# CS322:Big Data

# Final Class Project Report

**Project (FPL Analytics / YACS coding):** <u>YACS coding</u>   **Date:** <u>06/12/2020</u>

| SNo | Name | SRN | Class/Section |
|---|---|---|---|
| 1 | Yashashvini R | PES1201800087 | 5D |
| 2 | Greeshma C R | PES1201800196 | 5B |
| 3 | Navyadhara Gana Sai G | PES1201800230 | 5A |
| 4 | Ruchitha M | PES1201800252 | 5F |

## Introduction

Big data workloads consist of multiple jobs from different applications. These workloads are too large to run on a single machine. Therefore, they are run on clusters of interconnected machines. A scheduling framework is used to manage and allocate the resources of the cluster. And one efficeint way is to build a framework for scheduling on multiple machines. And what we implement here is YACS.

YACS – Yet Another Centralized Scheduling Framework, has one Master, workers which bind together to perform the scheduling tasks in the most efficient way.

## Related work

1. https://arxiv.org/abs/0906.0350
2. Process Scheduling (rutgers.edu)]

## Design

The design of this framework is quite simple but effective.

1. It has one Master/Driver processes and 3 worker processes mimicing 3 degenerate machines running on the same machine through different threads.
2. Threading locks are used to prevent any kind of race conditions or dead locks.
3. The master and the workers are connected through ports. Each worker operates on a different port.
4. Each of the three workers have a certain number of slots available in them.
5. A driver sends the queue of requests/tasks/jobs to be scheduled and the master schedules these queue of tasks using three different scheduling algorithms based on selection and they are
    i. Round Robin Scheduling.
    ii. Least Loaded Scheduling.
    iii. Random Selection Scheduling.
6. Both map and reduce tasks are scheduled and sent to workers.
7. Master schedules the tasks using any one of the above algorithms and sends them to worker to update the time and the slots available
8. The worker acknowledge the connections and update the master through port connections.
9. Log files are maintained to analyse the flow of tasks to and fro worker and master.
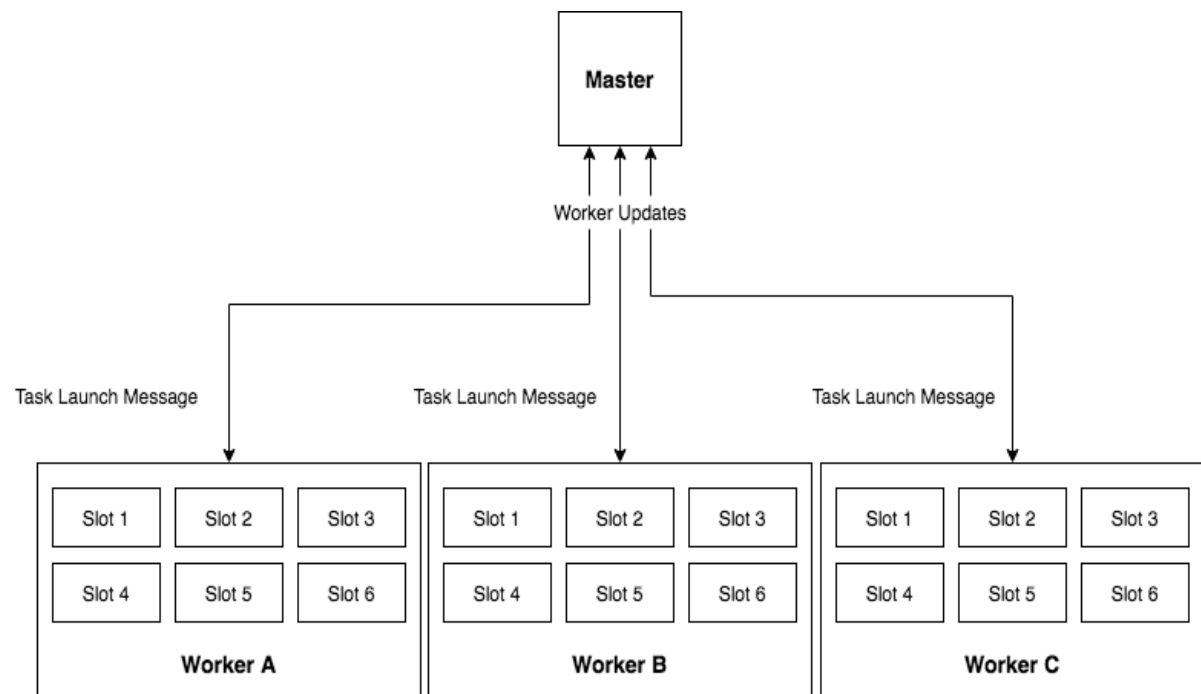
## Results

```
---------------------------------- TaskCompletion Stats ----------------------------------

---------------------------Mean and Median for RoundRobin Algorithm are -> 2.45,2.00----------------------------

---------------------------Mean and Median for LeastLoaded Algorithm are -> 2.24,2.00----------------------------

---------------------------Mean and Median for RandomSelection Algorithm are -> 2.44,2.00----------------------------
```

```
---------------------------------- JobCompletion Stats ----------------------------------

---------------------------Mean and Median for RoundRobin Algorithm are -> 6.00,6.00----------------------------

---------------------------Mean and Median for LeastLoaded Algorithm are -> 8.20,9.00----------------------------

---------------------------Mean and Median for RandomSelection Algorithm are -> 5.75,6.00----------------------------
```

