Team size:- 2

Project Ideas for team with 3 members:

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

8. Distributed Mutual Exclusion
    a. Implement the Ricart-Agarwala's mutual exclusion protocol that should handle
        i. Handle node failures
        ii. Record state of different nodes
        iii. Add new node to system
    b. Team size:- 3

9. Watch Party Application
    a. Multiple users can together watch on YouTube (or any other OTT platform of your choice). The users can be on different servers.
    b. Include functionalities like Group Chat, etc.
    c. Team size:- 3

10. Finding Large Primes
    a. Implements a distributed system for finding large primes.
    b. 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.
    c. Team size:- 3

11. File Synchronizer
    a. Implement a file synchronizer using client and server methodology

b. Team size:- 3

12. Implement 2 phase commit protocol
    a. Keep 3-4 servers either on local host, or on laptops of different team members and implement 2 phase commit protocol algorithm.
    b. Consider a simple sql table to exist in all sites (servers) and where a simple query would be executed.
    c. 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.
    d. Team size:- 3

13. Distributed Chess engine:
    a. Distributed the load of finding the best possible move over multiple interconnected computers
    b. Team size: - 3

14. Implement Leader Election
    a. Experiment with various algorithms for various topologies
    b. Study which algorithms perform best for different topologies in terms of number of messages
    c. Team Size:- 3

15. Searchable Encryption over Blockchain
    a. Apply the concept of searchable encryption over the blockchain network wherein the user will upload data in encrypted form to the blockchain which will later on allow the owner to search for keywords over the data.
    b. Create a node of the blockchain network to store the data
    c. Create an application allowing the user to upload files, search for keywords and download the files after decryption.
    d. Team size:- 3

Project ideas for team with 4 members:

16. Map-reduce framework.
    a. Implement a distributed Map-reduce framework capable of handling worker failure.
    b. Team size:- 4

17. Real Time Collaborative Text Editor
    a. CRDTs + WebRTC. Implementation of CRDT must be done from scratch
    b. Maintaining Consistency among users, reducing latency
    c. Team size:- 4

18. Implement Raft, a replicated state machine protocol.
    a. Leader Election: Implement Raft leader election and heartbeats
    b. Log: Implement the leader and follower code to append new log entries
    c. (Optional) If a Raft-based server reboots it should resume service where it left off.
    d. This requires that Raft keep a persistent state that survives a reboot.
    e. Team size:- 4

19. Distributed Cache
    a. CAP Theorem (CP, AP cache)
    b. Location of Cache , LRU behavior, consensus
    c. Adding of nodes, failure of nodes
    d. Cache operations for CRUD operations in DB
    e. Team size:- 4

20. Distributed File system
    a. Distributed File system with semantics similar to Frangipani
    b. Highly available, strongly consistent
    c. Team size:- 4

21. Domain Name System (DNS) using Distributed Hash Tables (DHT)
    a. Prototyping an alternate architecture for DNS
    b. Compare pros and cons
    c. BONUS: caching among peers
    d. Team size:- 3

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

23. Distributed messaging queue:
    a. There might be multiple producers and consumers interacting with the queue at the same time.
    b. Think about Exactly-once, at-most-once, at-least-once delivery in case of failure
    c. Team size:- 4

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

25. Sharded Key Value Storage
    a. Build a sharded key value storage that partitions the keys over a set of replicas with read write support. AP/CP system possible.
    b. Team size:- 4

26. Distributed File Sharing System
    a. Semantics of Dropbox for consistency
    b. Team size:- 4

27. Distributed Web Crawler and Indexing
    a. 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.
    b. Calculate page ranks of each node using a map reduce approach
    c. Team size:- 4