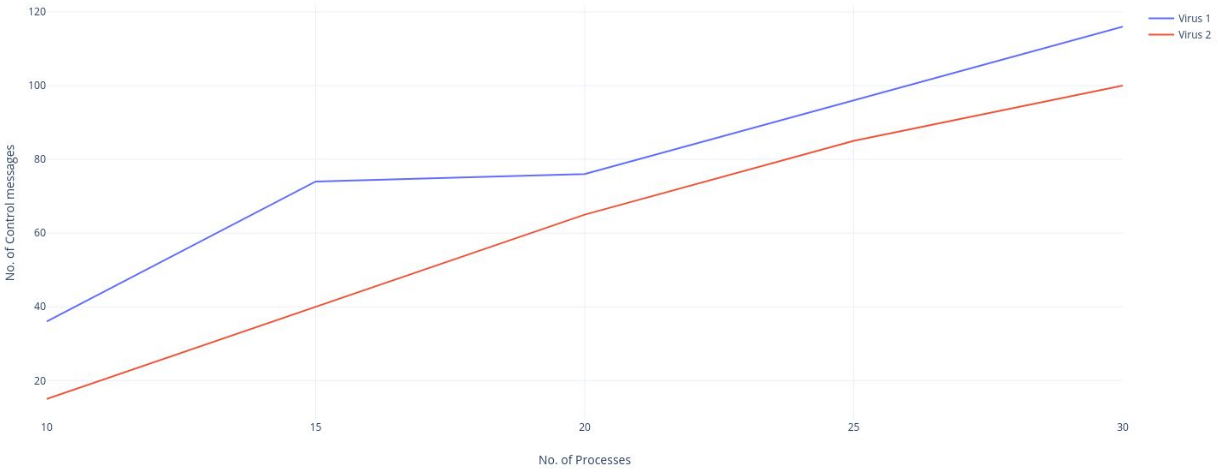
Assignment 2 - Distributed Termination Detection.

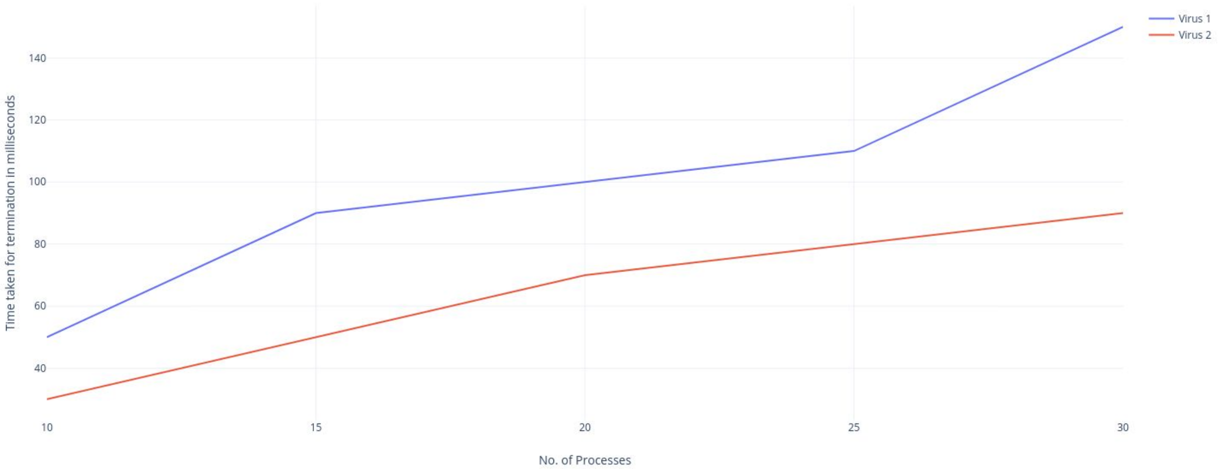
CS17BTECH11022

For this assignment, I have used OpenMPI to implement both the algorithms. The design for the first algorithm is as follows. On receiving a blue message, a process checks if all it's children have terminated, after which it accordingly passes a token to it's parent. And if it's the root, it announces termination. I used MPI routines to create the Distributed Graph Topology. Instead of using threads, one for listening and one for sending events, I used Non-blocking receiving with a single thread to account for parallel computation and IO. We can emulate the behaviour of two threads with this Non blocking Receive and Blocking Send.

> If a process sends a red message, it marks it's token as a black token. Also, a process changes its token color to black when it receives a black token. Therefore when the root detects termination and it's token is black, it sends a restart message, after which all processes change their token color to white and reset other data structures.

For the second algorithm, I have implemented "A message optimal" algorithm in the book by Chandrasekhar and Venkatesan. An extra node is created which is designated as dparent in the algorithm. When this new extra process receives acknowledgements from all the processes, it declares termination. In this algorithm, red messages are treated as basic messages, and the number of control messages the number of red messages.




Virus 1
Virus 2
Time taken for termination in milliseconds
140
120
100
80
60
40
10
15
20
25
30
No. of Processes