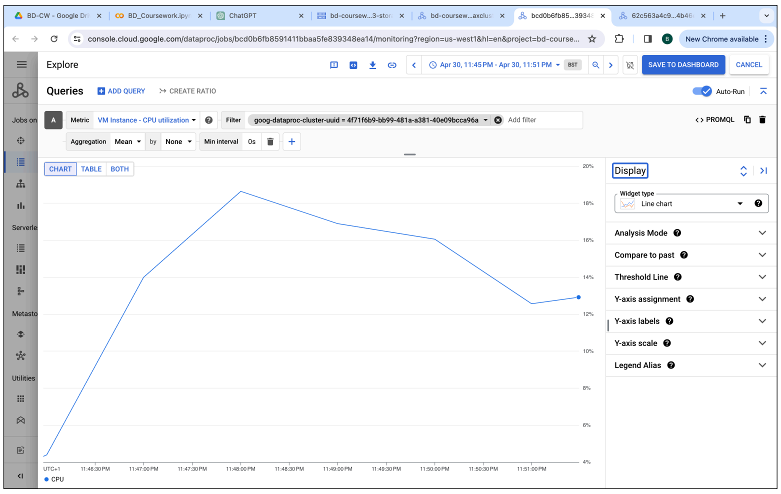
INM432 Big Data Coursework Report

- **Student name, ID:** Baibhav Datta (230059246)
- **Google Drive folder:** https://colab.research.google.com/drive/1Zm61lzjU77k7T-cHY4Rgh8WvGAbXygbg?usp=sharing

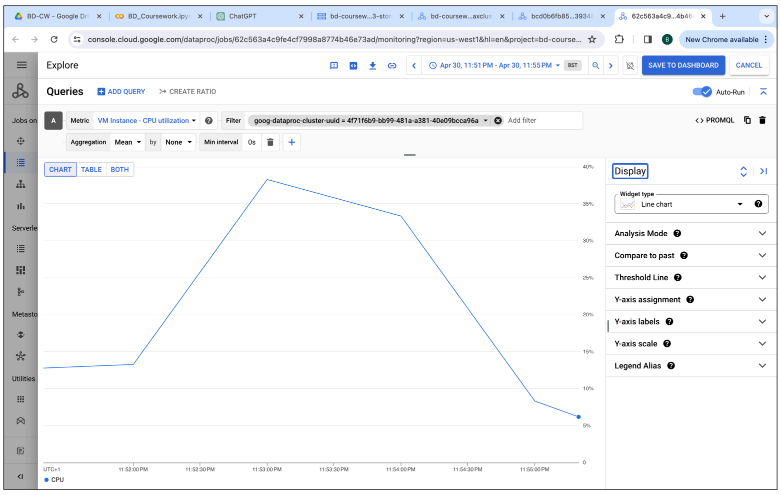
TASK 1d.i) Improving Parallelisation

To address the observation that all computation was initially concentrated on only two nodes, I made changes to the parallelisation strategy by adjusting the number of partitions. By utilising the second parameter in the initial call to parallelize function, I increased the number of partitions from 2 to 16.

Observations:
<u>Before Change (2 partitions)</u>:
Processing Time: 5 minutes 31 seconds
Cluster Utilisation: Concentrated on only two nodes
Screenshot: CPU Utilisation- Lower CPU Utilisation (approx. 18%) signifies inefficient cluster utilisation.


console.cloud.google.com/dataproc/jobs/bcd0b6fb8591411bbaa5fe839348ea14/monitoring?region=us-west1&hl=en&project=bd-course...
Explore
Apr 30, 11:45 PM - Apr 30, 11:51 PM
SAVE TO DASHBOARD
CANCEL
Queries
Metric VM Instance - CPU utilization
Filter goog-dataproc-cluster-uuid = 4f71f6b9-bb99-481a-a381-40e09bcca96a
Aggregation Mean by None
Min interval 0s
CHART TABLE BOTH
Display
Widget type Line chart
20%
18%
16%
14%
12%
10%
8%
6%
4%
11:46:30 PM
11:47:00 PM
11:47:30 PM
11:48:00 PM
11:48:30 PM
11:49:00 PM
11:49:30 PM
11:50:00 PM
11:50:30 PM
11:51:00 PM
CPU

<u>After Change (16 partitions)</u>:
Processing Time: 2 minutes 29 seconds
Cluster Utilisation: Utilisation spread across multiple nodes
Screenshot: CPU Utilisation- Higher CPU Utilisation (approx. 37%) signifies efficient cluster utilisation.