**Worker Vs No of Jobs**
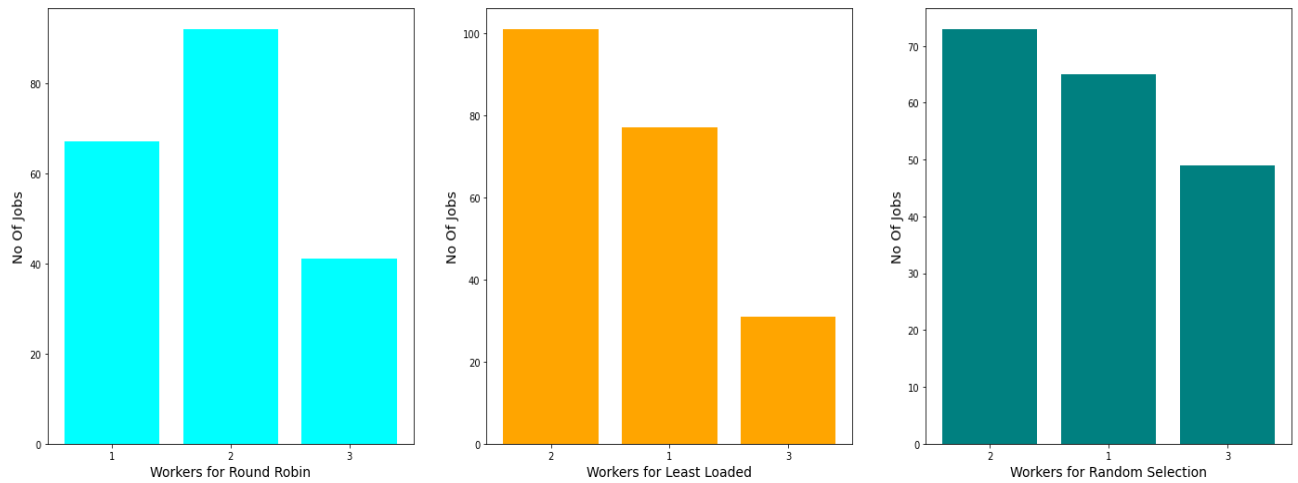


**Worker Vs Time in Secs**
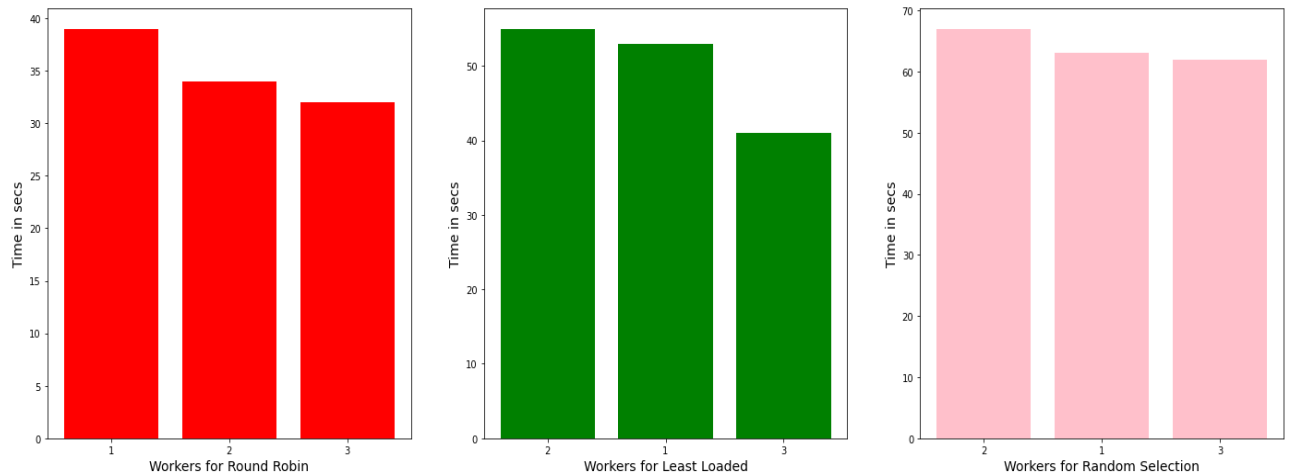


## Problems
1. Faced few problems while binding ports
2. Numpy caused problems with installation.

## Conclusion
YACS was overall a very good project working on. We were able to learn about socket programming and also how seamlessly computer handles large amounts of data through an excellent framework.

## EVALUATIONS:

| SNo | Name | SRN | Contribution (Individual) |
|---|---|---|---|
| 1 | Yashashvini R | PES1201800087 | Master[Connection functions between master and workers, Scheduling Algorithms], Worker |
| 2 | Greeshma C R | PES1201800196 | Master[Socket Programming], log file analysis |
| 3 | Navyadhara Gana Sai G | PES1201800230 | Master [Connection functions between master and workers, Scheduling Algorithms], Worker |
| 4 | Ruchitha M | PES1201800252 | Master[Socket Programming], log file analysis |

## (Leave this for the faculty)

| Date | Evaluator | Comments | Score |
|---|---|---|---|
|  |  |  |  |

## CHECKLIST:

| SNo | Item | Status |
|---|---|---|
| 1. | Source code documented | Yes |
| 2. | Source code uploaded to GitHub – (access link for the same, to be added in status →) | navydhara79/BD_0087_0196_0230_0252_Project (github.com) |
| 3. | Instructions for building and running the code. Your code must be usable out of the box. | BD_0087_0196_0230_0252_Project/README.md at main · navydhara79/BD_0087_0196_0230_0252_Project (github.com) |