Group 7 - Load Balancing: Description & Feature Set

- **PROJECT DESCRIPTION:** We would be designing a distributed system which aims to maintain similar amount of load at all of its nodes. The system would comprise of multiple nodes and multiple clients each of whom can run a job on any of these nodes. Each node would have a load balancer and will be able to transfer the job to other nodes based on the decision of its load balancer. The client gets the feedback from the node it connected to irrespective of where the job actually got executed.

- **FEATURE SET**

CLIENT

- Connects to a node in the distributed system.
- Sends a job to execute and waits for getting back the result from the same node.
- At any time instant, connection to only a single node is allowed. However, multiple requests can be made to the same node in the ongoing session.
- Jobs can be heterogeneous. Along with the job, client also sends its resource requirements (memory, space, etc.) (optional).

NODE

- Allows connection to multiple clients at the same point of time.
- Receives jobs from the connected clients (or nodes in case of redistribution) and executes them if it needs to (decided by load balancing policies).
- Nodes are homogeneous. All nodes have similar set of resources.
- Nodes are connected in an arbitrary topology.
- If the system is small, each node sends its current load status to all the other nodes on an interrupt. The interrupt is invoked whenever the load at the node changes by more than x% (say, 10%) of the total capacity. In this way, each node keeps track of load at all the nodes in the system. If the system is large, it sends the load information only to its neighbours and only the load at neighbors is known.
- After completing the job, if the job arrived at the current node from some other node, the output is propagated to the source node. If it is the source node, it sends the output to the client.

LOAD DEFINITION: Overall CPU utilization of the node.

TRANSFER POLICY: Each time a job arrives at a node, status of nodes for which information is accessible are used to check for load imbalance (variance of loads). A threshold on the variance is used for deciding if the arrived job should be transferred to some other node. If the job has been sent by the client, it can be forwarded to some other node based on the policy. If the system is large, a TTL must be set by the node which receives it from client. In case of small systems, if the job has been forwarded by another node, then the current node should execute it

if it is the target, else it should forward the job towards the target. While in case of large systems, each time a job is received, TTL is decremented by 1. If TTL becomes 0, the current node executes the job, else the transfer policy is followed to decide if the job needs to be further transferred.

LOCATION POLICY: The node with smallest load is selected as the target for job execution. In case of small systems, job gets executed on the target node only. While in large systems, the target can further propagate the job based on transfer policy.

FAULT TOLERANCE (Discuss with Sir)

ASSUMPTIONS

- Asynchronous System.
- Reliable Network.
- All nodes have unique IDs assigned.
- No redistribution of jobs once it starts executing.