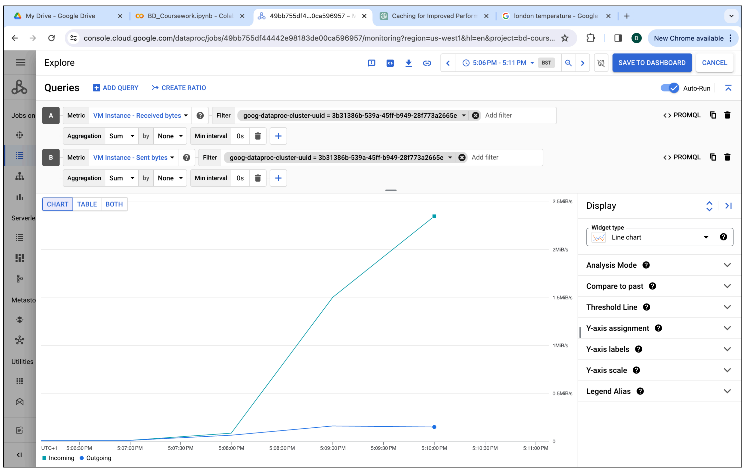

console.cloud.google.com/dataproc/jobs/62c563a4c9fe4cf7998a8774b46e73ad/monitoring?region=us-west1&hl=en&project=bd-course...
Explore
Apr 30, 11:51 PM - Apr 30, 11:55 PM
SAVE TO DASHBOARD
CANCEL
Queries
Metric VM Instance - CPU utilization
Filter goog-dataproc-cluster-uuid = 4f71f6b9-bb99-481a-a381-40e09bcca96a
Aggregation Mean by None
Min interval 0s
CHART TABLE BOTH
Display
Widget type Line chart
40%
35%
30%
25%
20%
15%
10%
5%
0
11:52:00 PM
11:52:30 PM
11:53:00 PM
11:53:30 PM
11:54:00 PM
11:54:30 PM
11:55:00 PM
CPU

Interpretation:
The adjustment in the number of partitions significantly improved cluster utilisation and reduced processing time. By distributing the workload across more partitions, the job executed faster and efficiently utilized the available cluster resources. This enhancement demonstrates the importance of optimising parallelisation strategies to achieve better performance in distributed computing environments.

TASK 1d.ii) Experiments with Cluster Configurations

Cluster with 8 machines

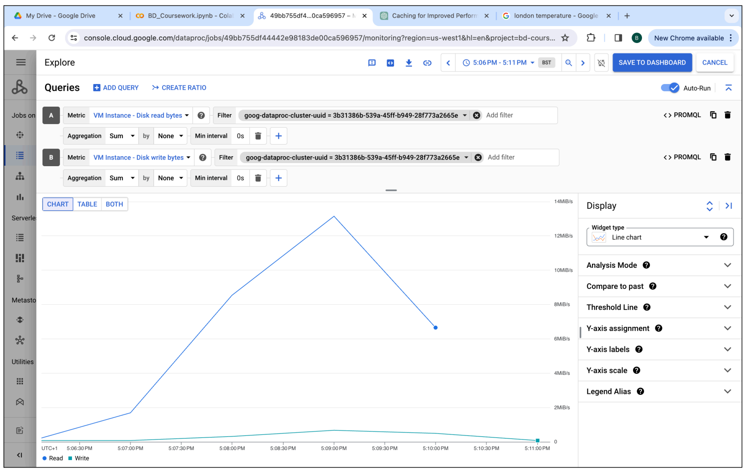
Configuration: 1 master and 7 worker nodes, 1 CPU each, master disk (SSD) size 500GB, worker disk (Standard) size 585GB, machine type Standard.


