CS322:Big Data

Final Class Project Report

Project (FPL Analytics / YACS coding): YACS Coding **Date: 01/12/2020**

SNo	Name	SRN	Class/Section
1	Gaurika Poplai	PES1201800374	5 I
2	Keerthana Mahadev	PES1201800768	5 E
3	Nikita J. Raj	PES1201800808	5 H
4	Rahul KR	PES1201802064	5 J

Introduction

The aim of this project was to simulate a centralized job scheduler with 1 master process and 3 worker process (forming a cluster). The master process runs on one machine while the other 3 workers work on 3 different machines. Each of the worker machines have fixed no. of slots depending on the machine's capacity.

Each job is divided into multiple tasks (map and reduce tasks), such that each task can be executed on one of the slots completely. Map tasks are executed first followed by the reduce tasks.

The master process manages the cluster by keeping track of all the slots in each worker machine. It processes the job requests and assigns the tasks to one of the available slots based on the chosen scheduling algorithm.

The worker process running on each worker machine is responsible for starting the execution of the tasks in their respective slots by allocating resources and freeing the resources once the task is complete. It's also responsible for sending updates to the worker once the task is completed.

Related work

- https://realpython.com/intro-to-python-threading/#starting-a-thread To understand threading concept and implementation in python
- https://realpython.com/python-sockets/ To get a handle on socket connection in python

Design

Master has 2 threads:

1) One thread listens for job requests through port 5000
2) Another thread to listen for updates (on task completion) from workers through port 5001

Master can schedule tasks using the various scheduling algorithms

Scheduling algorithms:

1) Least loaded:

   Master node checks the no. of available slots in each machine and picks the one with the highest no. of free slots for task launch

2) Random:

   Master node selects a machine at random to launch a task. If the machine has no free slots, selects another machine at random.

3) Round Robin:

   worker machines are ordered based on their worker_id. Round robin method is used to pick a worker node for task launch. If the selected worker has no free slots, the next worker in ordering is picked

Worker has 2 threads:

1) One thread to listen for task launch messages from master through port 4000
2) Another thread to start task executions in their respective slots and send updates to the master

Results

Discuss the results you got. What inferences could you draw from the results? Was any result unexpected? Any fine-tuning done to parameters so that the results changed?

Problems

Mention problems faced and how were they solved

FPL:

- streaming the data
- Converting dstreams to dataframes
- Tracking and updating the data over the course of different computations

Switched to YACS

YACS:

- Deciding on an efficient data structure
- Connection reset by peer error, keeping track of socket connections
- Thinking along the lines of multithreading, monitoring the activities of each thread

Conclusion

Main learning from this project:

- This project provided a clear and practical understanding of how hadoop and spark execute map-reduce tasks using Yarn.
- Understood Yarn's master-slave architecture and its implementation.
- Improved understanding of socket connections
- Proper implementation of multithreading, giving insight into its widespread applications

EVALUATIONS:

SNo	Name	SRN	Contribution (Individual)
1	Gaurika Poplai	PES1201800374	
2	Keerthana Mahadev	PES1201800768	
3	Nikita J. Raj	PES1201800808	
4	Rahul KR	PES1201802064	

(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.	