CS425 MP2 Report

Melissa Jin (mqjin2), Thomas Kao (tkao4)

Design:

We implemented a failure detection system using the heartbeating style of failure detection in a virtual ring. We choose to have nodes send out a heartbeat to four other machines in the ring. This way we can guarantee completeness for up to 4 machines failing simultaneously. Each node maintains a local copy of the membership list which is constructed as a doubly circular linked list. We also use a map to retrieve the pointer to a node in constant time when given the machine id (VM number). Our design can easily scale to a large number of machines because inserting and removing nodes from the membership list is done in constant time since we only need to change pointer references.

Each node detects its neighbors by determining the next four nodes in the ring that are alive. The direction that it searches depends on the function that it is performing. For example, when searching for neighbors to gossip to, the searches for the next four neighbors that are alive in the clockwise direction. Similarly, when searching for nodes to listen to, the node searches for the next four neighbors that are alive in the counterclockwise direction.

In each heartbeat, the machine sends its ID and the version of its own membership list. This is so that when the neighbors receive the heartbeat, they can then merge the received membership list with their own membership lists. Each node in a membership list contains a heartbeat counter which counts the number of times the node sends out a heartbeat. We use the value of the heartbeat counter to get the most updated status of the node when merging two membership lists. In addition, we record the timestamp whenever a machine joins, which allows us to determine whether the two membership lists have the same incarnation of each machine.

We marshalled the heartbeat messages by using protocol buffers. We only heartbeat the machine id and membership list, and not the entire machine list structure, in order to minimize the size of the sent messages. Using protocol buffers, we transform the machine's membership list from a doubly circular linked list to an array where each element represents one machine. Each machine element contains an id, timestamp, heartbeat counter, and status of the machine.

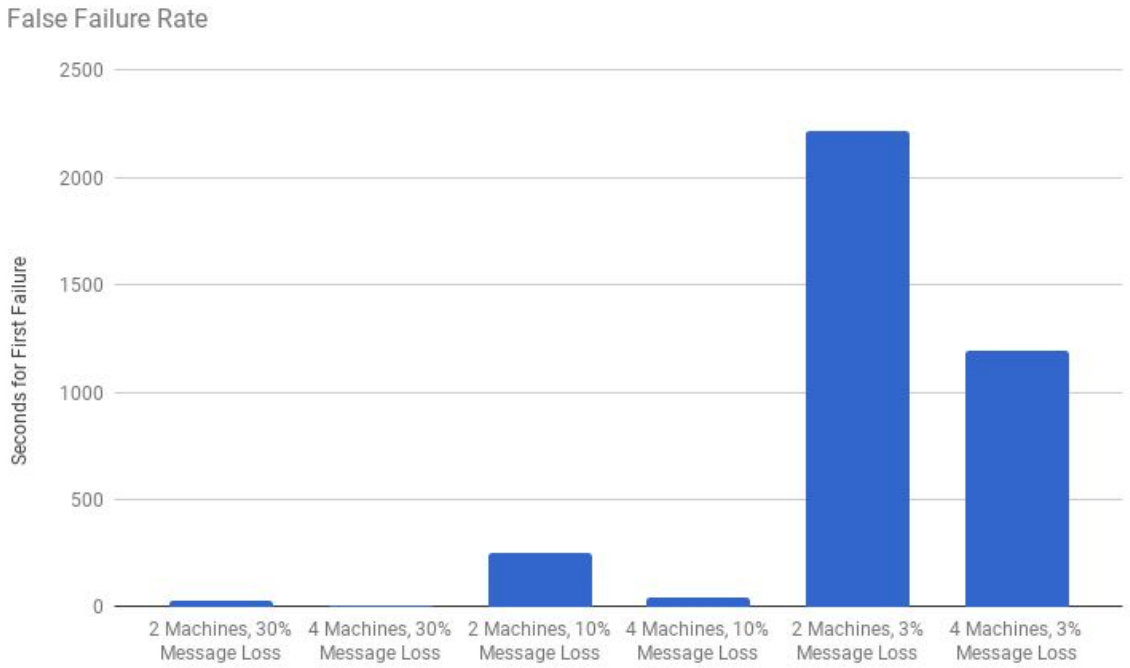
In order to stay within the time limit for detecting and propagating failures, we set time intervals for gossiping, as well as time deadlines for listening. For gossiping, we choose a time interval such that a machine sends messages frequently enough that the listening machine knows that it is alive, but not so much that it congests the bandwidth. When a machine is listening, it sets a deadline for receiving messages when it realizes that it has a new neighbor to listen to. Then, every time it receives a message from that neighbor, it pushes the deadline further.

We used MP1 to help us debug for MP2 by displaying the grep output of the log files from all machines on one machine. This way, we can easily check when the membership lists of all the machines are updated to make sure our system meets the time requirements.

Measurements:

We set each machine to send a heartbeat to the other machines every 400ms. With 4 machines in the system and no joins/fails/leaves, each machine is sending a heartbeat the 3 other machines. Therefore, it sends 3 messages and receives 3 messages every 400ms. This is equivalent to sending 7.5 and receiving 7.5 messages per second which is a total of 15 messages per second. When a machine joins the system, it sends out a message to 5 entry machines requesting the current membership list, and if all five machines are up, they send a response to the new machine. So that is an extra 10 total messages to the bandwidth. When a machine fails, there are no extra messages that are sent by the other machines, but due to the failed machine, the bandwidth usage is reduced by the number of messages one machine sends/receives per second. When a machine leaves, it continues to heartbeat during its cleanup period, which means the bandwidth stays the same during that period. After the machine leaves,the bandwidth usage is reduced by the number of messages one machine sends/receives per second.

The plots show that with higher message loss rates, the time that it takes to detect a failure decreases dramatically. This shows that with more dropped packets, it is more likely that a machine won't receive a heartbeat from one of its neighbors within the 2 second deadline which results in a false failure. In addition, when more machines are added, the failure rate increases. This is because there is a higher volume of messages being sent, so more of them are being dropped. This makes it much more likely that any listening machine will miss a heartbeat.



	2, 30% Loss	4, 30% Loss	2, 10% Loss	4, 10% Loss	2, 3% Loss	4, 3% Loss
Average	28.6s	254.4s	2216.8s	3.44s	44.6s	1194.8s
Std	7.903s	89.09s	643.26s	0.367s	19.68s	197.4s