INM432 BIG DATA

Report

Collab link : https://colab.research.google.com/drive/1iNaVdcUPaiayz80bbklYEvTL7pX-bSqO?usp=sharing

Vivek Kanna Jayaprakash
MSc Data Science
City University of London
London, United Kingdom
Email : Vivek.Jayaprakash@city.ac.uk

I. TASK 1

1D : Optimisation, experiments, and discussion
Experiment with cluster configurations
Part 1
In addition to the experiments testing the cluster script with various VMS as per the attached

We tried both the set of result on our system to identify the difference and tabulated below the findings.

Resources	Logging Initialization time	Server start time
2vCPUs with 6 workers	9970 MS	10113 MS
8 vCPUs	9252 MS	9349 MS

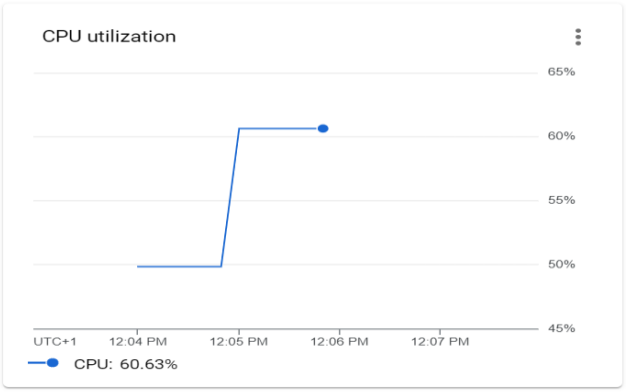

Figure 1: 2vCPUs with 6 workers

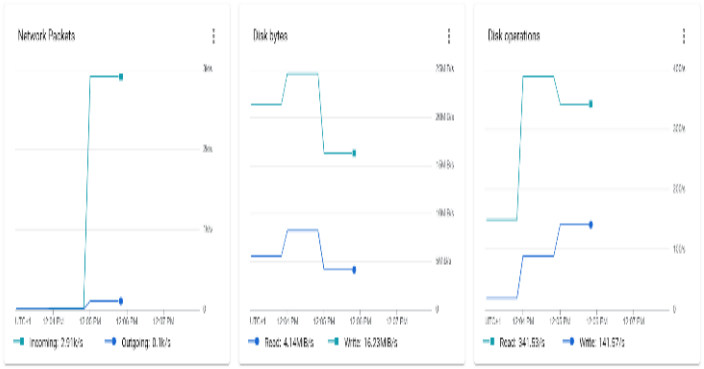

Figure 2 : 2vCPUs with 6 workers STATS

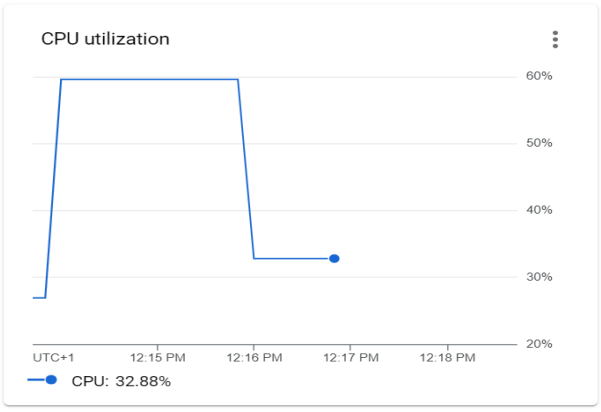

Figure 3 : CPU utilisation 8 vCPUs

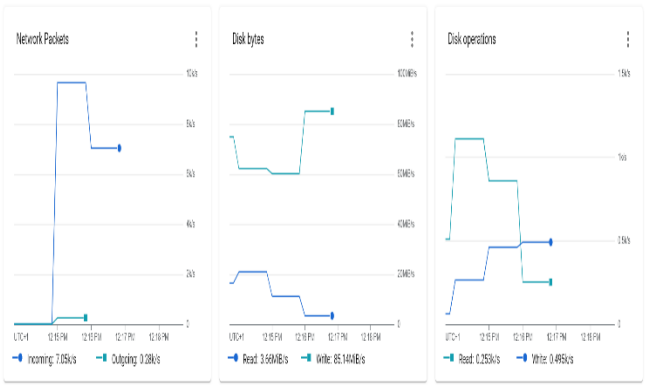

Figure 4: 8 vCPUs Figure 2 : 2vCPUs with 6 workers STATs

Part 2 : Explain the difference between the use if spark and most standard applications.

1.Spark is a very easily programmable concept and doesn't require hard coding like MapReduce.

2.The interactive model of spark compared to MapReduce which is basically a batch processing solutions.

3.In terms of processing of data, the spark can use do various tasks like machine learning , streaming and batch processing therefore it is called as the cluster computation engine.

4.Excutest task 10 to 1000 times faster

5.Used abstraction RDD

6.MapReduce is a disk based concept while space used a very less latency level by caching the scripts.

II. TASK 2

Section 2D: Improving the efficiency:

Cache is technique for improve the accuracy and reducing the time as mentioned in the report. It has efficiency contributor. We understand from our tasks in 2a that the RDD and speed test results are to be mainly avoided repeatedly timing function calls. It contributes to the decrease in the timing and in running of the script in the cluster. This helps us to save in the short term feed of the memory.

This speed reduced gives way to a free access of the underlying layer in the storage area .

We also parallelise the same so it also helps us to enhance the efficiency by running the speed test within the function.

Type	Run time
Non cached script	1 hour 12 minutes
Cached scripted	14 minutes

2 D theory

The basic concept which is identified interpreting the results is that the model performs better on the cloud cluster than on a local scale. And faster too And further caching it makes the cluster even more faster to run on the cloud.

This is why every company is taking their data to the cloud for various advantages especially considering the fast and convincing nature of the analytics

