Distributed Computing: Autumn 2018
Programming Assignment 3: Implementing Distributed Mutual Exclusion Algorithm

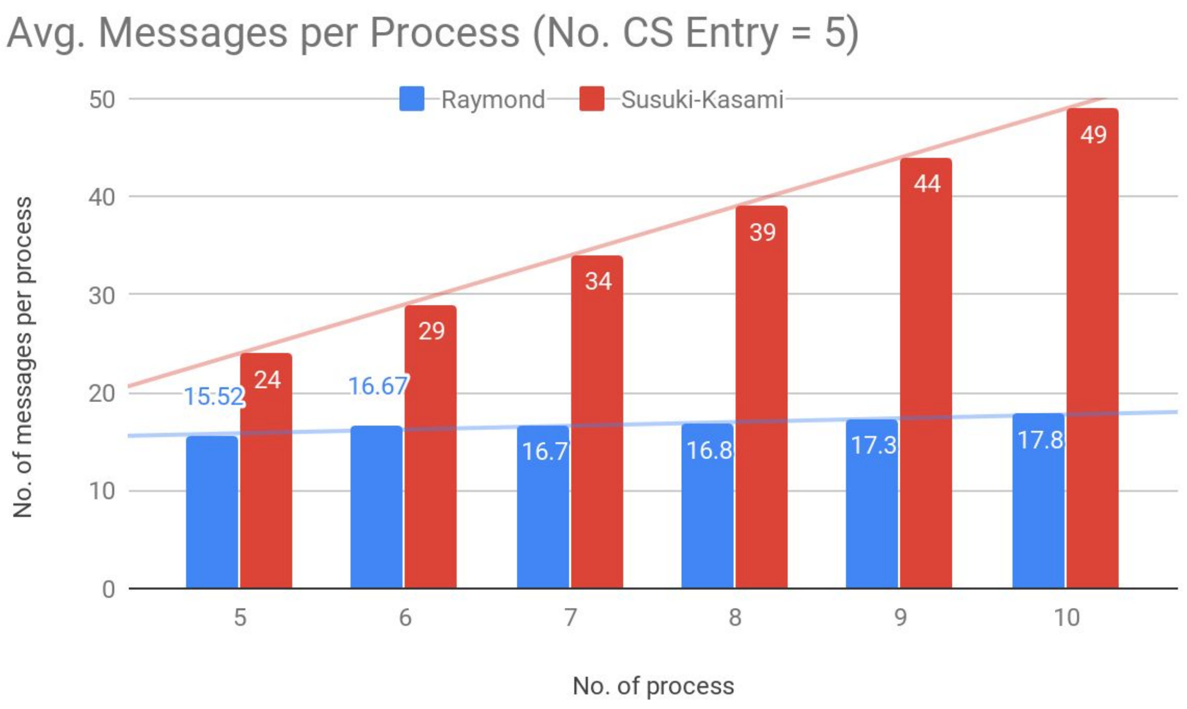
Ganesh Vernekar - CS15BTECH11018

Program Input Parameters

- Common
    - k (no. of CS entry per process) = 5
    - alpha =10, beta = 2
- Raymond:
    - Randomly generated tree for each n (no. of process)
        - Script used to generate is given in the submission.

(NOTE: Inputs used for the tests is given with the submission)

1. Avg. Messages per Process



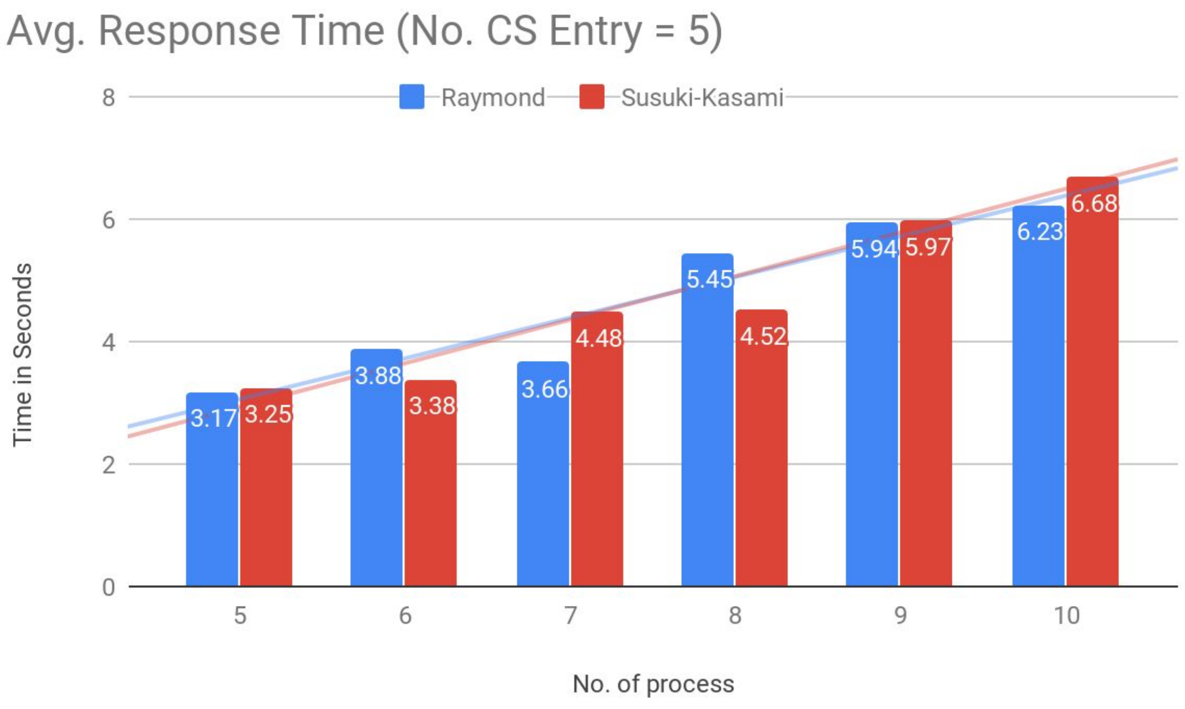
Observations

- Raymond:
    - No. of messages did not increase significantly with the increase of processes.

- As the tokens are passed along the tree, the number of messages required to deliver a token doesn't increase a lot in average when more processes are added.

- Suzuki-Kasami:
  - The messages complexity increased significantly with increase in number of processes. This is because each process has to broadcast to every process when it wants to enter CS. (Note that no. of messages for each CS entry would be the value in the graph divided by 5. The value in the graph is for 5 entries.)
- Comparison:
  - Suzuki-Kasami requires significantly more messages than Raymond as each process has to send request to every other process. Whereas in Raymond, in a balanced tree, it would be exponentially less.
  - Looking at the trend line in the graph, Raymond looks more scalable with respect to message complexity.

2. Avg. Response Time



Observations

- The message sending time was very less (less than 1 millisecond to few milliseconds), hence it's difficult to any consistent difference between the 2 algorithms. Sometimes Raymond is better, sometimes Suzuki-Kasami.
- In both the algorithms, response time increased with increase in no. of process. This is due to more process in the queue for CS on an average as no. of processes increase.

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.
- **ProgAssn3-CS15BTECH11018-Raymond.cpp** contains the Kerry Raymond implementation.
- **ProgAssn3-CS15BTECH11018-SuzukiKasami.cpp** contains the Lai-Yang implementation.
- **common.h** contains the elements used in both the implementations.
- **gen.py** is script to generate inputs.
- **sort_logs.py** to sort the logs generated based on the timestamp of the logs.
- **Makefile** to build and run programs.

Note

- 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 or get stuck, and needs to be re-run.*