Big Data Report

Vighnesh Mane

vighnesh.mane@city.ac.uk

1d i.

Fig. 1 Metrics of cluster with 1 master with 1 vCPU + 3 workers with 1 vCPU (maximal) only on two nodes

As illustrated in Fig. 1 during starting stage the CPU was utilized only approximately 5 percent. The largest amount of CPU power used by the cluster is more than 15 percent. Normally the CPU utilized was 3.43 percent.

Metrics	Network Bytes	Network packets	Disk Bytes	Disk Operations
Maximum incoming	68.98 KiB/s	94.53/s	N/A	N/A
Maximum outgoing	68.46 KiB/s	90.45/s	N/A	N/A
Maximum Read	N/A	N/A	391.28 KiB/s	234.85 KiB/s
Maximum Write	N/A	N/A	20.82/s	5.1/s

Table. 1 Maximum parameters of cluster with 1 master with 1 vCPU + 3 workers with 1 vCPU (maximal) only on two nodes

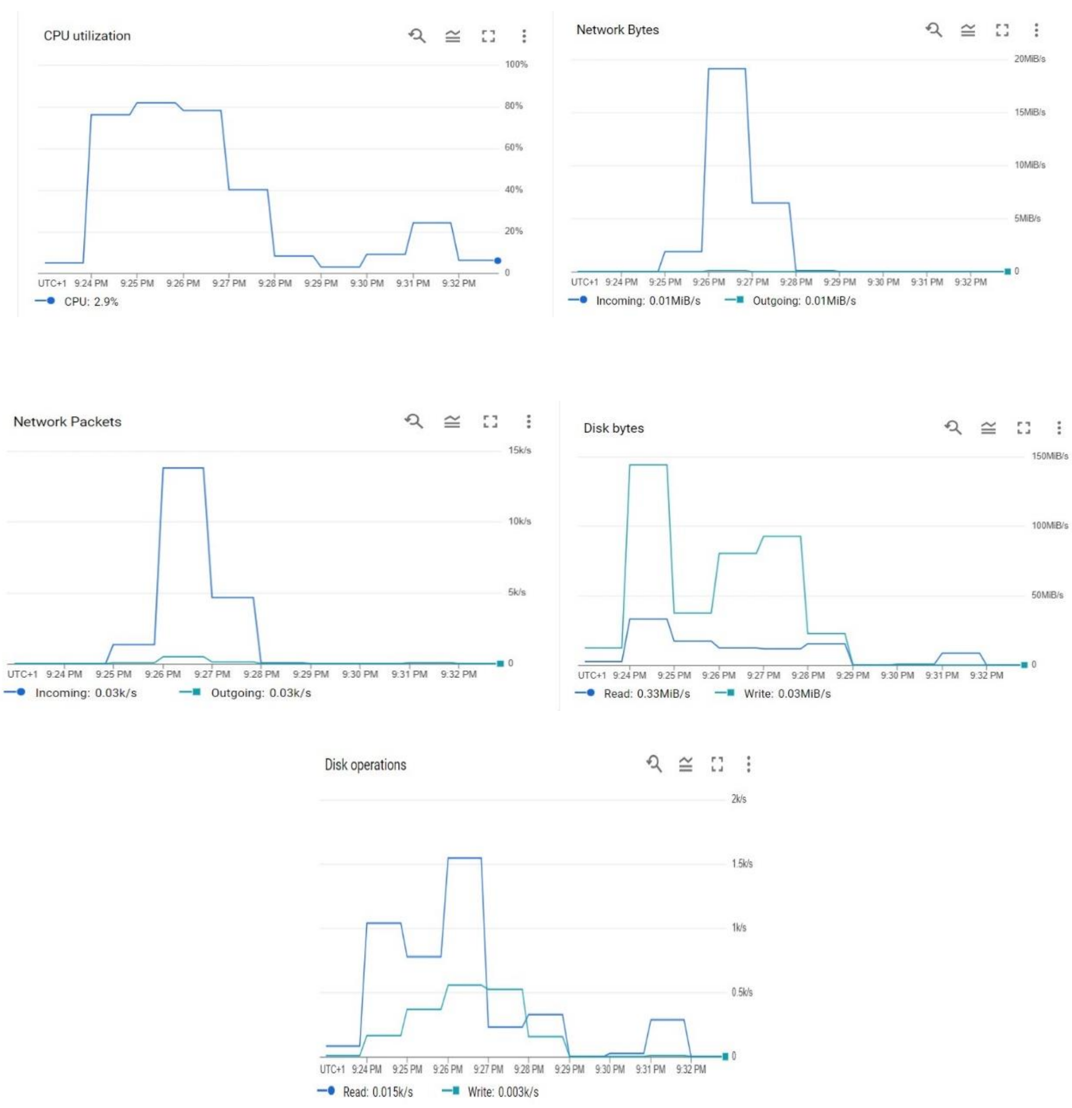

Fig. 2 Metrics of cluster with 1 master with 1 vCPU + 3 workers with 1 vCPU (maximal) after using second parameter

As illustrated in Fig. 2 the largest amount of CPU power used by the cluster is more than 85 percent. Normally the CPU utilized is 2.9 percent.

