Distributed Systems Project Topics

Broad Guidelines for each topic are below. Talk to your assigned TA for more details; the design may also be changed in consultation with the TAs (but finally with my permission) if the TAs find it necessary. This document may be updated again in the next one week or so, so please check.

Other than correctness, you should think of performance, fault tolerance, and cost for your design.

1. P2P File Sharing –1
   a. Replicated Peers to store location of files and answers file search queries
   b. Normal node registers/searches in peers
   c. Replication among peers
   d. Files shared directly between nodes storing
   e. Known super-peer to find peer addresses
   f. New peer selection if number of peers is down too much
   g. Incentives/penalties to foster collaboration

2. P2P File Sharing –2
   a. No peers, distributed storage and search
   b. Caching of entries at multiple nodes for performance
   c. Incentives/penalties to foster collaboration

3. Online Retail Store
   a. Sells goods from multiple vendors (access to/from multiple database)
   b. Replicated servers to store and search who sells what and to log transactions
   c. Web based interface, load balanced to servers

4. Online Travel Agency
   a. Web based interface for travel planning
   b. Agency handles multiple hotels and multiple airlines spread over multiple cities
   c. User can ask for only hotels or only airtickets or both
   d. Trips can span over cities
   e. A trip is confirmed if all sectors are confirmed
   f. Provision to cancel

5. Building a NFS-like DFS integrated with Linux using FUSE or similar

6. Cluster Middleware –1
   a. Web based interface to submit executable codes to run, input file and output file name. Results return in output file
   b. Jobs submitted at a single server

   c. Server where jobs are submitted monitors backend servers, load balances jobs between them.
   d. Node may fail, needs strategy to handle, monitor, re-run (if needed) running jobs.
   e. Node may recover and rejoin system

7. Cluster Middleware –2
   a. Multiple machines, jobs can be submitted at any machine
   b. Machines do distributed load monitoring/balancing , and jobs are migrated before start to appropriate machine
   c. Jobs started at appropriate machines with load balancing
   d. Results to go back in original machine
   e. Node may fail, needs strategy to handle, monitor, re-run (if needed) running jobs.
   f. Node may recover and rejoin system

8. Distributed Chat Server
   a. Central server only to maintain group information
   b. Multiple groups, changes possible at anytime.
   c. Chat between group members is totally distributed (no central server)
   d. Ordering to be maintained
   e. Strategies for handling group join/leave, that happens on central server but needs to be known by all members
   f. Voice chat if possible

9. Replication with Single Master Operation
   a. Set of replicated servers
   b. Mostly read, infrequent writes
   c. Data can be changed at any replica but needs to be done in a mutually exclusive manner
   d. Lazy replication of change between replicas
   e. Should replicate only the data changed, not all data stored

10. Network Game - 1
    a. 2-3 persons
    b. No big graphics but real time
    c. Talk to TA and decide

11. Stock Market Game

    a. Bank - The one bank holds the cash of any player. A player can request a new account, deposit and withdraw money from the account. The bank has a maximum of 10 accounts. Once every minute during the game, all money in an account accumulates 1% interest.

b. Company - A company has a name and issues stock. Once a minute, the company's price of the stock is adjusted based on a random process and the law of supply and demand. Initially each company has 1000 shares of stock, which starts at $30 a share. There are several companies for example three to ten in the game. A company buys or sells shares to a player by accepting or rejecting the player's bid. The process of deciding to accept or reject may be random or based on sound business practices.

c. Player - There can be two types of players – human and computer. A player starts with $1000 and buys and sells stock from companies or other players by trading through the stock exchange. The goal of a player is to make money. A player buys or sells shares from another player or a company by posting bids on the stock exchange. The company or the other player may accept or reject the player's bid. The process of deciding how the computer-based player accepts or rejects may be random or based on sound business practices. There are two to ten players.

d. Stock exchange - Players and companies must register with the stock exchange. All trading of stocks must go through the centralized stock exchange. The stock exchange posts current bids and the prices of recent transactions on each stock.

12. Time Synchronization between

    a. NTP-like system with sync from multiple hierarchical servers.
    b. Should study NTP to see how corrections are made
    c. Should allow users to set parameters
    d. Should handle spurious clocks

13. Replication Topology Maintenance

    a. Nodes are connected but not completely connected
    b. Need to build and monitor a spanning tree to ensure replication paths between nodes
    c. Construction of spanning tree will depend on constraints such as link capacity, bandwidth, degree etc. Should handle dynamic changes including node/link failures, addition/removal of replicas etc.
    d. Should have provision for inputing user preferences

14. Distributed Publish-Subscribe Framework

    a. Multiple published servers and multiple subscribed devices

    b. Concurrency control

    c. Pull and push models for handling offline devices

15. Implement RPC framework

16. Comparison of Checkpointing/Recovery Protocols

    a. Take 3-4 algorithms (asynchronous and synchronous) and implement them (decide on algorithms based on talk with TA)

    b. Compare their performances under different parameter changes

17. Network Games - 2

    a. Minimal graphics/ no graphics(you can use 'curces' library to implement in command line)
    b. Multiple players with dedicated lanes to start the race. During, the can switch and also have to survive random obstacles in their path.
    c. Real-time updates for all players.

18. Distributed Log System

    a. Logs collected at different server
    b. API to log events with arbitrary info, send/receive of messages
    c. Different types of query: type, events in time range (others, plus combinations with AND and OR)
    d. Forming process traces from distributed logs

19. Distributed Storage

    a. Web based interface to offer storage service for files
    b. Files to be stored at different servers
    c. Fast search to find a file
    d. Should replicate for fault tolerance, handle writes to replicas