Explore — 5:06 PM - 5:11 PM

A Metric: VM Instance - Received bytes; Filter: goog-dataproc-cluster-uuid = 3b31386b-539a-45ff-b949-28f773a2665e; Aggregation: Sum by None; Min interval: 0s

B Metric: VM Instance - Sent bytes; Filter: goog-dataproc-cluster-uuid = 3b31386b-539a-45ff-b949-28f773a2665e; Aggregation: Sum by None; Min interval: 0s

Widget type: Line chart


Explore — 5:06 PM - 5:11 PM

A Metric: VM Instance - Disk read bytes; Filter: goog-dataproc-cluster-uuid = 3b31386b-539a-45ff-b949-28f773a2665e; Aggregation: Sum by None; Min interval: 0s

B Metric: VM Instance - Disk write bytes; Filter: goog-dataproc-cluster-uuid = 3b31386b-539a-45ff-b949-28f773a2665e; Aggregation: Sum by None; Min interval: 0s

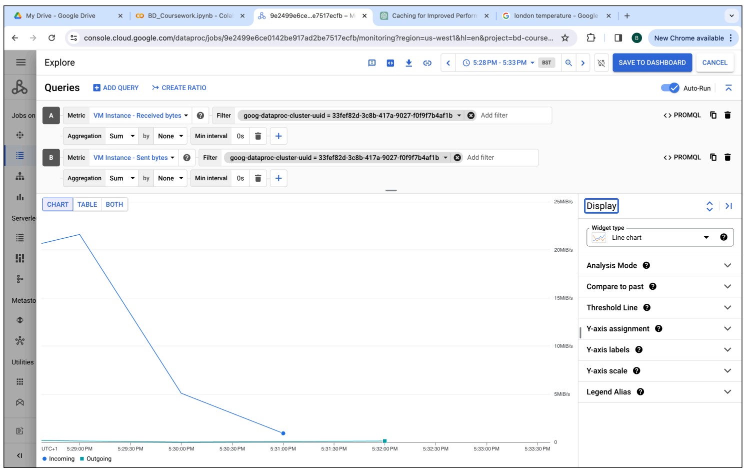
Widget type: Line chart

Network Bytes

Disk Bytes

Cluster with 4 machines () double resources

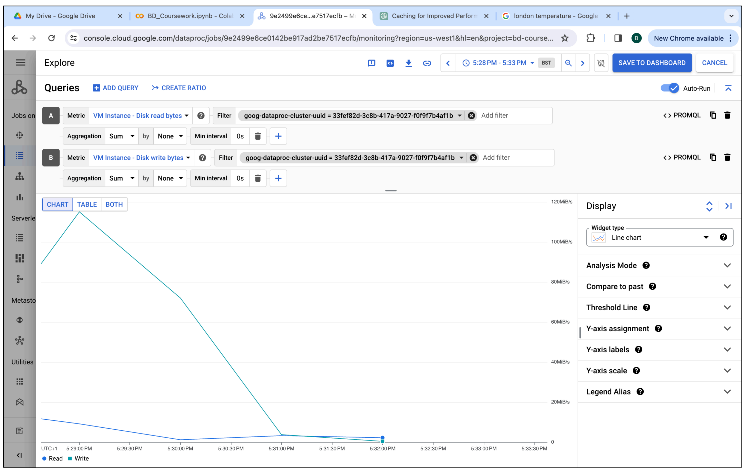
Configuration: 1 master and 3 worker nodes, 2 CPU each, master disk (SSD) size 500GB(max capacity), worker disk (Standard) size 1365GB, machine type High Memory.


Explore — 5:28 PM - 5:33 PM

A Metric: VM Instance - Received bytes; Filter: goog-dataproc-cluster-uuid = 33fef82d-3c8b-417a-9027-f0f9f7b4af1b; Aggregation: Sum by None; Min interval: 0s

B Metric: VM Instance - Sent bytes; Filter: goog-dataproc-cluster-uuid = 33fef82d-3c8b-417a-9027-f0f9f7b4af1b; Aggregation: Sum by None; Min interval: 0s

Widget type: Line chart


Explore — 5:28 PM - 5:33 PM

A Metric: VM Instance - Disk read bytes; Filter: goog-dataproc-cluster-uuid = 33fef82d-3c8b-417a-9027-f0f9f7b4af1b; Aggregation: Sum by None; Min interval: 0s

