

CS322:Big Data

Final Class Project Report

**Project (FPL Analytics / YACS coding): YACS coding**  **Date: 03/12/2020**

|  |  |  |  |
| --- | --- | --- | --- |
| SNo | Name | SRN | Class/Section |
| 1 | Prithwish Nag | PES1201800405 | I |
| 2 | D Aishwarya | PES1201801075 | H |
| 3 | Varshini G | PES1201801630 | H |
| 4 | Siri L Y | PES1201801928 | H |

## Introduction

It is a centralized scheduling framework. The framework 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 processrunning on each of them **.** The Master process makes scheduling decisions while the Worker processes execute the tasks and inform the Master when a task completes its execution.

Threads are used to concurrently execute the connection between master and worker , , and , master and client. Scheduling algorithms are used to find the free slots in the worker machine so that the tasks can be assigned to it. TCP connection is maintained for the transaction of messages between client, master and worker. Log info is calculated from the workers and master to analyse the data.

## Related work

<http://www.uobabylon.edu.iq/download/M.S%202013-2014/Operating_System_Concepts,_8th_Edition%5BA4%5D.pdf> (**Operating System Concepts**)

**StackOverflow** - For error correction.

**TutorialsPoint** - threading and socket modules.

## Design

* Client - request.py
* Master - Master.py < config file > < Scheduling algorithm>
* Worker - Worker.py < port > < worker ID>

Master creates a connection to the client using a thread to listen for incoming jobs from the client. Another thread is used for creating connections between Master and Worker to check for any acknowledgement of the completed task. Three scheduling algorithms are implemented namely, Random , RoundRobin and Least loaded , using a thread which also schedules the map tasks to be sent to the Worker before the reduce tasks. A timer is set to log the number of tasks going on in the worker.

Worker creates a connection to the Master for receiving the task to be done and writes the information log in a file. Using one thread for completing each incoming tasks and logging it to a file.

Five terminals are opened , each acting as , Client , Master and three Workers. The log files produced by Master and Workers are then used to find mean, median and perform the specified analysis.

## Results

Each Worker terminal specifies the task done by it and the Master terminal specifies the sending tasks and acknowledgement from the Worker of the completed task.

Performance of three algorithms are compared by using the log files generated and the graph plotted.

## Problems

Errors were caused due to threading.

## Conclusion

Learnt about threads and socket connections.

Learnt the working of Master Slave architecture.

## EVALUATIONS:

|  |  |  |  |
| --- | --- | --- | --- |
| SNo | Name | SRN | Contribution (Individual) |
| 1 | Prithwish Nag | PES1201800405 | Master Worker |
| 2 | D Aishwarya | PES1201801075 | Master Worker |
| 3 | Varshini G | PES1201801630 | Analytics |
| 4 | Siri LY | PES1201801928 | Analytics |

## (Leave this for the faculty)

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

## CHECKLIST:

|  |  |  |
| --- | --- | --- |
| SNo | Item | Status |
| 1. | Source code documented |  |
| 2. | Source code uploaded to GitHub – (access link for the same, to be added in status 🡪) |  |
| 3. | Instructions for building and running the code. Your code must be usable out of the box. |  |