Assignment 5
Implement Dining Philosopher's using Conditional Variables

Name : Shivashish Suman
Roll No. : CS17BTECH11037

In this assignment, we have implemented the solution to Dining Philosopher's Problem using conditional variables.

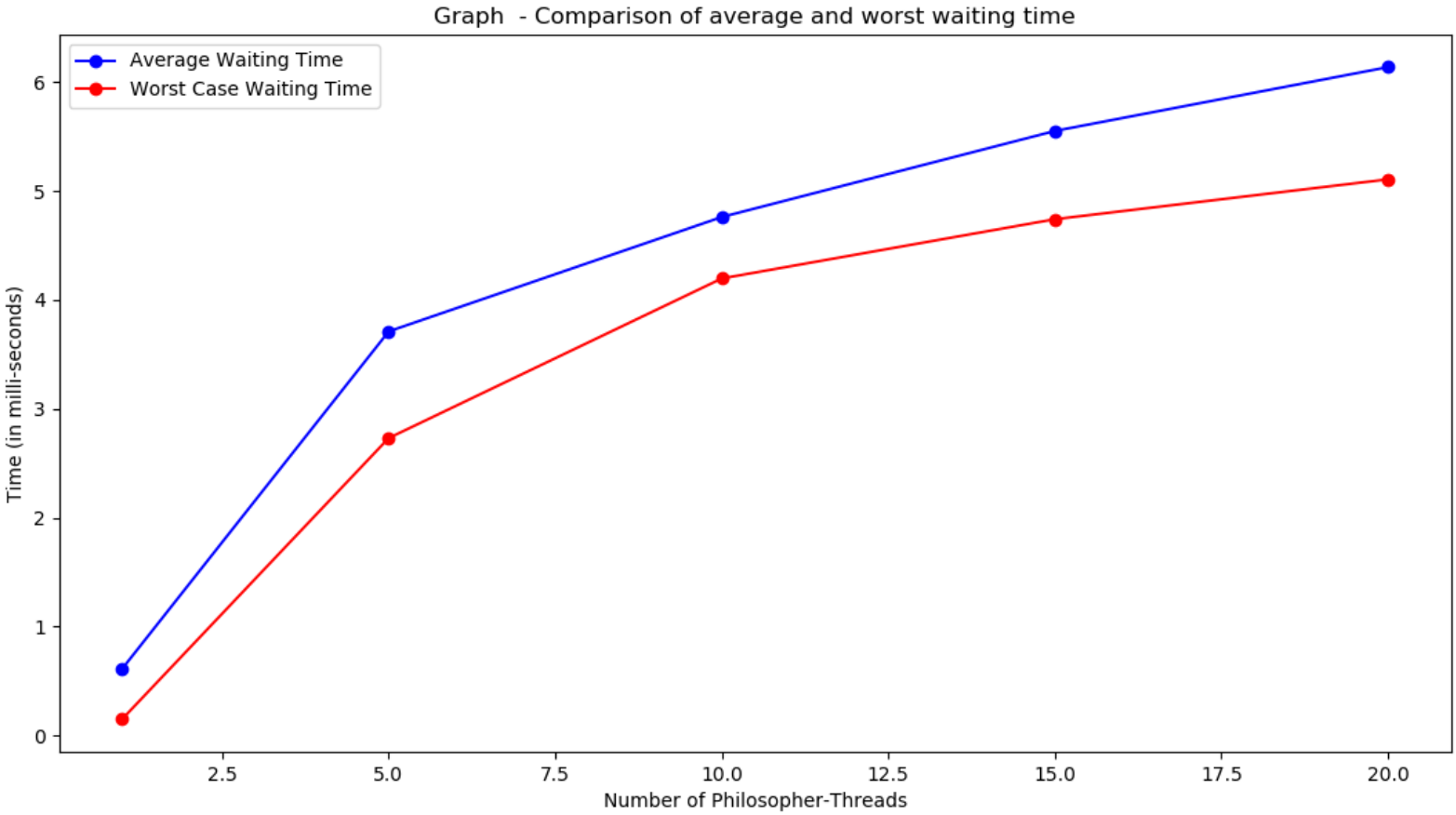
We have created an enumeration for storing the states of philosophers which are 'Thinking', 'Eating' and 'Hungry' in the 'state' array.
We have an array of conditional variables for philosophers whose limit is given by 'philnum' which has been declared as a Macro with value 20.

We have created a class called 'myMonitor' which has functions of pickup , putdown and test. Each philosopher, before starting to eat, must invoke the operation pickup() .This act may result in the suspension of the philosopher process. After the successful completion of the operation, the philosopher may eat. Following this, the philosopher invokes the putdown() operation.
In pickup function whether both its are neighbours are eating or not. If both are not eating, then philosopher will pickup the chopstick otherwise it waits on the condition variable.This is achieved using test function.
In putdown function, it checks for its neighbours who are hungry and if one of them is hungry then it gives the chopstick to that philosopher.

We have used mutex lock to allow only one thread to enter the critical section.When a process waits on the conditional variable

then it signals the mutex so that other process can enter the critical section.
We have noted the time at which philosopher request to enter the critical section and time at which it enters. The difference of these two give us the waiting time.

Graph :



We observe that for n=1 the average waiting time as well as worst waiting time is very low(< 1 millisecond) as comapared to other values of n because single philosopher does not need to wait. Also worst case waiting time is slightly smaller as compared to average waiting time for all values of n.