B Metric: VM Instance - Disk write bytes; Filter: goog-dataproc-cluster-uuid = 33fef82d-3c8b-417a-9027-f0f9f7b4af1b; Aggregation: Sum by None; Min interval: 0s

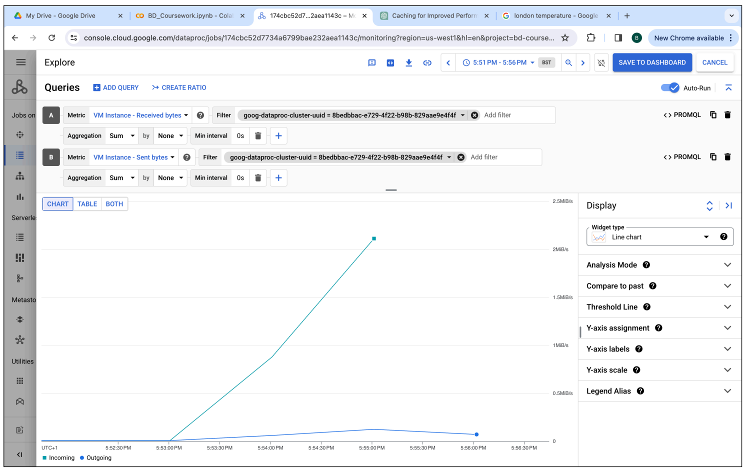
Widget type: Line chart

Network Bytes

Disk Bytes

Cluster with 1 machine

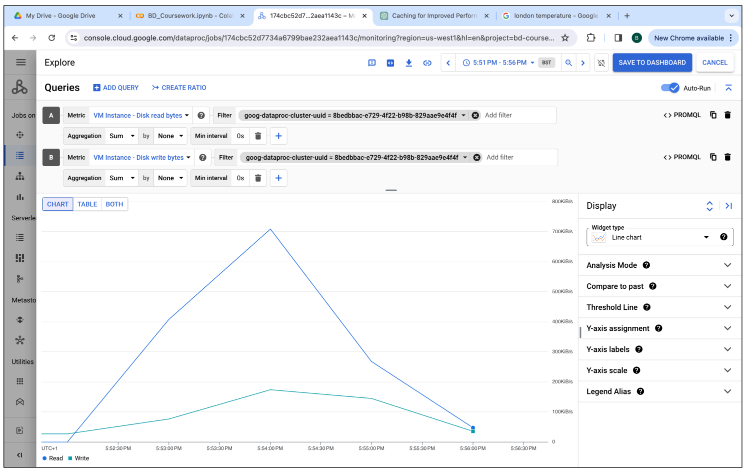
Configuration: 1 master machine, 8 CPUs, master disk (SSD) size 500GB, machine type High Memory.


Explore — 5:51 PM - 5:56 PM

A Metric: VM Instance - Received bytes; Filter: goog-dataproc-cluster-uuid = 8bedbbac-e729-4f22-b98b-829aae9e4f4f; Aggregation: Sum by None; Min interval: 0s

B Metric: VM Instance - Sent bytes; Filter: goog-dataproc-cluster-uuid = 8bedbbac-e729-4f22-b98b-829aae9e4f4f; Aggregation: Sum by None; Min interval: 0s

Widget type: Line chart


Explore — 5:51 PM - 5:56 PM

A Metric: VM Instance - Disk read bytes; Filter: goog-dataproc-cluster-uuid = 8bedbbac-e729-4f22-b98b-829aae9e4f4f; Aggregation: Sum by None; Min interval: 0s

B Metric: VM Instance - Disk write bytes; Filter: goog-dataproc-cluster-uuid = 8bedbbac-e729-4f22-b98b-829aae9e4f4f; Aggregation: Sum by None; Min interval: 0s

Widget type: Line chart

Network Bytes

Disk Bytes

Interpretation:

Cluster configurations show varying patterns in disk I/O and network bandwidth allocation. In an 8-machine cluster, network bytes gradually peaked at 2.5 MiB/s, while disk bytes stabilised at 7 MiB/s after peaking at 13 MiB/s. Conversely, a 4-machine cluster initially had higher network bytes at 21 MiB/s but stabilised at 120 MiB/s for disk bytes. In contrast, a single machine exhibited modest network bytes at 2 MiB/s and disk bytes at 700 KiB/s, indicating under-utilisation. Overall, higher resource clusters displayed increased activity, but efficiency varied based on workload characteristics.

