Project Topics (Broad outline) for IDS630E

1. 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

2. 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

3. 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

4. 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

5. 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

6. 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

7. 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

8. 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

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

11. Implement RPC framework

12. 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

13. Network Game