CS322: Big Data

Final Class Project Report

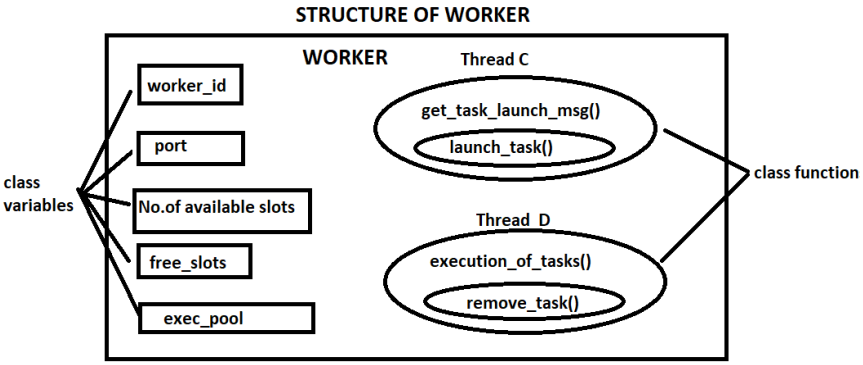
Project (FPL Analytics / YACS coding): YACS Date:

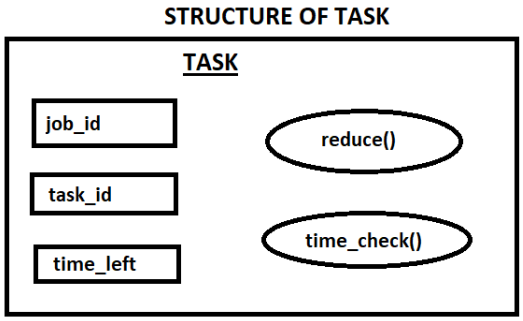
SNo	Name	SRN	Class/Section
1	Sai Eashwar K S	PES1201801910	5B
2	Ritvik Sanjeev Patil	PES1201801587	5A
3	Monisha Chandra	PES1201802102	5E
4	Teja P	PES1201800408	5B

Introduction

YACS(Yet Another Centralized Scheduler) - Consists of one Master, which runs on a dedicated machine and manages the resources of the rest of the machines in the cluster. The other machines in the cluster have one Worker process running on each of them. The Master process schedules tasks while the Worker processes execute the tasks by listening for Task Launch messages from the Master. Once received, the Worker adds the task to the execution pool (Consisting of all currently running tasks) of the machine it runs on.

The project simulates a framework consisting of 1 Master & 3 Workers. The value of available slots is decremented when a slot is said to have been allocated, and incremented when a slot is said to have been freed on a task's completion. The Master allocates tasks to Workers based on 3 Scheduling Algorithms – Random, Round Robin & Least Loaded





Related work

Referred books:

Operating System Concepts by Abraham Silberschatz, Peter B Galvin, Gerg Gagna.

Core Python Applications Programming, 3rd edition by Wesley Chun.

Referred Websites:

RealPython

StackOverflow

GeeksForGeeks

Assumptions

Since config file is not given as input to the worker program, we have hard coded the number of slots for each worker.

Design

The Master listens to requests on port 5000 & updates from Workers on port 5001. The Master and Workers each have 2 threads arising from them.

Master Thread 1: getRequests()