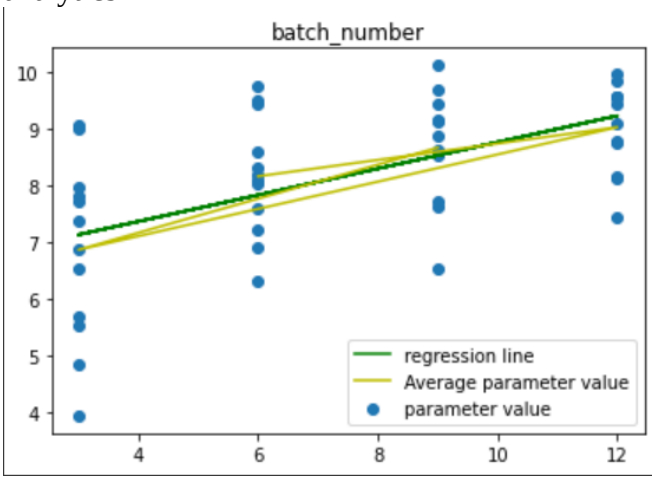

Figure 5: Linear regression on cloud result : batch number

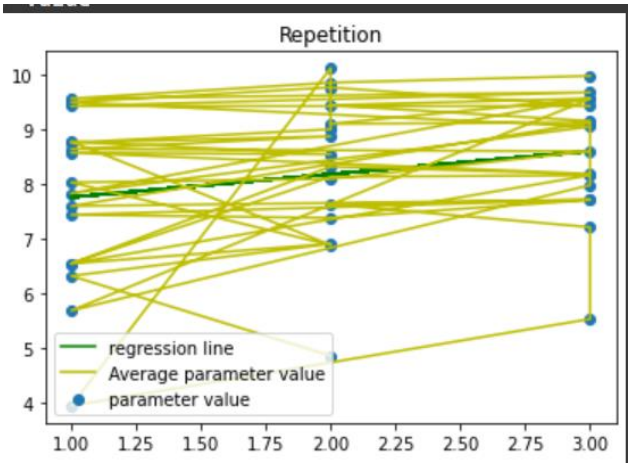

Figure 6: Figure 5: Linear regression on cloud result : repetition

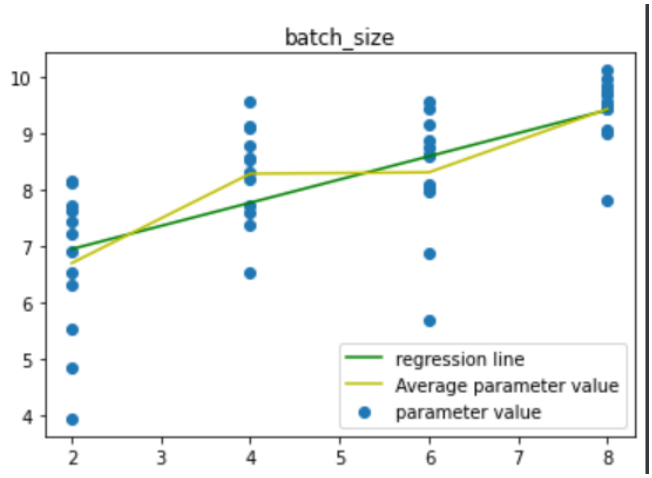

Figure 7: Figure 5: Linear regression on cloud result : batch size

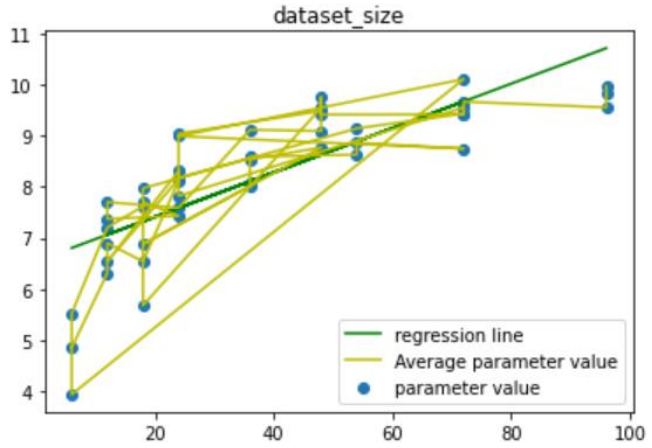

Figure 8: Figure 5: Linear regression on cloud result : datset_size

III. TASK 3

3C : Distributed learning

Using the various different batch strategies and batch sizes.

1. Standard machine

Compute Engine machine name: n1-standard-4
BATCH SIZE 64

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1	4.9362	0.2344	1.5918	0.2422
2	1.5863	0.2533	1.5806	0.2467
3	1.5671	0.2560	1.5815	0.2467
4	1.5585	0.2567	1.5465	0.2422
5	1.5244	0.2887	1.5253	0.3504

2. complex_model_l_gpu

Compute Engine machine name: n1-standard-32-k80x8
using only 1 master 8 x GPU K80
Batch size 128

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1	8.5651	0.2158	1.6054	0.2556
2	1.5768	0.3006	1.5805	0.2768
3	1.5479	0.3177	1.5961	0.2522
4	1.5585	0.2567	1.5465	0.2422
5	1.5244	0.2887	1.5253	0.2504

3. complex_model_l_gpu

Compute Engine machine name: n1-standard-32-k80x8
8 K80 GPU,
BATCH SIZE 128 - B

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1	4.5836	0.2374	1.619	0.2511
2	1.5443	0.2662	1.603	0.3466
3	1.58627	0.26777	1.600	0.2488
4	1.5850	0.26599	1.5969	0.2455
5	1.5838	0.2652	1.5815	0.2611

4. complex_model_l_gpu

Compute Engine machine name: n1-standard-32-k80x8
8 K80 GPU,
BATCH SIZE 64

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1	2.662	0.2209	1.6049	0.2354
2	1.602	0.2511	1.6054	0.2388
3	1.602	0.2526	1.6022	0.2354
4	1.599	0.2601	1.6001	0.2354
5	1.5968	0.2652	1.5865	0.2522

IV. TASK 4

Big data analytics is a concept that is growing and evolving rapidly as the days progress to a fully sustainable AI society.

