Big Data Coursework - Questions

Data Processing and Machine Learning in the Cloud

This is the **INM432 Big Data coursework 2024**. This coursework contains extended elements of **theory** and **practice**, mainly around parallelisation of tasks withSpark and a bit about parallel training using TensorFlow.

Code and Report

Your tasks parallelization of tasks in PySpark, extension, evaluation, and theoretical reflection.
Please complete and submit the **coding tasks** in a copy of **this notebook**. Write your code in the **indicated cells** and **include** the **output** in the submitted notebook.
Make sure that **your code contains comments** on its **stucture** and explanations of its **purpose**.

Provide also a **report** with the **textual answers in a separate document**.
Include **screenshots** from the Google Cloud web interface (don't use the SCREENSHOT function that Google provides, but take a picture of the graphs you see for the VMs) and result tables, as well as written text about the analysis.

Submission

Download and submit **your version of this notebook** as an **.ipynb** file and also submit a **shareable link** to your notebook on Colab in your report (created with the Colab 'Share' function) (**and don't change the online version after submission**).

Further, provide your **report as a PDF document**. **State the number of words** in the document at the end. The report should **not have more than 2000 words**.

Please also submit **a PDF of your Jupyter notebook**.

Introduction and Description

This coursework focuses on parallelisation and scalability in the cloud with Spark and TesorFlow/Keras. We start with code based on **lessons 3 and 4** of the *Fast and Lean Data Science* course by Martin Gorner. The course is based on Tensorflow for data processing and MachineLearning. Tensorflow's data processing approach is somewhat similar to that of Spark, but you don't need to study Tensorflow, just make sure you understand the high-level structure. What we will do here is **parallelising pre-processing**, and **measuring** performance, and we will perform **evaluation** and **analysis** on the cloud performance, as well as **theoretical discussion**.

This coursework contains **3 sections**.

Section 0

This section just contains some necessary code for setting up the environment. It has no tasks for you (but do read the code and comments).

Section 1

Section 1 is about preprocessing a set of image files. We will work with a public dataset “Flowers” (3600 images, 5 classes). This is not a vast dataset, but it keeps the tasks more manageable for development and you can scale up later, if you like.

In **'Getting Started'** we will work through the data preprocessing code from *Fast and Lean Data Science* which uses TensorFlow's `tf.data` package. There is no task for you here, but you will need to re-use some of this code later.

In **Task 1** you will **parallelise the data preprocessing in Spark**, using Google Cloud (GC) Dataproc. This involves adapting the code from 'Getting Started' to use Spark and running it in the cloud.

Section 2

In **Section 2** we are going to **measure the speed of reading data** in the cloud. In **Task 2** we will **paralellize the measuring** of different configurations **using Spark**.

Section 3

This section is about the theoretical discussion, based on one paper, in **Task 3**. The answers should be given in the PDF report.

General points

For **all coding tasks**, take the **time of the operations** and for the cloud operations, get performance **information from the web interfaces** for your reporting and analysis.

The **tasks** are **mostly independent** of each other. The later tasks can mostly be addressed without needing the solution to the earlier ones.

Section 0: Set-up

As usual, you need to run the **imports and authentication every time you work with this notebook**. Use the **local Spark** installation for development before you send jobs to the cloud.

Read through this section once and **fill in the project ID the first time**, then you can just step straight throught this at the beginning of each session - except for the two authentication cells.

Imports

We import some **packages that will be needed throughout**. For the **code that runs in the cloud**, we will need **separate import sections** that will need to be partly different from the one below.

```
import os, sys, math
import numpy as np
import scipy as sp
import scipy.stats
```

```
import time
import datetime
import string
import random
from matplotlib import pyplot as plt
import tensorflow as tf
print("Tensorflow version " + tf.__version__)
import pickle
```

```
Tensorflow version 2.15.0
```

Cloud and Drive authentication

This is for **authenticating with with GCS Google Drive**, so that we can create and use our own buckets and access Dataproc and AI-Platform.

This section **starts with the two interactive authentications**.

First, we mount Google Drive for persistent local storage and create a directory `DB-CW` thay you can use for this work. Then we'll set up the cloud environment, including a storage bucket.

```
print('Mounting google drive...')
from google.colab import drive
drive.mount('/content/drive')
# %cd "/content/drive/MyDrive"
# !mkdir BD-CW
# %cd "/content/drive/MyDrive/BD-CW"
```

```
Mounting google drive...
Mounted at /content/drive
```

Next, we authenticate with the GCS to enable access to Dataproc and AI-Platform.

```
import sys
if 'google.colab' in sys.modules:
    from google.colab import auth
    auth.authenticate_user()
```

It is useful to **create a new Google Cloud project** for this coursework. You can do this on the GC Console page by clicking on the entry at the top, right of the *Google Cloud Platform* and choosing *New Project*. **Copy** the **generated project ID** to the next cell. Also **enable billing** and the **Compute, Storage and Dataproc** APIs like we did during the labs.

We also specify the **default project and region**. The REGION should be `us-central1` as that seems to be the only one that reliably works with the free credit. This way we don't have to specify this information every time we access the cloud.

```
# bd-coursework-421223
```

```
PROJECT = 'bd-coursework-421223'  ### USE YOUR GOOGLE CLOUD PROJECT ID HERE. ###
!gcloud config set project $PROJECT
# REGION = 'us-central1'
REGION = 'us-west1'
CLUSTER = '{}-cluster'.format(PROJECT)
!gcloud config set compute/region $REGION
!gcloud config set dataproc/region $REGION

!gcloud config list # show some information
```

```
Updated property [core/project].
WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
Updated property [dataproc/region].
[component_manager]
disable_update_check = True
[compute]
region = us-west1
[core]
account = baibhav.datta2024@gmail.com
project = bd-coursework-421223
[dataproc]
region = us-west1

Your active configuration is: [default]
```

With the cell below, we **create a storage bucket** that we will use later for **global storage**. If the bucket exists you will see a "ServiceException: 409 ...", which does not cause any problems. **You must create your own bucket to have write access.**

```
BUCKET = 'gs://{}-storage'.format(PROJECT)
!gsutil mb $BUCKET
```

```
Creating gs://bd-coursework-421223-storage/...
ServiceException: 409 A Cloud Storage bucket named 'bd-coursework-421223-storage' already exists. Try another name. Bucket names must be globally unique across all Google Cloud projects, including those outside of your organization.
```

The cell below just **defines some routines for displaying images** that will be **used later**. You can see the code by double-clicking, but you don't need to study this.

```
#@title Utility functions for image display **[RUN THIS TO ACTIVATE]**
{ display-mode: "form" }
def display_9_images_from_dataset(dataset):
  plt.figure(figsize=(13,13))
  subplot=331
  for i, (image, label) in enumerate(dataset):
```

```
    plt.subplot(subplot)
    plt.axis('off')
    plt.imshow(image.numpy().astype(np.uint8))
    plt.title(str(label.numpy()), fontsize=16)
    # plt.title(label.numpy().decode(), fontsize=16)
    subplot += 1
    if i==8:
      break
  plt.tight_layout()
  plt.subplots_adjust(wspace=0.1, hspace=0.1)
  plt.show()

def display_training_curves(training, validation, title, subplot):
  if subplot%10==1: # set up the subplots on the first call
    plt.subplots(figsize=(10,10), facecolor='#F0F0F0')
    plt.tight_layout()
  ax = plt.subplot(subplot)
  ax.set_facecolor('#F8F8F8')
  ax.plot(training)
  ax.plot(validation)
  ax.set_title('model '+ title)
  ax.set_ylabel(title)
  ax.set_xlabel('epoch')
  ax.legend(['train', 'valid.'])

def dataset_to_numpy_util(dataset, N):
    dataset = dataset.batch(N)
    for images, labels in dataset:
        numpy_images = images.numpy()
        numpy_labels = labels.numpy()
        break;
    return numpy_images, numpy_labels

def title_from_label_and_target(label, correct_label):
  correct = (label == correct_label)
  return "{} [{}{}{}]".format(CLASSES[label], str(correct), ', shoud
be ' if not correct else '',
                              CLASSES[correct_label] if not correct
else ''), correct

def display_one_flower(image, title, subplot, red=False):
    plt.subplot(subplot)
    plt.axis('off')
    plt.imshow(image)
    plt.title(title, fontsize=16, color='red' if red else 'black')
    return subplot+1

def display_9_images_with_predictions(images, predictions, labels):
  subplot=331
  plt.figure(figsize=(13,13))
```

```
  classes = np.argmax(predictions, axis=-1)
  for i, image in enumerate(images):
    title, correct = title_from_label_and_target(classes[i],
labels[i])
    subplot = display_one_flower(image, title, subplot, not correct)
    if i >= 8:
      break;

  plt.tight_layout()
  plt.subplots_adjust(wspace=0.1, hspace=0.1)
  plt.show()
```

Install Spark locally for quick testing

You can use the cell below to **install Spark locally on this Colab VM** (like in the labs), to do quicker small-scale interactive testing. Using Spark in the cloud with **Dataproc is still required for the final version**.

```
%cd
!apt-get update -qq
!apt-get install openjdk-8-jdk-headless -qq >> /dev/null # send any
output to null device
!tar -xzf "/content/drive/My Drive/Big_Data/data/spark/spark-3.5.0-
bin-hadoop3.tgz" # unpack

!pip install -q findspark
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/root/spark-3.5.0-bin-hadoop3"
import findspark
findspark.init()
import pyspark
print(pyspark.__version__)
sc = pyspark.SparkContext.getOrCreate()
print(sc)
```

```
/root
3.5.0

/usr/lib/python3.10/subprocess.py:1796: RuntimeWarning: os.fork() was
called. os.fork() is incompatible with multithreaded code, and JAX is
multithreaded, so this will likely lead to a deadlock.
  self.pid = _posixsubprocess.fork_exec(

<SparkContext master=local[*] appName=pyspark-shell>
```

```
os.environ["PATH"] += ":/root/spark-3.5.0-bin-hadoop3/bin"
```

Section 1: Data pre-processing

This section is about the **pre-processing of a dataset** for deep learning. We first look at a ready-made solution using Tensorflow and then we build a implement the same process with Spark. The tasks are about **parallelisation** and **analysis** the performance of the cloud implementations.

1.1 Getting started

In this section, we get started with the data pre-processing. The code is based on lecture 3 of the 'Fast and Lean Data Science' course.

This code is using the TensorFlow `tf.data` package, which supports map functions, similar to Spark. Your **task** will be to **re-implement the same approach in Spark**.

We start by **setting some variables for the *Flowers* dataset**.

```
GCS_PATTERN = 'gs://flowers-public/*/*.jpg' # glob  pattern for input
files
PARTITIONS = 16 # no of partitions we will use later
TARGET_SIZE = [192, 192] # target resolution for the images
CLASSES = [b'daisy', b'dandelion', b'roses', b'sunflowers', b'tulips']
    # labels for the data
```

We **read the image files** from the public GCS bucket that contains the *Flowers* dataset. **TensorFlow** has **functions** to execute glob patterns that we use to calculate the the number of images in total and per partition (rounded up as we cannont deal with parts of images).

```
nb_images = len(tf.io.gfile.glob(GCS_PATTERN)) # number of images
partition_size = math.ceil(1.0 * nb_images / PARTITIONS) # images per
partition (float)
print("GCS_PATTERN matches {} images, to be divided into {} partitions
with up to {} images each.".format(nb_images, PARTITIONS,
partition_size))
```

```
GCS_PATTERN matches 3670 images, to be divided into 16 partitions with
up to 230 images each.
```

Map functions

In order to read use the images for learning, they need to be **preprocessed** (decoded, resized, cropped, and potentially recompressed). Below are **map functions** for these steps. You **don't need to study** the **internals of these functions** in detail.

```
def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
```

```
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2

def resize_and_crop_image(image, label):
    # Resizes and cropd using "fill" algorithm:
    # always make sure the resulting image is cut out from the source
image
    # so that it fills the TARGET_SIZE entirely with no black bars
    # and a preserved aspect ratio.
    w = tf.shape(image)[0]
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,
                    lambda: tf.image.resize(image, [w*tw/w, h*tw/w]),
# if true
                    lambda: tf.image.resize(image, [w*th/h, h*th/h])
# if false
                    )
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def recompress_image(image, label):
    # this reduces the amount of data, but takes some time
    image = tf.cast(image, tf.uint8)
    image = tf.image.encode_jpeg(image, optimize_size=True,
chroma_downsampling=False)
    return image, label
```

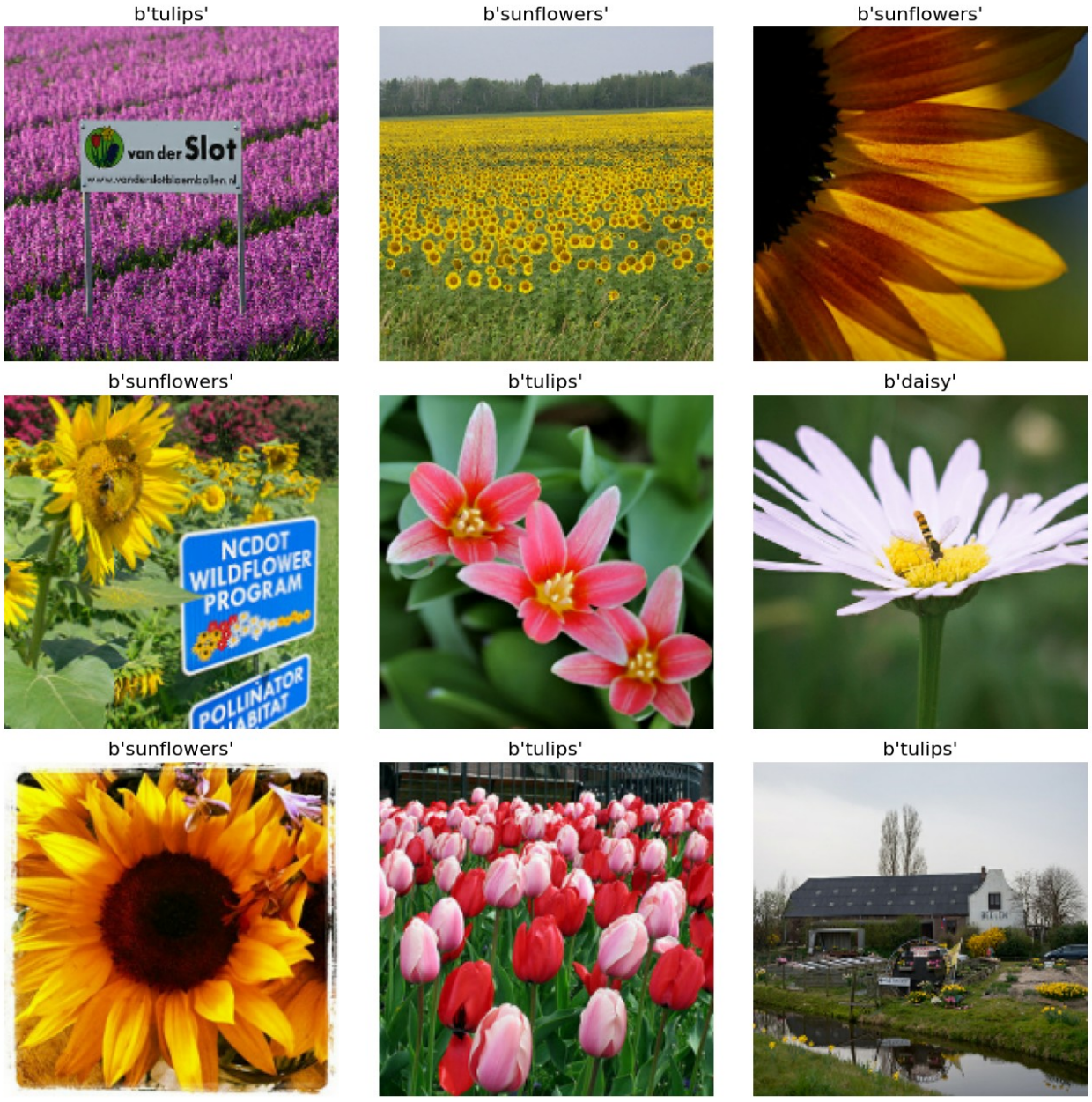
With `tf.data`, we can apply decoding and resizing as map functions.

```
dsetFiles = tf.data.Dataset.list_files(GCS_PATTERN) # This also
shuffles the images
dsetDecoded = dsetFiles.map(decode_jpeg_and_label)
dsetResized = dsetDecoded.map(resize_and_crop_image)
```

We can also look at some images using the image display function defined above (the one with the hidden code).

```
display_9_images_from_dataset(dsetResized)
```

```
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
<class 'tensorflow.python.framework.ops.EagerTensor'>
```



Now, let's test continuous reading from the dataset. We can see that reading the first 100 files already takes some time.

```
sample_set = dsetResized.batch(10).take(10) # take 10 batches of 10
images for testing
for image, label in sample_set:
    print("Image batch shape {}, {})".format(image.numpy().shape,
        [lbl.decode('utf8') for lbl in label.numpy()]))
```

```
Image batch shape (10, 192, 192, 3), ['roses', 'roses', 'daisy',
'tulips', 'dandelion', 'tulips', 'tulips', 'roses', 'roses',
'dandelion'])
Image batch shape (10, 192, 192, 3), ['dandelion', 'tulips',
'dandelion', 'daisy', 'dandelion', 'roses', 'sunflowers', 'dandelion',
'tulips', 'tulips'])
Image batch shape (10, 192, 192, 3), ['daisy', 'tulips', 'daisy',
'sunflowers', 'tulips', 'roses', 'dandelion', 'daisy', 'dandelion',
'daisy'])
Image batch shape (10, 192, 192, 3), ['dandelion', 'dandelion',
'sunflowers', 'roses', 'roses', 'sunflowers', 'dandelion', 'tulips',
'tulips', 'daisy'])
Image batch shape (10, 192, 192, 3), ['dandelion', 'dandelion',
'sunflowers', 'dandelion', 'daisy', 'roses', 'sunflowers', 'roses',
'sunflowers', 'tulips'])
Image batch shape (10, 192, 192, 3), ['daisy', 'dandelion',
'dandelion', 'roses', 'tulips', 'dandelion', 'daisy', 'roses',
'tulips', 'roses'])
Image batch shape (10, 192, 192, 3), ['dandelion', 'sunflowers',
'daisy', 'daisy', 'dandelion', 'roses', 'roses', 'daisy', 'daisy',
'daisy'])
Image batch shape (10, 192, 192, 3), ['roses', 'sunflowers',
'dandelion', 'roses', 'dandelion', 'daisy', 'sunflowers', 'roses',
'daisy', 'tulips'])
Image batch shape (10, 192, 192, 3), ['sunflowers', 'tulips',
'sunflowers', 'tulips', 'tulips', 'tulips', 'tulips', 'tulips',
'tulips', 'tulips'])
Image batch shape (10, 192, 192, 3), ['dandelion', 'daisy',
'dandelion', 'daisy', 'daisy', 'daisy', 'dandelion', 'sunflowers',
'dandelion', 'roses'])
```

1.2 Improving Speed

Using individual image files didn't look very fast. The 'Lean and Fast Data Science' course introduced **two techniques to improve the speed**.

Recompress the images

By **compressing** the images in the **reduced resolution** we save on the size. This **costs some CPU time** upfront, but **saves network and disk bandwith**, especially when the data are **read multiple times**.

```
# This is a quick test to get an idea how long recompressions takes.
dataset4 = dsetResized.map(recompress_image)
test_set = dataset4.batch(10).take(10)
for image, label in test_set:
    print("Image batch shape {}, {})".format(image.numpy().shape,
[lbl.decode('utf8') for lbl in label.numpy()]))
```

```
Image batch shape (10,), ['roses', 'tulips', 'roses', 'dandelion',
'tulips', 'daisy', 'roses', 'daisy', 'dandelion', 'dandelion'])
Image batch shape (10,), ['tulips', 'roses', 'dandelion',
'sunflowers', 'roses', 'sunflowers', 'dandelion', 'sunflowers',
'dandelion', 'sunflowers'])
Image batch shape (10,), ['sunflowers', 'dandelion', 'dandelion',
'tulips', 'roses', 'sunflowers', 'roses', 'tulips', 'tulips',
'dandelion'])
Image batch shape (10,), ['daisy', 'daisy', 'tulips', 'sunflowers',
'sunflowers', 'tulips', 'daisy', 'dandelion', 'roses', 'roses'])
Image batch shape (10,), ['tulips', 'tulips', 'dandelion', 'tulips',
'tulips', 'tulips', 'daisy', 'roses', 'tulips', 'tulips'])
Image batch shape (10,), ['daisy', 'sunflowers', 'daisy', 'tulips',
'dandelion', 'daisy', 'roses', 'dandelion', 'tulips', 'daisy'])
Image batch shape (10,), ['sunflowers', 'dandelion', 'sunflowers',
'tulips', 'tulips', 'dandelion', 'roses', 'tulips', 'dandelion',
'tulips'])
Image batch shape (10,), ['dandelion', 'sunflowers', 'roses',
'tulips', 'sunflowers', 'sunflowers', 'daisy', 'dandelion', 'tulips',
'sunflowers'])
Image batch shape (10,), ['dandelion', 'tulips', 'daisy', 'daisy',
'dandelion', 'sunflowers', 'dandelion', 'roses', 'roses', 'daisy'])
Image batch shape (10,), ['daisy', 'dandelion', 'daisy', 'dandelion',
'daisy', 'tulips', 'roses', 'sunflowers', 'tulips', 'dandelion'])
```

Write the dataset to TFRecord files

By writing **multiple preprocessed samples into a single file**, we can make further speed gains. We distribute the data over **partitions** to facilitate **parallelisation** when the data are used. First we need to **define a location** where we want to put the file.

```
GCS_OUTPUT = BUCKET + '/tfrecords-jpeg-192x192-2/flowers'  # prefix
for output file names
```

Now we can **write the TFRecord files** to the bucket.

Running the cell takes some time and **only needs to be done once** or not at all, as you can use the publicly available data for the next few cells. For convenience I have commented out the call to `write_tfrecords` at the end of the next cell. You don't need to run it (it takes some time), but you'll need to use the code below later (but there is no need to study it in detail).

There is a **ready-made pre-processed data** versions available here: `gs://flowers-public/tfrecords-jpeg-192x192-2/`, that we can use for testing.

```
# functions for writing TFRecord entries
# Feature values are always stored as lists, a single data element
will be a list of size 1
def _bytestring_feature(list_of_bytestrings):
    return
tf.train.Feature(bytes_list=tf.train.BytesList(value=list_of_bytestrin
gs))

def _int_feature(list_of_ints): # int64
    return
tf.train.Feature(int64_list=tf.train.Int64List(value=list_of_ints))

def to_tfrecord(tfrec_filewriter, img_bytes, label): # Create tf data
records
    class_num = np.argmax(np.array(CLASSES)==label) # 'roses' => 2
(order defined in CLASSES)
    one_hot_class = np.eye(len(CLASSES))[class_num]     # [0, 0, 1, 0,
0] for class #2, roses
    feature = {
        "image": _bytestring_feature([img_bytes]), # one image in the
list
        "class": _int_feature([class_num]) #,      # one class in the
list
    }
    return
tf.train.Example(features=tf.train.Features(feature=feature))

def write_tfrecords(GCS_PATTERN,GCS_OUTPUT,partition_size): # write
the images to files.
    print("Writing TFRecords")
    tt0 = time.time()
    filenames = tf.data.Dataset.list_files(GCS_PATTERN)
    dataset1 = filenames.map(decode_jpeg_and_label)
    dataset2 = dataset1.map(resize_and_crop_image)
    dataset3 = dataset2.map(recompress_image)
    dataset4 = dataset3.batch(partition_size) # partitioning: there
will be one "batch" of images per file
    for partition, (image, label) in enumerate(dataset4):
        # batch size used as partition size here
        partition_size = image.numpy().shape[0]
        # good practice to have the number of records in the filename
        filename = GCS_OUTPUT + "{:02d}-{}.tfrec".format(partition,
partition_size)
        # You need to change GCS_OUTPUT to your own bucket to actually
create new files
        with tf.io.TFRecordWriter(filename) as out_file:
            for i in range(partition_size):
                example = to_tfrecord(out_file,
                                      image.numpy()[i], # re-compressed
image: already a byte string
```

```
                                    label.numpy()[i] #
                                    )
                out_file.write(example.SerializeToString())
        print("Wrote file {} containing {} records".format(filename,
partition_size))
    print("Total time: "+str(time.time()-tt0))

write_tfrecords(GCS_PATTERN,GCS_OUTPUT,partition_size) # uncomment to
run this cell
```

Test the TFRecord files

We can now **read from the TFRecord files**. By default, we use the files in the public bucket. Comment out the 1st line of the cell below to use the files written in the cell above.

```
# GCS_OUTPUT = 'gs://flowers-public/tfrecords-jpeg-192x192-2/'
# remove the line above to use your own files that you generated above

def read_tfrecord(example):
    features = {
        "image": tf.io.FixedLenFeature([], tf.string),  # tf.string =
bytestring (not text string)
        "class": tf.io.FixedLenFeature([], tf.int64) #,   # shape []
means scalar
    }
    # decode the TFRecord
    example = tf.io.parse_single_example(example, features)
    image = tf.image.decode_jpeg(example['image'], channels=3)
    image = tf.reshape(image, [*TARGET_SIZE, 3])
    class_num = example['class']
    return image, class_num

def load_dataset(filenames):
    # read from TFRecords. For optimal performance, read from multiple
    # TFRecord files at once and set the option
experimental_deterministic = False
    # to allow order-altering optimizations.
    option_no_order = tf.data.Options()
    option_no_order.experimental_deterministic = False

    dataset = tf.data.TFRecordDataset(filenames)
    dataset = dataset.with_options(option_no_order)
    dataset = dataset.map(read_tfrecord)
    return dataset


filenames = tf.io.gfile.glob(GCS_OUTPUT + "*.tfrec")
datasetTfrec = load_dataset(filenames)
```

Let's have a look **if reading from the TFRecord** files is **quicker**.

```
batched_dataset = datasetTfrec.batch(10)
sample_set = batched_dataset.take(10)
for image, label in sample_set:
    print("Image batch shape {}, {})".format(image.numpy().shape, \
                        [str(lbl) for lbl in label.numpy()]))
```

```
Image batch shape (10, 192, 192, 3), ['3', '2', '4', '0', '0', '3',
'1', '0', '1', '4'])
Image batch shape (10, 192, 192, 3), ['3', '1', '1', '0', '1', '3',
'2', '4', '2', '2'])
Image batch shape (10, 192, 192, 3), ['1', '2', '3', '1', '3', '3',
'0', '1', '3', '0'])
Image batch shape (10, 192, 192, 3), ['3', '4', '4', '3', '1', '3',
'0', '2', '4', '3'])
Image batch shape (10, 192, 192, 3), ['0', '3', '2', '4', '0', '1',
'0', '4', '1', '4'])
Image batch shape (10, 192, 192, 3), ['1', '4', '1', '1', '0', '4',
'0', '2', '0', '2'])
Image batch shape (10, 192, 192, 3), ['3', '4', '3', '2', '3', '4',
'1', '1', '1', '0'])
Image batch shape (10, 192, 192, 3), ['4', '0', '1', '0', '3', '1',
'3', '0', '1', '0'])
Image batch shape (10, 192, 192, 3), ['2', '2', '2', '1', '4', '4',
'3', '4', '2', '4'])
Image batch shape (10, 192, 192, 3), ['3', '1', '2', '0', '3', '1',
'2', '3', '4', '1'])
```

Wow, we have a **massive speed-up**! The repackageing is worthwhile :-)

Task 1: Write TFRecord files to the cloud with Spark (40%)

Since recompressing and repackaging is very effective, we would like to be able to do it inparallel for large datasets. This is a relatively straightforward case of **parallelisation**. We will **use Spark to implement** the same process as above, but in parallel.