Metrics	Network Bytes	Network packets	Disk Bytes	Disk Operations
Maximum incoming	19.17 MiB/s	13.83 K/s	N/A	N/A
Maximum outgoing	0.09 MiB/s	0.53 K/s	N/A	N/A
Maximum Read	N/A	N/A	33.56 MiB/s	1.54 K/s
Maximum Write	N/A	N/A	144.07 MiB/s	0.56 K/s

Table. 2 Maximum parameters of cluster with 1 master with 1 vCPU + 3 workers with 1 vCPU (maximal) after using second parameter

The overall scenario seems to be that when the second parameter (16 partitions) is added the utilization of the CPU increased tremendously. Various parameters for network bytes, network packets, disk bytes and disk operation also increased when the parameter is introduced to the cluster.

Cluster	Initialization time	Server connect time
1 master with 1 vCPU + 3 workers with 1 vCPU (maximal) only on two nodes	8826ms	9103ms
1 master with 1 vCPU + 3 workers with 1 vCPU (maximal) after using second parameter	10564ms	10957ms

Table. 3 Configurations of the CPU for 1 master with 1 vCPU + 3 workers with 1 vCPU before and after adding second parameter

1d ii.

Cluster	Initialization time	Server connect time
1 master with 2 vCPU + 3 workers with 1 vCPU	6335ms	6465ms
1 master with 8 vCPU	5673ms	5759ms

Table. 4 Configurations of the CPU for 1 master with 2 vCPU + 3 workers with 1 vCPU and 1 master with 8 vCPU

As illustrated in the Table. 4 the 1 master with 2 vCPU + 3 workers with 1 vCPU cluster took 6335ms for initialization and to connect to the server it took 6465ms whereas 1 master with 8 vCPU took 5673ms for initialization and 5759ms for the connecting to the server.