In order to support the revolution of BD analytics a huge number of evolving techniques having been evolving like MapReduce, SQL, DL, and in memory analyst.

The execution of the said concepts all similar form a structural point of view.it all requires a cluster of virtual machines.

All the analytical jobs have varied amount of behavioural patterns and resource requirements which include the Use of vCPUs disks and networks etc

The configuration in the analytical part of the big data cloud cannot be easily unified i.e VM instance types and VM numbers

In order to chose the right cloud configuration for any said job is an herculean task as there is very minimal amount of error for any quality deviation and the amount of financials involved.

In our experiments conducted above we are running our scripts locally in our instance and then creating a cluster on the cloud and running the same on the cloud. We found out that it is highly difficult hosting the perfect cluster based on the various different configurations available also because of the said restrictions. We went with a very simple approach that is to move from building a very small cluster to a bigger one and progressing s the stage goes. The different combinations of RAM and CPU, num of workers were crucial in setting up the same.

Choosing the right cloud configuration for an application is essential to serve quality and be in the utmost of commercial competitiveness.

This is similar to a trial and error method in which we may get stuck in the spot for any local minima leading to inaccuracy and therefore we won't be getting the configurations in an optimal way.

A bad configuration can result in 12 times more costs with very poor optimization of the performance.

But as we understand from the above experiments that its very difficult to achieve high accuracy with cheap overhead costs. The process is further more made difficult by the fact that the configuration must me made adaptable for various applications and executions.

Alipourfoud [1] In the book reference suggests a technical workflow to find the maximum effect configuration knowns as cherry pick.

Cherry pic leverages Bayesian optimisation modelling in which high frequency and high accurate models are build for various executions and are accurate enough to differentiate the best and the close to the best configuration.

In the same book the author has experiment with the technique in which cherry pick has a 45-90% chance of finding the optimal configurations which saves a high amount of costs compared to the experiments in use in the current trend.

The concept of Cherry picking is very simple

We run the configurations in the cloud in which a black box modelling is present which automatically detects the results and updates the same between the cloud in which the cost is configured.

Based on this results provided by the black box the user can choose the best modelling model which can give us high performance and low cost.

The above steps are iterated in a loop till we get the best configuration combination result.

The concept of cherry pick would have been very useful in our process for picking the perfect cloud cognizations which thereby would have made the performance better with lower incurring costs.

Data paralysation is a concept of paralysing the data across various processors in various parallel running computing devices or environments. The main job is to distribute the said given data across various different nodes in which the data acts parallel

It can be applied in regular data structures also.

Its is process which is very limited by the memory capacity and the communications overheard. The entire model is replicated foe each node on the computing environments. This is why the training of large or deeper neural networks have a memory capacity limit.

The videos and high definition pictures comprises the memory capacity and would limit the number of samples which is therefor processed by the GPU.

We understood the same in the experiments with effect to the table mentioned

Parallelism strategy	Conv	Pooling	BNorm / LNorm	ReLU	FC
Data	-	-	-	-	-
Spatial	✓	✓	-	-	✓
Filter / Channel	✓	-	-	-	✓

Figure 9: different parallelism techniques

Even with many algorithm combinations it is very hard to get the communication between the system and the said process is still a bottleneck when the model size goes up.

Hybrid parallelisation is technique suggested in the paper[2] to avoid the same.
It generally suggests a combination of two or more type of the parallelism techniques (of data + spatial parallelism etc.)