1a) Create the script (14%)

Re-implement the pre-processing in Spark, using Spark mechanisms for **distributing** the workload **over multiple machines**.

You need to:

i) **Copy** over the **mapping functions** (see section 1.1) and **adapt** the resizing and recompression functions **to Spark** (only one argument). (3%)

ii) **Replace** the TensorFlow **Dataset objects with RDDs**, starting with an RDD that contains the list of image filenames. (3%)

iii) **Sample** the the RDD to a smaller number at an appropriate position in the code. Specify a sampling factor of 0.02 for short tests. (1%)

iv) Then **use the functions from above** to write the TFRecord files. (3%)

v) The code for **writing to the TFRecord files** needs to be put into a function, that can be applied to every partition with the 'RDD.mapPartitionsWithIndex' function. The return value of that function is not used here, but you should return the filename, so that you have a list of the created TFRecord files. (4%)

```
#TASK 1.a.i

#The functions to be mapped to rdd

def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2

def resize_and_crop_image(data):
    image, label=data
    # Resizes and cropd using "fill" algorithm:
    # always make sure the resulting image is cut out from the source
image
    # so that it fills the TARGET_SIZE entirely with no black bars
    # and a preserved aspect ratio.
    w = tf.shape(image)[0]
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,
                    lambda: tf.image.resize(image, [w*tw/w, h*tw/w]),
# if true
                    lambda: tf.image.resize(image, [w*th/h, h*th/h])
# if false
                    )
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def recompress_image(data):
    image, label=data
    # this reduces the amount of data, but takes some time
    image = tf.cast(image, tf.uint8)
```

```
    image = tf.image.encode_jpeg(image, optimize_size=True,
chroma_downsampling=False)
    return image, label

GCS_PATTERN = 'gs://flowers-public/*/*.jpg' # glob  pattern for input
files
PARTITIONS = 16 # no of partitions we will use later
TARGET_SIZE = [192, 192] # target resolution for the images
CLASSES = [b'daisy', b'dandelion', b'roses', b'sunflowers', b'tulips']
    # labels for the data


nb_images = len(tf.io.gfile.glob(GCS_PATTERN)) # number of images
partition_size = math.ceil(1.0 * nb_images / PARTITIONS) # images per
partition (float)

#TASK 1.a.ii Creating the RDDs, Applying the functions of
preprocessing

### TASK 1d ###
dsetRDD = sc.parallelize(tf.io.gfile.glob(GCS_PATTERN),PARTITIONS)

dsetDecoded=dsetRDD.map(decode_jpeg_and_label)
dsetResized=dsetDecoded.map(resize_and_crop_image)
dsetRecompressed = dsetResized.map(recompress_image)

#TASK 1.a.iii Sampling the RDD

dsetSampled=dsetRecompressed.sample(False,0.02)

#TASK 1.a.iv and 1.a.v Adapting the write_tfrecords function
appropriate for RDDs

GCS_OUTPUT = BUCKET + '/tfrecordsNEW-jpeg-192x192-2/flowers'  # prefix
for output file names
# functions for writing TFRecord entries
# Feature values are always stored as lists, a single data element
will be a list of size 1
def _bytestring_feature(list_of_bytestrings):
    return
tf.train.Feature(bytes_list=tf.train.BytesList(value=list_of_bytestrin
gs))

def _int_feature(list_of_ints): # int64
    return
tf.train.Feature(int64_list=tf.train.Int64List(value=list_of_ints))

def to_tfrecord(tfrec_filewriter, img_bytes, label): # Create tf data
records
    class_num = np.argmax(np.array(CLASSES)==label) # 'roses' => 2
(order defined in CLASSES)
```

```
    one_hot_class = np.eye(len(CLASSES))[class_num]     # [0, 0, 1, 0,
0] for class #2, roses
    feature = {
        "image": _bytestring_feature([img_bytes]), # one image in the
list
        "class": _int_feature([class_num]) #,      # one class in the
list
    }
    return
tf.train.Example(features=tf.train.Features(feature=feature))

def write_tfrecords(partition_index, iterator):
  partition=partition_index
  global partition_size
  filename = GCS_OUTPUT + "{:02d}-{}.tfrec".format(partition,
partition_size)
  with tf.io.TFRecordWriter(filename) as out_file:
    for image,label in iterator:
      example = to_tfrecord(out_file,image.numpy(),
                            label.numpy().decode('utf-8')
                                          )
      out_file.write(example.SerializeToString())
  return [filename]

#Applying the function to write tfrecords to each partition

TFRecord_filenames=dsetSampled.mapPartitionsWithIndex(write_tfrecords)

#collecting the filenames

output_files = TFRecord_filenames.collect()
```

1b) Testing (3%)

i) Read from the TFRecord Dataset, using `load_dataset` and `display_9_images_from_dataset` to test.

```
def read_tfrecord(example):
    features = {
        "image": tf.io.FixedLenFeature([], tf.string),  # tf.string =
bytestring (not text string)
        "class": tf.io.FixedLenFeature([], tf.int64) #,   # shape []
means scalar
    }
    # decode the TFRecord
    example = tf.io.parse_single_example(example, features)
    image = tf.image.decode_jpeg(example['image'], channels=3)
    image = tf.reshape(image, [*TARGET_SIZE, 3])
    class_num = example['class']
    return image, class_num
```

```
def load_dataset(filenames):
    # read from TFRecords. For optimal performance, read from multiple
    # TFRecord files at once and set the option
experimental_deterministic = False
    # to allow order-altering optimizations.
    option_no_order = tf.data.Options()
    option_no_order.experimental_deterministic = False

    dataset = tf.data.TFRecordDataset(filenames)
    dataset = dataset.with_options(option_no_order)
    dataset = dataset.map(read_tfrecord)
    return dataset

### CODING TASK ###

#TASK 1.b.i  Reading from the tfrecord dataset

filenames = tf.io.gfile.glob(GCS_OUTPUT + "*.tfrec")
datasetTfrec = load_dataset(filenames)

#displaying 9 images from the loaded data
display_9_images_from_dataset(datasetTfrec)
```

ii) Write your code above into a file using the *cell magic* `%%writefile spark_write_tfrec.py` at the beginning of the file. Then, run the file locally in Spark.

```
### CODING TASK ###

#TASK 1.b.ii

%%writefile /content/spark_write_tfrec.py

import subprocess
subprocess.call(['pip', 'install', 'tensorflow'])
subprocess.call(['pip', 'install', 'findspark'])
```

```
subprocess.call(['pip', 'install', 'pyspark'])
subprocess.call(['pip', 'install', 'py4j'])

import os, sys, math
import numpy as np
import time
import datetime
import string
import random
import tensorflow as tf
print("Tensorflow version " + tf.__version__)
import pickle
import argparse

PROJECT = 'bd-coursework-421223'
subprocess.call(['gcloud', 'config', 'set', 'project', PROJECT])
REGION = 'us-west1'
CLUSTER = '{}-cluster'.format(PROJECT)
subprocess.call(['gcloud', 'config', 'set', 'compute/region', REGION])
subprocess.call(['gcloud', 'config', 'set', 'dataproc/region',
REGION])

BUCKET = 'gs://{}-storage'.format(PROJECT)

os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

import findspark
findspark.init()
import pyspark
sc = pyspark.SparkContext.getOrCreate()

def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2

def resize_and_crop_image(data):
    image, label=data
    # Resizes and cropd using "fill" algorithm:
    # always make sure the resulting image is cut out from the source
image
    # so that it fills the TARGET_SIZE entirely with no black bars
    # and a preserved aspect ratio.
    w = tf.shape(image)[0]
```

```
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,
                    lambda: tf.image.resize(image, [w*tw/w, h*tw/w]),
# if true
                    lambda: tf.image.resize(image, [w*th/h, h*th/h])
# if false
                    )
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def recompress_image(data):
    image, label=data
    # this reduces the amount of data, but takes some time
    image = tf.cast(image, tf.uint8)
    image = tf.image.encode_jpeg(image, optimize_size=True,
chroma_downsampling=False)
    return image, label

# functions for writing TFRecord entries
# Feature values are always stored as lists, a single data element
will be a list of size 1
def _bytestring_feature(list_of_bytestrings):
    return
tf.train.Feature(bytes_list=tf.train.BytesList(value=list_of_bytestrin
gs))

def _int_feature(list_of_ints): # int64
    return
tf.train.Feature(int64_list=tf.train.Int64List(value=list_of_ints))

def to_tfrecord(tfrec_filewriter, img_bytes, label): # Create tf data
records
    class_num = np.argmax(np.array(CLASSES)==label) # 'roses' => 2
(order defined in CLASSES)
    one_hot_class = np.eye(len(CLASSES))[class_num]     # [0, 0, 1, 0,
0] for class #2, roses
    feature = {
        "image": _bytestring_feature([img_bytes]), # one image in the
list
        "class": _int_feature([class_num]) #,      # one class in the
list
    }
    return
tf.train.Example(features=tf.train.Features(feature=feature))
```

```
def write_tfrecords(partition_index, iterator):
  partition=partition_index
  global partition_size
  filename = GCS_OUTPUT + "{:02d}-{}.tfrec".format(partition,
partition_size)
  with tf.io.TFRecordWriter(filename) as out_file:
    for image,label in iterator:
      example = to_tfrecord(out_file,image.numpy(),
                            label.numpy().decode('utf-8')
                                        )
      out_file.write(example.SerializeToString())
  return [filename]

GCS_PATTERN = 'gs://flowers-public/*/*.jpg' # glob  pattern for input
files
PARTITIONS = 16 # no of partitions we will use later
TARGET_SIZE = [192, 192] # target resolution for the images
CLASSES = [b'daisy', b'dandelion', b'roses', b'sunflowers', b'tulips']
    # labels for the data


nb_images = len(tf.io.gfile.glob(GCS_PATTERN)) # number of images
partition_size = math.ceil(1.0 * nb_images / PARTITIONS) # images per
partition (float)

### TASK 1d ###
dsetRDD = sc.parallelize(tf.io.gfile.glob(GCS_PATTERN),PARTITIONS)

dsetDecoded=dsetRDD.map(decode_jpeg_and_label)
dsetResized=dsetDecoded.map(resize_and_crop_image)
dsetRecompressed = dsetResized.map(recompress_image)
dsetSampled=dsetRecompressed.sample(False,0.02)

GCS_OUTPUT = BUCKET + '/tfrecords-jpeg-192x192-2/flowers'  # prefix
for output file names

TFRecord_filenames=dsetSampled.mapPartitionsWithIndex(write_tfrecords)
output_filenames = TFRecord_filenames.collect()

#new

def save(object,bucket,filename):
    with open(filename,mode='wb') as f:
        pickle.dump(object,f)
    print("Saving {} to {}".format(filename,bucket))
    proc = subprocess.run(["gsutil","cp", filename,
bucket],stderr=subprocess.PIPE)
    print("gstuil returned: " + str(proc.returncode))
    print(str(proc.stderr))
```

```
def output(argv):
    # Parse the provided arguments
    global output_filenames
    parser = argparse.ArgumentParser() # get a parser object
    parser.add_argument('--out_bucket', metavar='out_bucket',
required=True,
                        help='The bucket URL for the result.') # add a
required argument
    parser.add_argument('--out_file', metavar='out_file',
required=True,
                        help='The filename for the result.') # add a
required argument
    args = parser.parse_args(argv) # read the value
    save(output_filenames,args.out_bucket,args.out_file)

output(["--out_bucket", BUCKET, "--out_file","filenames.pkl"])
```

```
Writing /content/spark_write_tfrec.py
```

```
os.environ["PATH"] += ":/root/spark-3.5.0-bin-hadoop3/bin"
```

```
!spark-submit /content/spark_write_tfrec.py
```

```
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.10/dist-packages (2.15.0)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.5.26 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.5.4)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes~=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: numpy<2.0.0,>=1.23.5 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.25.2)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in
```

```
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.4.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.36.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.62.2)
Requirement already satisfied: tensorboard<2.16,>=2.15 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.2)
Requirement already satisfied: tensorflow-estimator<2.16,>=2.15.0
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: keras<2.16,>=2.15.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.15.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.43.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.27.0)
Requirement already satisfied: google-auth-oauthlib<2,>=0.5 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (1.2.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.6)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (2.31.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.10/dist-packages (from
tensorboard<2.16,>=2.15->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from tensorboard<2.16,>=2.15-
>tensorflow) (3.0.2)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (0.4.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
```

```
/usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.16,>=2.15->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth-
oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.16,>=2.15->tensorflow) (2024.2.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.16,>=2.15->tensorflow) (2.1.5)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow) (0.6.0)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.10/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5-
>tensorboard<2.16,>=2.15->tensorflow) (3.2.2)
Requirement already satisfied: findspark in
/usr/local/lib/python3.10/dist-packages (2.0.1)
Requirement already satisfied: pyspark in
/usr/local/lib/python3.10/dist-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in
/usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
2024-04-29 20:08:05.164700: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable
to register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
2024-04-29 20:08:05.164753: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
2024-04-29 20:08:05.166106: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable
to register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
2024-04-29 20:08:06.457951: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Could not find TensorRT
Tensorflow version 2.15.0
```

```
Updated property [core/project].
WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
Updated property [dataproc/region].
24/04/29 20:08:14 INFO SparkContext: Running Spark version 3.5.0
24/04/29 20:08:14 INFO SparkContext: OS info Linux, 6.1.58+, amd64
24/04/29 20:08:14 INFO SparkContext: Java version 1.8.0_402
24/04/29 20:08:14 WARN NativeCodeLoader: Unable to load native-hadoop
library for your platform... using builtin-java classes where
applicable
24/04/29 20:08:15 INFO ResourceUtils:
==============================================================
24/04/29 20:08:15 INFO ResourceUtils: No custom resources configured
for spark.driver.
24/04/29 20:08:15 INFO ResourceUtils:
==============================================================
24/04/29 20:08:15 INFO SparkContext: Submitted application:
spark_write_tfrec.py
24/04/29 20:08:15 INFO ResourceProfile: Default ResourceProfile
created, executor resources: Map(cores -> name: cores, amount: 1,
script: , vendor: , memory -> name: memory, amount: 1024, script: ,
vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ),
task resources: Map(cpus -> name: cpus, amount: 1.0)
24/04/29 20:08:15 INFO ResourceProfile: Limiting resource is cpu
24/04/29 20:08:15 INFO ResourceProfileManager: Added ResourceProfile
id: 0
24/04/29 20:08:15 INFO SecurityManager: Changing view acls to: root
24/04/29 20:08:15 INFO SecurityManager: Changing modify acls to: root
24/04/29 20:08:15 INFO SecurityManager: Changing view acls groups to:
24/04/29 20:08:15 INFO SecurityManager: Changing modify acls groups
to:
24/04/29 20:08:15 INFO SecurityManager: SecurityManager:
authentication disabled; ui acls disabled; users with view
permissions: root; groups with view permissions: EMPTY; users with
modify permissions: root; groups with modify permissions: EMPTY
24/04/29 20:08:16 INFO Utils: Successfully started service
'sparkDriver' on port 33183.
24/04/29 20:08:16 INFO SparkEnv: Registering MapOutputTracker
24/04/29 20:08:16 INFO SparkEnv: Registering BlockManagerMaster
24/04/29 20:08:16 INFO BlockManagerMasterEndpoint: Using
org.apache.spark.storage.DefaultTopologyMapper for getting topology
information
24/04/29 20:08:16 INFO BlockManagerMasterEndpoint:
BlockManagerMasterEndpoint up
24/04/29 20:08:16 INFO SparkEnv: Registering
BlockManagerMasterHeartbeat
24/04/29 20:08:16 INFO DiskBlockManager: Created local directory at
/tmp/blockmgr-fcf4b63d-4f10-46fa-be9c-6ebeb3e85a54
24/04/29 20:08:16 INFO MemoryStore: MemoryStore started with capacity
```

```
366.3 MiB
24/04/29 20:08:16 INFO SparkEnv: Registering OutputCommitCoordinator
24/04/29 20:08:16 INFO JettyUtils: Start Jetty 0.0.0.0:4040 for
SparkUI
24/04/29 20:08:16 WARN Utils: Service 'SparkUI' could not bind on port
4040. Attempting port 4041.
24/04/29 20:08:16 INFO Utils: Successfully started service 'SparkUI'
on port 4041.
24/04/29 20:08:16 INFO Executor: Starting executor ID driver on host
f8df7a97da49
24/04/29 20:08:16 INFO Executor: OS info Linux, 6.1.58+, amd64
24/04/29 20:08:16 INFO Executor: Java version 1.8.0_402
24/04/29 20:08:16 INFO Executor: Starting executor with user classpath
(userClassPathFirst = false): ''
24/04/29 20:08:16 INFO Executor: Created or updated repl class loader
org.apache.spark.util.MutableURLClassLoader@6954158d for default.
24/04/29 20:08:17 INFO Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port
46425.
24/04/29 20:08:17 INFO NettyBlockTransferService: Server created on
f8df7a97da49:46425
24/04/29 20:08:17 INFO BlockManager: Using
org.apache.spark.storage.RandomBlockReplicationPolicy for block
replication policy
24/04/29 20:08:17 INFO BlockManagerMaster: Registering BlockManager
BlockManagerId(driver, f8df7a97da49, 46425, None)
24/04/29 20:08:17 INFO BlockManagerMasterEndpoint: Registering block
manager f8df7a97da49:46425 with 366.3 MiB RAM, BlockManagerId(driver,
f8df7a97da49, 46425, None)
24/04/29 20:08:17 INFO BlockManagerMaster: Registered BlockManager
BlockManagerId(driver, f8df7a97da49, 46425, None)
24/04/29 20:08:17 INFO BlockManager: Initialized BlockManager:
BlockManagerId(driver, f8df7a97da49, 46425, None)
24/04/29 20:08:21 INFO SparkContext: Starting job: collect at
/content/spark_write_tfrec.py:128
24/04/29 20:08:21 INFO DAGScheduler: Got job 0 (collect at
/content/spark_write_tfrec.py:128) with 16 output partitions
24/04/29 20:08:21 INFO DAGScheduler: Final stage: ResultStage 0
(collect at /content/spark_write_tfrec.py:128)
24/04/29 20:08:21 INFO DAGScheduler: Parents of final stage: List()
24/04/29 20:08:21 INFO DAGScheduler: Missing parents: List()
24/04/29 20:08:21 INFO DAGScheduler: Submitting ResultStage 0
(PythonRDD[1] at collect at /content/spark_write_tfrec.py:128), which
has no missing parents
24/04/29 20:08:21 INFO MemoryStore: Block broadcast_0 stored as values
in memory (estimated size 9.3 KiB, free 366.3 MiB)
24/04/29 20:08:21 INFO MemoryStore: Block broadcast_0_piece0 stored as
bytes in memory (estimated size 6.0 KiB, free 366.3 MiB)
24/04/29 20:08:21 INFO BlockManagerInfo: Added broadcast_0_piece0 in
```

```
memory on f8df7a97da49:46425 (size: 6.0 KiB, free: 366.3 MiB)
24/04/29 20:08:21 INFO SparkContext: Created broadcast 0 from
broadcast at DAGScheduler.scala:1580
24/04/29 20:08:21 INFO DAGScheduler: Submitting 16 missing tasks from
ResultStage 0 (PythonRDD[1] at collect at
/content/spark_write_tfrec.py:128) (first 15 tasks are for partitions
Vector(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14))
24/04/29 20:08:21 INFO TaskSchedulerImpl: Adding task set 0.0 with 16
tasks resource profile 0
24/04/29 20:08:21 INFO TaskSetManager: Starting task 0.0 in stage 0.0
(TID 0) (f8df7a97da49, executor driver, partition 0, PROCESS_LOCAL,
20380 bytes)
24/04/29 20:08:21 INFO TaskSetManager: Starting task 1.0 in stage 0.0
(TID 1) (f8df7a97da49, executor driver, partition 1, PROCESS_LOCAL,
20246 bytes)
24/04/29 20:08:21 INFO Executor: Running task 0.0 in stage 0.0 (TID 0)
24/04/29 20:08:21 INFO Executor: Running task 1.0 in stage 0.0 (TID 1)
2024-04-29 20:08:25.017615: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable
to register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
2024-04-29 20:08:25.017782: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
2024-04-29 20:08:25.020296: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable
to register cuDNN factory: Attempting to register factory for plugin
cuDNN when one has already been registered
2024-04-29 20:08:25.020307: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable
to register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
2024-04-29 20:08:25.020409: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to
register cuFFT factory: Attempting to register factory for plugin
cuFFT when one has already been registered
2024-04-29 20:08:25.022769: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable
to register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
2024-04-29 20:08:27.882842: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Could not find TensorRT
2024-04-29 20:08:28.164238: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Could not find TensorRT
24/04/29 20:09:37 INFO PythonRunner: Times: total = 74508, boot =
1255, init = 8751, finish = 64502
```

```
24/04/29 20:09:37 INFO Executor: Finished task 0.0 in stage 0.0 (TID
0). 1491 bytes result sent to driver
24/04/29 20:09:37 INFO TaskSetManager: Starting task 2.0 in stage 0.0
(TID 2) (f8df7a97da49, executor driver, partition 2, PROCESS_LOCAL,
20477 bytes)
24/04/29 20:09:37 INFO Executor: Running task 2.0 in stage 0.0 (TID 2)
24/04/29 20:09:37 INFO TaskSetManager: Finished task 0.0 in stage 0.0
(TID 0) in 75562 ms on f8df7a97da49 (executor driver) (1/16)
24/04/29 20:09:37 INFO PythonAccumulatorV2: Connected to
AccumulatorServer at host: 127.0.0.1 port: 47419
24/04/29 20:09:41 INFO PythonRunner: Times: total = 78632, boot =
1271, init = 8572, finish = 68789
24/04/29 20:09:41 INFO Executor: Finished task 1.0 in stage 0.0 (TID
1). 1448 bytes result sent to driver
24/04/29 20:09:41 INFO TaskSetManager: Starting task 3.0 in stage 0.0
(TID 3) (f8df7a97da49, executor driver, partition 3, PROCESS_LOCAL,
21302 bytes)
24/04/29 20:09:41 INFO TaskSetManager: Finished task 1.0 in stage 0.0
(TID 1) in 79558 ms on f8df7a97da49 (executor driver) (2/16)
24/04/29 20:09:41 INFO Executor: Running task 3.0 in stage 0.0 (TID 3)
24/04/29 20:10:41 INFO PythonRunner: Times: total = 60688, boot = 244,
init = 290, finish = 60154
24/04/29 20:10:41 INFO Executor: Finished task 3.0 in stage 0.0 (TID
3). 1405 bytes result sent to driver
24/04/29 20:10:41 INFO TaskSetManager: Starting task 4.0 in stage 0.0
(TID 4) (f8df7a97da49, executor driver, partition 4, PROCESS_LOCAL,
21193 bytes)
24/04/29 20:10:41 INFO Executor: Running task 4.0 in stage 0.0 (TID 4)
24/04/29 20:10:41 INFO TaskSetManager: Finished task 3.0 in stage 0.0
(TID 3) in 60742 ms on f8df7a97da49 (executor driver) (3/16)
24/04/29 20:10:45 INFO PythonRunner: Times: total = 67951, boot = 181,
init = 170, finish = 67600
24/04/29 20:10:45 INFO Executor: Finished task 2.0 in stage 0.0 (TID
2). 1448 bytes result sent to driver
24/04/29 20:10:45 INFO TaskSetManager: Starting task 5.0 in stage 0.0
(TID 5) (f8df7a97da49, executor driver, partition 5, PROCESS_LOCAL,
21168 bytes)
24/04/29 20:10:45 INFO Executor: Running task 5.0 in stage 0.0 (TID 5)
24/04/29 20:10:45 INFO TaskSetManager: Finished task 2.0 in stage 0.0
(TID 2) in 68045 ms on f8df7a97da49 (executor driver) (4/16)
24/04/29 20:11:49 INFO PythonRunner: Times: total = 67247, boot = 183,
init = 174, finish = 66890
24/04/29 20:11:49 INFO Executor: Finished task 4.0 in stage 0.0 (TID
4). 1405 bytes result sent to driver
24/04/29 20:11:49 INFO TaskSetManager: Starting task 6.0 in stage 0.0
(TID 6) (f8df7a97da49, executor driver, partition 6, PROCESS_LOCAL,
20920 bytes)
24/04/29 20:11:49 INFO TaskSetManager: Finished task 4.0 in stage 0.0
(TID 4) in 67267 ms on f8df7a97da49 (executor driver) (5/16)
```

```
24/04/29 20:11:49 INFO Executor: Running task 6.0 in stage 0.0 (TID 6)
24/04/29 20:11:52 INFO PythonRunner: Times: total = 67090, boot = 170,
init = 177, finish = 66743
24/04/29 20:11:52 INFO Executor: Finished task 5.0 in stage 0.0 (TID
5). 1405 bytes result sent to driver
24/04/29 20:11:52 INFO TaskSetManager: Starting task 7.0 in stage 0.0
(TID 7) (f8df7a97da49, executor driver, partition 7, PROCESS_LOCAL,
20379 bytes)
24/04/29 20:11:52 INFO Executor: Running task 7.0 in stage 0.0 (TID 7)
24/04/29 20:11:52 INFO TaskSetManager: Finished task 5.0 in stage 0.0
(TID 5) in 67114 ms on f8df7a97da49 (executor driver) (6/16)
24/04/29 20:12:56 INFO PythonRunner: Times: total = 67158, boot = 160,
init = 162, finish = 66836
24/04/29 20:12:56 INFO Executor: Finished task 6.0 in stage 0.0 (TID
6). 1405 bytes result sent to driver
24/04/29 20:12:56 INFO TaskSetManager: Starting task 8.0 in stage 0.0
(TID 8) (f8df7a97da49, executor driver, partition 8, PROCESS_LOCAL,
20254 bytes)
24/04/29 20:12:56 INFO Executor: Running task 8.0 in stage 0.0 (TID 8)
24/04/29 20:12:56 INFO TaskSetManager: Finished task 6.0 in stage 0.0
(TID 6) in 67187 ms on f8df7a97da49 (executor driver) (7/16)
24/04/29 20:13:00 INFO PythonRunner: Times: total = 68606, boot = 166,
init = 168, finish = 68272
24/04/29 20:13:00 INFO Executor: Finished task 7.0 in stage 0.0 (TID
7). 1448 bytes result sent to driver
24/04/29 20:13:00 INFO TaskSetManager: Starting task 9.0 in stage 0.0
(TID 9) (f8df7a97da49, executor driver, partition 9, PROCESS_LOCAL,
20977 bytes)
24/04/29 20:13:00 INFO TaskSetManager: Finished task 7.0 in stage 0.0
(TID 7) in 68637 ms on f8df7a97da49 (executor driver) (8/16)
24/04/29 20:13:00 INFO Executor: Running task 9.0 in stage 0.0 (TID 9)
24/04/29 20:14:08 INFO PythonRunner: Times: total = 67853, boot = 159,
init = 146, finish = 67548
24/04/29 20:14:08 INFO Executor: Finished task 9.0 in stage 0.0 (TID
9). 1405 bytes result sent to driver
24/04/29 20:14:08 INFO TaskSetManager: Starting task 10.0 in stage 0.0
(TID 10) (f8df7a97da49, executor driver, partition 10, PROCESS_LOCAL,
21460 bytes)
24/04/29 20:14:08 INFO Executor: Running task 10.0 in stage 0.0 (TID
10)
24/04/29 20:14:08 INFO TaskSetManager: Finished task 9.0 in stage 0.0
(TID 9) in 67878 ms on f8df7a97da49 (executor driver) (9/16)
24/04/29 20:14:09 INFO PythonRunner: Times: total = 72624, boot = 264,
init = 264, finish = 72096
24/04/29 20:14:09 INFO Executor: Finished task 8.0 in stage 0.0 (TID
8). 1405 bytes result sent to driver
24/04/29 20:14:09 INFO TaskSetManager: Starting task 11.0 in stage 0.0
(TID 11) (f8df7a97da49, executor driver, partition 11, PROCESS_LOCAL,
21390 bytes)
```

```
24/04/29 20:14:09 INFO TaskSetManager: Finished task 8.0 in stage 0.0
(TID 8) in 72682 ms on f8df7a97da49 (executor driver) (10/16)
24/04/29 20:14:09 INFO Executor: Running task 11.0 in stage 0.0 (TID
11)
24/04/29 20:15:24 INFO PythonRunner: Times: total = 75204, boot = 168,
init = 175, finish = 74861
24/04/29 20:15:24 INFO Executor: Finished task 10.0 in stage 0.0 (TID
10). 1405 bytes result sent to driver
24/04/29 20:15:24 INFO TaskSetManager: Starting task 12.0 in stage 0.0
(TID 12) (f8df7a97da49, executor driver, partition 12, PROCESS_LOCAL,
21010 bytes)
24/04/29 20:15:24 INFO Executor: Running task 12.0 in stage 0.0 (TID
12)
24/04/29 20:15:24 INFO TaskSetManager: Finished task 10.0 in stage 0.0
(TID 10) in 75220 ms on f8df7a97da49 (executor driver) (11/16)
24/04/29 20:15:30 INFO PythonRunner: Times: total = 81208, boot = 180,
init = 170, finish = 80858
24/04/29 20:15:30 INFO Executor: Finished task 11.0 in stage 0.0 (TID
11). 1405 bytes result sent to driver
24/04/29 20:15:30 INFO TaskSetManager: Starting task 13.0 in stage 0.0
(TID 13) (f8df7a97da49, executor driver, partition 13, PROCESS_LOCAL,
20694 bytes)
24/04/29 20:15:30 INFO TaskSetManager: Finished task 11.0 in stage 0.0
(TID 11) in 81233 ms on f8df7a97da49 (executor driver) (12/16)
24/04/29 20:15:30 INFO Executor: Running task 13.0 in stage 0.0 (TID
13)
24/04/29 20:16:37 INFO PythonRunner: Times: total = 66845, boot = 327,
init = 316, finish = 66202
24/04/29 20:16:37 INFO Executor: Finished task 13.0 in stage 0.0 (TID
13). 1405 bytes result sent to driver
24/04/29 20:16:37 INFO TaskSetManager: Starting task 14.0 in stage 0.0
(TID 14) (f8df7a97da49, executor driver, partition 14, PROCESS_LOCAL,
20474 bytes)
24/04/29 20:16:37 INFO TaskSetManager: Finished task 13.0 in stage 0.0
(TID 13) in 66874 ms on f8df7a97da49 (executor driver) (13/16)
24/04/29 20:16:37 INFO Executor: Running task 14.0 in stage 0.0 (TID
14)
24/04/29 20:16:45 INFO PythonRunner: Times: total = 81232, boot = 168,
init = 147, finish = 80917
24/04/29 20:16:45 INFO Executor: Finished task 12.0 in stage 0.0 (TID
12). 1405 bytes result sent to driver
24/04/29 20:16:45 INFO TaskSetManager: Starting task 15.0 in stage 0.0
(TID 15) (f8df7a97da49, executor driver, partition 15, PROCESS_LOCAL,
20860 bytes)
24/04/29 20:16:45 INFO Executor: Running task 15.0 in stage 0.0 (TID
15)
24/04/29 20:16:45 INFO TaskSetManager: Finished task 12.0 in stage 0.0
(TID 12) in 81245 ms on f8df7a97da49 (executor driver) (14/16)
24/04/29 20:17:48 INFO PythonRunner: Times: total = 71293, boot = 179,
```

```
init = 147, finish = 70967
24/04/29 20:17:48 INFO Executor: Finished task 14.0 in stage 0.0 (TID
14). 1405 bytes result sent to driver
24/04/29 20:17:48 INFO TaskSetManager: Finished task 14.0 in stage 0.0
(TID 14) in 71320 ms on f8df7a97da49 (executor driver) (15/16)
24/04/29 20:18:05 INFO PythonRunner: Times: total = 80261, boot = 188,
init = 179, finish = 79894
24/04/29 20:18:05 INFO Executor: Finished task 15.0 in stage 0.0 (TID
15). 1405 bytes result sent to driver
24/04/29 20:18:05 INFO TaskSetManager: Finished task 15.0 in stage 0.0
(TID 15) in 80275 ms on f8df7a97da49 (executor driver) (16/16)
24/04/29 20:18:05 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose
tasks have all completed, from pool
24/04/29 20:18:05 INFO DAGScheduler: ResultStage 0 (collect at
/content/spark_write_tfrec.py:128) finished in 584.234 s
24/04/29 20:18:05 INFO DAGScheduler: Job 0 is finished. Cancelling
potential speculative or zombie tasks for this job
24/04/29 20:18:05 INFO TaskSchedulerImpl: Killing all running tasks in
stage 0: Stage finished
24/04/29 20:18:05 INFO DAGScheduler: Job 0 finished: collect at
/content/spark_write_tfrec.py:128, took 584.369249 s
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'Copying file://filenames.pkl [Content-Type=application/octet-
stream]...\n/ [0 files][    0.0 B/  1.3 KiB]
\r/ [1 files][  1.3 KiB/  1.3 KiB]
\r-\r\nOperation completed over 1 objects/1.3 KiB.
\n'
24/04/29 20:18:09 INFO SparkContext: Invoking stop() from shutdown
hook
24/04/29 20:18:09 INFO SparkContext: SparkContext is stopping with
exitCode 0.
24/04/29 20:18:09 INFO SparkUI: Stopped Spark web UI at
http://f8df7a97da49:4041
24/04/29 20:18:10 INFO MapOutputTrackerMasterEndpoint:
MapOutputTrackerMasterEndpoint stopped!
24/04/29 20:18:10 INFO MemoryStore: MemoryStore cleared
24/04/29 20:18:10 INFO BlockManager: BlockManager stopped
24/04/29 20:18:10 INFO BlockManagerMaster: BlockManagerMaster stopped
24/04/29 20:18:10 INFO
OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:
OutputCommitCoordinator stopped!
24/04/29 20:18:10 INFO SparkContext: Successfully stopped SparkContext
24/04/29 20:18:10 INFO ShutdownHookManager: Shutdown hook called
24/04/29 20:18:10 INFO ShutdownHookManager: Deleting directory
/tmp/spark-853f7c96-d4dd-4e01-8995-211597a2101c
24/04/29 20:18:10 INFO ShutdownHookManager: Deleting directory
/tmp/spark-b1c0800f-de3a-4a9e-9dd9-f04f59bed2f2
24/04/29 20:18:10 INFO ShutdownHookManager: Deleting directory
```

```
/tmp/spark-b1c0800f-de3a-4a9e-9dd9-f04f59bed2f2/pyspark-cf55f8e9-1a02-44c9-b55e-971fb0d14ef8
```