TASK 1d.iii)

In standard applications, data is typically stored centrally, but in Spark, especially when using Google Cloud Storage (GCS), data is distributed across multiple nodes. GCS is a cloud-based object storage service where data is distributed across Google's infrastructure. Spark processes data directly from these distributed locations, leveraging parallelism across the cluster.

In standard applications, data is stored centrally in a local file system or a database. However, in Spark, especially when using services like GCS, data is distributed across multiple nodes in a cluster. Spark employs a parallelisation approach known as data parallelism. This involves partitioning large datasets into smaller chunks, with each partition processed independently by different executors on various nodes in the Spark cluster.

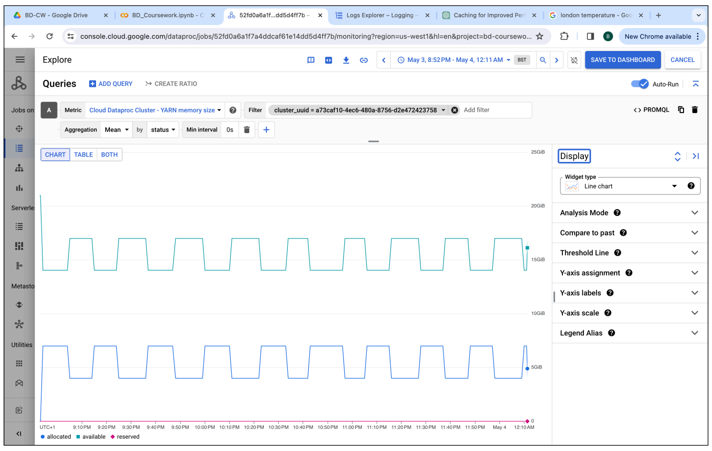
TASK 2c

Using caching in Spark improves performance by storing intermediate RDDs in memory, avoiding recomputation. This is especially beneficial when accessing the same intermediate results multiple times. Caching the RDD containing these results speeds up subsequent operations as data is quickly retrieved from memory.

In this project, caching reduced processing time from 3 hours and 18 minutes to 24 minutes and 37 seconds, showcasing its effectiveness in enhancing efficiency and resource utilisation.

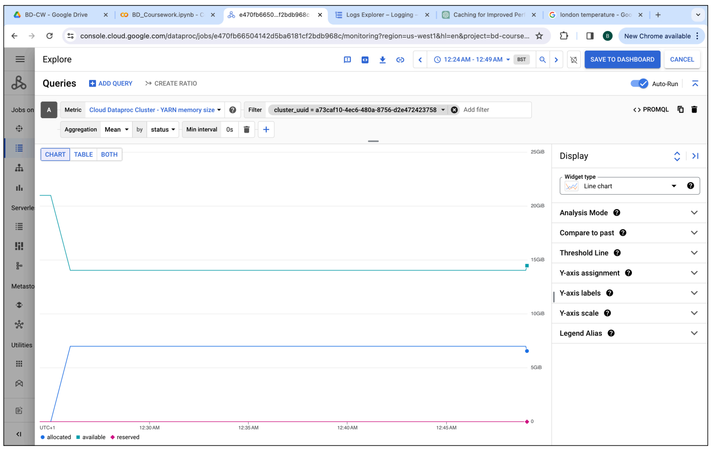
Without caching, repeated recomputation occurs, prolonging processing time and potentially causing fluctuations in resource utilisation. Caching plays a critical role in mitigating computational redundancies and optimising performance in Spark-based data processing tasks.

The graphs for Memory Utilisation before and after caching are as follows-



console.cloud.google.com/dataproc/jobs/52fd0a6a1f7a4ddcaf61e14dd5d4ff7b/monitoring?region=us-west1&hl=en&project=bd-coursewo...