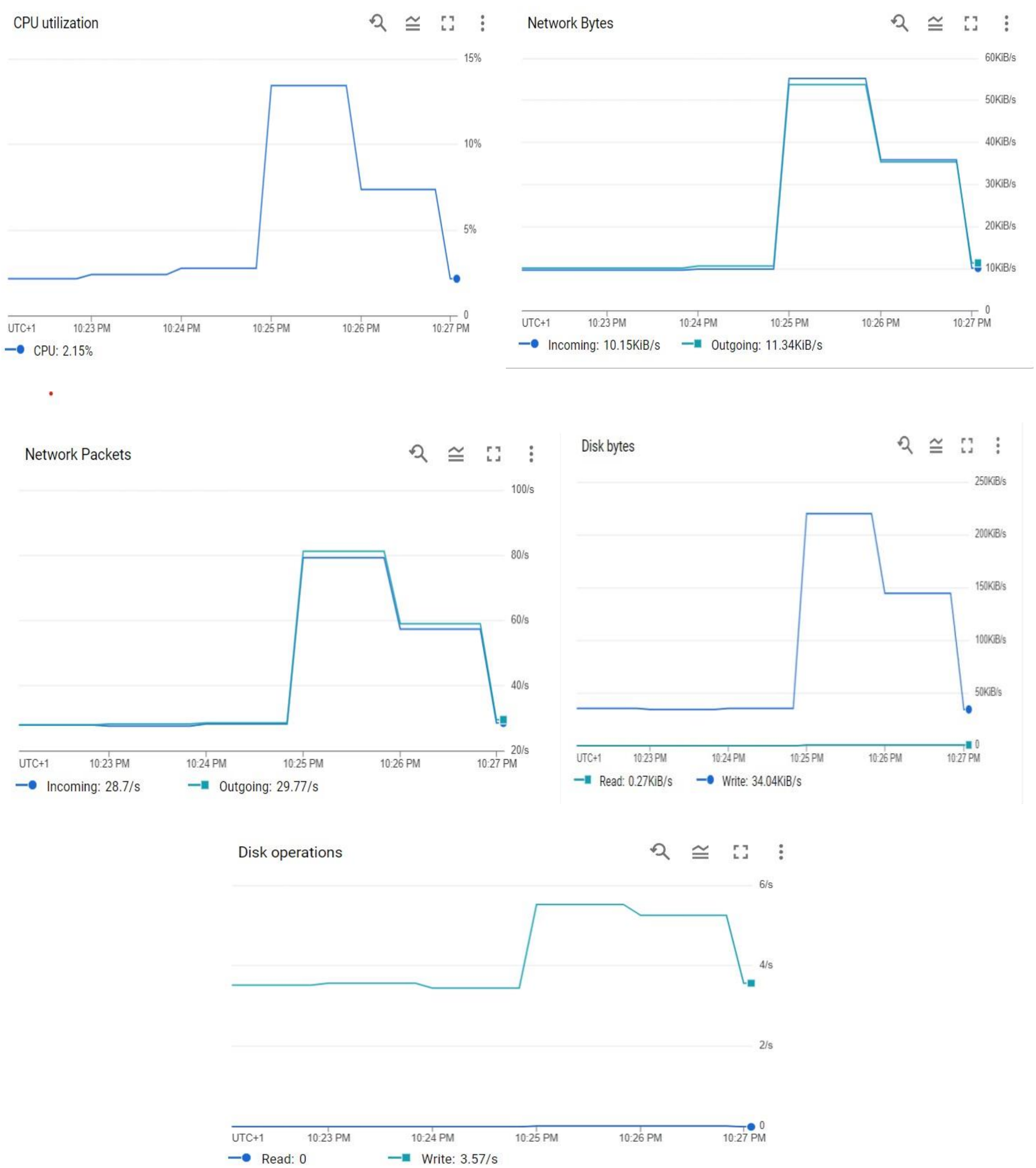

Fig. 3 Metrics of cluster with the 1 master with 2 vCPU + 3 workers with 1 vCPU

Metrics	Network Bytes	Network packets	Disk Bytes	Disk Operations
Maximum incoming	55.1 KiB/s	81.17 /s	N/A	N/A
Maximum outgoing	53.66 KiB/s	79.23 /s	N/A	N/A
Maximum Read	N/A	N/A	0.04 KiB/s	0.02 /s
Maximum Write	N/A	N/A	220.09 KiB/s	5.52 /s

Table. 5 Maximum parameters of cluster with the 1 master with 2 vCPU + 3 workers with 1 vCPU