1c) Set up a cluster and run the script. (6%)

Following the example from the labs, set up a cluster to run PySpark jobs in the cloud. You need to set up so that TensorFlow is installed on all nodes in the cluster.

i) Single machine cluster

Set up a cluster with a single machine using the maximal SSD size (100) and 8 vCPUs.

Enable **package installation** by passing a flag `--initialization-actions` with argument `gs://goog-dataproc-initialization-actions-$REGION/python/pip-install.sh` (this is a public script that will read metadata to determine which packages to install). Then, the **packages are specified** by providing a `--metadata` flag with the argument `PIP_PACKAGES=tensorflow==2.4.0`.

Note: consider using `PIP_PACKAGES="tensorflow numpy"` or `PIP_PACKAGES=tensorflow` in case an older version of tensorflow is causing issues.

When the cluster is running, run your script to check that it works and keep the output cell output. (3%)

```
### CODING TASK ###

#TASK 1.c.i  Creating a single machine cluster with the specified configurations

!gcloud dataproc clusters create $CLUSTER \
    --image-version 1.5-ubuntu18 \
    --single-node \
    --master-machine-type=n1-standard-8 \
    --master-boot-disk-type pd-ssd \
    --master-boot-disk-size=100 \
    --initialization-actions gs://goog-dataproc-initialization-actions-$REGION/python/pip-install.sh \
    --metadata PIP_PACKAGES=tensorflow \
    --max-idle 3600s

Waiting on operation [projects/bd-coursework-421223/regions/us-central1/operations/85db41ce-94e5-385d-aa77-3b88e7d190aa].

WARNING: Don't create production clusters that reference initialization actions located in the gs://goog-dataproc-initialization-actions-REGION public buckets. These scripts are provided as reference implementations, and they are synchronized with ongoing GitHub repository changes—a new version of a initialization action in public buckets may break your cluster creation. Instead, copy the following initialization actions from public buckets into
```

```
your bucket :
gs://goog-dataproc-initialization-actions-us-west1/python/pip-
install.sh
WARNING: The firewall rules for specified network or subnetwork would
allow ingress traffic from 0.0.0.0/0, which could be a security risk.
WARNING: Unable to validate the staging bucket lifecycle configuration
of the bucket 'dataproc-staging-us-central1-832943544474-dttjnjdu' due
to an internal error, Please make sure that the provided bucket
doesn't have any delete rules set.
Created [https://dataproc.googleapis.com/v1/projects/bd-coursework-
421223/regions/us-central1/clusters/bd-coursework-421223-cluster]
Cluster placed in zone [us-central1-c].
```

Run the script in the cloud and test the output.

```
### CODING TASK ###
#Running the script in the cloud

FILENAME = 'filenames.pkl'
!gcloud dataproc jobs submit pyspark --cluster $CLUSTER \
    /content/spark_write_tfrec.py \
    -- --out_bucket $BUCKET --out_file $FILENAME
```

```
Job [892e35e330d94cb0b53209366ae5c98a] submitted.
Waiting for job output...
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1;
platform_machine != "arm64" or platform_system != "Darwin" in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
```

```
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: werkzeug>=1.0.1 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4; python_version
< "3.10" in /opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
```

```
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4; python_version < "3.10"->markdown>=2.6.8-
>tensorboard<2.12,>=2.11->tensorflow) (3.11.0)
Requirement already satisfied: pyasn1>=0.1.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from rsa<5,>=3.1.4-
>google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.5.1)
Requirement already satisfied: findspark in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.0.1)
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
Requirement already satisfied: py4j in
/opt/conda/miniconda3/lib/python3.7/site-packages (0.10.7)
2024-04-30 17:35:20.729861: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-04-30 17:35:20.902217: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 17:35:20.902323: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-04-30 17:35:21.785098: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 17:35:21.785263: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 17:35:21.785291: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
Tensorflow version 2.11.0
```

```
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: You do not appear to have access to project [bd-coursework-
421223] or it does not exist.
Updated property [core/project].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Updated property [dataproc/region].
24/04/30 17:35:30 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/04/30 17:35:30 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/04/30 17:35:30 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/04/30 17:35:30 INFO org.spark_project.jetty.util.log: Logging
initialized @14988ms to org.spark_project.jetty.util.log.Slf4jLog
24/04/30 17:35:30 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/04/30 17:35:30 INFO org.spark_project.jetty.server.Server: Started
@15120ms
24/04/30 17:35:30 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@6950cf86{HTTP/1.1, (http/1.1)}{0.0.0.0:33091}
24/04/30 17:35:31 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at
bd-coursework-421223-cluster-m/10.128.0.7:8032
24/04/30 17:35:32 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
cluster-m/10.128.0.7:10200
24/04/30 17:35:32 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
```

```
24/04/30 17:35:32 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/04/30 17:35:32 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/04/30 17:35:32 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/04/30 17:35:35 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714495107482_0001
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://filenames.pkl [Content-Type=application/octet-stream]...\n/ [0
files][    0.0 B/  1.4 KiB]
\r/ [1 files][  1.4 KiB/  1.4 KiB]
\r\nOperation completed over 1 objects/1.4 KiB.
\n'
24/04/30 17:37:52 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@6950cf86{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [892e35e330d94cb0b53209366ae5c98a] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-central1-832943544474-
dttjnjdu/google-cloud-dataproc-metainfo/6e30590a-5359-46cc-81b1-
aadf5d5d9e24/jobs/892e35e330d94cb0b53209366ae5c98a/
driverOutputResourceUri: gs://dataproc-staging-us-central1-
832943544474-dttjnjdu/google-cloud-dataproc-metainfo/6e30590a-5359-
46cc-81b1-aadf5d5d9e24/jobs/892e35e330d94cb0b53209366ae5c98a/
driveroutput
jobUuid: 6e361dc9-dc72-3c93-a89b-57afa7f5bac5
placement:
  clusterName: bd-coursework-421223-cluster
  clusterUuid: 6e30590a-5359-46cc-81b1-aadf5d5d9e24
pysparkJob:
  args:
  - --out_bucket
  - gs://bd-coursework-421223-storage
  - --out_file
  - filenames.pkl
  mainPythonFileUri: gs://dataproc-staging-us-central1-832943544474-
dttjnjdu/google-cloud-dataproc-metainfo/6e30590a-5359-46cc-81b1-
aadf5d5d9e24/jobs/892e35e330d94cb0b53209366ae5c98a/staging/
spark_write_tfrec.py
```

```
reference:
  jobId: 892e35e330d94cb0b53209366ae5c98a
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-04-30T17:37:54.923555Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-04-30T17:35:14.823316Z'
- state: SETUP_DONE
  stateStartTime: '2024-04-30T17:35:14.870553Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-04-30T17:35:15.143802Z'
yarnApplications:
- name: spark_write_tfrec.py
  progress: 1.0
  state: FINISHED
  trackingUrl:
http://bd-coursework-421223-cluster-m:8088/proxy/application_1714495107482_0001/
```

```
# checking the output
import pickle
%cd /content/drive/MyDrive/BD-CW
!gsutil cp $BUCKET/$FILENAME .
!ls -l

with open(FILENAME,mode='rb') as f:
    fnames = pickle.load(f)

fnames
```

```
/content/drive/MyDrive/BD-CW
Copying gs://bd-coursework-421223-storage/filenames.pkl...
/ [1 files][  1.4 KiB/  1.4 KiB]

Operation completed over 1 objects/1.4 KiB.

total 152
-rw------- 1 root root 152765 Apr 30 17:37 BD_Coursework.ipynb
-rw------- 1 root root   1416 Apr 30 17:38 filenames.pkl
-rw------- 1 root root     27 Apr 29 18:44 upgradepip.sh

['gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers00-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers01-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
```

```
02-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
03-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
04-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
05-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
06-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
07-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
08-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
09-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
10-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
11-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
12-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
13-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
14-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
15-230.tfrec']
```

In the free credit tier on Google Cloud, there are normally the following **restrictions** on compute machines:

- max 100GB of *SSD persistent disk*
- max 2000GB of *standard persistent disk*
- max 8 *vCPU*s
- no GPUs

See here for details The **disks are virtual** disks, where **I/O speed is limited in proportion to the size**, so we should allocate them evenly. This has mainly an effect on the **time the cluster needs**

to start, as we are reading the data mainly from the bucket and we are not writing much to disk at all.

ii) Maximal cluster

Use the **largest possible cluster** within these constraints, i.e. **1 master and 7 worker nodes**. Each of them with 1 (virtual) CPU. The master should get the full *SSD* capacity and the 7 worker nodes should get equal shares of the *standard* disk capacity to maximise throughput.

Once the cluster is running, test your script. (3%)

```
# Note: There was an error running the script regarding the pip
version, so creating this script for upgrading pip,
# which will be called while creating the cluster

%%writefile /content/upgradepip.sh

pip install --upgrade pip
```

```
Writing /content/upgradepip.sh
```

```
#uploading the script for upgrading pip to our bucket

!gsutil cp /content/upgradepip.sh $BUCKET/upgradepip.sh
```

```
Copying file:///content/upgradepip.sh [Content-Type=text/x-sh]...
/ [1 files][   27.0 B/   27.0 B]

Operation completed over 1 objects/27.0 B.
```

```
### CODING TASK ###
#TASK 1.c.ii
#creating the max cluster

MAXCLUSTER='{}-maxcluster'.format(PROJECT)

!gcloud dataproc clusters create $MAXCLUSTER \
    --image-version 1.5-ubuntu18 \
    --master-machine-type=n1-standard-1 \
    --master-boot-disk-type pd-ssd \
    --master-boot-disk-size=500 \
    --num-workers=7 \
    --worker-machine-type=n1-standard-1 \
    --worker-boot-disk-type=pd-standard \
    --worker-boot-disk-size=585 \
    --initialization-actions $BUCKET/upgradepip.sh,gs://goog-dataproc-
initialization-actions-$REGION/python/pip-install.sh \
    --metadata PIP_PACKAGES=tensorflow \
    --max-idle 3600s
```

```
Waiting on operation
[projects/bd-coursework-421223/regions/us-west1/operations/b8982c2f-
```

```
ab92-366f-a577-7bf3cc6e64a8].

WARNING: Creating clusters using the n1-standard-1 machine type is not
recommended. Consider using a machine type with higher memory.
WARNING: Don't create production clusters that reference
initialization actions located in the gs://goog-dataproc-
initialization-actions-REGION public buckets. These scripts are
provided as reference implementations, and they are synchronized with
ongoing GitHub repository changes—a new version of a initialization
action in public buckets may break your cluster creation. Instead,
copy the following initialization actions from public buckets into
your bucket :
gs://goog-dataproc-initialization-actions-us-west1/python/pip-
install.sh
WARNING: For PD-Standard without local SSDs, we strongly recommend
provisioning 1TB or larger to ensure consistently high I/O
performance. See
https://cloud.google.com/compute/docs/disks/performance for
information on disk I/O performance.
WARNING: The firewall rules for specified network or subnetwork would
allow ingress traffic from 0.0.0.0/0, which could be a security risk.
WARNING: The specified custom staging bucket 'dataproc-staging-us-
west1-832943544474-ngdqyb5y' is not using uniform bucket level access
IAM configuration. It is recommended to update bucket to enable the
same. See https://cloud.google.com/storage/docs/uniform-bucket-level-
access.
Created [https://dataproc.googleapis.com/v1/projects/bd-coursework-
421223/regions/us-west1/clusters/bd-coursework-421223-maxcluster]
Cluster placed in zone [us-west1-c].
```

```
### CODING TASK ###

#Running the script in maximal cluster

FILENAME = 'filenames.pkl'
!gcloud dataproc jobs submit pyspark --cluster $MAXCLUSTER \
    /content/spark_write_tfrec.py \
    -- --out_bucket $BUCKET --out_file $FILENAME

Job [49bb755df44442e98183de00ca596957] submitted.
Waiting for job output...
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
```

```
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
```

```
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl.metadata (352
bytes)
Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: py4j in
/opt/conda/miniconda3/lib/python3.7/site-packages (0.10.7)
```

```
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-05-04 16:07:23.770306: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-05-04 16:07:24.677351: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:07:24.677466: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-05-04 16:07:26.228811: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:07:26.229008: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:07:26.229055: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
Tensorflow version 2.11.0
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: You do not appear to have access to project [bd-coursework-
421223] or it does not exist.
Updated property [core/project].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
```

```
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Updated property [dataproc/region].
24/05/04 16:07:40 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/05/04 16:07:40 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/05/04 16:07:41 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/05/04 16:07:41 INFO org.spark_project.jetty.util.log: Logging
initialized @26809ms to org.spark_project.jetty.util.log.Slf4jLog
24/05/04 16:07:41 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/05/04 16:07:41 INFO org.spark_project.jetty.server.Server: Started
@27087ms
24/05/04 16:07:41 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:39555}
24/05/04 16:07:44 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at
bd-coursework-421223-maxcluster-m/10.138.15.205:8032
24/05/04 16:07:44 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
maxcluster-m/10.138.15.205:10200
24/05/04 16:07:44 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/05/04 16:07:44 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/05/04 16:07:44 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/05/04 16:07:44 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/05/04 16:07:48 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714838172205_0001
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
```

```
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://filenames.pkl [Content-Type=application/octet-stream]...\n/ [0
files][    0.0 B/  1.3 KiB]
\r/ [1 files][  1.3 KiB/  1.3 KiB]
\r\nOperation completed over 1 objects/1.3 KiB.
\n'
24/05/04 16:10:00 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [49bb755df44442e98183de00ca596957] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/3b31386b-539a-45ff-b949-
28f773a2665e/jobs/49bb755df44442e98183de00ca596957/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/3b31386b-539a-45ff-b949-
28f773a2665e/jobs/49bb755df44442e98183de00ca596957/driveroutput
jobUuid: bc87ee2d-5ce6-314a-9e5f-cbaa8b0369a2
placement:
  clusterName: bd-coursework-421223-maxcluster
  clusterUuid: 3b31386b-539a-45ff-b949-28f773a2665e
pysparkJob:
  args:
  - --out_bucket
  - gs://bd-coursework-421223-storage
  - --out_file
  - filenames.pkl
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/3b31386b-539a-45ff-b949-
28f773a2665e/jobs/49bb755df44442e98183de00ca596957/staging/
spark_write_tfrec.py
reference:
  jobId: 49bb755df44442e98183de00ca596957
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-05-04T16:10:02.829635Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-05-04T16:07:12.175805Z'
- state: SETUP_DONE
  stateStartTime: '2024-05-04T16:07:12.203303Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-05-04T16:07:12.719320Z'
yarnApplications:
- name: spark_write_tfrec.py
```

```
  progress: 1.0
  state: FINISHED
  trackingUrl:
http://bd-coursework-421223-maxcluster-m:8088/proxy/application_171483
8172205_0001/

## checking the output
import pickle
%cd /content/drive/MyDrive/BD-CW
!gsutil cp $BUCKET/$FILENAME .
!ls -l

with open(FILENAME,mode='rb') as f:
    fnames = pickle.load(f)

fnames

/content/drive/MyDrive/BD-CW
Copying gs://bd-coursework-421223-storage/filenames.pkl...
/ [1 files][  1.4 KiB/  1.4 KiB]

Operation completed over 1 objects/1.4 KiB.

total 471
-rw------- 1 root root 475989 Apr 30 22:35 BD_Coursework.ipynb
-rw------- 1 root root   1416 Apr 30 22:42 filenames.pkl
drwx------ 2 root root   4096 Apr 30 19:56 screenshots
-rw------- 1 root root     27 Apr 29 18:44 upgradepip.sh

['gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/
flowers00-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
01-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
02-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
03-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
04-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
05-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
06-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
```

```
07-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
08-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
09-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
10-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
11-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
12-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
13-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
14-230.tfrec',

'gs://bd-coursework-421223-storage/tfrecordsNEW-jpeg-192x192-2/flowers
15-230.tfrec']
```

1d) Optimisation, experiments, and discussion (17%)

i) Improve parallelisation

If you implemented a straightfoward version, you will **probably** observe that **all the computation** is done on only **two nodes**. This can be adressed by using the **second parameter** in the initial call to **parallelize**. Make the **suitable change** in the code you have written above and mark it up in comments as `### TASK 1d ###`.

Demonstrate the difference in cluster utilisation before and after the change based on different parameter values with **screenshots from Google Cloud** and measure the **difference in the processing time**. (6%)

ii) Experiment with cluster configurations.

