CS 425 MP-2

Ragini Gupta and Yinfang Chen

NETIDs: Raginig2, Yinfang3.

Algorithm: In the designed algorithm for distributed membership, we create a ring topology for all servers where each server maintains a neighbor list with which it communicates periodically (in the ping-ack communication framework). Each server's neighbor list contain two predecessors and two successors from the ring structure since the question mentions there will be atmost three simultaneous failures of the machines.

Machine-01 is treated as the introducer. If a machine joins the ring, it will first submit "join" request to the introducer which is followed by a multicast message by the introducer to the remaining nodes in the ring structure. Once the machine joins the ring, it will start pinging its immediate neighbors periodically at an interval of 1 second. The timestamp is recorded by each machine such that. if sender machine does not receive an ACK message to the PING request within a timeout interval of 2 second, that neighbor is marked as failed and updated membership list will be communicated to the sender node's neighbor list. Using this communication scheme, a message is communicated across nodes within a time-frame of 3 seconds and a failure detection will take atmost 2 seconds time to get reported considering 2 seconds of the time-out interval. This ensures the 5 second completeness for failure detection when there are upto 3 simultaneous failures.

Scalability: The design is scalable with an order of O(N) where N is the number of machines.

Message Format: Three types of message formats are used for communication (Ping, Ack, Join, Leave). For the PING and ACK message, the content consists of the membership list of the node. For Join/Leave, content consist of additional time-stamp and status value for the sender node (status can be Joining, Running, Failed, Left (voluntarily)).The data content is dumped into JSON format before forwarding it into the ring. The metadata of the message contains a string of what type the message is including ping or ack or leave or join.

Use of MP1-: We logged the membership entries on each of the machine and use the grep command for benchmarking different measurements for bandwidth usage and false positive rate.

Background bandwidth usage:

The size of a message is ~128 Bytes. Since message is pinged to four of the node's neighbors in addition to receiving ACKs, receiving ping messages and responding to pings with ACKs (i.e. send PING+ Receive PING + Send ACK + Receive ACK on a single machine): the total bandwidth for one machine is 16*128= 2048 Bytes/second and for N=6 machines, the bandwidth usage is approx. 12.28 kBps.

Average Bandwidth:

Approximately, 6.8 kBps for leaving and 5.1kBps for failing node. When a node leaves, there is a sudden rise in the bandwidth usage due to the overhead of sending leave messages to the introducer.

Approximately, 16.3 kBps for joining a process. When a fifth node joins, there is a sharp increase in the number bandwidth usage due to an overhead for sending join messages to the introducer and from introducer node to all other machines in the ring.

False Positive rate:

We simulated the false positive rate by controlling the rate of packet drops using uniform distribution for dropping packets when the probability distribution is < 0.03% and <0.3%

The False positive rates is determined when the message is dropped but the status is shown alive in the neighbor's membership list. It is the ratio of number of times nodes were detected alive in the presence of dropped messages to the total number of entries in the log. To measure the average false positive rate, 5 data points are considered. For confidence interval, a confidence level of 90% is considered which gives z=1.645. CI= $\frac{z* \sigma}{\sqrt{n}}$ where n is 5 and σ is Standard Deviation.

	N=2		**N=6**	
	Packet Drop=3%	Packet Drop=30%	Packet Drop=3%	Packet Drop=30%
Average FP rate	0.00294	0.0322	0.009244855	0.04093576
Std (σ)	0.0011	0.0262	0.0079	0.042
Confidence Interval	0.0018	0.019	0.0058	0.031

As the message drop rate increases, the false positive rate also increases. This is expected because the node will not consider the network delay due to message drop loss. It will only consider that its membership list entries are stale because the node has failed instead of the poor network condition. Subsequently, it will inaccurately flag the node as failed when it is actually alive. As the probability of packet drop rate increases, the chances of dropping the messages also increase, and the possibility of the node being marked as failed increases proportionally. This implies that the false positive rate also increases by increasing message drop %.

Git: https://gitlab.engr.illinois.edu/yinfang3/cs425-mp/-/tree/main/mp2