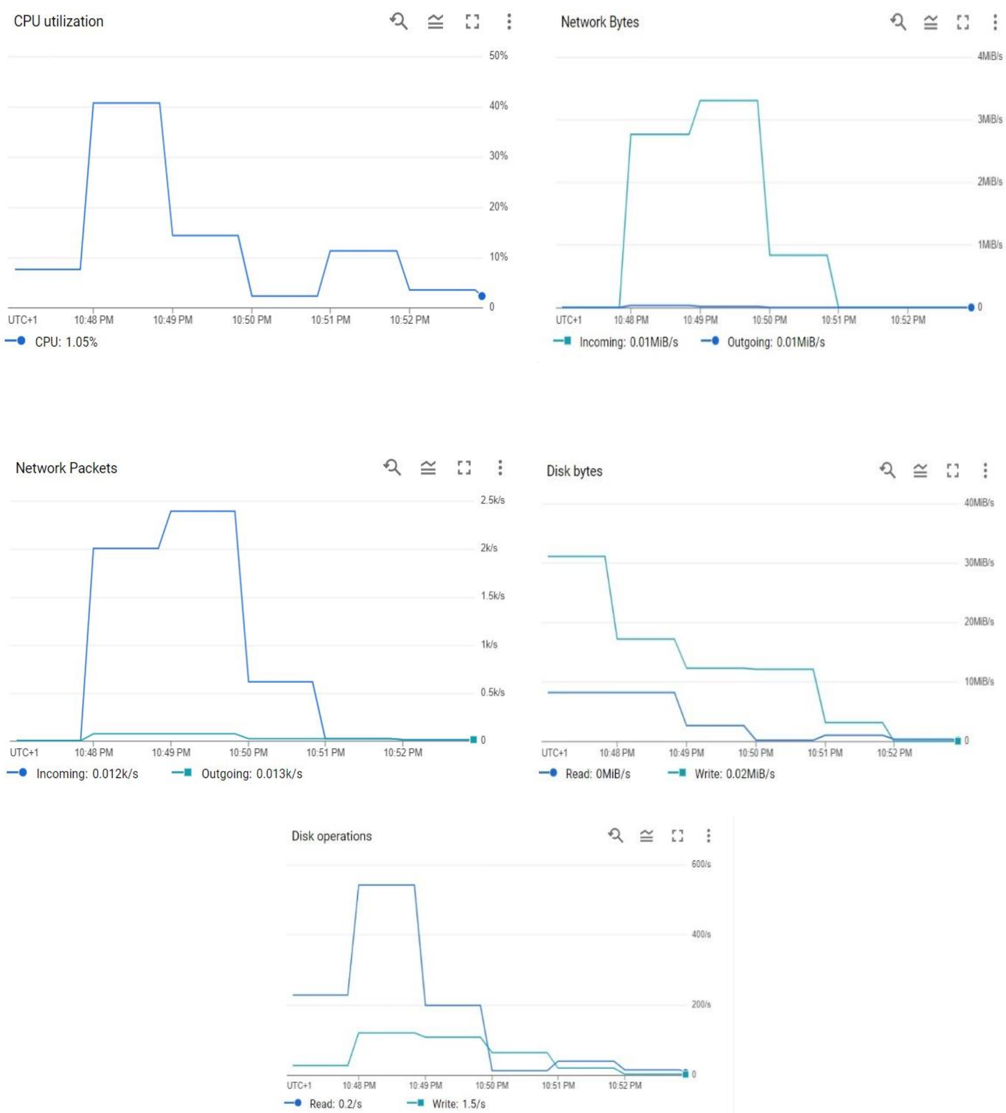

Fig. 4 Metrics of cluster with 1 master with 8 vCPU

Metrics	Network Bytes	Network packets	Disk Bytes	Disk Operations
Maximum incoming	3.31 MiB/s	2.393 K/s	N/A	N/A
Maximum outgoing	0.02 MiB/s	0.08 K/s	N/A	N/A
Maximum Read	N/A	N/A	8.17 MiB/s	541.87 /s
Maximum Write	N/A	N/A	17.25 MiB/s	121.62 /s

Table. 6 Maximum parameters of cluster with 1 master with 8 vCPU

1d iii.

On a lab session, a single task was tested on a single machine, but now, all jobs and workers are operating in the cloud. Lab sessions were done locally, however the cloud is also being used to distribute this strain. Spark differs from most standard applications in that it depends on extra settings in the cloud and more external elements to affect the durations of processes, while most standard apps process locally.

Error for part 1d:

```
### CODING TASK ###
# cluster with a single machine using the maximal SSD size (100) and 1 master with 1 vCPU + 7 workers with 1 vCPU
!gcloud dataproc clusters create $CLUSTER \
  --bucket $PROJECT-storage \
  --image-version 1.5-ubuntu18 \
  --master-machine-type n1-standard-1 \
  --master-boot-disk-type pd-ssd --master-boot-disk-size 100 \
  --num-workers 7 --worker-machine-type n1-standard-1 --worker-boot-disk-size 100 \
  --initialization-actions gs://goog-dataproc-initialization-actions-$REGION/python/pip-install.sh \
  --metadata PIP_PACKAGES=tensorflow==2.4.0

ERROR: (gcloud.dataproc.clusters.create) INVALID_ARGUMENT: Insufficient 'IN_USE_ADDRESSES' quota. Requested 8.0, available 4.0.
```

Fig. 5 Unable to create Cluster 1 master and 1vCPU + 7 workers with 1vCPU (maximal cluster)

2d.