In addition to the experiments above (using 8 VMs),test your program with 4 machines with double the resources each (2 vCPUs, memory, disk) and 1 machine with eightfold resources. Discuss the results in terms of disk I/O and network bandwidth allocation in the cloud. (7%)

iii) Explain the difference between this use of Spark and most standard applications like e.g. in our labs in terms of where the data is stored. What kind of parallelisation approach is used here? (4%)

Write the code below and your answers in the report.

```
#TASK 1d.i

#Before change(2 partitions)

### CODING TASK ###

#TASK 1.b.ii

%%writefile /content/spark_write_tfrec_2partitions.py

import subprocess
subprocess.call(['pip', 'install', 'tensorflow'])
subprocess.call(['pip', 'install', 'findspark'])
subprocess.call(['pip', 'install', 'pyspark'])

import os, sys, math
import numpy as np
# import scipy as sp
# import scipy.stats
import time
import datetime
import string
import random
import tensorflow as tf
print("Tensorflow version " + tf.__version__)
import pickle
import argparse

PROJECT = 'bd-coursework-421223'
subprocess.call(['gcloud', 'config', 'set', 'project', PROJECT])
REGION = 'us-west1'
CLUSTER = '{}-cluster'.format(PROJECT)
subprocess.call(['gcloud', 'config', 'set', 'compute/region', REGION])
subprocess.call(['gcloud', 'config', 'set', 'dataproc/region',
REGION])

BUCKET = 'gs://{}-storage'.format(PROJECT)

# os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
# os.environ["SPARK_HOME"] = "/root/spark-3.5.0-bin-hadoop3"
# os.environ['SPARK_HOME'] = '/usr/lib/spark'
os.environ["PATH"] += ":/root/spark-3.5.0-bin-hadoop3/bin"

import findspark
findspark.init()
import pyspark
sc = pyspark.SparkContext.getOrCreate()

def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
```

```
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2

def resize_and_crop_image(data):
    image, label=data
    # Resizes and cropd using "fill" algorithm:
    # always make sure the resulting image is cut out from the source
image
    # so that it fills the TARGET_SIZE entirely with no black bars
    # and a preserved aspect ratio.
    w = tf.shape(image)[0]
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,
                    lambda: tf.image.resize(image, [w*tw/w, h*tw/w]),
# if true
                    lambda: tf.image.resize(image, [w*th/h, h*th/h])
# if false
                    )
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def recompress_image(data):
    image, label=data
    # this reduces the amount of data, but takes some time
    image = tf.cast(image, tf.uint8)
    image = tf.image.encode_jpeg(image, optimize_size=True,
chroma_downsampling=False)
    return image, label

# functions for writing TFRecord entries
# Feature values are always stored as lists, a single data element
will be a list of size 1
def _bytestring_feature(list_of_bytestrings):
    return
tf.train.Feature(bytes_list=tf.train.BytesList(value=list_of_bytestrin
gs))

def _int_feature(list_of_ints): # int64
```

```
    return
tf.train.Feature(int64_list=tf.train.Int64List(value=list_of_ints))

def to_tfrecord(tfrec_filewriter, img_bytes, label): # Create tf data
records
    class_num = np.argmax(np.array(CLASSES)==label) # 'roses' => 2
(order defined in CLASSES)
    one_hot_class = np.eye(len(CLASSES))[class_num]     # [0, 0, 1, 0,
0] for class #2, roses
    feature = {
        "image": _bytestring_feature([img_bytes]), # one image in the
list
        "class": _int_feature([class_num]) #,      # one class in the
list
    }
    return
tf.train.Example(features=tf.train.Features(feature=feature))

def write_tfrecords(partition_index, iterator):
  partition=partition_index
  global partition_size
  filename = GCS_OUTPUT + "{:02d}-{}.tfrec".format(partition,
partition_size)
  with tf.io.TFRecordWriter(filename) as out_file:
    for image,label in iterator:
      example = to_tfrecord(out_file,image.numpy(),
                            label.numpy().decode('utf-8')
                                        )
      out_file.write(example.SerializeToString())
  return [filename]

GCS_PATTERN = 'gs://flowers-public/*/*.jpg' # glob  pattern for input
files
PARTITIONS = 16 # no of partitions we will use later
TARGET_SIZE = [192, 192] # target resolution for the images
CLASSES = [b'daisy', b'dandelion', b'roses', b'sunflowers', b'tulips']
    # labels for the data
nb_images = len(tf.io.gfile.glob(GCS_PATTERN)) # number of images


### TASK 1d ###
dsetRDD = sc.parallelize(tf.io.gfile.glob(GCS_PATTERN))
dsetDecoded=dsetRDD.map(decode_jpeg_and_label)
dsetResized=dsetDecoded.map(resize_and_crop_image)
dsetRecompressed = dsetResized.map(recompress_image)
dsetSampled=dsetRecompressed.sample(False,0.02)

partition_size = math.ceil(1.0 * nb_images /
dsetSampled.getNumPartitions()) # images per partition (float)
```

```
GCS_OUTPUT = BUCKET + '/tfrecords-jpeg-192x192-2/flowers'  # prefix
for output file names

TFRecord_filenames=dsetSampled.mapPartitionsWithIndex(write_tfrecords)
output_filenames = TFRecord_filenames.collect()

#new

def save(object,bucket,filename):
    with open(filename,mode='wb') as f:
        pickle.dump(object,f)
    print("Saving {} to {}".format(filename,bucket))
    proc = subprocess.run(["gsutil","cp", filename,
bucket],stderr=subprocess.PIPE)
    print("gstuil returned: " + str(proc.returncode))
    print(str(proc.stderr))

def output(argv):
    # Parse the provided arguments
    global output_filenames
    parser = argparse.ArgumentParser() # get a parser object
    parser.add_argument('--out_bucket', metavar='out_bucket',
required=True,
                        help='The bucket URL for the result.') # add a
required argument
    parser.add_argument('--out_file', metavar='out_file',
required=True,
                        help='The filename for the result.') # add a
required argument
    args = parser.parse_args(argv) # read the value
    save(output_filenames,args.out_bucket,args.out_file)

output(["--out_bucket", BUCKET, "--out_file","filenames.pkl"])
```

```
Overwriting /content/spark_write_tfrec_2partitions.py
```

```
#Running the script in maximal cluster

FILENAME = 'filenames.pkl'
!gcloud dataproc jobs submit pyspark --cluster $MAXCLUSTER --region
$REGION \
    /content/spark_write_tfrec_2partitions.py \
    -- --out_bucket $BUCKET --out_file $FILENAME
```

```
Job [bcd0b6fb8591411bbaa5fe839348ea14] submitted.
Waiting for job output...
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
```

```
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
```

```
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: findspark in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.0.1)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-04-30 22:46:06.569175: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
```

```
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-04-30 22:46:06.924540: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 22:46:06.924659: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-04-30 22:46:08.821948: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 22:46:08.822548: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 22:46:08.822618: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
Tensorflow version 2.11.0
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: You do not appear to have access to project [bd-coursework-
421223] or it does not exist.
Updated property [core/project].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.
```

```
If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Updated property [dataproc/region].
24/04/30 22:46:21 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/04/30 22:46:21 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/04/30 22:46:21 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/04/30 22:46:22 INFO org.spark_project.jetty.util.log: Logging
initialized @23198ms to org.spark_project.jetty.util.log.Slf4jLog
24/04/30 22:46:22 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/04/30 22:46:22 INFO org.spark_project.jetty.server.Server: Started
@23459ms
24/04/30 22:46:22 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:41155}
24/04/30 22:46:24 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at
bd-coursework-421223-maxcluster-m/10.138.0.16:8032
24/04/30 22:46:25 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
maxcluster-m/10.138.0.16:10200
24/04/30 22:46:25 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/04/30 22:46:25 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/04/30 22:46:25 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/04/30 22:46:25 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/04/30 22:46:28 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714515430789_0002
24/04/30 22:46:43 WARN org.apache.spark.scheduler.TaskSetManager:
Stage 0 contains a task of very large size (133 KB). The maximum
recommended task size is 100 KB.
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
```

```
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://filenames.pkl [Content-Type=application/octet-stream]...\n/ [0
files][    0.0 B/  180.0 B]
\r/ [1 files][  180.0 B/  180.0 B]
\r\nOperation completed over 1 objects/180.0 B.
\n'
24/04/30 22:51:27 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [bcd0b6fb8591411bbaa5fe839348ea14] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/4f71f6b9-bb99-481a-a381-
40e09bcca96a/jobs/bcd0b6fb8591411bbaa5fe839348ea14/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/4f71f6b9-bb99-481a-a381-
40e09bcca96a/jobs/bcd0b6fb8591411bbaa5fe839348ea14/driveroutput
jobUuid: e6d81cd5-aa64-3515-8f21-978470196b02
placement:
  clusterName: bd-coursework-421223-maxcluster
  clusterUuid: 4f71f6b9-bb99-481a-a381-40e09bcca96a
pysparkJob:
  args:
  - --out_bucket
  - gs://bd-coursework-421223-storage
  - --out_file
  - filenames.pkl
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/4f71f6b9-bb99-481a-a381-
40e09bcca96a/jobs/bcd0b6fb8591411bbaa5fe839348ea14/staging/
spark_write_tfrec_2partitions.py
reference:
  jobId: bcd0b6fb8591411bbaa5fe839348ea14
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-04-30T22:51:29.006101Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-04-30T22:45:57.890131Z'
- state: SETUP_DONE
  stateStartTime: '2024-04-30T22:45:57.936635Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-04-30T22:45:58.159164Z'
yarnApplications:
- name: spark_write_tfrec_2partitions.py
  progress: 1.0
  state: FINISHED
  trackingUrl:
```

```
http://bd-coursework-421223-maxcluster-m:8088/proxy/application_1714515430789_0002/

#After change(16 partitions)

#Running the script in maximal cluster

FILENAME = 'filenames.pkl'
!gcloud dataproc jobs submit pyspark --cluster $MAXCLUSTER --region $REGION \
    /content/spark_write_tfrec.py \
    -- --out_bucket $BUCKET --out_file $FILENAME

Job [62c563a4c9fe4cf7998a8774b46e73ad] submitted.
Waiting for job output...
Requirement already satisfied: tensorflow in /opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (1.62.2)
Requirement already satisfied: h5py>=2.9.0 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (2.11.0)
Requirement already satisfied: libclang>=13.0.0 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: numpy>=1.20 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow) (3.3.0)
```

```
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
```

```
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: findspark in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.0.1)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: py4j in
/opt/conda/miniconda3/lib/python3.7/site-packages (0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-04-30 22:51:45.474349: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-04-30 22:51:45.693749: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 22:51:45.693875: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-04-30 22:51:47.492683: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 22:51:47.492897: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-04-30 22:51:47.492968: W
```

```
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
Tensorflow version 2.11.0
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: You do not appear to have access to project [bd-coursework-
421223] or it does not exist.
Updated property [core/project].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Updated property [dataproc/region].
24/04/30 22:52:00 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/04/30 22:52:00 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/04/30 22:52:00 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/04/30 22:52:00 INFO org.spark_project.jetty.util.log: Logging
initialized @23853ms to org.spark_project.jetty.util.log.Slf4jLog
24/04/30 22:52:00 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/04/30 22:52:00 INFO org.spark_project.jetty.server.Server: Started
@24106ms
24/04/30 22:52:01 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:42483}
24/04/30 22:52:03 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at
bd-coursework-421223-maxcluster-m/10.138.0.16:8032
```

```
24/04/30 22:52:03 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
maxcluster-m/10.138.0.16:10200
24/04/30 22:52:03 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/04/30 22:52:03 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/04/30 22:52:03 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/04/30 22:52:03 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/04/30 22:52:07 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714515430789_0003
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://filenames.pkl [Content-Type=application/octet-stream]...\n/ [0
files][    0.0 B/  1.4 KiB]
\r/ [1 files][  1.4 KiB/  1.4 KiB]
\r\nOperation completed over 1 objects/1.4 KiB.
\n'
24/04/30 22:54:01 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [62c563a4c9fe4cf7998a8774b46e73ad] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/4f71f6b9-bb99-481a-a381-
40e09bcca96a/jobs/62c563a4c9fe4cf7998a8774b46e73ad/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/4f71f6b9-bb99-481a-a381-
40e09bcca96a/jobs/62c563a4c9fe4cf7998a8774b46e73ad/driveroutput
jobUuid: ca73baea-cc92-39af-8c87-724e36900db6
placement:
  clusterName: bd-coursework-421223-maxcluster
  clusterUuid: 4f71f6b9-bb99-481a-a381-40e09bcca96a
pysparkJob:
  args:
  - --out_bucket
  - gs://bd-coursework-421223-storage
  - --out_file
  - filenames.pkl
```

```
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/4f71f6b9-bb99-481a-a381-
40e09bcca96a/jobs/62c563a4c9fe4cf7998a8774b46e73ad/staging/
spark_write_tfrec.py
reference:
  jobId: 62c563a4c9fe4cf7998a8774b46e73ad
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-04-30T22:54:04.127053Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-04-30T22:51:35.339371Z'
- state: SETUP_DONE
  stateStartTime: '2024-04-30T22:51:35.374932Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-04-30T22:51:35.564801Z'
yarnApplications:
- name: spark_write_tfrec.py
  progress: 1.0
  state: FINISHED
  trackingUrl:
http://bd-coursework-421223-maxcluster-m:8088/proxy/application_171451
5430789_0003/
```

```
## TASK 1.d.ii

#EXPERIMENT 1
#cluster with 4 machines  (double resources-vCPUs, memory, disk)

EXPERIMENT_CLUSTER1='{}-experiment-cluster1'.format(PROJECT)

!gcloud dataproc clusters create $EXPERIMENT_CLUSTER1 \
--image-version 1.5-ubuntu18 \
    --master-machine-type=n1-highmem-2 \
    --master-boot-disk-type pd-ssd \
    --master-boot-disk-size=500 \
    --num-workers=3 \
    --worker-machine-type=n1-highmem-2 \
    --worker-boot-disk-type=pd-standard \
    --worker-boot-disk-size=1365 \
    --initialization-actions $BUCKET/upgradepip.sh,gs://goog-dataproc-
initialization-actions-$REGION/python/pip-install.sh \
    --metadata PIP_PACKAGES=tensorflow \
    --max-idle 3600s

Waiting on operation
[projects/bd-coursework-421223/regions/us-west1/operations/09ffa8d5-
746b-3d84-8567-611f128289b1].
```

```
WARNING: Don't create production clusters that reference
initialization actions located in the gs://goog-dataproc-
initialization-actions-REGION public buckets. These scripts are
provided as reference implementations, and they are synchronized with
ongoing GitHub repository changes—a new version of a initialization
action in public buckets may break your cluster creation. Instead,
copy the following initialization actions from public buckets into
your bucket :
gs://goog-dataproc-initialization-actions-us-west1/python/pip-
install.sh
WARNING: Failed to validate permissions required for google cloud
dataproc service agent service account: 'service-
832943544474@dataproc-accounts.iam.gserviceaccount.com'. Cluster
creation could still be successful if required permissions have been
granted to the respective service accounts as mentioned in the
document https://cloud.google.com/dataproc/docs/concepts/configuring-
clusters/service-accounts#dataproc_service_accounts_2.
WARNING: The firewall rules for specified network or subnetwork would
allow ingress traffic from 0.0.0.0/0, which could be a security risk.
WARNING: The specified custom staging bucket 'dataproc-staging-us-
west1-832943544474-ngdqyb5y' is not using uniform bucket level access
IAM configuration. It is recommended to update bucket to enable the
same. See https://cloud.google.com/storage/docs/uniform-bucket-level-
access.
Created [https://dataproc.googleapis.com/v1/projects/bd-coursework-
421223/regions/us-west1/clusters/bd-coursework-421223-experiment-
cluster1] Cluster placed in zone [us-west1-c].
```

```
#Running the script in experiment1 cluster

FILENAME = 'filenames.pkl'
!gcloud dataproc jobs submit pyspark --cluster $EXPERIMENT_CLUSTER1 --
region $REGION \
    /content/spark_write_tfrec.py \
    -- --out_bucket $BUCKET --out_file $FILENAME
```

```
Job [9e2499e6ce0142be917ad2be7517ecfb] submitted.
Waiting for job output...
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
```

```
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
```

```
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl.metadata (352
bytes)
Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: py4j in
/opt/conda/miniconda3/lib/python3.7/site-packages (0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
```

```
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-05-04 16:29:57.421784: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-05-04 16:29:57.559235: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:29:57.559265: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-05-04 16:29:58.327711: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:29:58.327867: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:29:58.327904: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
Tensorflow version 2.11.0
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: You do not appear to have access to project [bd-coursework-
421223] or it does not exist.
Updated property [core/project].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.
```

```
WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Updated property [dataproc/region].
24/05/04 16:30:08 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/05/04 16:30:08 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/05/04 16:30:08 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/05/04 16:30:09 INFO org.spark_project.jetty.util.log: Logging
initialized @18085ms to org.spark_project.jetty.util.log.Slf4jLog
24/05/04 16:30:09 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/05/04 16:30:09 INFO org.spark_project.jetty.server.Server: Started
@18233ms
24/05/04 16:30:09 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@62a83bf4{HTTP/1.1, (http/1.1)}{0.0.0.0:34499}
24/05/04 16:30:11 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at bd-coursework-421223-experiment-
cluster1-m/10.138.15.210:8032
24/05/04 16:30:11 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
experiment-cluster1-m/10.138.15.210:10200
24/05/04 16:30:11 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/05/04 16:30:11 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/05/04 16:30:11 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/05/04 16:30:11 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/05/04 16:30:14 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714840051807_0001
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
```

```
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://filenames.pkl [Content-Type=application/octet-stream]...\n/ [0
files][    0.0 B/  1.3 KiB]
\r/ [1 files][  1.3 KiB/  1.3 KiB]
\r\nOperation completed over 1 objects/1.3 KiB.
\n'
24/05/04 16:32:24 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@62a83bf4{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [9e2499e6ce0142be917ad2be7517ecfb] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/33fef82d-3c8b-417a-9027-
f0f9f7b4af1b/jobs/9e2499e6ce0142be917ad2be7517ecfb/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/33fef82d-3c8b-417a-9027-
f0f9f7b4af1b/jobs/9e2499e6ce0142be917ad2be7517ecfb/driveroutput
jobUuid: 8abde196-0e6d-347c-81ff-06f09c61ebac
placement:
  clusterName: bd-coursework-421223-experiment-cluster1
  clusterUuid: 33fef82d-3c8b-417a-9027-f0f9f7b4af1b
pysparkJob:
  args:
  - --out_bucket
  - gs://bd-coursework-421223-storage
  - --out_file
  - filenames.pkl
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/33fef82d-3c8b-417a-9027-
f0f9f7b4af1b/jobs/9e2499e6ce0142be917ad2be7517ecfb/staging/
spark_write_tfrec.py
reference:
  jobId: 9e2499e6ce0142be917ad2be7517ecfb
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-05-04T16:32:25.878213Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-05-04T16:29:49.037805Z'
- state: SETUP_DONE
  stateStartTime: '2024-05-04T16:29:49.066384Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-05-04T16:29:49.364778Z'
yarnApplications:
- name: spark_write_tfrec.py
  progress: 1.0
  state: FINISHED
```

```
  trackingUrl: http://bd-coursework-421223-experiment-cluster1-
m:8088/proxy/application_1714840051807_0001/



#EXPERIMENT 2
#cluster with 1 machine (eighfold resources)

EXPERIMENT_CLUSTER2='{}-experiment-cluster2'.format(PROJECT)

!gcloud dataproc clusters create $EXPERIMENT_CLUSTER2 \
    --image-version 1.5-ubuntu18 \
    --single-node \
    --master-machine-type=n1-highmem-8 \
    --master-boot-disk-type pd-ssd \
    --master-boot-disk-size=500 \
    --initialization-actions $BUCKET/upgradepip.sh,gs://goog-dataproc-
initialization-actions-$REGION/python/pip-install.sh \
    --metadata PIP_PACKAGES=tensorflow \
    --max-idle 3600s

Waiting on operation
[projects/bd-coursework-421223/regions/us-west1/operations/9693e848-
f421-34e2-95c8-80dd5476ee8a].

WARNING: Don't create production clusters that reference
initialization actions located in the gs://goog-dataproc-
initialization-actions-REGION public buckets. These scripts are
provided as reference implementations, and they are synchronized with
ongoing GitHub repository changes—a new version of a initialization
action in public buckets may break your cluster creation. Instead,
copy the following initialization actions from public buckets into
your bucket :
gs://goog-dataproc-initialization-actions-us-west1/python/pip-
install.sh
WARNING: The firewall rules for specified network or subnetwork would
allow ingress traffic from 0.0.0.0/0, which could be a security risk.
WARNING: The specified custom staging bucket 'dataproc-staging-us-
west1-832943544474-ngdqyb5y' is not using uniform bucket level access
IAM configuration. It is recommended to update bucket to enable the
same. See https://cloud.google.com/storage/docs/uniform-bucket-level-
access.
Created [https://dataproc.googleapis.com/v1/projects/bd-coursework-
421223/regions/us-west1/clusters/bd-coursework-421223-experiment-
cluster2] Cluster placed in zone [us-west1-c].

#Running the script in experiment2 cluster

FILENAME = 'filenames.pkl'
!gcloud dataproc jobs submit pyspark --cluster $EXPERIMENT_CLUSTER2 --
region $REGION \
```

```
    /content/spark_write_tfrec.py \
    -- --out_bucket $BUCKET --out_file $FILENAME

Job [174cbc52d7734a6799bae232aea1143c] submitted.
Waiting for job output...
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
```

```
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl.metadata (352
bytes)
Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Installing collected packages: findspark
Successfully installed findspark-2.0.1
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
```

```
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: py4j in
/opt/conda/miniconda3/lib/python3.7/site-packages (0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-05-04 16:53:13.318766: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-05-04 16:53:13.458928: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:53:13.458968: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-05-04 16:53:14.194581: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:53:14.194708: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-04 16:53:14.194727: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
Tensorflow version 2.11.0
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
```

```
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: You do not appear to have access to project [bd-coursework-
421223] or it does not exist.
Updated property [core/project].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

WARNING: Property validation for compute/region was skipped.
Updated property [compute/region].
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it
by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Updated property [dataproc/region].
24/05/04 16:53:23 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/05/04 16:53:23 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/05/04 16:53:23 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/05/04 16:53:23 INFO org.spark_project.jetty.util.log: Logging
initialized @15928ms to org.spark_project.jetty.util.log.Slf4jLog
24/05/04 16:53:24 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/05/04 16:53:24 INFO org.spark_project.jetty.server.Server: Started
@16030ms
24/05/04 16:53:24 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@600eaa78{HTTP/1.1, (http/1.1)}{0.0.0.0:34389}
24/05/04 16:53:25 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at bd-coursework-421223-experiment-
cluster2-m/10.138.15.213:8032
24/05/04 16:53:25 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
experiment-cluster2-m/10.138.15.213:10200
24/05/04 16:53:25 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/05/04 16:53:25 INFO
```

```
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/05/04 16:53:25 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/05/04 16:53:25 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/05/04 16:53:28 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714841158679_0001
Saving filenames.pkl to gs://bd-coursework-421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://filenames.pkl [Content-Type=application/octet-stream]...\n/ [0
files][    0.0 B/  1.3 KiB]
\r/ [1 files][  1.3 KiB/  1.3 KiB]
\r\nOperation completed over 1 objects/1.3 KiB.
\n'
24/05/04 16:55:20 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@600eaa78{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [174cbc52d7734a6799bae232aea1143c] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/8bedbbac-e729-4f22-b98b-
829aae9e4f4f/jobs/174cbc52d7734a6799bae232aea1143c/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/8bedbbac-e729-4f22-b98b-
829aae9e4f4f/jobs/174cbc52d7734a6799bae232aea1143c/driveroutput
jobUuid: 522302cc-72ed-396a-bfb7-df4fe98da9b9
placement:
  clusterName: bd-coursework-421223-experiment-cluster2
  clusterUuid: 8bedbbac-e729-4f22-b98b-829aae9e4f4f
pysparkJob:
  args:
  - --out_bucket
  - gs://bd-coursework-421223-storage
  - --out_file
  - filenames.pkl
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/8bedbbac-e729-4f22-b98b-
829aae9e4f4f/jobs/174cbc52d7734a6799bae232aea1143c/staging/
spark_write_tfrec.py
reference:
  jobId: 174cbc52d7734a6799bae232aea1143c
```

```
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-05-04T16:55:22.726104Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-05-04T16:53:06.347757Z'
- state: SETUP_DONE
  stateStartTime: '2024-05-04T16:53:06.374884Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-05-04T16:53:06.752773Z'
yarnApplications:
- name: spark_write_tfrec.py
  progress: 1.0
  state: FINISHED
  trackingUrl: http://bd-coursework-421223-experiment-cluster2-
m:8088/proxy/application_1714841158679_0001/
```

Section 2: Speed tests

We have seen that **reading from the pre-processed TFRecord files** is **faster** than reading individual image files and decoding on the fly. This task is about **measuring this effect** and **parallelizing the tests with PySpark**.

2.1 Speed test implementation

Here is **code for time measurement** to determine the **throughput in images per second**. It doesn't render the images but extracts and prints some basic information in order to make sure the image data are read. We write the information to the null device for longer measurements `null_file=open("/dev/null", mode='w')`. That way it will not clutter our cell output.

We use batches (`dset2 = dset1.batch(batch_size)`) and select a number of batches with (`dset3 = dset2.take(batch_number)`). Then we use the `time.time()` to take the **time measurement** and take it multiple times, reading from the same dataset to see if reading speed changes with mutiple readings.

We then **vary** the size of the batch (`batch_size`) and the number of batches (`batch_number`) and **store the results for different values**. Store also the **results for each repetition** over the same dataset (repeat 2 or 3 times).

The speed test should be combined in a **function** `time_configs()` that takes a configuration, i.e. a dataset and arrays of `batch_sizes`, `batch_numbers`, and `repetitions` (an array of integers starting from 1), as **arguments** and runs the time measurement for each combination of batch_size and batch_number for the requested number of repetitions.

```
# Here are some useful values for testing your code, use higher values
later for actually testing throughput
batch_sizes = [2,4]
batch_numbers = [3,6]
repetitions = [1]

def time_configs(dataset, batch_sizes, batch_numbers, repetitions):
    dims = [len(batch_sizes),len(batch_numbers),len(repetitions)]
    print(dims)
    results = np.zeros(dims)
    params = np.zeros(dims + [3])
    print( results.shape )
    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        for bsi,bs in enumerate(batch_sizes):
            for dsi, ds in enumerate(batch_numbers):
                batched_dataset = dataset.batch(bs)
                timing_set = batched_dataset.take(ds)
                for ri,rep in enumerate(repetitions):
                    print("bs: {}, ds: {}, rep: {}".format(bs,ds,rep))
                    t0 = time.time()
                    for image, label in timing_set:
                        #print("Image batch shape
{}".format(image.numpy().shape),
                        print("Image batch shape {},
{})".format(image.numpy().shape,
                              [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
images
                    results[bsi,dsi,ri] = ( bs * ds) / td
                    params[bsi,dsi,ri] = [ bs, ds, rep ]
    print("total time: "+str(time.time()-tt))
    return results, params

for ri,rep in enumerate([3]):
  print(ri,rep)
```

```
0 3
```

Let's try this function with a **small number** of configurations of batch_sizes batch_numbers and repetions, so that we get a set of parameter combinations and corresponding reading speeds. Try reading from the image files (dataset4) and the TFRecord files (datasetTfrec).

```
[res,par] = time_configs(dataset4, batch_sizes, batch_numbers,
repetitions)
print(res)
print(par)
```

```
print("==============")

[res,par] = time_configs(datasetTfrec, batch_sizes, batch_numbers,
repetitions)
print(res)
print(par)
```

Task 2: Parallelising the speed test with Spark in the cloud. (36%)

As an exercise in **Spark programming and optimisation** as well as **performance analysis**, we will now implement the **speed test** with multiple parameters in parallel with Spark. Runing multiple tests in parallel would **not be a useful approach on a single machine, but it can be in the cloud** (you will be asked to reason about this later).

2a) Create the script (14%)

Your task is now to **port the speed test above to Spark** for running it in the cloud in Dataproc. **Adapt the speed testing** as a Spark program that performs the same actions as above, but **with Spark RDDs in a distributed way**. The distribution should be such that **each parameter combination (except repetition)** is processed in a separate Spark task.

