Distributed Computing: Autumn 2018
Programming Assignment 2: Implementing Distributed Snapshots

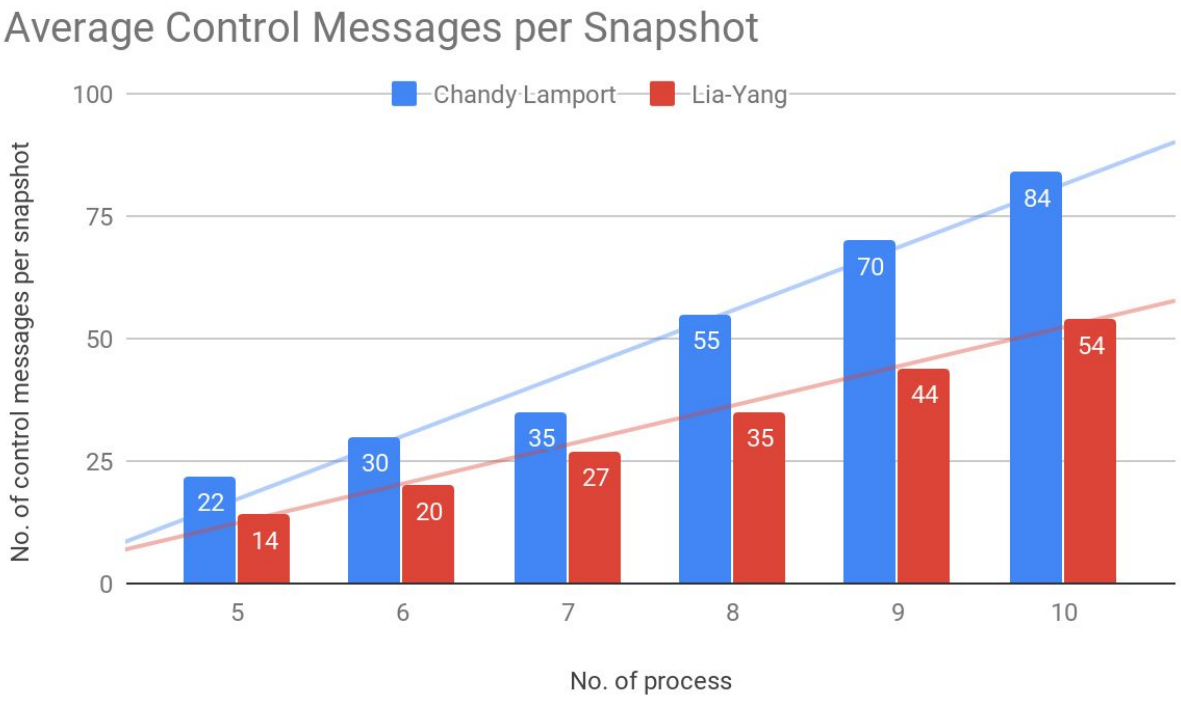
Ganesh Vernekar - CS15BTECH11018

Graph

(Input files on which it was tested are given with the submission)

Program Input Parameters

- N=5,6,7,8,9,10 (No. of Processes)
- A=1000000, T=15000.
- Random Strongly Connected Graph for sending messages.
- Spanning Tree for sending terminate messages and snapshot (and markers in Lai-Yang).



Observations

- No. of control messages per snapshot increases at higher rate in Chandy Lamport compared to Lai-Yang.
- No. of control messages depend highly on graph topology in Chandy Lamport as marker is sent on all channels. Graph topology didn't matter much for Lai-Yang as every control message is sent over a spanning tree.
- No. of control messages was not uniform across all processes in Lai-Yang. The leaf nodes usually send and get less control messages compared to those which are near the root. That is because the nodes in between root and leaf has to forward the control messages through them.
    - Hence because of this, some processes are highly loaded with control messages during snapshot, and some are very less loaded.
- No. of control messages was slightly uniform across all processes in Chandy Lamport because markers are sent across all channels.

About files

- **Readme.md** contains the instructions on how to run the program and use existing tests inputs.
    - **inp{5,10}** are the test inputs on which the graph is based.
- **CL.cpp** contains the Chandy Lamport implementation.
- **LY.cpp** contains the Lai-Yang implementation.
- **common.h** contains the elements used in both the implementations.
- **channel.h** is golang style channel implemented in C++.
- **gen.py** is script to generate inputs.

Note

- It is assumed that messages will be transferred in FIFO order for Chandy Lamport.
- Fault tolerance is not taken care, it is assumed that mess sent is always received and without any error. So sometimes due to small errors in MPI message sending and receiving, the program might crash and needs to be re-run.