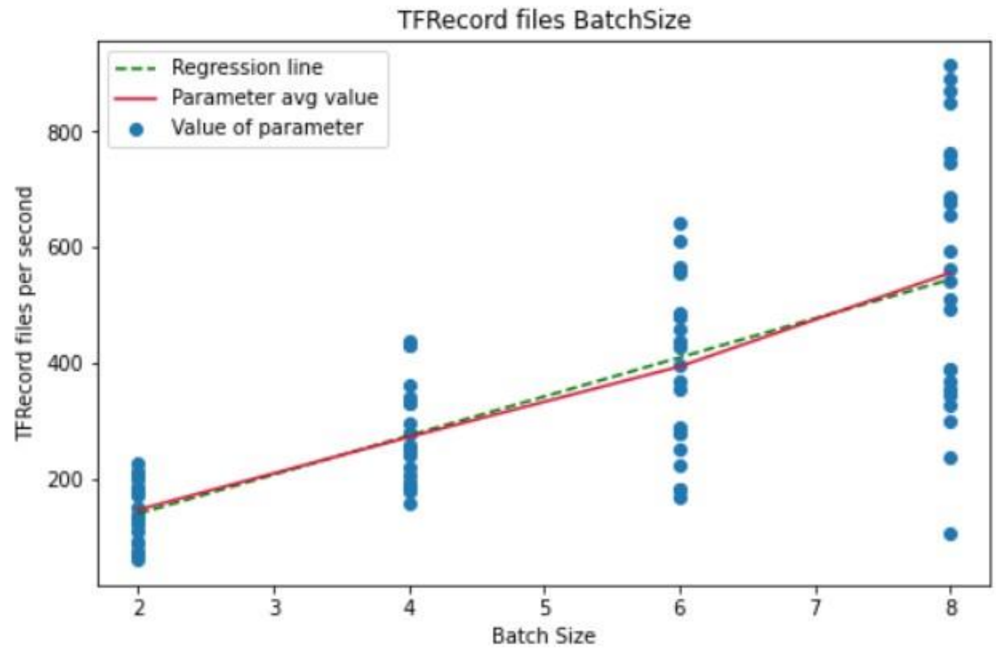

Fig. 6 Batch Size of the reading TFRecord files

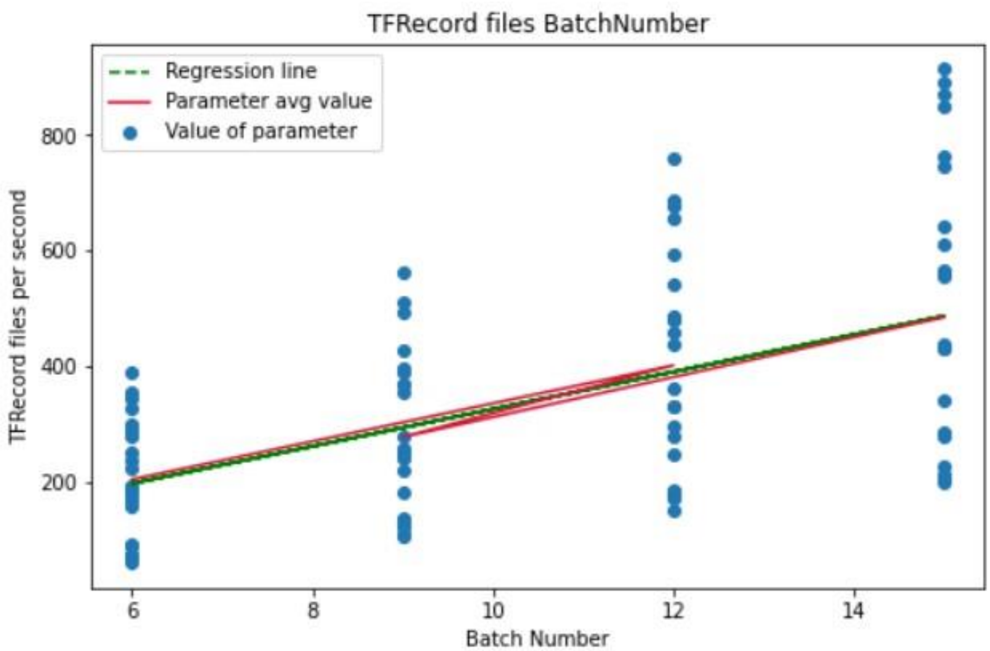

Fig. 7 Batch Number of the reading TFRecord files

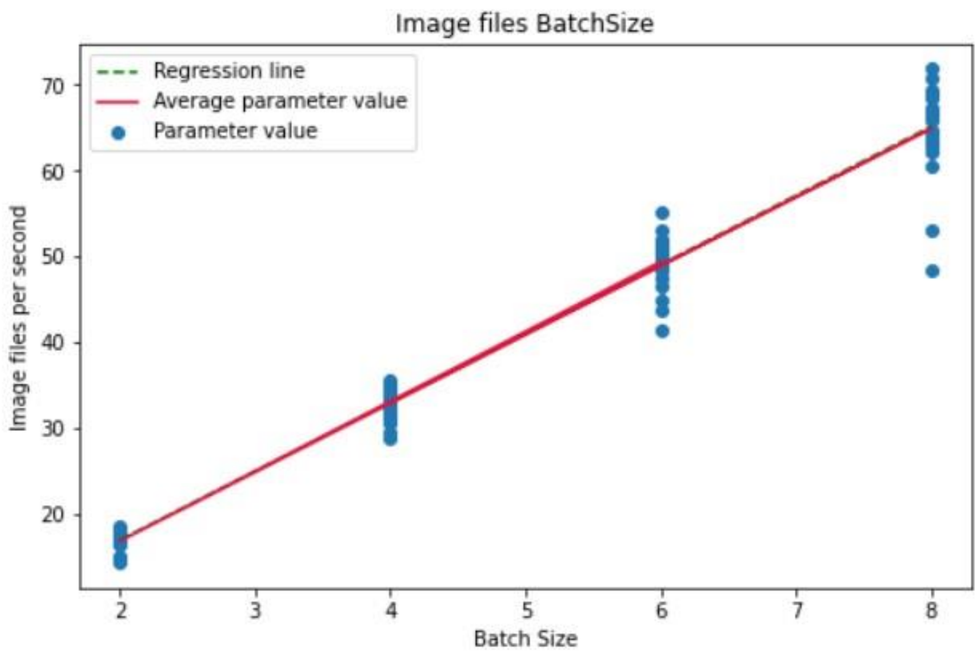

Fig. 8 Batch Size of the Image files

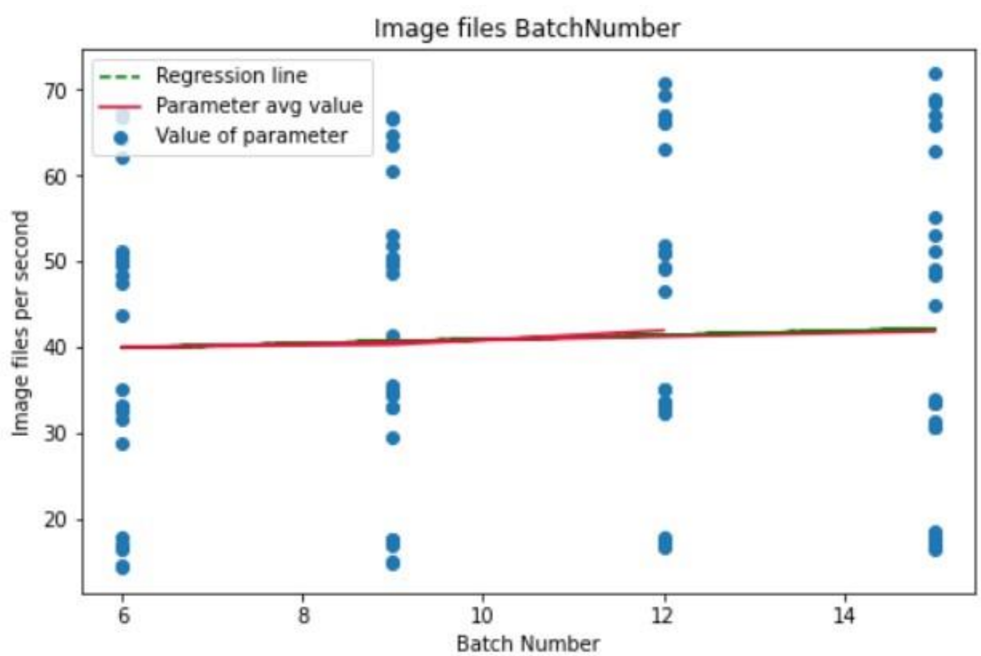

Fig. 9 Batch Number of the Image files

Parameters	Slope	Intercept	P-value
TFRecord files Batch Size	67.55	3.67	0.53
TFRecord files Batch Number	32.24	2.91	0.27
TFRecord files Repetitions	2.57	335.45	8.63
TFRecord files Dataset size	6.50	-0.08	0.88
Image files Batch Size	8.04	0.83	0.97
Image files Batch Number	0.24	38.51	0.001
Image files Repetitions	-0.05	41.18	5.76
Image files Dataset Size	0.49	15.02	0.65

