National College of Ireland

Optimizing GPU Resource Allocation and Scheduling using a Hybrid Scheduler

MSc Research Project
Cloud Computing

Sai Nitish Kavali
Student ID: 22125809

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Sai Nitish Kavali
Student ID:	22125809
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	14/12/2023
Project Title:	Optimizing GPU Resource Allocation and Scheduling using a Hybrid Scheduler
Word Count:	358
Page Count:	3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Nitish Kavali
Date:	29th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing GPU Resource Allocation and Scheduling using a Hybrid Scheduler

Sai Nitish Kavali
22125809

1 Introduction

This is the Hybrid Scheduler configuration manual, a powerful GPU resource manager. This manual guides you through hybrid scheduler setup and configuration to maximize GPU utilization and system efficiency. The purpose of this configuration manual is to give a detailed and brief explanation of the setup and running of the project "Optimizing GPU Resource Allocation and Scheduling using a Hybrid Scheduler."

2 System Specification

- Programming Languages: Python, HTML
- Libraries and Frameworks: Flask, GPUStat, Pytorch, Watchmen,

2.1 System configuaration

Configuration of the system is as follows: g4dn.12xlarge EC2 Instance

- Windows Server
- vCPU 48
- 4 NVIDIA Tesla GPUs

3 Implementation Steps

- First, Download the code artifact submitted in the moodle
- Create an EC2 instance in the AWS with the server g4dn.12xlarge or any large server than the specified as the scheduler requires 4 GPU to effectively display scheduling.
- Install all the required libraries and Python 3 to execute the code without any issues
- Start the server with the command python server.py.
- After starting the server you can find a URL that redirects to the frontend web app.

```
APScheduler
Flask
gpustat
pydantic
requests
pyreadline
torchvision
```

Figure 1: Requirements

```python
import os
import sys
import json
import queue
import logging
import readline
import datetime
import argparse
import threading
from collections import OrderedDict

from flask import Flask, jsonify, request, render_template
from flask.helpers import make_response
from json import JSONEncoder
from apscheduler.schedulers.blocking import BlockingScheduler

from listener import (
    is_single_gpu_totally_free,
    check_gpus_existence,
    check_req_gpu_num,
    GPUInfo
)
from client import (
    ClientStatus,
    ClientMode,
    ClientModel,
    ClientCollection
)
```

Figure 2: Requirements

```
from __future__ import print_function

import sys
import argparse

import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms
from torch.optim.lr_scheduler import StepLR

from watchmen import WatchClient
from watchmen.client import ClientMode
```

Figure 3: Requirements

- Now open new terminals in order to simulate the workload on the GPUs.
- After starting the server and viewing the frontend now we move on to training the CNN models which will start the GPUs so that we can test the scheduler.
- Now we start increasing the load on the GPU using python mnist.py –id=”single” –cuda=0 –wait
- run the commands in different terminals with different parameters which help in different kinds of scheduling. (NOTE: Change the ID every time you run the command).
- In the URL you can see how the scheduler is working and what all the tasks are running on different GPUs and their utilization.

This concludes the configuration manual for the Hybrid Scheduler. Please refer to the readme.txt file for any doubts related to the parameters related to commands and reach out to me for any difficulties.