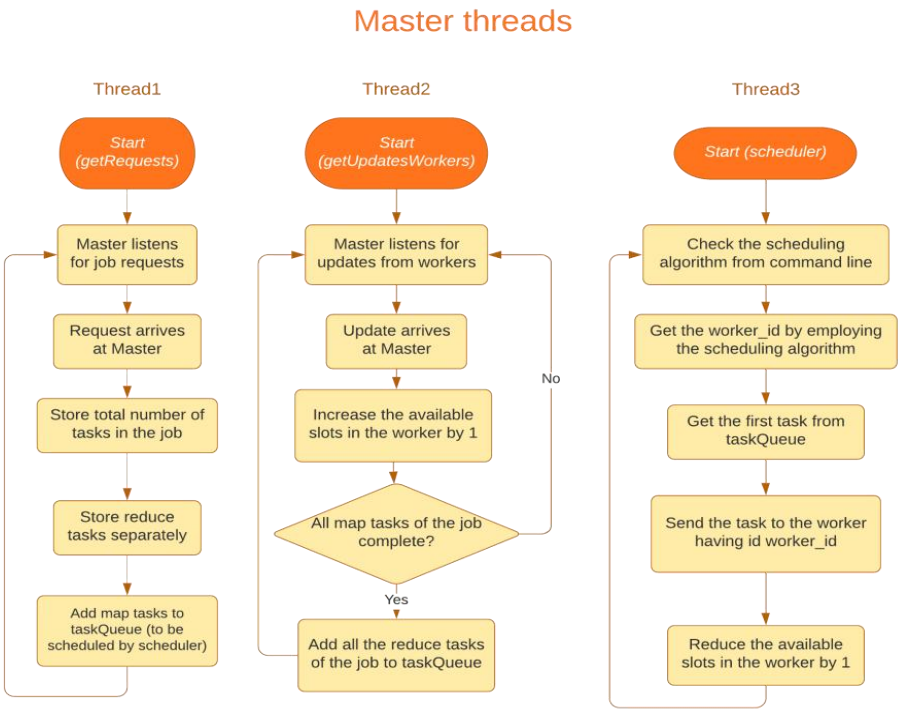
A thread is started to get requests. Once the request arrives at the Master, it parses the job request. If Map Task is not scheduled the Master continues to listen to job requests, otherwise it finds a free slot based on the scheduling algorithm and sends the task to the Worker.

Master Thread 2: getUpdates()

Master listens for updates from Workers. On arrival, free slots are updated. If task satisfies any dependencies they are updated and if they are ready to run the tasks are scheduled. Otherwise, Master continues to listen to updates.

Master Thread 3:scheduler

We check which scheduling algorithm to use from the command line argument. The worker_id is then decided according to the scheduling algorithm that we chose. Then, we get the first task from task queue and send the task to the worker having corresponding worker_id . Every time a task is received, number of slots will be reduced by 1

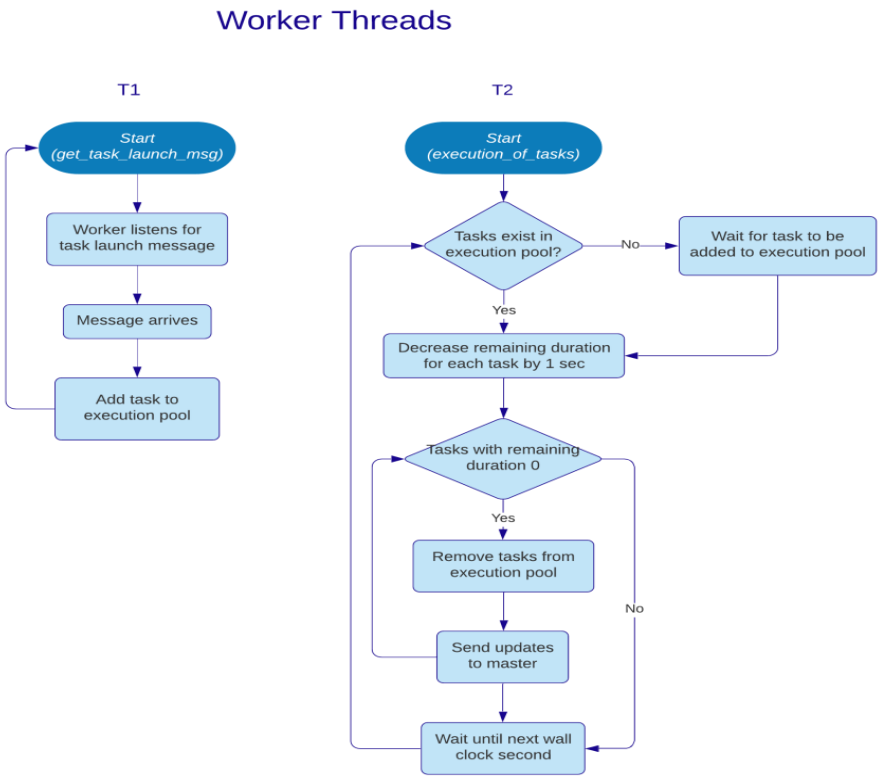


Worker Thread 1

Worker listens for task launch message and on arrival, the task is added to the execution pool