Table. 7 Output of each parameter

As shown in Figs. 6 and 7, the batch size of TfRecord files is 0.53, while the batch number of TfRecord files is 0.27, demonstrating a relationship between the two. If a considerable amount of data is available, it is feasible to improve the P-value even more by picking better combinations. The P-value for batch size is 0.97, and the P-value for batch number is 0.001, which is exceptionally low, as shown in Figures 8 and 9. A comparison of the model results acquired when testing it locally against those received when testing it in the cloud revealed that the model's performance may be greatly improved when testing it locally. It was discovered that the cloud has a high latency of 6488 milliseconds while connecting to servers. It has been possible to save costs by running the tests in parallel and finishing them all on one machine. Because processors don't have to wait for the previous test to be finished before going on to the next, they are able to move on to the next test more quickly. As a rule, systems optimised for throughput tend to be slower than those optimised for speed when it comes to little jobs. It is thus more useful to use storage space to reduce latency while performing little tasks than rely on memory. Because of this, cloud service providers would be able to better control efficiency by tying throughput to the disc's storage capacity. There is a bottleneck in this circumstance since doing regressions other than linear regression on the cloud side will increase latency and be less cost efficient. To put it simply, linear modelling is used to examine the connection between the number of batches and the size of each batch, as well as other factors such as repetitions and the size of the dataset. In order to get the most accurate results, the variables should be in large numbers, but this will take longer and cost cloud service providers more money.

3c.

```
ERROR: (gcloud.ai-platform.jobs.submit.training) HttpError accessing <https://ml.googleapis.com/v1/projects/crypto-avenue-347713/jobs?alt=json>:
  "error": {
    "code": 429,
    "message": "Quota failure for project crypto-avenue-347713. The request for 1 K80 accelerators exceeds the allowed maximum of 0 A100, 0 K80,
    "status": "RESOURCE_EXHAUSTED",
    "details": [
      {
        "@type": "type.googleapis.com/google.rpc.QuotaFailure",
        "violations": [
          {
            "subject": "crypto-avenue-347713",
            "description": "The request for 1 K80 accelerators exceeds the allowed maximum of 0 A100, 0 K80, 0 P100, 0 P4, 0 T4, 0 TPU_V2, 0 TPU_
          }
        ]
      }
    ]
  }
}
>
This may be due to network connectivity issues. Please check your network settings, and the status of the service you are trying to reach.
```

Fig. 10 Error getting GPU quota

```
API [ml.googleapis.com] not enabled on project [844609536274]. Would you like to
 enable and retry (this will take a few minutes)? (y/N)?  y

Enabling service [ml.googleapis.com] on project [844609536274]...
Operation "operations/acf.p2-844609536274-477c4759-59a6-4d49-870e-cad4d62a7c20" finished successfully.
ERROR: (gcloud.ai-platform.jobs.submit.training) HttpError accessing <https://ml.googleapis.com/v1/projects/sanguine-city-349200/jobs?alt=json>:
  "error": {
    "code": 429,
    "message": "Quota failure for project sanguine-city-349200. The request for 1 K80 accelerators exceeds the allowed maximum of 0 A100, 0 K80,
    "status": "RESOURCE_EXHAUSTED",
    "details": [
      {
        "@type": "type.googleapis.com/google.rpc.QuotaFailure",
        "violations": [
          {
            "subject": "sanguine-city-349200",
            "description": "The request for 1 K80 accelerators exceeds the allowed maximum of 0 A100, 0 K80, 0 P100, 0 P4, 0 T4, 0 TPU_V2, 0 TPU_
          }
        ]
      }
    ]
  }
}
>
This may be due to network connectivity issues. Please check your network settings, and the status of the service you are trying to reach.
```

Fig. 10 Error getting GPU quota from another gmail account

```
Epoch 1/5
46/46 [==============================] - 82s 2s/step - loss: 4.0761 - accuracy: 0.2463 - val_loss: 1.5874 - val_accuracy: 0.2484
Epoch 2/5
46/46 [==============================] - 78s 2s/step - loss: 1.5929 - accuracy: 0.2582 - val_loss: 1.6046 - val_accuracy: 0.2375
Epoch 3/5
46/46 [==============================] - 78s 2s/step - loss: 1.6030 - accuracy: 0.2561 - val_loss: 1.6000 - val_accuracy: 0.2469
Epoch 4/5
46/46 [==============================] - 78s 2s/step - loss: 1.5779 - accuracy: 0.2677 - val_loss: 1.5263 - val_accuracy: 0.3266
Epoch 5/5
46/46 [==============================] - 78s 2s/step - loss: 1.5643 - accuracy: 0.2721 - val_loss: 1.5559 - val_accuracy: 0.3203
Wall clock time = 444.70842385292053
Savingtask3a2.pkl to gs://crypto-avenue-347713-storage
```