More specifically:

- i) combine the previous cells to have the code to create a dataset and create a list of parameter combinations in an RDD (2%)
- ii) get a Spark context and create the dataset and run timing test for each combination in parallel (2%)
- iii) transform the resulting RDD to the structure (parameter_combination, images_per_second) and save these values in an array (2%)
- iv) create an RDD with all results for each parameter as (parameter_value,images_per_second) and collect the result for each parameter (2%)
- v) create an RDD with the average reading speeds for each parameter value and collect the results. Keep associativity in mind when implementing the average. (3%)
- vi) write the results to a pickle file in your bucket (2%)
- vii) Write your code it into a file using the *cell magic* `%%writefile spark_job.py` (1%)

Important: The task here is not to parallelize the pre-processing, but to run multiple speed tests in parallel using Spark.

```
### CODING TASK

#TASK 2a.i

# Function to generate parameter combinations

def generate_param_combinations(batch_sizes, batch_numbers,
```

```
repetitions):
    param_combinations = []
    for bs in batch_sizes:
        for bn in batch_numbers:
            for rep in repetitions:
                param_combinations.append((bs, bn, rep))
    return param_combinations

#defining the parameters
batch_sizes = [10,20,30,40]
batch_numbers = [3,6,9,12]
repetitions = [1,2,3]

param_combinations = generate_param_combinations(batch_sizes,
batch_numbers, repetitions)
# Creating RDD from parameter combinations
param_combinations_rdd = sc.parallelize(param_combinations)
param_combinations_rdd.collect()
```

```
[(10, 3, 1),
 (10, 3, 2),
 (10, 3, 3),
 (10, 6, 1),
 (10, 6, 2),
 (10, 6, 3),
 (10, 9, 1),
 (10, 9, 2),
 (10, 9, 3),
 (10, 12, 1),
 (10, 12, 2),
 (10, 12, 3),
 (20, 3, 1),
 (20, 3, 2),
 (20, 3, 3),
 (20, 6, 1),
 (20, 6, 2),
 (20, 6, 3),
 (20, 9, 1),
 (20, 9, 2),
 (20, 9, 3),
 (20, 12, 1),
 (20, 12, 2),
 (20, 12, 3),
 (30, 3, 1),
 (30, 3, 2),
 (30, 3, 3),
 (30, 6, 1),
 (30, 6, 2),
 (30, 6, 3),
 (30, 9, 1),
```

```
 (30, 9, 2),
 (30, 9, 3),
 (30, 12, 1),
 (30, 12, 2),
 (30, 12, 3),
 (40, 3, 1),
 (40, 3, 2),
 (40, 3, 3),
 (40, 6, 1),
 (40, 6, 2),
 (40, 6, 3),
 (40, 9, 1),
 (40, 9, 2),
 (40, 9, 3),
 (40, 12, 1),
 (40, 12, 2),
 (40, 12, 3)]
```

```
# Below are the helper functions required to handle the two types of
data: tf image dataset & tfrecord files
def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2

def resize_and_crop_image(image, label):
    w = tf.shape(image)[0]
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,lambda: tf.image.resize(image,
[w*tw/w, h*tw/w]),lambda: tf.image.resize(image, [w*th/h, h*th/h]))
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def read_tfrecord(example):
    features = {
        "image": tf.io.FixedLenFeature([], tf.string),  # tf.string =
bytestring (not text string)
        "class": tf.io.FixedLenFeature([], tf.int64) #,   # shape []
```

```
means scalar
    }
    # decode the TFRecord
    example = tf.io.parse_single_example(example, features)
    image = tf.image.decode_jpeg(example['image'], channels=3)
    image = tf.reshape(image, [*TARGET_SIZE, 3])
    class_num = example['class']
    return image, class_num

def load_dataset(filenames):
    # read from TFRecords. For optimal performance, read from multiple
    # TFRecord files at once and set the option
experimental_deterministic = False
    # to allow order-altering optimizations.
    option_no_order = tf.data.Options()
    option_no_order.experimental_deterministic = False

    dataset = tf.data.TFRecordDataset(filenames)
    dataset = dataset.with_options(option_no_order)
    dataset = dataset.map(read_tfrecord)
    return dataset

#function for running speed test on Tensorflow Datasets
TARGET_SIZE = [192, 192]

def img_time_configs(params):
    batch_size, batch_number, repetitions=params

    GCS_PATTERN = 'gs://flowers-public/*/*.jpg'
    dsetFiles = tf.data.Dataset.list_files(GCS_PATTERN)
    dsetDecoded = dsetFiles.map(decode_jpeg_and_label)
    dsetResized = dsetDecoded.map(resize_and_crop_image)

    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        batched_dataset = dsetResized.batch(batch_size)
        timing_set = batched_dataset.take(batch_number)
        results=[]
        params=[]
        for i in range(repetitions):
                    t0 = time.time()
                    for image, label in timing_set:
                        print("Image batch shape {},
{})".format(image.numpy().shape,
                                [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
images
                    result = ( batch_size * batch_number) / td
```

```
                    dataset_size=batch_size * batch_number
                    param = [ batch_size, batch_number, repetitions,
dataset_size ]
                    results.append(result)
                    params.append(param)
    return results, params

#function for running speed test on tfrecord files
def tfrec_time_configs(params):
    batch_size, batch_number, repetitions=params

    tfrecord='gs://flowers-public/tfrecords-jpeg-192x192-2/'
    filenames = tf.io.gfile.glob(tfrecord + "*.tfrec")
    datasetTfrec = load_dataset(filenames)

    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        batched_dataset = datasetTfrec.batch(batch_size)
        timing_set = batched_dataset.take(batch_number)
        results=[]
        params=[]
        for i in range(repetitions):
                    t0 = time.time()
                    for image, label in timing_set:
                        print("Image batch shape {},
{})".format(image.numpy().shape,
                              [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
images
                    result = ( batch_size * batch_number) / td
                    dataset_size=batch_size * batch_number
                    param = [ batch_size, batch_number, repetitions,
dataset_size]
                    results.append(result)
                    params.append(param)
    return results, params


#TASK 2a.ii Running the speed test for each parameter combination in
parallel

#image dataset

img_time_configs_result=param_combinations_rdd.map(img_time_configs)
# Applying flatMap to flatten the nested lists
img_time_configs_flattened = img_time_configs_result.flatMap(lambda x:
```

```
[((params, scores)) for params, scores in zip(x[0], x[1])]).cache()
img_time_configs_flattened.collect()
```

```
[(11.506540704386644, [10, 3, 1, 30]),
 (11.44209819522213, [10, 3, 2, 30]),
 (16.173841151537214, [10, 3, 2, 30]),
 (15.911086461800037, [10, 3, 3, 30]),
 (15.90383867482167, [10, 3, 3, 30]),
 (15.953059666863604, [10, 3, 3, 30]),
 (15.943580389601085, [10, 6, 1, 60]),
 (13.886570765556781, [10, 6, 2, 60]),
 (14.784576417102102, [10, 6, 2, 60]),
 (18.486438861548855, [10, 6, 3, 60]),
 (19.901354513090336, [10, 6, 3, 60]),
 (15.173163751573188, [10, 6, 3, 60]),
 (15.095590680250087, [10, 9, 1, 90]),
 (12.772512281308659, [10, 9, 2, 90]),
 (14.468060218121265, [10, 9, 2, 90]),
 (20.191533403878584, [10, 9, 3, 90]),
 (21.39066775797669, [10, 9, 3, 90]),
 (21.85055445250778, [10, 9, 3, 90]),
 (20.519937730367797, [10, 12, 1, 120]),
 (22.488931612339258, [10, 12, 2, 120]),
 (22.69493967663408, [10, 12, 2, 120]),
 (22.774661089251982, [10, 12, 3, 120]),
 (32.32016165587854, [10, 12, 3, 120]),
 (29.952560696433448, [10, 12, 3, 120]),
 (11.518448939332739, [20, 3, 1, 60]),
 (19.505800508816154, [20, 3, 2, 60]),
 (13.480963215826288, [20, 3, 2, 60]),
 (20.289861802870753, [20, 3, 3, 60]),
 (22.720057520083774, [20, 3, 3, 60]),
 (20.127273141707278, [20, 3, 3, 60]),
 (23.271397286753963, [20, 6, 1, 120]),
 (29.75123802297704, [20, 6, 2, 120]),
 (30.127501060383878, [20, 6, 2, 120]),
 (20.495681164263168, [20, 6, 3, 120]),
 (21.63843365569517, [20, 6, 3, 120]),
 (23.49565750699023, [20, 6, 3, 120]),
 (22.400870381583267, [20, 9, 1, 180]),
 (26.303819285665384, [20, 9, 2, 180]),
 (32.33059400527036, [20, 9, 2, 180]),
 (20.98763325543186, [20, 9, 3, 180]),
 (17.347271233414823, [20, 9, 3, 180]),
 (22.48577719523931, [20, 9, 3, 180]),
 (27.634448679641768, [20, 12, 1, 240]),
 (11.64866503853764, [20, 12, 2, 240]),
 (19.72640300427239, [20, 12, 2, 240]),
 (23.454121131140358, [20, 12, 3, 240]),
 (23.227688269986153, [20, 12, 3, 240]),
```

```
(22.55305384025872, [20, 12, 3, 240]),
(14.186070353239343, [30, 3, 1, 90]),
(17.70065276097462, [30, 3, 2, 90]),
(18.881288936459658, [30, 3, 2, 90]),
(14.342978039348418, [30, 3, 3, 90]),
(16.97820397869985, [30, 3, 3, 90]),
(21.798044947283632, [30, 3, 3, 90]),
(17.823045790448685, [30, 6, 1, 180]),
(14.775015266233066, [30, 6, 2, 180]),
(18.240971250445227, [30, 6, 2, 180]),
(22.66118517764788, [30, 6, 3, 180]),
(22.53229045029261, [30, 6, 3, 180]),
(21.979194002081776, [30, 6, 3, 180]),
(26.294980724959593, [30, 9, 1, 270]),
(29.292553767151137, [30, 9, 2, 270]),
(21.197439959820663, [30, 9, 2, 270]),
(24.086302676593604, [30, 9, 3, 270]),
(30.820915422453368, [30, 9, 3, 270]),
(22.46898863249355, [30, 9, 3, 270]),
(23.190564570621994, [30, 12, 1, 360]),
(29.324344479909318, [30, 12, 2, 360]),
(23.005321065164967, [30, 12, 2, 360]),
(24.236595357647335, [30, 12, 3, 360]),
(17.518995781249753, [30, 12, 3, 360]),
(21.9896493148058, [30, 12, 3, 360]),
(19.14391941057981, [40, 3, 1, 120]),
(29.079553923964628, [40, 3, 2, 120]),
(29.238516967340665, [40, 3, 2, 120]),
(24.805685140851068, [40, 3, 3, 120]),
(22.232822283758647, [40, 3, 3, 120]),
(21.558734104702115, [40, 3, 3, 120]),
(23.097335745540768, [40, 6, 1, 240]),
(33.33865643652299, [40, 6, 2, 240]),
(23.26473942045424, [40, 6, 2, 240]),
(33.51591997355572, [40, 6, 3, 240]),
(37.8987249412958, [40, 6, 3, 240]),
(26.47626195196675, [40, 6, 3, 240]),
(33.85964585750585, [40, 9, 1, 360]),
(33.90155225362838, [40, 9, 2, 360]),
(17.513134856013377, [40, 9, 2, 360]),
(30.46944104319882, [40, 9, 3, 360]),
(34.192500531926314, [40, 9, 3, 360]),
(26.88139077508018, [40, 9, 3, 360]),
(23.244616629191356, [40, 12, 1, 480]),
(37.04668806331751, [40, 12, 2, 480]),
(33.65814508633085, [40, 12, 2, 480]),
(23.365825646347943, [40, 12, 3, 480]),
(23.398567624992516, [40, 12, 3, 480]),
(34.243463564281875, [40, 12, 3, 480])]
```

```
#tfrec data
tfrec_time_configs_result=param_combinations_rdd.map(tfrec_time_config
s)
# Applying flatMap to flatten the nested lists
tfrec_time_configs_flattened =
tfrec_time_configs_result.flatMap(lambda x: [((params, scores)) for
params, scores in zip(x[0], x[1])]).cache()
tfrec_time_configs_flattened.collect()
```

```
[(101.15718584416555, [10, 3, 1, 30]),
 (121.21273306135818, [10, 3, 2, 30]),
 (87.66105733225118, [10, 3, 2, 30]),
 (140.8579600472403, [10, 3, 3, 30]),
 (160.37951950811336, [10, 3, 3, 30]),
 (182.56927829277743, [10, 3, 3, 30]),
 (225.75894775463996, [10, 6, 1, 60]),
 (235.49415658608754, [10, 6, 2, 60]),
 (292.036693415029, [10, 6, 2, 60]),
 (188.61655244365298, [10, 6, 3, 60]),
 (295.0217111988521, [10, 6, 3, 60]),
 (262.2953662842932, [10, 6, 3, 60]),
 (263.45689351937153, [10, 9, 1, 90]),
 (247.85726620367785, [10, 9, 2, 90]),
 (287.80040697483815, [10, 9, 2, 90]),
 (334.20719415033716, [10, 9, 3, 90]),
 (370.60085491398826, [10, 9, 3, 90]),
 (361.1500166947783, [10, 9, 3, 90]),
 (494.59236526901975, [10, 12, 1, 120]),
 (538.8492791668183, [10, 12, 2, 120]),
 (550.0744592605222, [10, 12, 2, 120]),
 (584.1804969259857, [10, 12, 3, 120]),
 (542.5762093566268, [10, 12, 3, 120]),
 (534.5996475772797, [10, 12, 3, 120]),
 (297.6030992754403, [20, 3, 1, 60]),
 (395.2540285848908, [20, 3, 2, 60]),
 (437.77918490324464, [20, 3, 2, 60]),
 (364.6068923151816, [20, 3, 3, 60]),
 (297.6668133365506, [20, 3, 3, 60]),
 (435.5290037416367, [20, 3, 3, 60]),
 (624.2259194721601, [20, 6, 1, 120]),
 (521.519600122682, [20, 6, 2, 120]),
 (627.5868312079872, [20, 6, 2, 120]),
 (512.2965258007615, [20, 6, 3, 120]),
 (729.3152225258723, [20, 6, 3, 120]),
 (557.5913021706907, [20, 6, 3, 120]),
 (536.2508630021393, [20, 9, 1, 180]),
 (687.8877766904839, [20, 9, 2, 180]),
 (730.8500893990275, [20, 9, 2, 180]),
 (612.043572476087, [20, 9, 3, 180]),
 (684.1811301057028, [20, 9, 3, 180]),
```

```
(769.8309170369796, [20, 9, 3, 180]),
(728.6305364806401, [20, 12, 1, 240]),
(560.1383105699969, [20, 12, 2, 240]),
(832.1364440847584, [20, 12, 2, 240]),
(649.4347534925491, [20, 12, 3, 240]),
(570.0777106120688, [20, 12, 3, 240]),
(564.9984957814528, [20, 12, 3, 240]),
(277.5928110246643, [30, 3, 1, 90]),
(278.25516742676484, [30, 3, 2, 90]),
(546.444028505791, [30, 3, 2, 90]),
(432.38365340340283, [30, 3, 3, 90]),
(269.0965964378259, [30, 3, 3, 90]),
(393.0726922476181, [30, 3, 3, 90]),
(545.8967938516224, [30, 6, 1, 180]),
(524.6713015341763, [30, 6, 2, 180]),
(704.0435085481204, [30, 6, 2, 180]),
(642.8719884057773, [30, 6, 3, 180]),
(654.920543767073, [30, 6, 3, 180]),
(667.2470761668155, [30, 6, 3, 180]),
(407.41072729477054, [30, 9, 1, 270]),
(415.5706719479853, [30, 9, 2, 270]),
(629.1487457437287, [30, 9, 2, 270]),
(555.6165415894747, [30, 9, 3, 270]),
(421.5407579351889, [30, 9, 3, 270]),
(844.2515021843176, [30, 9, 3, 270]),
(919.08575296339, [30, 12, 1, 360]),
(553.0887524720415, [30, 12, 2, 360]),
(547.6466509016524, [30, 12, 2, 360]),
(595.3634381773226, [30, 12, 3, 360]),
(578.0401104361555, [30, 12, 3, 360]),
(619.9692796988572, [30, 12, 3, 360]),
(364.8918409868939, [40, 3, 1, 120]),
(520.7684730695053, [40, 3, 2, 120]),
(569.0130157606809, [40, 3, 2, 120]),
(367.0504635929658, [40, 3, 3, 120]),
(545.0705167983002, [40, 3, 3, 120]),
(368.41504735511836, [40, 3, 3, 120]),
(564.1160631076476, [40, 6, 1, 240]),
(368.3357812626925, [40, 6, 2, 240]),
(371.35288879182565, [40, 6, 2, 240]),
(364.7762747364645, [40, 6, 3, 240]),
(374.06962610241203, [40, 6, 3, 240]),
(604.1239199673042, [40, 6, 3, 240]),
(551.5641897926524, [40, 9, 1, 360]),
(553.6279759416757, [40, 9, 2, 360]),
(565.3155820792704, [40, 9, 2, 360]),
(560.1612724915991, [40, 9, 3, 360]),
(565.1526580842929, [40, 9, 3, 360]),
(563.1767214483152, [40, 9, 3, 360]),
```

```
 (747.8493978991662, [40, 12, 1, 480]),
 (1040.557743843663, [40, 12, 2, 480]),
 (756.8589202820422, [40, 12, 2, 480]),
 (1011.31331803587, [40, 12, 3, 480]),
 (756.0869238650055, [40, 12, 3, 480]),
 (1000.4297950557568, [40, 12, 3, 480])]


#TASK 2a.iii transforming the resulting rdd to the structure
( parameter_combination, images_per_second ) and
# saving as arrays

#image dataset
img_time_configs_transformed = img_time_configs_flattened.map(lambda
x: (x[1], x[0])).cache()        ### TASK 2c ###

img_flattened_list = [(params[0], params[1], params[2], params[3],
score) for params, score in img_time_configs_transformed.collect()]
img_time_configs_array = np.array(img_flattened_list)

img_time_configs_array

array([[ 10.        ,   3.        ,   1.        ,  30.        ,
         11.5065407 ],
       [ 10.        ,   3.        ,   2.        ,  30.        ,
         11.4420982 ],
       [ 10.        ,   3.        ,   2.        ,  30.        ,
         16.17384115],
       [ 10.        ,   3.        ,   3.        ,  30.        ,
         15.91108646],
       [ 10.        ,   3.        ,   3.        ,  30.        ,
         15.90383867],
       [ 10.        ,   3.        ,   3.        ,  30.        ,
         15.95305967],
       [ 10.        ,   6.        ,   1.        ,  60.        ,
         15.94358039],
       [ 10.        ,   6.        ,   2.        ,  60.        ,
         13.88657077],
       [ 10.        ,   6.        ,   2.        ,  60.        ,
         14.78457642],
       [ 10.        ,   6.        ,   3.        ,  60.        ,
         18.48643886],
       [ 10.        ,   6.        ,   3.        ,  60.        ,
         19.90135451],
       [ 10.        ,   6.        ,   3.        ,  60.        ,
         15.17316375],
       [ 10.        ,   9.        ,   1.        ,  90.        ,
         15.09559068],
       [ 10.        ,   9.        ,   2.        ,  90.        ,
```

```
        12.77251228],
       [ 10.        ,   9.        ,   2.        ,  90.        ,
        14.46806022],
       [ 10.        ,   9.        ,   3.        ,  90.        ,
        20.1915334 ],
       [ 10.        ,   9.        ,   3.        ,  90.        ,
        21.39066776],
       [ 10.        ,   9.        ,   3.        ,  90.        ,
        21.85055445],
       [ 10.        ,  12.        ,   1.        , 120.        ,
        20.51993773],
       [ 10.        ,  12.        ,   2.        , 120.        ,
        22.48893161],
       [ 10.        ,  12.        ,   2.        , 120.        ,
        22.69493968],
       [ 10.        ,  12.        ,   3.        , 120.        ,
        22.77466109],
       [ 10.        ,  12.        ,   3.        , 120.        ,
        32.32016166],
       [ 10.        ,  12.        ,   3.        , 120.        ,
        29.9525607 ],
       [ 20.        ,   3.        ,   1.        ,  60.        ,
        11.51844894],
       [ 20.        ,   3.        ,   2.        ,  60.        ,
        19.50580051],
       [ 20.        ,   3.        ,   2.        ,  60.        ,
        13.48096322],
       [ 20.        ,   3.        ,   3.        ,  60.        ,
        20.2898618 ],
       [ 20.        ,   3.        ,   3.        ,  60.        ,
        22.72005752],
       [ 20.        ,   3.        ,   3.        ,  60.        ,
        20.12727314],
       [ 20.        ,   6.        ,   1.        , 120.        ,
        23.27139729],
       [ 20.        ,   6.        ,   2.        , 120.        ,
        29.75123802],
       [ 20.        ,   6.        ,   2.        , 120.        ,
        30.12750106],
       [ 20.        ,   6.        ,   3.        , 120.        ,
        20.49568116],
       [ 20.        ,   6.        ,   3.        , 120.        ,
        21.63843366],
       [ 20.        ,   6.        ,   3.        , 120.        ,
        23.49565751],
       [ 20.        ,   9.        ,   1.        , 180.        ,
        22.40087038],
       [ 20.        ,   9.        ,   2.        , 180.        ,
        26.30381929],
```

```
       [ 20.        ,   9.        ,   2.        , 180.        ,
         32.33059401],
       [ 20.        ,   9.        ,   3.        , 180.        ,
         20.98763326],
       [ 20.        ,   9.        ,   3.        , 180.        ,
         17.34727123],
       [ 20.        ,   9.        ,   3.        , 180.        ,
         22.4857772 ],
       [ 20.        ,  12.        ,   1.        , 240.        ,
         27.63444868],
       [ 20.        ,  12.        ,   2.        , 240.        ,
         11.64866504],
       [ 20.        ,  12.        ,   2.        , 240.        ,
         19.726403  ],
       [ 20.        ,  12.        ,   3.        , 240.        ,
         23.45412113],
       [ 20.        ,  12.        ,   3.        , 240.        ,
         23.22768827],
       [ 20.        ,  12.        ,   3.        , 240.        ,
         22.55305384],
       [ 30.        ,   3.        ,   1.        ,  90.        ,
         14.18607035],
       [ 30.        ,   3.        ,   2.        ,  90.        ,
         17.70065276],
       [ 30.        ,   3.        ,   2.        ,  90.        ,
         18.88128894],
       [ 30.        ,   3.        ,   3.        ,  90.        ,
         14.34297804],
       [ 30.        ,   3.        ,   3.        ,  90.        ,
         16.97820398],
       [ 30.        ,   3.        ,   3.        ,  90.        ,
         21.79804495],
       [ 30.        ,   6.        ,   1.        , 180.        ,
         17.82304579],
       [ 30.        ,   6.        ,   2.        , 180.        ,
         14.77501527],
       [ 30.        ,   6.        ,   2.        , 180.        ,
         18.24097125],
       [ 30.        ,   6.        ,   3.        , 180.        ,
         22.66118518],
       [ 30.        ,   6.        ,   3.        , 180.        ,
         22.53229045],
       [ 30.        ,   6.        ,   3.        , 180.        ,
         21.979194  ],
       [ 30.        ,   9.        ,   1.        , 270.        ,
         26.29498072],
       [ 30.        ,   9.        ,   2.        , 270.        ,
         29.29255377],
       [ 30.        ,   9.        ,   2.        , 270.        ,
```

```
        21.19743996],
       [ 30.        ,   9.        ,   3.        , 270.        ,
        24.08630268],
       [ 30.        ,   9.        ,   3.        , 270.        ,
        30.82091542],
       [ 30.        ,   9.        ,   3.        , 270.        ,
        22.46898863],
       [ 30.        ,  12.        ,   1.        , 360.        ,
        23.19056457],
       [ 30.        ,  12.        ,   2.        , 360.        ,
        29.32434448],
       [ 30.        ,  12.        ,   2.        , 360.        ,
        23.00532107],
       [ 30.        ,  12.        ,   3.        , 360.        ,
        24.23659536],
       [ 30.        ,  12.        ,   3.        , 360.        ,
        17.51899578],
       [ 30.        ,  12.        ,   3.        , 360.        ,
        21.98964931],
       [ 40.        ,   3.        ,   1.        , 120.        ,
        19.14391941],
       [ 40.        ,   3.        ,   2.        , 120.        ,
        29.07955392],
       [ 40.        ,   3.        ,   2.        , 120.        ,
        29.23851697],
       [ 40.        ,   3.        ,   3.        , 120.        ,
        24.80568514],
       [ 40.        ,   3.        ,   3.        , 120.        ,
        22.23282228],
       [ 40.        ,   3.        ,   3.        , 120.        ,
        21.5587341 ],
       [ 40.        ,   6.        ,   1.        , 240.        ,
        23.09733575],
       [ 40.        ,   6.        ,   2.        , 240.        ,
        33.33865644],
       [ 40.        ,   6.        ,   2.        , 240.        ,
        23.26473942],
       [ 40.        ,   6.        ,   3.        , 240.        ,
        33.51591997],
       [ 40.        ,   6.        ,   3.        , 240.        ,
        37.89872494],
       [ 40.        ,   6.        ,   3.        , 240.        ,
        26.47626195],
       [ 40.        ,   9.        ,   1.        , 360.        ,
        33.85964586],
       [ 40.        ,   9.        ,   2.        , 360.        ,
        33.90155225],
       [ 40.        ,   9.        ,   2.        , 360.        ,
        17.51313486],
```

```
       [ 40.        ,   9.        ,   3.        , 360.        ,
         30.46944104],
       [ 40.        ,   9.        ,   3.        , 360.        ,
         34.19250053],
       [ 40.        ,   9.        ,   3.        , 360.        ,
         26.88139078],
       [ 40.        ,  12.        ,   1.        , 480.        ,
         23.24461663],
       [ 40.        ,  12.        ,   2.        , 480.        ,
         37.04668806],
       [ 40.        ,  12.        ,   2.        , 480.        ,
         33.65814509],
       [ 40.        ,  12.        ,   3.        , 480.        ,
         23.36582565],
       [ 40.        ,  12.        ,   3.        , 480.        ,
         23.39856762],
       [ 40.        ,  12.        ,   3.        , 480.        ,
         34.24346356]])

#tfrec data
tfrec_time_configs_transformed =
tfrec_time_configs_flattened.map(lambda x: (x[1], x[0])).cache()
### TASK 2c ###

tfrec_flattened_list = [(params[0], params[1], params[2], params[3],
score) for params, score in tfrec_time_configs_transformed.collect()]
tfrec_time_configs_array = np.array(tfrec_flattened_list)

tfrec_time_configs_array

array([[1.00000000e+01, 3.00000000e+00, 1.00000000e+00,
3.00000000e+01,
        1.01157186e+02],
       [1.00000000e+01, 3.00000000e+00, 2.00000000e+00,
3.00000000e+01,
        1.21212733e+02],
       [1.00000000e+01, 3.00000000e+00, 2.00000000e+00,
3.00000000e+01,
        8.76610573e+01],
       [1.00000000e+01, 3.00000000e+00, 3.00000000e+00,
3.00000000e+01,