The other techniques such as spatial parallelism , model horizontal Parnellism , are other ways to improve the same.

The paper Introduces the concept of ParaDL which is oracle technique in which using the information obtained beforehand such aa the material, model and the computer configs and also user constrains are taken as input to calculates the computational and communication time to produce best performance.

The said models contributes to the strategy change and it would eventually increase the efficiency of the execution time and improve the strategies thereby guaranteeing us a full high performance.

This technique could have been very useful in our project and will be considered for the future work.

5B

In the concept of process, we perfume the batch operations as mentioned in the inline, streaming and the cloud concepts

We did batch sizes and process on the same static data to do our analysis. This was very simple technique to easily modify the cluster CPU size to the data required and would be high reliable and relevant to the facts we can change.

The Bayesian optimisation techniques for Cherry picking would have been very useful in this approach as this allows us to use the black box to easily experiment with our results.

This would have resulted in highly accurate global minimum in the training at all the above tasks.

We experimented with various codes from the stack overflow duly referenced in the task 3 AI cloud platform machine learning segment which is a process that could have been made fairly easy if we have chosen to use cherry pick.

REFERENCES

[1]. Alipourfard, O., Harry Liu, H. and Chen, J., 2022. *CherryPick: Adaptively Unearthing the Best Cloud Configurations for Big Data Analytics*.

[2]. Kahira, A., Nguyen, T., Gomez, L., Takano, R., Badia, R. and Wahib, M., 2021. An Oracle for Guiding Large-Scale Model/Hybrid Parallel Training of Convolutional Neural Networks. *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*,.

[3]. Pointer, I., 2022. *What is Apache Spark? The big data platform that crushed Hadoop*. [online] InfoWorld. Available at: <https://www.infoworld.com/article/3236869/what-is-apache-spark-the-big-data-platform-that-crushed-hadoop.html#:~:text=Apache%20Spark%20is%20a%20data,wit

h%20other%20distributed%20computing%20tools.> [Accessed 4 May 2022]

[4]. Tutorialspoint.com. 2022. *Data parallelism vs Task parallelism*. [online] Available at: <https://www.tutorialspoint.com/data-parallelism-vs-task-parallelism> [Accessed 4 May 2022].

[5]. Dreyfus, G., 2005. *Neural networks*. Berlin: Springer

[6]. GitHub. 2022. *GitHub - Mahrukh-Niazi/train_neural_network_TFD: Train a neural network on a subset of the Toronto Faces Dataset (TFD) and optimize it.*. [online] Available at: <https://github.com/Mahrukh-Niazi/train_neural_network_TFD> [Accessed 24 April 2022].

[7]. GitHub. 2022. *GitHub - Mahrukh-Niazi/train_neural_network_TFD: Train a neural network on a subset of the Toronto Faces Dataset (TFD) and optimize it.*. [online] Available at: https://github.com/Mahrukh-Niazi/train_neural_network_TFD> [Accessed 24 April 2022].

[8]. GitHub. 2022. *GitHub - shangeth/Lunar-Lander-Deep-RL: Deep Q network, DDQN, Dueling DQN for Lunar Lander Environment.*. [online] Available at: <https://github.com/ nar-Lander-Deep-RL> [Accessed 24 April 2022].

[9]. Businesstechweekly.com. 2022. *Big Data Basics: Understanding Big Data - Businesstechweekly.com*. [online] Available at: <https://www.businesstechweekly.com/operational-efficiency/data-management/big-data-basics-understanding-big-data/> [Accessed 4 May 2022].

[10]. SearchDataManagement. 2022. *Hadoop vs. Spark: In-Depth Big Data Framework Comparison*. [online] Available at: <https://www.techtarget.com/searchdatamanagement/feature/Hadoop-vs-Spark-Comparing-the-two-big-data-frameworks> [Accessed 4 May 2022].