Fig. 11 Performance of standard GPU (1xK80) in local environment

```
Epoch 1/5
46/46 [==============================] - 80s 2s/step - loss: 2.7179 - accuracy: 0.2571 - val_loss: 1.5620 - val_accuracy: 0.3578
Epoch 2/5
46/46 [==============================] - 74s 2s/step - loss: 1.5468 - accuracy: 0.2938 - val_loss: 1.5453 - val_accuracy: 0.3531
Epoch 3/5
46/46 [==============================] - 75s 2s/step - loss: 1.5199 - accuracy: 0.3115 - val_loss: 1.5312 - val_accuracy: 0.3516
Epoch 4/5
46/46 [==============================] - 75s 2s/step - loss: 1.5027 - accuracy: 0.3118 - val_loss: 1.5001 - val_accuracy: 0.3766
Epoch 5/5
46/46 [==============================] - 75s 2s/step - loss: 1.4899 - accuracy: 0.3179 - val_loss: 1.5337 - val_accuracy: 0.3313
Wall clock time = 378.4437572956085
```

Fig. 12 Performance of complex_model_l_gpu (8xK80) in local environment

The Fig.11 and Fig.12 illustrates the performance of the 1xK80 and 8xK80 GPUs respectively. As seen in both the figures the wall clock time for the 1xK80 is 443.70 and for 8xK80 is 378.44. The loss for 1xK80 is more than the 8xK80 whereas the accuracy for the 8xK80 is better than 1xK80. The validation loss for the 8xK80 is somewhat similar to the 1xK80 and the validation accuracy of the 8xK80 is more than 1xK80

4.

5a.

Several times throughout this testing, basic cloud tasks took far longer to complete than they had before. A source of the issue was identified as cloud noise, which is frequent in the environment. In our black box, a cost function is multiplied using adaptive algorithms to account for noise, and the result is shown. By cherry-picking, we are able to maximize our returns while also evaluating the likelihood of success in various combinations. This strategy may have been more successful in this circumstance, however there are a number of difficulties to overcome when adopting cherry picking, including a complicated performance model, a cost model, and the heterogeneity of the applications to name a few. Also included in the cherry-picking are the strawman answers, which assert that the two most effective methods for forecasting a near-optimal cloud design are modelling and search.

The second article proposes a technique for selecting models in two parts: the assessment environment and the Configurations of Experiments. The evaluation environment is discussed first, followed by the Configurations of Experiments. This coursework makes use of linear regression rather than other regression methods since it is physically impossible to utilise any other regression technique on the cloud, and we have been successful in job 2d by using linear regression instead of other regression methods.

5b.

The data for this research was gathered via the use of a batch processing speed test, which was carried out. The broad set of configuration options available for the CPU may be adjusted to meet the individual needs of the user. It has become feasible to use cherry picking in a straightforward way because to the introduction of Bayesian optimization approaches. When cherry-picking, it is critical to extract data; however, the pace at which micro batch processing is performed should be raised or decreased depending on a number of elements, including the situation and needs. Because it delivers the most important information, cherry picking is an excellent approach for influencing price movements. As a result of the interaction of numerous factors, network delay is experienced. Because of these external factors, as well as the large number of adjustable options available in GCP, maintaining a near-perfect setup on a consistent basis is difficult. Come into play systems such as Cherry Pick, which search through a large number of frameworks to identify the best potential configuration for a specific task. The technique described above has a monetary penalty in terms of calculation time.

In the second half of the second research, parallelism, CNN training, and the utility of the paraDL are all discussed. The basic parallelization processes are discussed in detail in the first article in this series. Data parallelism, as well as parallelism based on an existing pipeline, are examples of these techniques. Data parallelism, spatial parallax, pipeline parallax, and channels/filters parallax are some of the distributed training approaches that are covered in this paper. Distributed learning cannot be used by all students in this course's assignment 3c due to quota restrictions and the fact that it is not cost-effective in the long run. In this challenge, we utilised distributed training to create numerous mirrored algorithms with varying batch sizes, all of which were implemented via distributed training. The assumption made in this study, in addition to training duration and memory estimation, is that the data has already been loaded into the memory before to commencing training. In contrast to Task 2, data was sent to the cloud before the linear regression analysis was carried out in this task. This article addresses sparsification, which is a parallel strategy that only operates on the meaningful weight or gradient in a given situation. As opposed to doing linear regression on every parameter, the model in this coursework's task 2d only presented the most important batch sizes and batch numbers, which was sufficient. Another important parallel approach that has been considered is all-optimization. The goal of this strategy is to reduce the amount of time it takes for calculations to take place. There was a significant variation in calculation time between the activities completed locally and those completed in the cloud for the coursework project.

Word count:

1922

Colab link -
https://colab.research.google.com/drive/1oyU7oJJye7emxsc4CRfocu40GepVn4T?usp=sharing