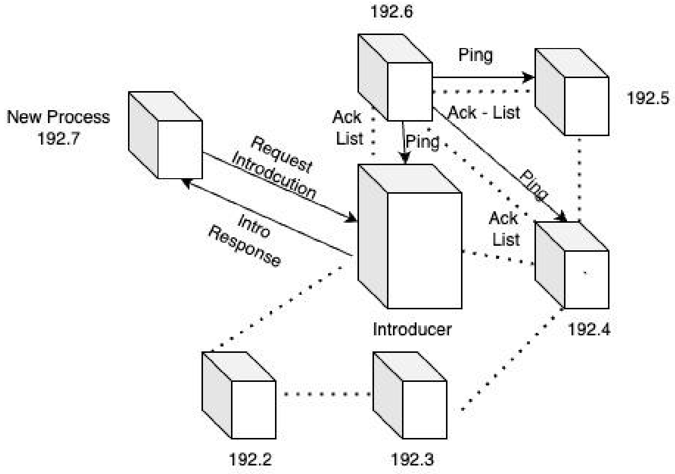
Distributed Group Membership - MP2

Rahul Shivaprasad (rahul11)
Prajwal Kiran Kumar (prajwal5)

Design



We have built a failure detector that works in a SWIM style mechanism. An introducer is responsible for joining processes into a cluster. A full membership list is maintained at all processes. The processes ping each 3 neighbors set as i+1, i-1 and (i+n/2)%n. Hence, it can scale to any N number of processes (we don't multicast) . The acknowledgements sent from these processes contains the full membership list that contains process ids, timestamps and incarnation numbers of all the processes in the cluster. This list is merged into the existing membership list. If the client process does not receive the acks by T-acktimeout, we mark the process as suspicious and start timers T-dead to kill the process. If in this duration, we receive acks informing that the suspicious process is not actually dead, we mark it back as alive and continue ping/acking. If we do not, once the timer runs out, we mark a process as dead and keep the process in the list for another time period T-cleanup. Once T-cleanup runs out, we remove the member from the membership list and continue with our ping/acking.

Message Format :

Our marshaled message format for the acknowledgement is as follows :

```
{
        “membershipID”: <ip-address>,
        “timestamp: <timestamp of when process came up>,
        “incarnation”: <incarnation number>,
        “status”: <status of the process> // 2 - Alive , 1 - Suspicious, 0 - Dead
}
```

Every process in the list maintains a neighbor list of upto 3 members. Hence, if the process and 2 of its neighbors fail, we will still be able to detect the failures of these processes, as the third neighbor can record the death of these process(es).

MP1 distributed log querier was heavily used to debug concurrency race conditions and deadlock situations in the code by adding required keywords in the logs to denote locking and unlocking of critical sections of the code that modify the global membership list. Also, we used the logs to verify and confirm membership list creation and neighbor ping ack responses by grepping the required IP Address and hostname.

Background Bandwidth Usage for N=6 is 817.74 Bytes/second on an average for N=6 nodes.
Average bandwidth usage for N=6 for
Leaves = 968.12 Bytes/s
Joins = 858.41 Bytes/s
Failures = 910.23 Bytes/s

The leave, join and failure average bandwidths are approximately equal, as more failures/leaves/joins in this design are not sending out lesser/more messages. The whole membership list is always piggybacked on the acknowledgements sent (no extra ping/pongs sent)