        1.40857960e+02],
       [1.00000000e+01, 3.00000000e+00, 3.00000000e+00,
3.00000000e+01,
        1.60379520e+02],
       [1.00000000e+01, 3.00000000e+00, 3.00000000e+00,
3.00000000e+01,
        1.82569278e+02],
       [1.00000000e+01, 6.00000000e+00, 1.00000000e+00,
6.00000000e+01,
```

```
        2.25758948e+02],
       [1.00000000e+01, 6.00000000e+00, 2.00000000e+00,
6.00000000e+01,
        2.35494157e+02],
       [1.00000000e+01, 6.00000000e+00, 2.00000000e+00,
6.00000000e+01,
        2.92036693e+02],
       [1.00000000e+01, 6.00000000e+00, 3.00000000e+00,
6.00000000e+01,
        1.88616552e+02],
       [1.00000000e+01, 6.00000000e+00, 3.00000000e+00,
6.00000000e+01,
        2.95021711e+02],
       [1.00000000e+01, 6.00000000e+00, 3.00000000e+00,
6.00000000e+01,
        2.62295366e+02],
       [1.00000000e+01, 9.00000000e+00, 1.00000000e+00,
9.00000000e+01,
        2.63456894e+02],
       [1.00000000e+01, 9.00000000e+00, 2.00000000e+00,
9.00000000e+01,
        2.47857266e+02],
       [1.00000000e+01, 9.00000000e+00, 2.00000000e+00,
9.00000000e+01,
        2.87800407e+02],
       [1.00000000e+01, 9.00000000e+00, 3.00000000e+00,
9.00000000e+01,
        3.34207194e+02],
       [1.00000000e+01, 9.00000000e+00, 3.00000000e+00,
9.00000000e+01,
        3.70600855e+02],
       [1.00000000e+01, 9.00000000e+00, 3.00000000e+00,
9.00000000e+01,
        3.61150017e+02],
       [1.00000000e+01, 1.20000000e+01, 1.00000000e+00,
1.20000000e+02,
        4.94592365e+02],
       [1.00000000e+01, 1.20000000e+01, 2.00000000e+00,
1.20000000e+02,
        5.38849279e+02],
       [1.00000000e+01, 1.20000000e+01, 2.00000000e+00,
1.20000000e+02,
        5.50074459e+02],
       [1.00000000e+01, 1.20000000e+01, 3.00000000e+00,
1.20000000e+02,
        5.84180497e+02],
       [1.00000000e+01, 1.20000000e+01, 3.00000000e+00,
1.20000000e+02,
        5.42576209e+02],
```

```
       [1.00000000e+01, 1.20000000e+01, 3.00000000e+00,
1.20000000e+02,
        5.34599648e+02],
       [2.00000000e+01, 3.00000000e+00, 1.00000000e+00,
6.00000000e+01,
        2.97603099e+02],
       [2.00000000e+01, 3.00000000e+00, 2.00000000e+00,
6.00000000e+01,
        3.95254029e+02],
       [2.00000000e+01, 3.00000000e+00, 2.00000000e+00,
6.00000000e+01,
        4.37779185e+02],
       [2.00000000e+01, 3.00000000e+00, 3.00000000e+00,
6.00000000e+01,
        3.64606892e+02],
       [2.00000000e+01, 3.00000000e+00, 3.00000000e+00,
6.00000000e+01,
        2.97666813e+02],
       [2.00000000e+01, 3.00000000e+00, 3.00000000e+00,
6.00000000e+01,
        4.35529004e+02],
       [2.00000000e+01, 6.00000000e+00, 1.00000000e+00,
1.20000000e+02,
        6.24225919e+02],
       [2.00000000e+01, 6.00000000e+00, 2.00000000e+00,
1.20000000e+02,
        5.21519600e+02],
       [2.00000000e+01, 6.00000000e+00, 2.00000000e+00,
1.20000000e+02,
        6.27586831e+02],
       [2.00000000e+01, 6.00000000e+00, 3.00000000e+00,
1.20000000e+02,
        5.12296526e+02],
       [2.00000000e+01, 6.00000000e+00, 3.00000000e+00,
1.20000000e+02,
        7.29315223e+02],
       [2.00000000e+01, 6.00000000e+00, 3.00000000e+00,
1.20000000e+02,
        5.57591302e+02],
       [2.00000000e+01, 9.00000000e+00, 1.00000000e+00,
1.80000000e+02,
        5.36250863e+02],
       [2.00000000e+01, 9.00000000e+00, 2.00000000e+00,
1.80000000e+02,
        6.87887777e+02],
       [2.00000000e+01, 9.00000000e+00, 2.00000000e+00,
1.80000000e+02,
        7.30850089e+02],
       [2.00000000e+01, 9.00000000e+00, 3.00000000e+00,
```

```
1.80000000e+02,
        6.12043572e+02],
       [2.00000000e+01, 9.00000000e+00, 3.00000000e+00,
1.80000000e+02,
        6.84181130e+02],
       [2.00000000e+01, 9.00000000e+00, 3.00000000e+00,
1.80000000e+02,
        7.69830917e+02],
       [2.00000000e+01, 1.20000000e+01, 1.00000000e+00,
2.40000000e+02,
        7.28630536e+02],
       [2.00000000e+01, 1.20000000e+01, 2.00000000e+00,
2.40000000e+02,
        5.60138311e+02],
       [2.00000000e+01, 1.20000000e+01, 2.00000000e+00,
2.40000000e+02,
        8.32136444e+02],
       [2.00000000e+01, 1.20000000e+01, 3.00000000e+00,
2.40000000e+02,
        6.49434753e+02],
       [2.00000000e+01, 1.20000000e+01, 3.00000000e+00,
2.40000000e+02,
        5.70077711e+02],
       [2.00000000e+01, 1.20000000e+01, 3.00000000e+00,
2.40000000e+02,
        5.64998496e+02],
       [3.00000000e+01, 3.00000000e+00, 1.00000000e+00,
9.00000000e+01,
        2.77592811e+02],
       [3.00000000e+01, 3.00000000e+00, 2.00000000e+00,
9.00000000e+01,
        2.78255167e+02],
       [3.00000000e+01, 3.00000000e+00, 2.00000000e+00,
9.00000000e+01,
        5.46444029e+02],
       [3.00000000e+01, 3.00000000e+00, 3.00000000e+00,
9.00000000e+01,
        4.32383653e+02],
       [3.00000000e+01, 3.00000000e+00, 3.00000000e+00,
9.00000000e+01,
        2.69096596e+02],
       [3.00000000e+01, 3.00000000e+00, 3.00000000e+00,
9.00000000e+01,
        3.93072692e+02],
       [3.00000000e+01, 6.00000000e+00, 1.00000000e+00,
1.80000000e+02,
        5.45896794e+02],
       [3.00000000e+01, 6.00000000e+00, 2.00000000e+00,
1.80000000e+02,
```

```
        5.24671302e+02],
       [3.00000000e+01, 6.00000000e+00, 2.00000000e+00,
1.80000000e+02,
        7.04043509e+02],
       [3.00000000e+01, 6.00000000e+00, 3.00000000e+00,
1.80000000e+02,
        6.42871988e+02],
       [3.00000000e+01, 6.00000000e+00, 3.00000000e+00,
1.80000000e+02,
        6.54920544e+02],
       [3.00000000e+01, 6.00000000e+00, 3.00000000e+00,
1.80000000e+02,
        6.67247076e+02],
       [3.00000000e+01, 9.00000000e+00, 1.00000000e+00,
2.70000000e+02,
        4.07410727e+02],
       [3.00000000e+01, 9.00000000e+00, 2.00000000e+00,
2.70000000e+02,
        4.15570672e+02],
       [3.00000000e+01, 9.00000000e+00, 2.00000000e+00,
2.70000000e+02,
        6.29148746e+02],
       [3.00000000e+01, 9.00000000e+00, 3.00000000e+00,
2.70000000e+02,
        5.55616542e+02],
       [3.00000000e+01, 9.00000000e+00, 3.00000000e+00,
2.70000000e+02,
        4.21540758e+02],
       [3.00000000e+01, 9.00000000e+00, 3.00000000e+00,
2.70000000e+02,
        8.44251502e+02],
       [3.00000000e+01, 1.20000000e+01, 1.00000000e+00,
3.60000000e+02,
        9.19085753e+02],
       [3.00000000e+01, 1.20000000e+01, 2.00000000e+00,
3.60000000e+02,
        5.53088752e+02],
       [3.00000000e+01, 1.20000000e+01, 2.00000000e+00,
3.60000000e+02,
        5.47646651e+02],
       [3.00000000e+01, 1.20000000e+01, 3.00000000e+00,
3.60000000e+02,
        5.95363438e+02],
       [3.00000000e+01, 1.20000000e+01, 3.00000000e+00,
3.60000000e+02,
        5.78040110e+02],
       [3.00000000e+01, 1.20000000e+01, 3.00000000e+00,
3.60000000e+02,
        6.19969280e+02],
```

```
       [4.00000000e+01, 3.00000000e+00, 1.00000000e+00,
1.20000000e+02,
        3.64891841e+02],
       [4.00000000e+01, 3.00000000e+00, 2.00000000e+00,
1.20000000e+02,
        5.20768473e+02],
       [4.00000000e+01, 3.00000000e+00, 2.00000000e+00,
1.20000000e+02,
        5.69013016e+02],
       [4.00000000e+01, 3.00000000e+00, 3.00000000e+00,
1.20000000e+02,
        3.67050464e+02],
       [4.00000000e+01, 3.00000000e+00, 3.00000000e+00,
1.20000000e+02,
        5.45070517e+02],
       [4.00000000e+01, 3.00000000e+00, 3.00000000e+00,
1.20000000e+02,
        3.68415047e+02],
       [4.00000000e+01, 6.00000000e+00, 1.00000000e+00,
2.40000000e+02,
        5.64116063e+02],
       [4.00000000e+01, 6.00000000e+00, 2.00000000e+00,
2.40000000e+02,
        3.68335781e+02],
       [4.00000000e+01, 6.00000000e+00, 2.00000000e+00,
2.40000000e+02,
        3.71352889e+02],
       [4.00000000e+01, 6.00000000e+00, 3.00000000e+00,
2.40000000e+02,
        3.64776275e+02],
       [4.00000000e+01, 6.00000000e+00, 3.00000000e+00,
2.40000000e+02,
        3.74069626e+02],
       [4.00000000e+01, 6.00000000e+00, 3.00000000e+00,
2.40000000e+02,
        6.04123920e+02],
       [4.00000000e+01, 9.00000000e+00, 1.00000000e+00,
3.60000000e+02,
        5.51564190e+02],
       [4.00000000e+01, 9.00000000e+00, 2.00000000e+00,
3.60000000e+02,
        5.53627976e+02],
       [4.00000000e+01, 9.00000000e+00, 2.00000000e+00,
3.60000000e+02,
        5.65315582e+02],
       [4.00000000e+01, 9.00000000e+00, 3.00000000e+00,
3.60000000e+02,
        5.60161272e+02],
       [4.00000000e+01, 9.00000000e+00, 3.00000000e+00,
```

```
3.60000000e+02,
        5.65152658e+02],
       [4.00000000e+01, 9.00000000e+00, 3.00000000e+00,
3.60000000e+02,
        5.63176721e+02],
       [4.00000000e+01, 1.20000000e+01, 1.00000000e+00,
4.80000000e+02,
        7.47849398e+02],
       [4.00000000e+01, 1.20000000e+01, 2.00000000e+00,
4.80000000e+02,
        1.04055774e+03],
       [4.00000000e+01, 1.20000000e+01, 2.00000000e+00,
4.80000000e+02,
        7.56858920e+02],
       [4.00000000e+01, 1.20000000e+01, 3.00000000e+00,
4.80000000e+02,
        1.01131332e+03],
       [4.00000000e+01, 1.20000000e+01, 3.00000000e+00,
4.80000000e+02,
        7.56086924e+02],
       [4.00000000e+01, 1.20000000e+01, 3.00000000e+00,
4.80000000e+02,
        1.00042980e+03]])



#TASK 2a.iv Extracting reading speed for each parameter value

#image dataset

### TASK 2c ###
img_batch_size_rdd = img_time_configs_transformed.map(lambda x: (x[0]
[0], x[1])).cache()
img_batch_number_rdd = img_time_configs_transformed.map(lambda x:
(x[0][1], x[1])).cache()
img_repetitions_rdd = img_time_configs_transformed.map(lambda x: (x[0]
[2], x[1])).cache()
img_dataset_size_rdd = img_time_configs_transformed.map(lambda x:
(x[0][3], x[1])).cache()

# Collect the results for each parameter
img_batch_size_results = img_batch_size_rdd.collect()
img_batch_number_results = img_batch_number_rdd.collect()
img_repetitions_results = img_repetitions_rdd.collect()
img_dataset_size_results = img_dataset_size_rdd.collect()

#tfrec data

### TASK 2c ###
tfrec_batch_size_rdd = tfrec_time_configs_transformed.map(lambda x:
```

```
(x[0][0], x[1])).cache()
tfrec_batch_number_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][1], x[1])).cache()
tfrec_repetitions_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][2], x[1])).cache()
tfrec_dataset_size_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][3], x[1])).cache()

# Collect the results for each parameter
tfrec_batch_size_results = tfrec_batch_size_rdd.collect()
tfrec_batch_number_results = tfrec_batch_number_rdd.collect()
tfrec_repetitions_results = tfrec_repetitions_rdd.collect()
tfrec_dataset_size_results = tfrec_dataset_size_rdd.collect()



#TASK 2a.v

#function for calculating average speed for each param value
def calc_average_speed(rdd):
  # Map each value to a tuple (param, (images_per_sec, count))
  mapped_rdd = rdd.map(lambda x: (x[0], (x[1], 1)))

  # Reduce by key to calculate sum of images_per_sec and count for
each param
  reduced_rdd = mapped_rdd.reduceByKey(lambda x, y: (x[0] + y[0], x[1]
+ y[1]))

  # Calculate the average images_per_sec for each param
  average_rdd = reduced_rdd.map(lambda x: (x[0], x[1][0] / x[1][1]))

  return average_rdd

#calculating the average speeds

#image dataset

img_batch_size_average = calc_average_speed(img_batch_size_rdd)
img_batch_number_average = calc_average_speed(img_batch_number_rdd)
img_repetitions_average = calc_average_speed(img_repetitions_rdd)
img_dataset_size_average=calc_average_speed(img_dataset_size_rdd)

# Collect the results for each parameter
img_batch_size_average_results = img_batch_size_average.collect()
img_batch_number_average_results = img_batch_number_average.collect()
img_repetitions_average_results = img_repetitions_average.collect()
img_dataset_size_average_results = img_dataset_size_average.collect()

#tfrec data

tfrec_batch_size_average = calc_average_speed(tfrec_batch_size_rdd)
```

```
tfrec_batch_number_average =
calc_average_speed(tfrec_batch_number_rdd)
tfrec_repetitions_average = calc_average_speed(tfrec_repetitions_rdd)
tfrec_dataset_size_average =
calc_average_speed(tfrec_dataset_size_rdd)

# Collect the results for each parameter
tfrec_batch_size_average_results = tfrec_batch_size_average.collect()
tfrec_batch_number_average_results =
tfrec_batch_number_average.collect()
tfrec_repetitions_average_results =
tfrec_repetitions_average.collect()
tfrec_dataset_size_average_results =
tfrec_dataset_size_average.collect()

#releasing the memory for cached rdds

img_time_configs_flattened.unpersist()
tfrec_time_configs_flattened.unpersist()
img_time_configs_transformed.unpersist()
tfrec_time_configs_transformed.unpersist()
img_batch_size_rdd.unpersist()
img_batch_number_rdd.unpersist()
img_repetitions_rdd.unpersist()
img_dataset_size_rdd.unpersist()
tfrec_batch_size_rdd.unpersist()
tfrec_batch_number_rdd.unpersist()
tfrec_repetitions_rdd.unpersist()
tfrec_dataset_size_rdd.unpersist()
```

```
PythonRDD[12] at RDD at PythonRDD.scala:53
```

```
#TASK 2a.vi writing the results to a pickle file in my bucket
import subprocess

results_list=[img_batch_size_results,img_batch_number_results,img_repe
titions_results,img_dataset_size_results,

tfrec_batch_size_results,tfrec_batch_number_results,tfrec_repetitions_
results,tfrec_dataset_size_results,

img_batch_size_average_results,img_batch_number_average_results,img_re
petitions_average_results,

img_dataset_size_average_results,tfrec_batch_size_average_results,tfre
c_batch_number_average_results,

tfrec_repetitions_average_results,tfrec_dataset_size_average_results]

filename=datetime.datetime.now().strftime("%y%m%d-%H%M")
+'_average_reading_speeds.pkl'
```

```
with open(filename,mode='wb') as f:
  pickle.dump(results_list,f)
print("Saving {} to {}".format(filename,BUCKET))
proc = subprocess.run(["gsutil","cp", filename,
BUCKET],stderr=subprocess.PIPE)
print("gstuil returned: " + str(proc.returncode))
print(str(proc.stderr))
```

```
Saving 240504-1810_average_reading_speeds.pkl to gs://bd-coursework-
421223-storage
gstuil returned: 0
b'Copying file://240504-1810_average_reading_speeds.pkl [Content-
Type=application/octet-stream]...\n/ [0 files][    0.0 B/ 10.4 KiB]
\r/ [1 files][ 10.4 KiB/ 10.4 KiB]
\r\nOperation completed over 1 objects/10.4 KiB.
\n'
```

```
#TASK 2a.vii

%%writefile spark_job.py

import subprocess
subprocess.call(['pip', 'install', 'pyspark'])
subprocess.call(['pip', 'install', 'tensorflow'])
import datetime
import tensorflow as tf
import time
import numpy as np
import pickle
import pyspark
sc = pyspark.SparkContext.getOrCreate()

PROJECT = 'bd-coursework-421223'
BUCKET = 'gs://{}-storage'.format(PROJECT)

#TASK 2a.i

#Helper functions
def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2
```

```
def resize_and_crop_image(image, label):
    w = tf.shape(image)[0]
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,lambda: tf.image.resize(image,
[w*tw/w, h*tw/w]),lambda: tf.image.resize(image, [w*th/h, h*th/h]))
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def read_tfrecord(example):
    features = {
        "image": tf.io.FixedLenFeature([], tf.string),  # tf.string =
bytestring (not text string)
        "class": tf.io.FixedLenFeature([], tf.int64) #,   # shape []
means scalar
    }
    # decode the TFRecord
    example = tf.io.parse_single_example(example, features)
    image = tf.image.decode_jpeg(example['image'], channels=3)
    image = tf.reshape(image, [*TARGET_SIZE, 3])
    class_num = example['class']
    return image, class_num

def load_dataset(filenames):
    # read from TFRecords. For optimal performance, read from multiple
    # TFRecord files at once and set the option
experimental_deterministic = False
    # to allow order-altering optimizations.
    option_no_order = tf.data.Options()
    option_no_order.experimental_deterministic = False

    dataset = tf.data.TFRecordDataset(filenames)
    dataset = dataset.with_options(option_no_order)
    dataset = dataset.map(read_tfrecord)
    return dataset

# Function to generate parameter combinations
def generate_param_combinations(batch_sizes, batch_numbers,
repetitions):
    param_combinations = []
    for bs in batch_sizes:
        for bn in batch_numbers:
            for rep in repetitions:
                param_combinations.append((bs, bn, rep))
    return param_combinations
```

```
batch_sizes = [10,20,30,40]
batch_numbers = [3,6,9,12]
repetitions = [1,2,3]

param_combinations = generate_param_combinations(batch_sizes,
batch_numbers, repetitions)
# Creating RDD from parameter combinations
param_combinations_rdd = sc.parallelize(param_combinations)
param_combinations_rdd.collect()

#function for Tensorflow Datasets
TARGET_SIZE = [192, 192]
def img_time_configs(params):
    batch_size, batch_number, repetitions=params

    GCS_PATTERN = 'gs://flowers-public/*/*.jpg'
    dsetFiles = tf.data.Dataset.list_files(GCS_PATTERN)
    dsetDecoded = dsetFiles.map(decode_jpeg_and_label)
    dsetResized = dsetDecoded.map(resize_and_crop_image)

    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        batched_dataset = dsetResized.batch(batch_size)
        timing_set = batched_dataset.take(batch_number)
        results=[]
        params=[]
        for i in range(repetitions):
                    t0 = time.time()
                    for image, label in timing_set:
                        print("Image batch shape {},
{})".format(image.numpy().shape,
                              [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
images
                    result = ( batch_size * batch_number) / td
                    dataset_size=batch_size * batch_number
                    param = [ batch_size, batch_number, repetitions,
dataset_size ]
                    results.append(result)
                    params.append(param)
    return results, params

#function for tfrecord files
def tfrec_time_configs(params):
    batch_size, batch_number, repetitions=params

    tfrecord='gs://flowers-public/tfrecords-jpeg-192x192-2/'
```

```
    filenames = tf.io.gfile.glob(tfrecord + "*.tfrec")
    datasetTfrec = load_dataset(filenames)

    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        batched_dataset = datasetTfrec.batch(batch_size)
        timing_set = batched_dataset.take(batch_number)
        results=[]
        params=[]
        for i in range(repetitions):
                    t0 = time.time()
                    for image, label in timing_set:
                        print("Image batch shape {},
{}".format(image.numpy().shape,
                                [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
images
                    result = ( batch_size * batch_number) / td
                    dataset_size=batch_size * batch_number
                    param = [ batch_size, batch_number, repetitions,
dataset_size]
                    results.append(result)
                    params.append(param)
    return results, params


#TASK 2a.ii

#image dataset

img_time_configs_result=param_combinations_rdd.map(img_time_configs)
# Apply flatMap to flatten the nested lists
img_time_configs_flattened = img_time_configs_result.flatMap(lambda x:
[((params, scores)) for params, scores in zip(x[0], x[1])])


#tfrec data
tfrec_time_configs_result=param_combinations_rdd.map(tfrec_time_config
s)
# Apply flatMap to flatten the nested lists
tfrec_time_configs_flattened =
tfrec_time_configs_result.flatMap(lambda x: [((params, scores)) for
params, scores in zip(x[0], x[1])])


#TASK 2a.iii

#image dataset
```

```
img_time_configs_transformed = img_time_configs_flattened.map(lambda
x: (x[1], x[0]))
#saving the results in an array
img_flattened_list = [(params[0], params[1], params[2], params[3],
score) for params, score in img_time_configs_transformed.collect()]
img_time_configs_array = np.array(img_flattened_list)

#tfrec data
tfrec_time_configs_transformed =
tfrec_time_configs_flattened.map(lambda x: (x[1], x[0]))
#saving the results in an array
tfrec_flattened_list = [(params[0], params[1], params[2], params[3],
score) for params, score in tfrec_time_configs_transformed.collect()]
tfrec_time_configs_array = np.array(tfrec_flattened_list)


#TASK 2a.iv

#image dataset
# Extract parameter values and result
img_batch_size_rdd = img_time_configs_transformed.map(lambda x: (x[0]
[0], x[1]))
img_batch_number_rdd = img_time_configs_transformed.map(lambda x:
(x[0][1], x[1]))
img_repetitions_rdd = img_time_configs_transformed.map(lambda x: (x[0]
[2], x[1]))
img_dataset_size_rdd = img_time_configs_transformed.map(lambda x:
(x[0][3], x[1]))

# Collect the results for each parameter
img_batch_size_results = img_batch_size_rdd.collect()
img_batch_number_results = img_batch_number_rdd.collect()
img_repetitions_results = img_repetitions_rdd.collect()
img_dataset_size_results = img_dataset_size_rdd.collect()

#tfrec data
# Extract parameter values and result
tfrec_batch_size_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][0], x[1]))
tfrec_batch_number_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][1], x[1]))
tfrec_repetitions_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][2], x[1]))
tfrec_dataset_size_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][3], x[1]))

# Collect the results for each parameter
tfrec_batch_size_results = tfrec_batch_size_rdd.collect()
tfrec_batch_number_results = tfrec_batch_number_rdd.collect()
tfrec_repetitions_results = tfrec_repetitions_rdd.collect()
```

```
tfrec_dataset_size_results = tfrec_dataset_size_rdd.collect()

#TASK 2a.v

def calc_average_speed(rdd):
  # Map each value to a tuple (param, (images_per_sec, count))
  mapped_rdd = rdd.map(lambda x: (x[0], (x[1], 1)))

  # Reduce by key to calculate sum of images_per_sec and count for
each param
  reduced_rdd = mapped_rdd.reduceByKey(lambda x, y: (x[0] + y[0], x[1]
+ y[1]))

  # Calculate the average images_per_sec for each param
  average_rdd = reduced_rdd.map(lambda x: (x[0], x[1][0] / x[1][1]))

  return average_rdd

#image dataset

img_batch_size_average = calc_average_speed(img_batch_size_rdd)
img_batch_number_average = calc_average_speed(img_batch_number_rdd)
img_repetitions_average = calc_average_speed(img_repetitions_rdd)
img_dataset_size_average=calc_average_speed(img_dataset_size_rdd)

# Collect the results for each parameter
img_batch_size_average_results = img_batch_size_average.collect()
img_batch_number_average_results = img_batch_number_average.collect()
img_repetitions_average_results = img_repetitions_average.collect()
img_dataset_size_average_results = img_dataset_size_average.collect()

#tfrec data

tfrec_batch_size_average = calc_average_speed(tfrec_batch_size_rdd)
tfrec_batch_number_average =
calc_average_speed(tfrec_batch_number_rdd)
tfrec_repetitions_average = calc_average_speed(tfrec_repetitions_rdd)
tfrec_dataset_size_average =
calc_average_speed(tfrec_dataset_size_rdd)

# Collect the results for each parameter
tfrec_batch_size_average_results = tfrec_batch_size_average.collect()
tfrec_batch_number_average_results =
tfrec_batch_number_average.collect()
tfrec_repetitions_average_results =
tfrec_repetitions_average.collect()
tfrec_dataset_size_average_results =
tfrec_dataset_size_average.collect()

#TASK 2a.vi
```

```
results_list=[img_batch_size_results,img_batch_number_results,img_repe
titions_results,img_dataset_size_results,

tfrec_batch_size_results,tfrec_batch_number_results,tfrec_repetitions_
results,tfrec_dataset_size_results,

img_batch_size_average_results,img_batch_number_average_results,img_re
petitions_average_results,

img_dataset_size_average_results,tfrec_batch_size_average_results,tfre
c_batch_number_average_results,

tfrec_repetitions_average_results,tfrec_dataset_size_average_results]

filename=datetime.datetime.now().strftime("%y%m%d-%H%M")
+'_average_reading_speeds.pkl'
with open(filename,mode='wb') as f:
  pickle.dump(results_list,f)
print("Saving {} to {}".format(filename,BUCKET))
proc = subprocess.run(["gsutil","cp", filename,
BUCKET],stderr=subprocess.PIPE)
print("gstuil returned: " + str(proc.returncode))
print(str(proc.stderr))

Overwriting spark_job.py
```