Metric: Cloud Dataproc Cluster - YARN memory size
Filter: cluster_uuid = a73caf10-4ec6-480a-8756-d2e472423758
Aggregation: Mean by status
Min interval: 0s
Time range: May 3, 8:52 PM - May 4, 12:11 AM BST
Widget type: Line chart



console.cloud.google.com/dataproc/jobs/e470fb66504142d5ba6181cf2bdb968c/monitoring?region=us-west1&hl=en&project=bd-course...

Metric: Cloud Dataproc Cluster - YARN memory size
Filter: cluster_uuid = a73caf10-4ec6-480a-8756-d2e472423758
Aggregation: Mean by status
Min interval: 0s
Time range: 12:24 AM - 12:49 AM BST
Widget type: Line chart

Memory Utilisation (before caching)

Memory Utilisation (after caching)

TASK 2d

The results of the linear regression performed on the parameters and reading speed data for image files and TFRecord files are presented below:

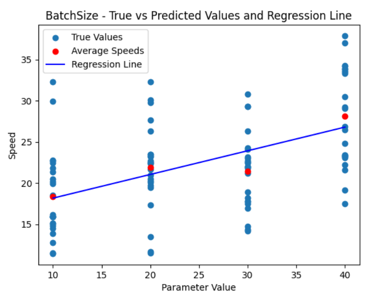
	R2 Score	Coefficient
BatchSize	0.263661	0.287634
BatchNumber	0.128679	0.669808
Repetition	0.018367	1.138747
DatasetSize	0.316659	0.028336

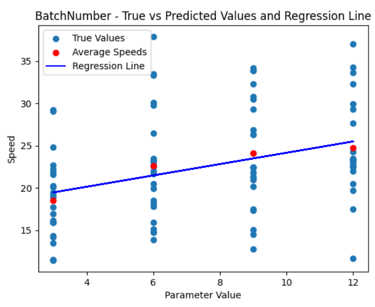
Image Files

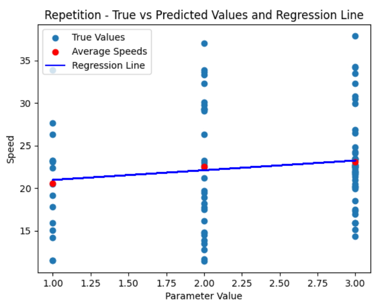
	R2 Score	Coefficient
BatchSize	0.199099	8.020421
BatchNumber	0.364004	36.148907
Repetition	0.002645	13.865529
DatasetSize	0.505498	1.148815

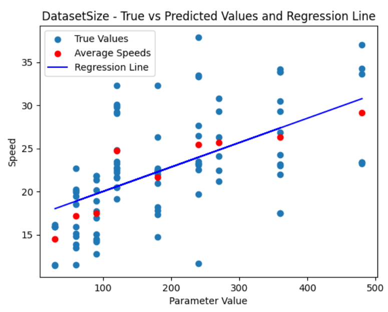
TFRecord Files

<u>Image Files</u>

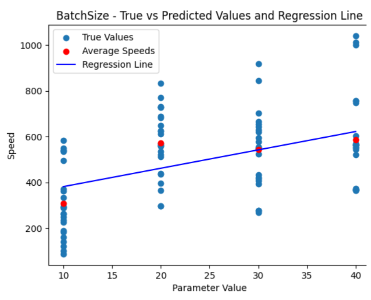


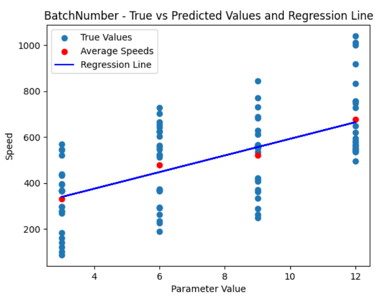


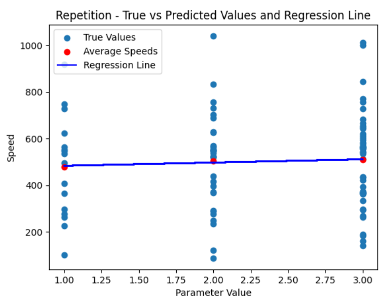


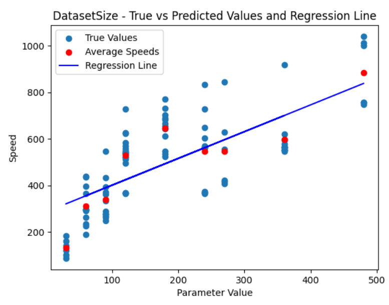


<u>TfRecord Files</u>









Interpretation:

In the regression analysis, Batch Size, Batch Number, and Dataset Size exhibit positive slopes, suggesting a direct relationship with reading speed. Conversely, Repetitions display a slope close to zero, indicating minimal impact on speed. These findings hold true for both image dataset files and TFRecord files.

R2 Score: R2 Score indicates predictability of speed from parameters. Scores vary, with TFRecord files generally more predictable.

For reading from image files, the R2 scores indicates that the regression models explain between 1.8% to 31.7% of the variance in speed. For reading from TFRecord files, the R2 scores indicates that the regression models explain between 0.3% to 50.5% of the variance in speed.

Coefficient: The coefficient represents the slope of the regression line and indicates the change in the speed for a unit change in the parameter value. In this case, the coefficients are relatively small, suggesting that the effect of each parameter on speed is modest.

TFRecord files show higher coefficients compared to image dataset, indicating that changes in parameters lead to more substantial speed changes in TFRecord files than in image files.

The regression results from the analysis have significant implications for large-scale machine learning applications, especially in cloud environments where data may be stored in distant physical locations.

- **Data Format Choice**: The selection between image files and TFRecord files significantly impacts ML performance. Higher TFRecord coefficient values suggest superior efficiency.
- **Data Transfer Latency**: Cloud data storage may introduce latency due to the physical distance between the client and the data center. For example, the round trip latency within the same data center is approximately 500,000 ns (0.5 ms), while sending a packet from California to the Netherlands and back takes 150,000,000 ns (150 ms). Therefore, minimising data transfer and access latency becomes crucial, especially for large-scale machine learning tasks where data is frequently accessed and processed.
- **Optimising Data Processing Pipelines**: Insights from regression analysis help optimise data processing pipelines. Developers can adjust parameters which enhances processing speed, minimises latency, and maximises throughput.
- **Resource Allocation and Scaling**: Regression insights guide resource allocation. Understanding parameter impacts optimises resource utilisation and workflow efficiency.

In a cloud environment, distributed storage, scalability, and network latency drive significant differences in behaviour compared to a single machine:

- **Distributed Storage**: Data is spread across multiple disks and servers, enabling parallel access for higher throughput.
- **Scalability**: Clouds allow dynamic resource allocation, facilitating parallel processing across nodes for increased throughput and faster processing times.
- **Network Latency**: Despite advantages, data transfer between nodes incurs latency, affecting data access and processing speed. Cloud providers optimise network infrastructure to mitigate latency and improve overall throughput.

Cloud providers tie throughput to the capacity of disk resources to ensure efficient resource utilisation and performance optimisation. By provisioning disk resources based on workload requirements, providers can allocate sufficient storage capacity to handle data-intensive tasks effectively. Additionally, tying throughput to disk capacity allows providers to manage resource allocation and scaling more effectively, ensuring that users have access to the necessary resources to meet their performance needs.

When conducting speed tests in parallel on the cloud, several considerations are crucial to accurately identify bottlenecks and optimise performance:

- **Request Rate**: Gradually ramping up request rates helps avoid issues and ensures smooth operation. Exponential backoff for specific response codes enhances reliability and resilience.
- **Access Distribution**: Randomising access patterns, especially after a common prefix, is key to effective scaling. Reordering bulk operations helps evenly distribute the load across key ranges, preventing hotspots and improving overall performance.
- **Naming Convention**: Avoiding sequential names and introducing randomness to object names can enhance load distribution and optimise performance. This prevents contention for resources and ensures efficient utilisation of cloud storage infrastructure.

Relating these considerations to the speed test results, we can infer that parallelisation may help identify potential bottlenecks in cloud storage performance. By varying parameters such as request rate, access patterns, and naming conventions, we can assess how these factors impact data access speed and overall throughput. Additionally, understanding the guidelines for optimising cloud storage performance allows us to interpret the results more effectively and make informed decisions for optimising large-scale machine learning applications in cloud environments.

