CS 425 Fall 2017 - Indranil Gupta

Hieu Huynh - hthuynh2, Rohith Nallandigal - nalland2

MP2 Report

Introduction:

In this machine problem, we designed a distributed membership system which maintains a dynamic membership list at each node in the system. The membership list may change over time as new nodes ask to join the group and nodes fail or voluntarily leave the group. The system can be divided into two major components: failure detection and dissemination.

Failure Detection:

A ring heartbeating protocol is used to maintain awareness of the state of each node in the system. The structure of the ring is communicated by the ordering of the membership list - each node in the membership list is sorted by its identification number. Then each node's heartbeat targets are calculated as its two predecessors and its two successors. By this "push" method, each node sends heartbeats to its targets every HB_TIME time units, a predetermined contstant.

Each node also receives heartbeats by its four closest neighbors. If node i detects that a heartbeat has not been received from node j within HB_TIMEOUT time units, it marks the node as failed. This failure is reflected in node i's membership list immediately. Then, node i disseminates information about node j's failure using the dissemination protocol. This algorithm scales to large N because each node has an $O(1)$ message load and the overall network has an $O(N)$ message load. Also, the time to detection, i.e. the time it takes for some node to detect a failure, is $O(1)$.

Dissemination

The dissemination of a node join or leave (collectively referred to as a membership change) is performed by the Gossip protocol. When a node learns of another node's failure, it communicates this information to GOSSIP_B randomly chosen nodes among the ones that are known to be alive. When nodes receive a gossip message, they update the membership list to reflect the changes contained in the gossip message. Duplicate gossip messages do not affect the membership list since if a membership change is already reflected, the update does not need to occur.

Also, gossip messages must be disseminated themselves. When nodes receive a gossip message, they randomly choose another GOSSIP_B nodes to relay the gossip message. This process continues for GOSSIP_C $\times \log N$ relays, upon which the nodes drop the message. The GOSSIP_B and GOSSIP_C parameters are adjusted to fit the specific network structure and minimize bandwidth but maintain a high probability of contacting all nodes eventually.

This algorithm scales to large N because each node has an $O(bc \log N)$ message load per membership change. The total time to detection, i.e. the expected time taken to disseminate the message to the whole network, is $O(\log N)$, for suitable constant b, c.

False Positive Rates:

In our distributed system, four heartbeats are sent out in every timeout interval; that is, for a node j to detect node i as failed, i must fail to send four consecutive heartbeats to j. In the case of a group with just two machines i and j, the probability of false positive is extremely low. As an example, consider a message loss rate of 10%. The probability of a node losing four consecutive heartbeats is 0.1^4, or 10^{-4}. Then, the time to detect a failure at a specific node, say i, can be modeled with a geometric distribution with mean $\frac{1}{p}$, or 10000 seconds. The time to deact a failure at *some* node can be modeled by a negative binomial distribution with mean $\frac{1-10^{-4}}{2(10^{-4})}$, or 5000 seconds. For this reason, the false positive rates with only two VMs in the group are reported as expected values.

Message Loads:

The data summaries for average message loads for various circumstaces are given below. In the case of four nodes, the number of messages for each of node join, fail and leave is constant because we do not adjust the number of retransmissions in the gossip protocol based on the number of nodes in the group. Node joins require two more messages because the introducer needs to send the membership list to the node requesting to join.

	3%	10%	30%
VMs 1-2	617283s	5000s	61s
VMs 1-4	308641s	2499s	31s

(a) False Positive Detection Times

	Mean	Standard Deviation
Background	21.39	2.73msgs/sec
Node Join	29	0
Node Fail	27	0
Node Leave	27	0

(b) Message Loads