2b) Testing the code and collecting results (4%)

i) First, test locally with `%run`.

It is useful to create a **new filename argument**, so that old results don't get overwritten.

You can for instance use `datetime.datetime.now().strftime("%y%m%d-%H%M")` to get a string with the current date and time and use that in the file name.

```
### CODING TASK
%run spark_job.py

Saving 240502-2155_average_reading_speeds.pkl to gs://bd-coursework-
421223-storage
gstuil returned: 0
b'Copying file://240502-2155_average_reading_speeds.pkl [Content-
Type=application/octet-stream]...\n/ [0 files][    0.0 B/  1.6 KiB]
\r/ [1 files][  1.6 KiB/  1.6 KiB]
\r-\r\nOperation completed over 1 objects/1.6 KiB.
\n'
```

ii) Cloud

If you have a cluster running, you can run the speed test job in the cloud.

While you run this job, switch to the Dataproc web page and take **screenshots of the CPU and network load** over time. They are displayed with some delay, so you may need to wait a little. These images will be useful in the next task. Again, don't use the SCREENSHOT function that Google provides, but just take a picture of the graphs you see for the VMs.

```
### CODING TASK ###

#Running the script in maximal cluster created in Task 1c.ii

!gcloud dataproc jobs submit pyspark --cluster $MAXCLUSTER \
    spark_job.py
```

```
Job [52fd0a6a1f7a4ddcaf61e14dd5d4ff7b] submitted.
Waiting for job output...
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
```

```
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
```

```
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
```

```
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-05-03 19:52:56.163230: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-05-03 19:52:56.628975: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-03 19:52:56.629080: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-05-03 19:52:58.698269: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-03 19:52:58.698468: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-03 19:52:58.698503: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
24/05/03 19:53:03 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/05/03 19:53:03 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/05/03 19:53:03 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/05/03 19:53:04 INFO org.spark_project.jetty.util.log: Logging
```

```
initialized @14779ms to org.spark_project.jetty.util.log.Slf4jLog
24/05/03 19:53:04 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/05/03 19:53:04 INFO org.spark_project.jetty.server.Server: Started
@15075ms
24/05/03 19:53:04 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:40899}
24/05/03 19:53:06 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at
bd-coursework-421223-maxcluster-m/10.138.15.201:8032
24/05/03 19:53:07 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
maxcluster-m/10.138.15.201:10200
24/05/03 19:53:07 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/05/03 19:53:07 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/05/03 19:53:07 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/05/03 19:53:07 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/05/03 19:53:11 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714757697174_0005
Saving 240503-2311_average_reading_speeds.pkl to gs://bd-coursework-
421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://240503-2311_average_reading_speeds.pkl [Content-
Type=application/octet-stream]...\n/ [0 files][    0.0 B/ 12.8 KiB]
\r/ [1 files][ 12.8 KiB/ 12.8 KiB]
\r\nOperation completed over 1 objects/12.8 KiB.
\n'
24/05/03 23:11:16 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@45915c69{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [52fd0a6a1f7a4ddcaf61e14dd5d4ff7b] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/a73caf10-4ec6-480a-8756-
d2e472423758/jobs/52fd0a6a1f7a4ddcaf61e14dd5d4ff7b/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
```

```
ngdqyb5y/google-cloud-dataproc-metainfo/a73caf10-4ec6-480a-8756-
d2e472423758/jobs/52fd0a6a1f7a4ddcaf61e14dd5d4ff7b/driveroutput
jobUuid: 1f56305e-75d7-3718-adaa-f5a9cf4e3208
placement:
  clusterName: bd-coursework-421223-maxcluster
  clusterUuid: a73caf10-4ec6-480a-8756-d2e472423758
pysparkJob:
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/a73caf10-4ec6-480a-8756-
d2e472423758/jobs/52fd0a6a1f7a4ddcaf61e14dd5d4ff7b/staging/
spark_job.py
reference:
  jobId: 52fd0a6a1f7a4ddcaf61e14dd5d4ff7b
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-05-03T23:11:18.167670Z'
statusHistory:
- state: PENDING
  stateStartTime: '2024-05-03T19:52:47.481729Z'
- state: SETUP_DONE
  stateStartTime: '2024-05-03T19:52:47.521367Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-05-03T19:52:47.779439Z'
yarnApplications:
- name: spark_job.py
  progress: 1.0
  state: FINISHED
  trackingUrl:
http://bd-coursework-421223-maxcluster-m:8088/proxy/application_171475
7697174_0005/
```

2c) Improve efficiency (6%)

If you implemented a straightfoward version of 2a), you will **probably have an inefficiency** in your code.

Because we are reading multiple times from an RDD to read the values for the different parameters and their averages, caching existing results is important. Explain **where in the process caching can help**, and **add a call to `RDD.cache()`** to your code, if you haven't yet. Measure the the effect of using caching or not using it.

Make the **suitable change** in the code you have written above and mark them up in comments as `### TASK 2c ###`.

Explain in your report what the **reasons for this change** are and **demonstrate and interpret its effect**

```
#writing a new script using RDD.cache()

%%writefile spark_job2c.py

import subprocess
subprocess.call(['pip', 'install', 'pyspark'])
subprocess.call(['pip', 'install', 'tensorflow'])
import datetime
import tensorflow as tf
import time
import pickle
import numpy as np
import pyspark
sc = pyspark.SparkContext.getOrCreate()

PROJECT = 'bd-coursework-421223'
BUCKET = 'gs://{}-storage'.format(PROJECT)

#TASK 2a.i

#Helper functions
def decode_jpeg_and_label(filepath):
    # extracts the image data and creates a class label, based on the
filepath
    bits = tf.io.read_file(filepath)
    image = tf.image.decode_jpeg(bits)
    # parse flower name from containing directory
    label = tf.strings.split(tf.expand_dims(filepath, axis=-1),
sep='/')
    label2 = label.values[-2]
    return image, label2

def resize_and_crop_image(image, label):
    w = tf.shape(image)[0]
    h = tf.shape(image)[1]
    tw = TARGET_SIZE[1]
    th = TARGET_SIZE[0]
    resize_crit = (w * th) / (h * tw)
    image = tf.cond(resize_crit < 1,lambda: tf.image.resize(image,
[w*tw/w, h*tw/w]),lambda: tf.image.resize(image, [w*th/h, h*th/h]))
    nw = tf.shape(image)[0]
    nh = tf.shape(image)[1]
    image = tf.image.crop_to_bounding_box(image, (nw - tw) // 2, (nh -
th) // 2, tw, th)
    return image, label

def read_tfrecord(example):
    features = {
        "image": tf.io.FixedLenFeature([], tf.string),  # tf.string =
bytestring (not text string)
```

```
        "class": tf.io.FixedLenFeature([], tf.int64) #,   # shape [] means scalar
    }
    # decode the TFRecord
    example = tf.io.parse_single_example(example, features)
    image = tf.image.decode_jpeg(example['image'], channels=3)
    image = tf.reshape(image, [*TARGET_SIZE, 3])
    class_num = example['class']
    return image, class_num

def load_dataset(filenames):
    # read from TFRecords. For optimal performance, read from multiple
    # TFRecord files at once and set the option experimental_deterministic = False
    # to allow order-altering optimizations.
    option_no_order = tf.data.Options()
    option_no_order.experimental_deterministic = False

    dataset = tf.data.TFRecordDataset(filenames)
    dataset = dataset.with_options(option_no_order)
    dataset = dataset.map(read_tfrecord)
    return dataset

# Function to generate parameter combinations
def generate_param_combinations(batch_sizes, batch_numbers, repetitions):
    param_combinations = []
    for bs in batch_sizes:
        for bn in batch_numbers:
            for rep in repetitions:
                param_combinations.append((bs, bn, rep))
    return param_combinations

batch_sizes = [10,20,30,40]
batch_numbers = [3,6,9,12]
repetitions = [1,2,3]

param_combinations = generate_param_combinations(batch_sizes, batch_numbers, repetitions)
# Creating RDD from parameter combinations
param_combinations_rdd = sc.parallelize(param_combinations)
param_combinations_rdd.collect()

#function for Tensorflow Datasets
TARGET_SIZE = [192, 192]
def img_time_configs(params):
    batch_size, batch_number, repetitions=params

    GCS_PATTERN = 'gs://flowers-public/*/*.jpg'
    dsetFiles = tf.data.Dataset.list_files(GCS_PATTERN)
```

```
    dsetDecoded = dsetFiles.map(decode_jpeg_and_label)
    dsetResized = dsetDecoded.map(resize_and_crop_image)

    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        batched_dataset = dsetResized.batch(batch_size)
        timing_set = batched_dataset.take(batch_number)
        results=[]
        params=[]
        for i in range(repetitions):
                    t0 = time.time()
                    for image, label in timing_set:
                        print("Image batch shape {},
{}".format(image.numpy().shape,
                              [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
images
                    result = ( batch_size * batch_number) / td
                    dataset_size=batch_size * batch_number
                    param = [ batch_size, batch_number, repetitions,
dataset_size ]
                    results.append(result)
                    params.append(param)
    return results, params

#function for tfrecord files
def tfrec_time_configs(params):
    batch_size, batch_number, repetitions=params

    tfrecord='gs://flowers-public/tfrecords-jpeg-192x192-2/'
    filenames = tf.io.gfile.glob(tfrecord + "*.tfrec")
    datasetTfrec = load_dataset(filenames)

    with open("/dev/null",mode='w') as null_file: # for printing the
output without showing it
        tt = time.time() # for overall time taking
        batched_dataset = datasetTfrec.batch(batch_size)
        timing_set = batched_dataset.take(batch_number)
        results=[]
        params=[]
        for i in range(repetitions):
                    t0 = time.time()
                    for image, label in timing_set:
                        print("Image batch shape {},
{}".format(image.numpy().shape,
                              [str(lbl) for lbl in label.numpy()]),
null_file)
                    td = time.time() - t0 # duration for reading
```

```
images
                    result = ( batch_size * batch_number) / td
                    dataset_size=batch_size * batch_number
                    param = [ batch_size, batch_number, repetitions,
dataset_size]
                    results.append(result)
                    params.append(param)
    return results, params


#TASK 2a.ii

#image dataset

img_time_configs_result=param_combinations_rdd.map(img_time_configs)
# Applying flatMap to flatten the nested lists
img_time_configs_flattened = img_time_configs_result.flatMap(lambda x:
[((params, scores)) for params, scores in zip(x[0], x[1])])


#tfrec data
tfrec_time_configs_result=param_combinations_rdd.map(tfrec_time_config
s)
# Applying flatMap to flatten the nested lists
tfrec_time_configs_flattened =
tfrec_time_configs_result.flatMap(lambda x: [((params, scores)) for
params, scores in zip(x[0], x[1])])


#TASK 2a.iii

#image dataset
img_time_configs_transformed = img_time_configs_flattened.map(lambda
x: (x[1], x[0])).cache()         ### TASK 2c ###

#saving the results in an array
img_flattened_list = [(params[0], params[1], params[2], params[3],
score) for params, score in img_time_configs_transformed.collect()]
img_time_configs_array = np.array(img_flattened_list)

#tfrec data
tfrec_time_configs_transformed =
tfrec_time_configs_flattened.map(lambda x: (x[1], x[0])).cache()
### TASK 2c ###

#saving the results in an array
tfrec_flattened_list = [(params[0], params[1], params[2], params[3],
score) for params, score in tfrec_time_configs_transformed.collect()]
tfrec_time_configs_array = np.array(tfrec_flattened_list)
```

```
#TASK 2a.iv

#image dataset
# Extract parameter values and result
### TASK 2c ###
img_batch_size_rdd = img_time_configs_transformed.map(lambda x: (x[0]
[0], x[1])).cache()
img_batch_number_rdd = img_time_configs_transformed.map(lambda x:
(x[0][1], x[1])).cache()
img_repetitions_rdd = img_time_configs_transformed.map(lambda x: (x[0]
[2], x[1])).cache()
img_dataset_size_rdd = img_time_configs_transformed.map(lambda x:
(x[0][3], x[1])).cache()

# Collect the results for each parameter
img_batch_size_results = img_batch_size_rdd.collect()
img_batch_number_results = img_batch_number_rdd.collect()
img_repetitions_results = img_repetitions_rdd.collect()
img_dataset_size_results = img_dataset_size_rdd.collect()

#tfrec data
# Extract parameter values and result
### TASK 2c ###
tfrec_batch_size_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][0], x[1])).cache()
tfrec_batch_number_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][1], x[1])).cache()
tfrec_repetitions_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][2], x[1])).cache()
tfrec_dataset_size_rdd = tfrec_time_configs_transformed.map(lambda x:
(x[0][3], x[1])).cache()

# Collect the results for each parameter
tfrec_batch_size_results = tfrec_batch_size_rdd.collect()
tfrec_batch_number_results = tfrec_batch_number_rdd.collect()
tfrec_repetitions_results = tfrec_repetitions_rdd.collect()
tfrec_dataset_size_results = tfrec_dataset_size_rdd.collect()

#TASK 2a.v

def calc_average_speed(rdd):
  # Map each value to a tuple (param, (images_per_sec, count))
  mapped_rdd = rdd.map(lambda x: (x[0], (x[1], 1)))

  # Reduce by key to calculate sum of images_per_sec and count for
each param
  reduced_rdd = mapped_rdd.reduceByKey(lambda x, y: (x[0] + y[0], x[1]
+ y[1]))

  # Calculate the average images_per_sec for each param
```

```
  average_rdd = reduced_rdd.map(lambda x: (x[0], x[1][0] / x[1][1]))

  return average_rdd

#image dataset

img_batch_size_average = calc_average_speed(img_batch_size_rdd)
img_batch_number_average = calc_average_speed(img_batch_number_rdd)
img_repetitions_average = calc_average_speed(img_repetitions_rdd)
img_dataset_size_average=calc_average_speed(img_dataset_size_rdd)

# Collect the results for each parameter
img_batch_size_average_results = img_batch_size_average.collect()
img_batch_number_average_results = img_batch_number_average.collect()
img_repetitions_average_results = img_repetitions_average.collect()
img_dataset_size_average_results = img_dataset_size_average.collect()

#tfrec data

tfrec_batch_size_average = calc_average_speed(tfrec_batch_size_rdd)
tfrec_batch_number_average =
calc_average_speed(tfrec_batch_number_rdd)
tfrec_repetitions_average = calc_average_speed(tfrec_repetitions_rdd)
tfrec_dataset_size_average =
calc_average_speed(tfrec_dataset_size_rdd)

# Collect the results for each parameter
tfrec_batch_size_average_results = tfrec_batch_size_average.collect()
tfrec_batch_number_average_results =
tfrec_batch_number_average.collect()
tfrec_repetitions_average_results =
tfrec_repetitions_average.collect()
tfrec_dataset_size_average_results =
tfrec_dataset_size_average.collect()

#TASK 2a.vi
results_list=[img_batch_size_results,img_batch_number_results,img_repe
titions_results,img_dataset_size_results,

tfrec_batch_size_results,tfrec_batch_number_results,tfrec_repetitions_
results,tfrec_dataset_size_results,

img_batch_size_average_results,img_batch_number_average_results,img_re
petitions_average_results,

img_dataset_size_average_results,tfrec_batch_size_average_results,tfre
c_batch_number_average_results,

tfrec_repetitions_average_results,tfrec_dataset_size_average_results]
```

```
filename=datetime.datetime.now().strftime("%y%m%d-%H%M")
+'_average_reading_speeds.pkl'
with open(filename,mode='wb') as f:
  pickle.dump(results_list,f)
print("Saving {} to {}".format(filename,BUCKET))
proc = subprocess.run(["gsutil","cp", filename,
BUCKET],stderr=subprocess.PIPE)
print("gstuil returned: " + str(proc.returncode))
print(str(proc.stderr))

#releasing the memory for cached rdds
img_time_configs_transformed.unpersist()
tfrec_time_configs_transformed.unpersist()
img_batch_size_rdd.unpersist()
img_batch_number_rdd.unpersist()
img_repetitions_rdd.unpersist()
img_dataset_size_rdd.unpersist()
tfrec_batch_size_rdd.unpersist()
tfrec_batch_number_rdd.unpersist()
tfrec_repetitions_rdd.unpersist()
tfrec_dataset_size_rdd.unpersist()
```

```
Writing spark_job2c.py
```

```
### CODING TASK ###

#Running the script with improved efficiency in maximal cluster
created in Task 1c.ii

!gcloud dataproc jobs submit pyspark --cluster $MAXCLUSTER \
    spark_job2c.py
```

```
Job [e470fb66504142d5ba6181cf2bdb968c] submitted.
Waiting for job output...
Requirement already satisfied: pyspark in /usr/lib/spark/python
(2.4.8)
Requirement already satisfied: py4j==0.10.7 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyspark)
(0.10.7)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
Requirement already satisfied: tensorflow in
/opt/conda/miniconda3/lib/python3.7/site-packages (2.11.0)
Requirement already satisfied: absl-py>=1.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.1.0)
Requirement already satisfied: astunparse>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
```

```
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.62.2)
Requirement already satisfied: h5py>=2.9.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.8.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: libclang>=13.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(18.1.1)
Requirement already satisfied: numpy>=1.20 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.21.6)
Requirement already satisfied: opt-einsum>=2.3.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.3.0)
Requirement already satisfied: packaging in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(24.0)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(3.19.6)
Requirement already satisfied: setuptools in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(41.4.0)
Requirement already satisfied: six>=1.12.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.12.0)
Requirement already satisfied: tensorboard<2.12,>=2.11 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.2)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(2.3.0)
```

```
Requirement already satisfied: typing-extensions>=3.6.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(4.7.1)
Requirement already satisfied: wrapt>=1.11.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(1.16.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from tensorflow)
(0.34.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
astunparse>=1.6.0->tensorflow) (0.33.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.29.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.22.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: werkzeug>=1.0.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-
auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (2.0.0)
Requirement already satisfied: importlib-metadata>=4.4 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.7.0)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in
```

```
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
in /opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from
werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.5)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from importlib-
metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow)
(3.11.0)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from pyasn1-
modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11-
>tensorflow) (0.5.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/miniconda3/lib/python3.7/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-
>tensorboard<2.12,>=2.11->tensorflow) (3.2.2)
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv
2024-05-03 23:24:35.795819: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2024-05-03 23:24:36.678811: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libcudart.so.11.0'; dlerror:
libcudart.so.11.0: cannot open shared object file: No such file or
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-03 23:24:36.678935: I
tensorflow/compiler/xla/stream_executor/cuda/cudart_stub.cc:29] Ignore
above cudart dlerror if you do not have a GPU set up on your machine.
2024-05-03 23:24:39.148283: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror:
libnvinfer.so.7: cannot open shared object file: No such file or
```

```
directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-03 23:24:39.149712: W
tensorflow/compiler/xla/stream_executor/platform/default/dso_loader.cc
:64] Could not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH: :/usr/lib/hadoop/lib/native
2024-05-03 23:24:39.149765: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
24/05/03 23:24:43 INFO org.apache.spark.SparkEnv: Registering
MapOutputTracker
24/05/03 23:24:43 INFO org.apache.spark.SparkEnv: Registering
BlockManagerMaster
24/05/03 23:24:43 INFO org.apache.spark.SparkEnv: Registering
OutputCommitCoordinator
24/05/03 23:24:43 INFO org.spark_project.jetty.util.log: Logging
initialized @14524ms to org.spark_project.jetty.util.log.Slf4jLog
24/05/03 23:24:44 INFO org.spark_project.jetty.server.Server: jetty-
9.4.z-SNAPSHOT; built: unknown; git: unknown; jvm 1.8.0_382-b05
24/05/03 23:24:44 INFO org.spark_project.jetty.server.Server: Started
@14791ms
24/05/03 23:24:44 INFO
org.spark_project.jetty.server.AbstractConnector: Started
ServerConnector@51f17779{HTTP/1.1, (http/1.1)}{0.0.0.0:45141}
24/05/03 23:24:46 INFO org.apache.hadoop.yarn.client.RMProxy:
Connecting to ResourceManager at
bd-coursework-421223-maxcluster-m/10.138.15.201:8032
24/05/03 23:24:46 INFO org.apache.hadoop.yarn.client.AHSProxy:
Connecting to Application History server at bd-coursework-421223-
maxcluster-m/10.138.15.201:10200
24/05/03 23:24:46 INFO org.apache.hadoop.conf.Configuration: resource-
types.xml not found
24/05/03 23:24:46 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find
'resource-types.xml'.
24/05/03 23:24:46 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = memory-mb, units = Mi, type = COUNTABLE
24/05/03 23:24:46 INFO
org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource
type - name = vcores, units = , type = COUNTABLE
24/05/03 23:24:47 WARN org.apache.hadoop.hdfs.DataStreamer: Caught
exception
java.lang.InterruptedException
     at java.lang.Object.wait(Native Method)
     at java.lang.Thread.join(Thread.java:1257)
     at java.lang.Thread.join(Thread.java:1331)
```

```
	at
org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:9
80)
	at
org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:630)
	at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:807)
24/05/03 23:24:49 INFO
org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted
application application_1714757697174_0006
Saving 240503-2348_average_reading_speeds.pkl to gs://bd-coursework-
421223-storage
gstuil returned: 0
b'WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023.
Please use Python version 3.8 and up.\n\nIf you have a compatible
Python interpreter installed, you can use it by setting\nthe
CLOUDSDK_PYTHON environment variable to point to it.\n\nCopying
file://240503-2348_average_reading_speeds.pkl [Content-
Type=application/octet-stream]...\n/ [0 files][    0.0 B/ 12.8 KiB]
\r/ [1 files][ 12.8 KiB/ 12.8 KiB]
\r\nOperation completed over 1 objects/12.8 KiB.
\n'
24/05/03 23:48:58 INFO
org.spark_project.jetty.server.AbstractConnector: Stopped
Spark@51f17779{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
Job [e470fb66504142d5ba6181cf2bdb968c] finished successfully.
done: true
driverControlFilesUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/a73caf10-4ec6-480a-8756-
d2e472423758/jobs/e470fb66504142d5ba6181cf2bdb968c/
driverOutputResourceUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/a73caf10-4ec6-480a-8756-
d2e472423758/jobs/e470fb66504142d5ba6181cf2bdb968c/driveroutput
jobUuid: 5c4cb3f1-7fe8-3765-bbbf-d87fe1332353
placement:
  clusterName: bd-coursework-421223-maxcluster
  clusterUuid: a73caf10-4ec6-480a-8756-d2e472423758
pysparkJob:
  mainPythonFileUri: gs://dataproc-staging-us-west1-832943544474-
ngdqyb5y/google-cloud-dataproc-metainfo/a73caf10-4ec6-480a-8756-
d2e472423758/jobs/e470fb66504142d5ba6181cf2bdb968c/staging/
spark_job2c.py
reference:
  jobId: e470fb66504142d5ba6181cf2bdb968c
  projectId: bd-coursework-421223
status:
  state: DONE
  stateStartTime: '2024-05-03T23:49:04.400197Z'
statusHistory:
- state: PENDING
```

```
  stateStartTime: '2024-05-03T23:24:27.209978Z'
- state: SETUP_DONE
  stateStartTime: '2024-05-03T23:24:27.253787Z'
- details: Agent reported job success
  state: RUNNING
  stateStartTime: '2024-05-03T23:24:27.656304Z'
yarnApplications:
- name: spark_job2c.py
  progress: 1.0
  state: FINISHED
  trackingUrl:
http://bd-coursework-421223-maxcluster-m:8088/proxy/application_171475
7697174_0006/
```

2d) Retrieve, analyse and discuss the output (12%)

Run the tests over a wide range of different paramters and list the results in a table.

Perform a **linear regression** (e.g. using scikit-learn) over **the values for each parameter** and for the **two cases** (reading from image files/reading TFRecord files). List a **table** with the output and interpret the results in terms of the effects of overall.
Also, **plot** the output values, the averages per parameter value and the regression lines for each parameter and for the product of batch_size and batch_number

Discuss the **implications** of this result for **applications** like large-scale machine learning. Keep in mind that cloud data may be stored in distant physical locations. Use the numbers provided in the PDF latency-numbers document available on Moodle or here for your arguments.

How is the **observed** behaviour **similar or different** from what you'd expect from a **single machine**? Why would cloud providers tie throughput to capacity of disk resources?

By **parallelising** the speed test we are making **assumptions** about the limits of the bucket reading speeds. See here for more information. Discuss, **what we need to consider** in **speed tests** in parallel on the cloud, which bottlenecks we might be identifying, and how this relates to your results.

Discuss to what extent **linear modelling** reflects the **effects** we are observing. Discuss what could be expected from a theoretical perspective and what can be useful in practice.

