CS 425 MP2 Report

Yanchao Liang

Liang51

This program consists two parts:

* A fixed contact machine which enable machine to join.
* Machines which will join,

The contact keeps a member list and a socket. When a machine joins, contact adds this machine’s IP to the member list and send its member list back to the machine. At the same time, contact use a Hashmap to distinguishes successive incarnations of the same machine. After it update its member list, contact send a add message to the first machine in the list. This will eventually update all member lists in the group. In addition, there is a thread running backend, waiting remove message. When a machine fail or leave, a message will be send to contact to update its member list.

When a machine object is created, it will connect to the contact, in order to join the group. Then receive a member list from contact, and update its member list. When a machine receive a add or remove message, it will update its member list, then send two more add/remove message to next/previous two machines, and keep updating the member lists in the group. If the member already be added or removed, do nothing. Therefore the adding/removing will eventually end.

There are two thread running backend. One keeps sending heartbeat per second to the next one in the list, another keep receiving heartbeat. If the receiver didn’t receive any message in 3 sec, it detected the previous machine failed. Then it remove previous one from the list, send a remove list to the contact, and send remove message to previous machine in the updated member list. The remove message will eventually update all lists in the group.

This algorithm scales to large N because each machine only communicates to this adjacent machine, there is no broadcast. Therefore the traffic is distributed.

The messages for adding or removing is sent in string with the first char A (add) or R (remove) then concatenated with the IP to add or remove. Heartbeat message is the sender’s IP in String.

We wrote a writeLog class, and logging any important event to a log file, then use MP1 to query log and debug.

Bandwidth Usage (4 machines, assuming each string is 1KB):

* Join: 1 TCP message to get member list (4 members) from contact, 4\*2+1=9 UDP adding messages. Total is approximate 13KB
* Leave: 1+4\*2 UDP removing message. Total is approximate 9KB
* Fails: 1+4\*2 UDP removing message. Total is approximate 9KB
* Heartbeat is 4 UDP message per sec. Approximate 4KB per sec

AVERAGE FALSE RATE

Standard deviation

Confidence intervals

This basically is a sum of modified Geometry Distribution. Failure is cause by two contiguous UDP drops, since my program send heartbeat per sec and check time out every 3 sec

Therefore as drop rate increase, average, standard deviations and confidence intervals all increase.

As machine number increase, average, standard deviations and confidence intervals all increase as well.