Worker Thread 2

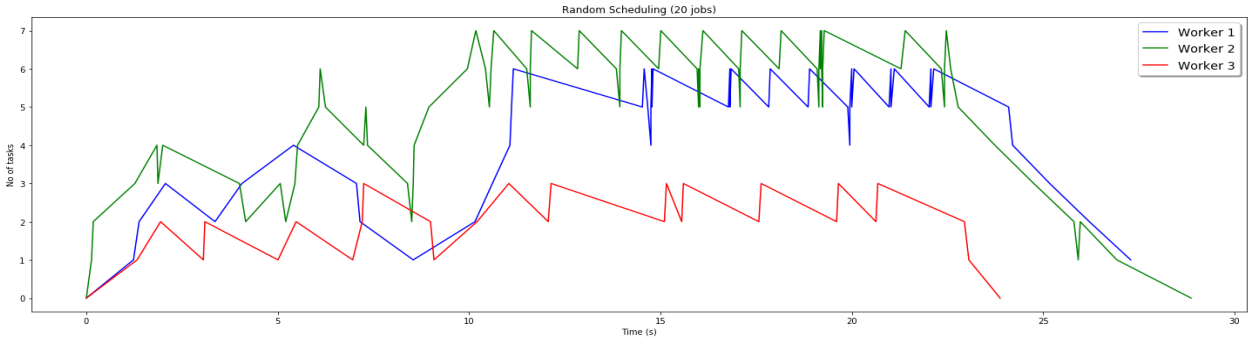
For all the tasks in the execution pool, the remaining duration is reduced by 1. If a task's remaining duration is 0, that task is removed from the execution pool and an update is sent to the master. This repeats every wall clock second.



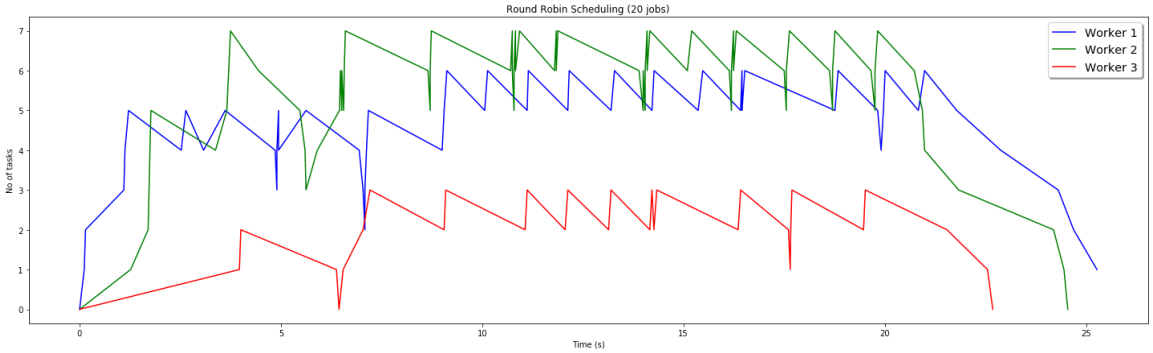
Results

Line plot between time and number of tasks for Random Scheduling, Round Robin scheduling and Least Loaded for 20 jobs with random.seed of 1 in requests.py

Random Scheduling:



Round Robin Scheduling:



Least Loaded Scheduling:

Challenges that we faced:

When all the slots were being utilized , the master stopped assigning further tasks. We solved this by having a separate thread for scheduler which continuously reads a list which contains the tasks to be scheduled

Note:- we faced numerous basic compiletime / runtime errors

Conclusion

We learned about the Master-Worker Framework and how to simulate them using threads and locks, along with socket programming.

EVALUATIONS:

SNo	Name	SRN	Contribution (Individual)
1	Sai Eashwar K S	PES1201801910	30%
2	Ritvik Sanjeev Patil	PES1201801587	30%
3	Monisha Chandra	PES1201802102	20%
4	Teja P	PES1201800408	20%

(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 →)	**YES** (https://github.com/SaiEashwarKS/YACS_BD_Project)
3.	Instructions for building and running the code. Your code must be usable out of the box.	**YES**