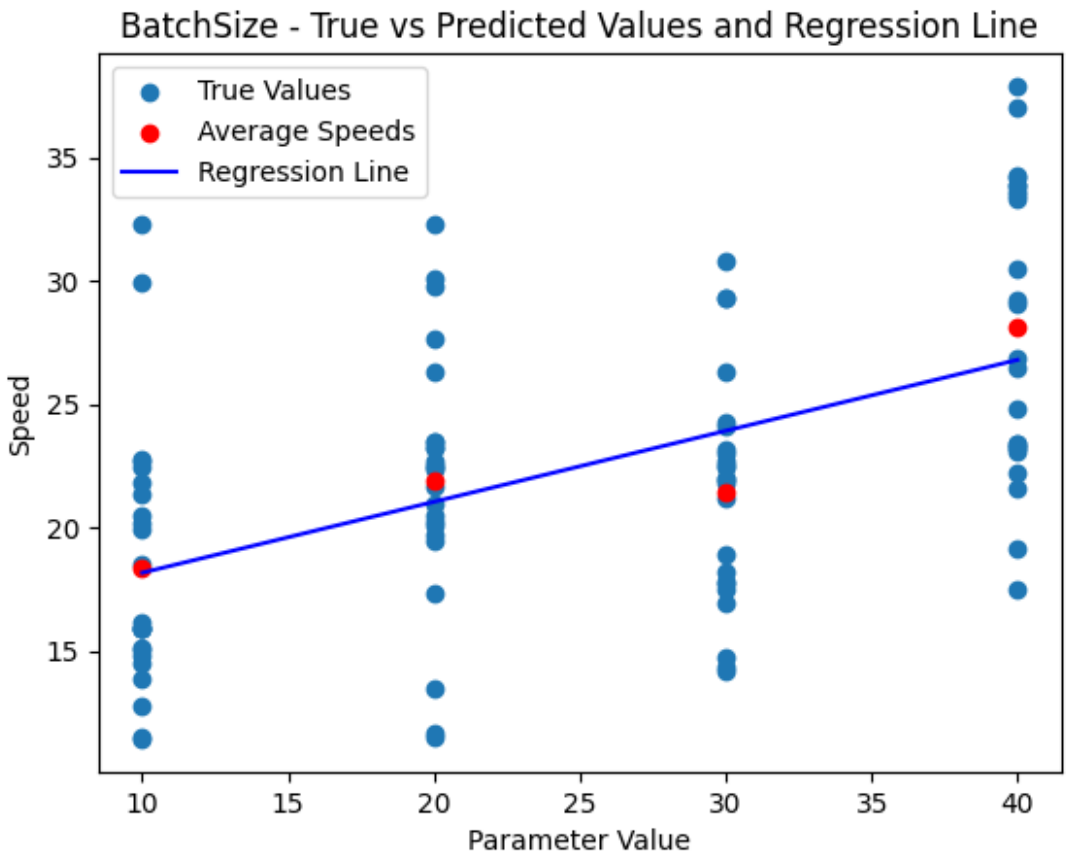
Write your **code below** and **include the output** in your submitted `ipynb` file. Provide the answer **text in your report**.

```
### CODING TASK ###

#Note: I have already run the test over a wide range of parameters
sufficient for regression modelling in TASK 2a,
#so I will be using those results here
```

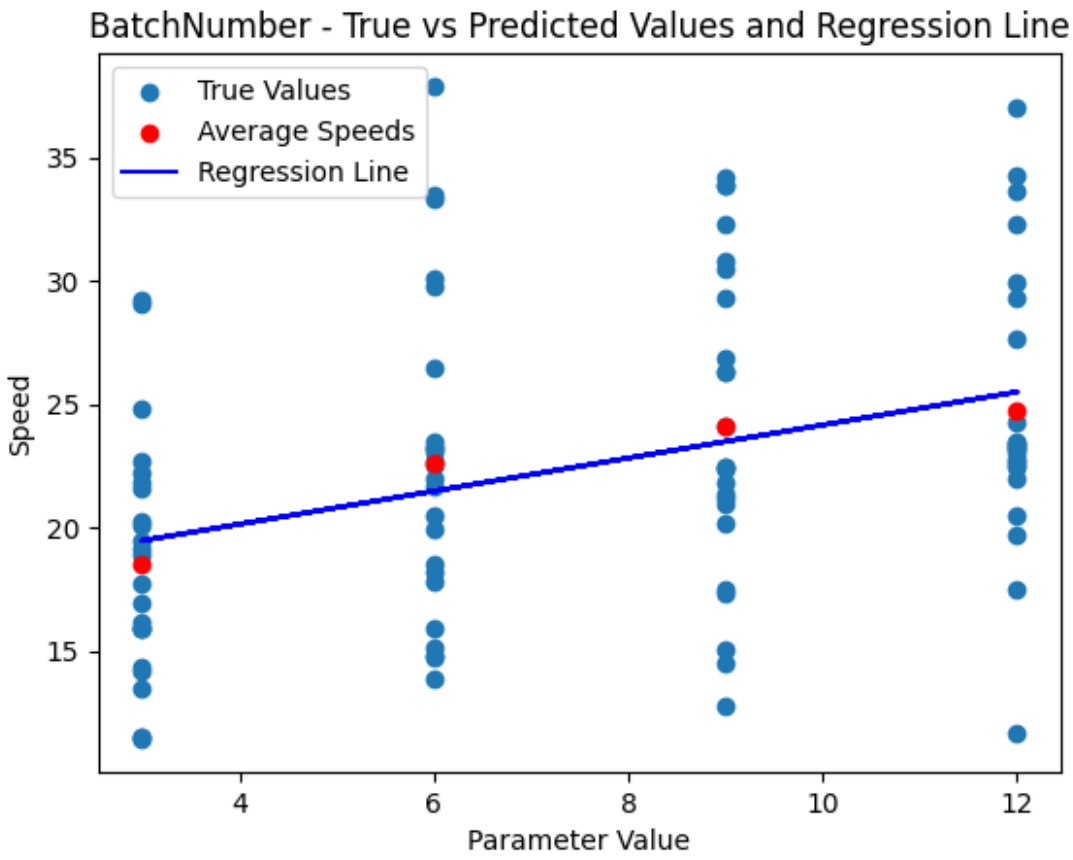
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import seaborn as sns
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score, median_absolute_error, mean_absolute_percentage_error
import pandas as pd
from sklearn.feature_selection import chi2
from sklearn.metrics import r2_score

#datasets to store the results
img_results_table=pd.DataFrame(columns=['R2 Score','Coefficient'])
tfrec_results_table=pd.DataFrame(columns=['R2 Score','Coefficient'])

#Performing regression and plotting the orignal data, average data,
and regression line

#tf dataset

img_batch_size_df=pd.DataFrame(img_batch_size_results,columns=['param_
value','speed'])
img_batch_number_df=pd.DataFrame(img_batch_number_results,columns=['pa
ram_value','speed'])
img_repetitions_df=pd.DataFrame(img_repetitions_results,columns=['para
m_value','speed'])
img_dataset_size_df=pd.DataFrame(img_dataset_size_results,columns=['pa
ram_value','speed'])

img_batch_size_average_df=pd.DataFrame(img_batch_size_average_results,
columns=['param_value','speed'])
img_batch_number_average_df=pd.DataFrame(img_batch_number_average_resu
lts,columns=['param_value','speed'])
img_repetitions_average_df=pd.DataFrame(img_repetitions_average_result
s,columns=['param_value','speed'])
img_dataset_size_average_df=pd.DataFrame(img_dataset_size_average_resu
lts,columns=['param_value','speed'])

for name, df, av in zip(['BatchSize', 'BatchNumber', 'Repetition',
'DatasetSize'],
                        [img_batch_size_df, img_batch_number_df,
img_repetitions_df, img_dataset_size_df],
                        [img_batch_size_average_df,
img_batch_number_average_df,
                         img_repetitions_average_df,
img_dataset_size_average_df]):
    print(name)

    X=df.param_value.values.reshape(-1, 1)
    y=df.speed.values.reshape(-1, 1)

    # Fit linear regression model to original data
```

```
    regr = LinearRegression()
    regr.fit(X,y)

    # Predict on test data
    y_pred = regr.predict(X)

    # Calculate R^2 score
    r2 = r2_score(y, y_pred)
    # calculating the coefficients (slopes) of the regression line
    coefficient = regr.coef_
    print(f"R2 Score: {r2}, Coef: {coefficient}")
    img_results_table.loc[len(img_results_table)]=[r2,coefficient[0]
[0]]

    # Plotting the speed for each param value
    plt.scatter(X, y, label='True Values')

    # Plotting average speeds
    plt.scatter(av.param_value, av.speed, color='red', label="Average
Speeds")

    # Plotting regression line
    plt.plot(X, y_pred, color='blue', label='Regression Line')

    plt.xlabel("Parameter Value")
    plt.ylabel("Speed")
    plt.title(f"{name} - True vs Predicted Values and Regression
Line")
    plt.legend()
    plt.show()

index_names = {0: 'BatchSize', 1:  'BatchNumber', 2: 'Repetition', 3:
'DatasetSize'}
img_results_table.rename(index=index_names, inplace=True)
```

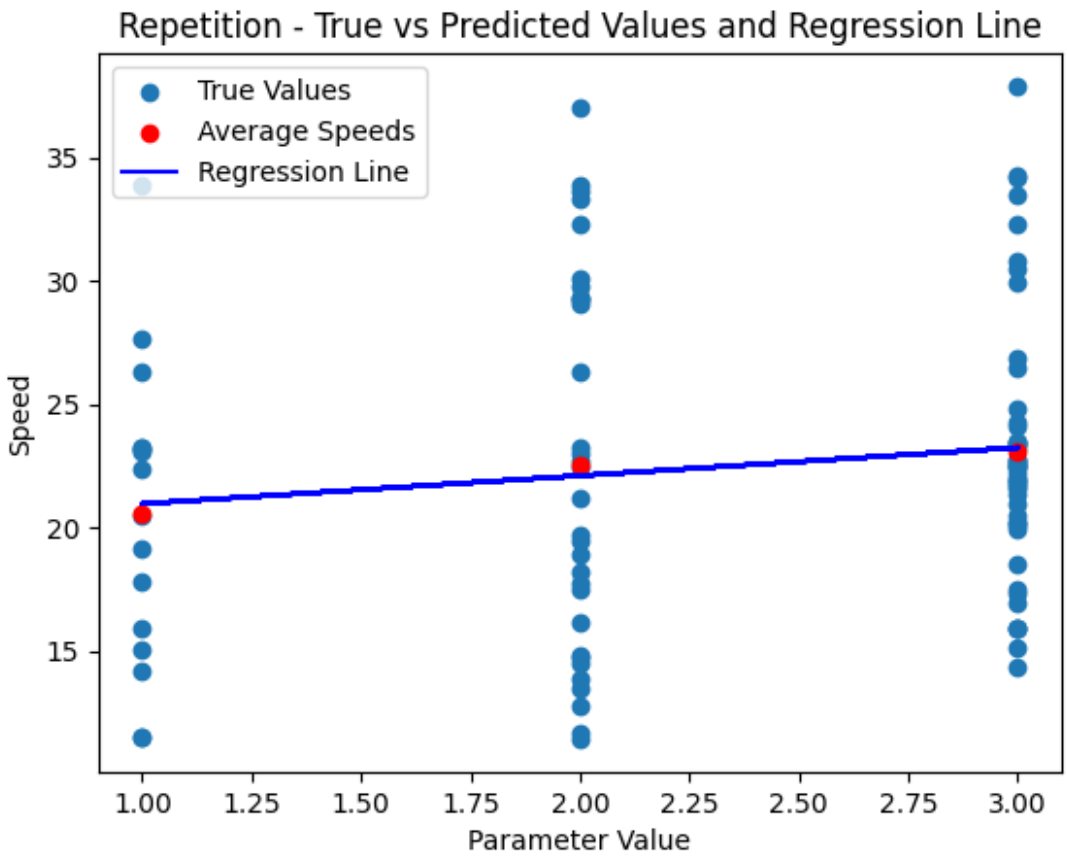
```
BatchSize
R2 Score: 0.2636610171662638, Coef: [[0.28763403]]
```




```
BatchNumber
R2 Score: 0.12867929339327278, Coef: [[0.66980809]]
```




```
Repetition
R2 Score: 0.018366955124968998, Coef: [[1.13874675]]
```




```
DatasetSize
R2 Score: 0.31665892644132276, Coef: [[0.02833613]]
```


```
img_results_table
```

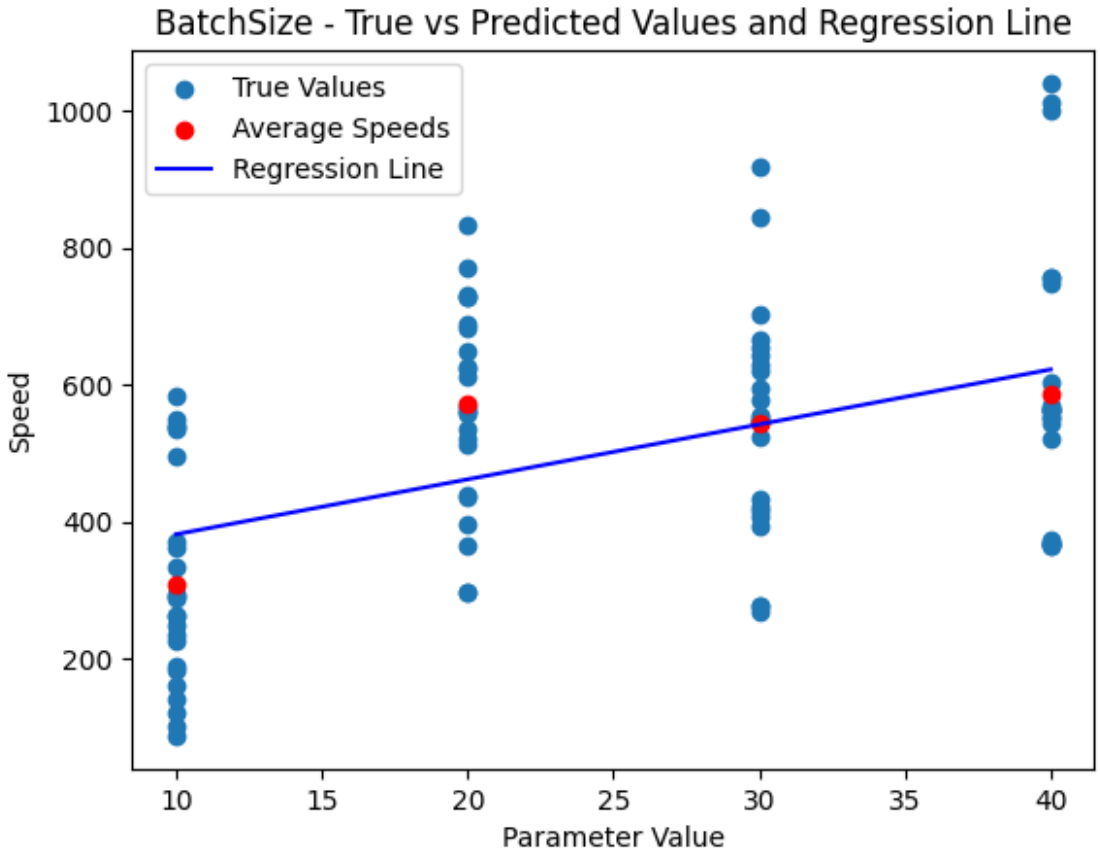

```
{"summary":"{\n  \"name\": \"img_results_table\",\n  \"rows\": 4,\n
\"fields\": [\n    {\n      \"column\": \"R2 Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.13468521187688925,\n        \"min\": 0.018366955124968998,\n
\"max\": 0.31665892644132276,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0.12867929339327278,\n
0.31665892644132276,\n          0.2636610171662638\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Coefficient\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.48322585612012575,\n        \"min\": 0.028336132326640052,\n
\"max\": 1.1387467480075142,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0.6698080877553467,\n
0.028336132326640052,\n          0.2876340324303218\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\
n}","type":"dataframe","variable_name":"img_results_table"}
```


```
#tfrecord files

tfrec_batch_size_df = pd.DataFrame(tfrec_batch_size_results,
```

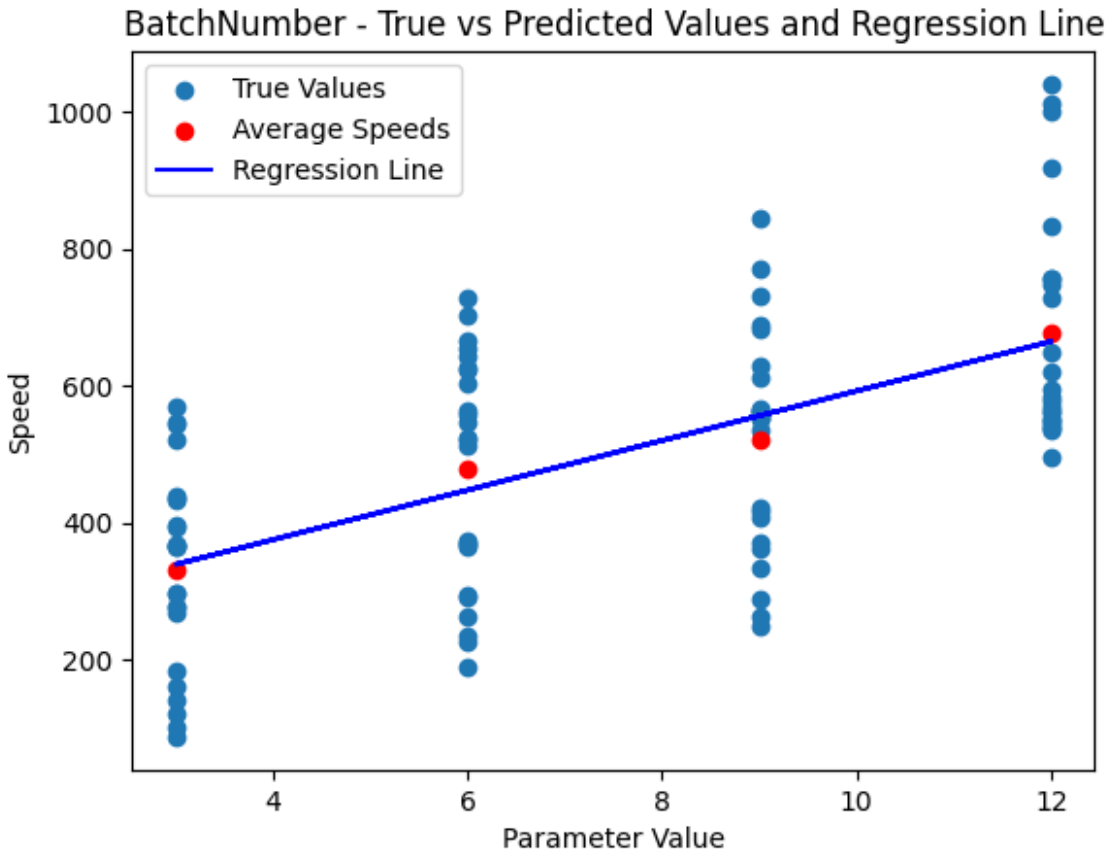
```
columns=['param_value', 'speed'])
tfrec_batch_number_df = pd.DataFrame(tfrec_batch_number_results,
columns=['param_value', 'speed'])
tfrec_repetitions_df = pd.DataFrame(tfrec_repetitions_results,
columns=['param_value', 'speed'])
tfrec_dataset_size_df = pd.DataFrame(tfrec_dataset_size_results,
columns=['param_value', 'speed'])

tfrec_batch_size_average_df =
pd.DataFrame(tfrec_batch_size_average_results, columns=['param_value',
'speed'])
tfrec_batch_number_average_df =
pd.DataFrame(tfrec_batch_number_average_results,
columns=['param_value', 'speed'])
tfrec_repetitions_average_df =
pd.DataFrame(tfrec_repetitions_average_results,
columns=['param_value', 'speed'])
tfrec_dataset_size_average_df =
pd.DataFrame(tfrec_dataset_size_average_results,
columns=['param_value', 'speed'])

for name, df, av in zip(['BatchSize', 'BatchNumber', 'Repetition',
'DatasetSize'],
                        [tfrec_batch_size_df, tfrec_batch_number_df,
tfrec_repetitions_df, tfrec_dataset_size_df],
                        [tfrec_batch_size_average_df,
tfrec_batch_number_average_df,
                         tfrec_repetitions_average_df,
tfrec_dataset_size_average_df]):
    print(name)

    X=df.param_value.values.reshape(-1, 1)
    y=df.speed.values.reshape(-1, 1)

    # Fit linear regression model to original data
    regr = LinearRegression()
    regr.fit(X,y)

    # Predict on test data
    y_pred = regr.predict(X)

    # Calculate R^2 score
    r2 = r2_score(y, y_pred)
    # calculating the coefficients (slopes) of the regression line
    coefficient = regr.coef_
    print(f"R2 Score: {r2}, Coef: {coefficient}")

tfrec_results_table.loc[len(tfrec_results_table)]=[r2,coefficient[0]
[0]]
```

```
    # Plotting the speed for each param value
    plt.scatter(X, y, label='True Values')

    # Plotting average speeds
    plt.scatter(av.param_value, av.speed, color='red', label="Average
Speeds")

    # Plotting regression line
    plt.plot(X, y_pred, color='blue', label='Regression Line')

    plt.xlabel("Parameter Value")
    plt.ylabel("Speed")
    plt.title(f"{name} - True vs Predicted Values and Regression
Line")
    plt.legend()
    plt.show()

tfrec_results_table.rename(index=index_names, inplace=True)
```

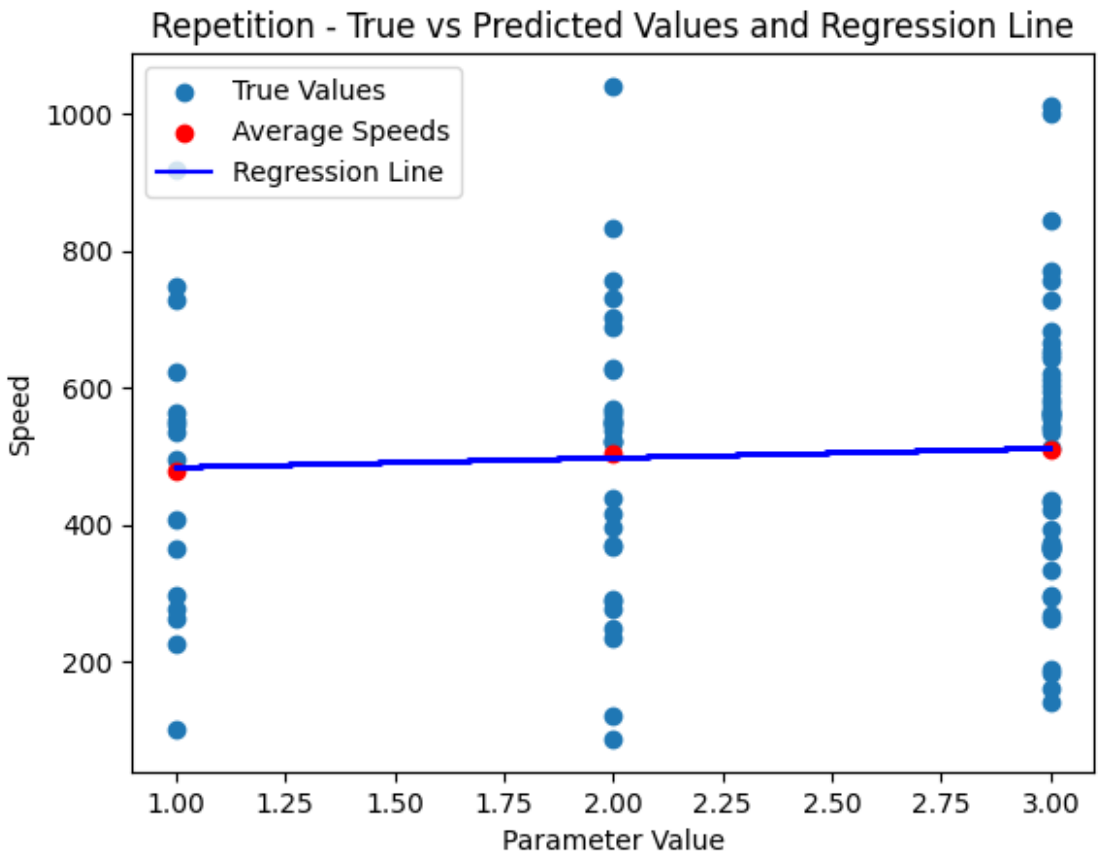
```
BatchSize
R2 Score: 0.19909858124736113, Coef: [[8.02042106]]
```



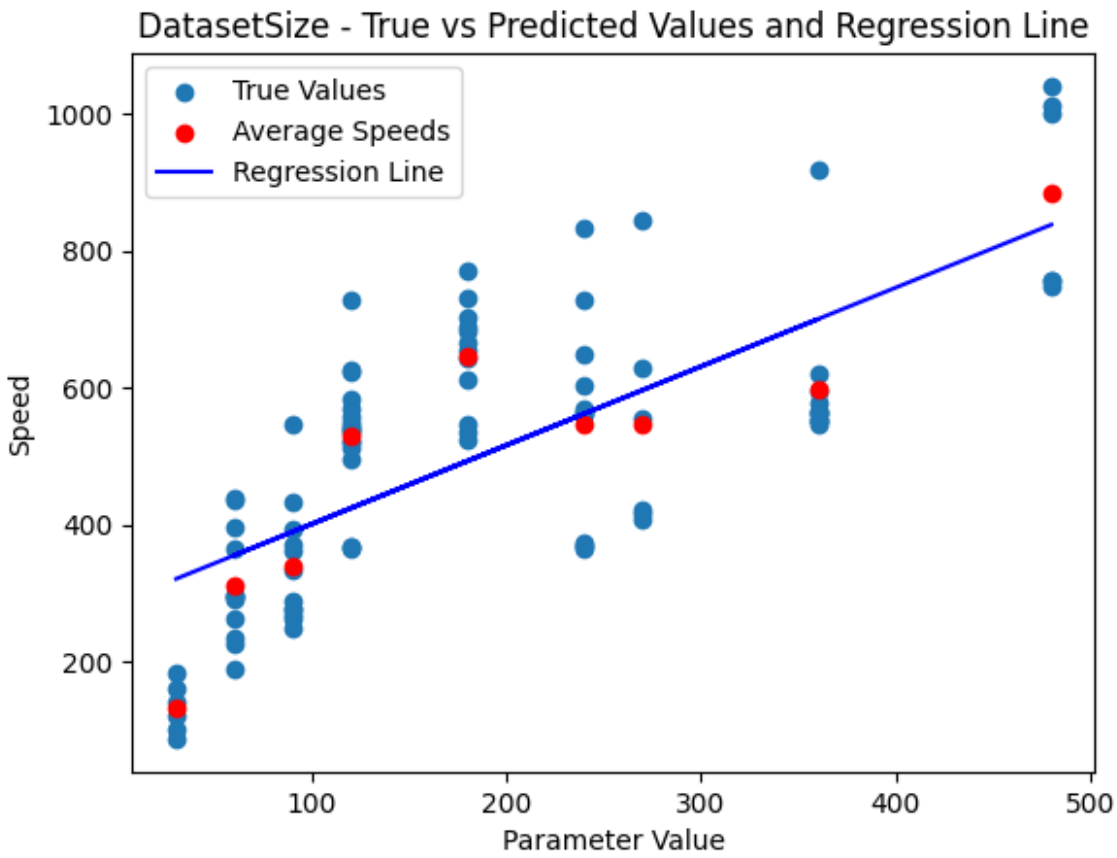
```
BatchNumber
R2 Score: 0.36400443433423857, Coef: [[36.14890683]]
```




```
Repetition
R2 Score: 0.002644625627744812, Coef: [[13.86552909]]
```




```
DatasetSize
R2 Score: 0.5054979036235667, Coef: [[1.14881517]]
```



tfrec_results_table


```
{"summary":"{\n  \"name\": \"tfrec_results_table\",\n  \"rows\": 4,\n
\"fields\": [\n    {\n      \"column\": \"R2 Score\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.21662782489109633,\n        \"min\": 0.002644625627744812,\n
\"max\": 0.5054979036235667,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0.36400443433423857,\n
0.5054979036235667,\n          0.19909858124736113\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Coefficient\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
15.154389134459278,\n        \"min\": 1.148815169801442,\n
\"max\": 36.14890683409005,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          36.14890683409005,\n
1.148815169801442,\n          8.02042106049325\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\
n}","type":"dataframe","variable_name":"tfrec_results_table"}
```

Section 3. Theoretical discussion

Task 3: Discussion in context. (24%)

In this task we refer an idea that is introduced in this paper:

- Alipourfard, O., Liu, H. H., Chen, J., Venkataraman, S., Yu, M., & Zhang, M. (2017). Cherrypick: Adaptively unearthing the best cloud configurations for big data analytics.. In USENIX NSDI 17 (pp. 469-482).

Alipourfard et al (2017) introduce the prediction an optimal or near-optimal cloud configuration for a given compute task.

3a) Contextualise

Relate the previous tasks and the results to this concept. (It is not necessary to work through the full details of the paper, focus just on the main ideas). To what extent and under what conditions do the concepts and techniques in the paper apply to the task in this coursework? (12%)

3b) Strategise

Define - as far as possible - concrete strategies for different application scenarios (batch, stream) and discuss the general relationship with the concepts above. (12%)

Provide the answers to these questions in your report.

Final cleanup

Once you have finshed the work, you can delete the buckets, to stop incurring cost that depletes your credit.

```
!gsutil -m rm -r $BUCKET/* # Empty your bucket
!gsutil rb $BUCKET # delete the bucket
```