Linear modelling offers insights into relationships between parameters and effects, assuming linearity and additive effects. However, real-world scenarios, like large-scale machine learning in the cloud, may not always adhere to these assumptions. Despite this, linear models serve as valuable approximations, aiding in trend identification and factor importance assessment. Theoretical considerations highlight assumptions of linearity and additivity. In practice, linear models offer coefficients and R-squared values for interpretation, but they may fall short in capturing nonlinearities. Therefore, complementing linear models with advanced techniques like polynomial regression or machine learning algorithms is advisable for capturing complex relationships effectively.

TASK 3a

The concepts and techniques discussed in the paper on CherryPick, which focuses on optimising cloud configurations for big data analytics, can be related to the coursework tasks on speed tests, cluster analysis, efficiency improvements, and regression analysis on reading speeds for image and TFRecord files.

Regression Analysis and Parameter Effects:

The regression analysis in the coursework aligns with the approach of building performance models to assess parameter impacts on job performance in the paper. Varying R2 scores and coefficients indicate different levels of explanatory power and parameter effects, akin to CherryPick's differentiation between optimal and near-optimal configurations based on performance models.

Optimising Machine Learning Performance:

Just as the results and analysis of this coursework highlights the importance of data format, network latency, and resource allocation in optimising machine learning performance, CherryPick emphasises the significance of selecting the right cloud configuration to enhance performance and reduce costs.

The considerations of data transfer latency and optimisation of data processing pipelines in this coursework tasks resonate with CherryPick's goal of minimising search costs and improving efficiency in cloud-based analytics jobs.

Real-World Complexity and Applicability:

While linear modelling offers insights into parameter effects, as observed in this coursework, the applicability of such models may be limited by real-world complexities. Similarly, CherryPick acknowledges the challenges of non-linear performance models and the variability in cloud environments.

TASK 3b

For different application scenarios such as batch processing and stream processing in cloud environments, the following concrete strategies can be defined to optimise performance and resource utilisation, considering the concepts discussed in the CherryPick paper:

Batch Processing:

Strategy- Cost-Performance Tradeoff: Applying Bayesian Optimisation techniques to identify near-optimal cloud configurations that balance cost and performance for batch processing tasks. Considering factors like data format, processing pipeline efficiency, and network latency to minimise cloud usage costs while ensuring job completion within acceptable timeframes.

Stream Processing:

Strategy- Real-Time Resource Allocation: Implementing dynamic resource allocation strategies for stream processing applications based on workload fluctuations and performance requirements. CherryPick's adaptivity in selecting cloud configurations can be leveraged to optimise resource utilisation for real-time data processing.

General Relationship with the Concepts:

- Performance Modelling: Both CherryPick paper and this coursework stress on the importance of accurate performance models in optimising cloud configurations for big data analytics tasks. The regression analysis in this coursework delves into the impact of various parameters on reading speeds, aiding in the development of predictive models for batch and stream processing tasks.
- Adaptivity and Optimisation: The paper CherryPick advocates for adaptively selecting optimal cloud configurations to minimise costs and ensure application performance. This coursework tasks, such as speed tests and efficiency improvements, offer insights into dynamically adjusting resource allocations based on workload characteristics. Leveraging strategies from this coursework enables real-time fine-tuning of cloud configurations to enhance task efficiency.
- Complexity Considerations: The paper CherryPick and this coursework recognise the complexities inherent in optimising cloud performance. The paper addresses challenges like non-linear performance impacts and variability in configurations, while this coursework explores techniques like cluster analysis and latency reduction. By considering data formats, network latencies, and resource utilisation, tailored strategies can mitigate bottlenecks and improve overall workflow efficiency.

- Cost-Performance Tradeoff: The CherryPick paper underscores the importance of balancing cost and performance in cloud configurations for big data analytics. This coursework tasks on efficiency improvements and regression analysis provide insights into optimising resource allocations for better performance without excessive costs.

WORD COUNT - 1999

REFERENCES -

[1] Omid Alipourfard, et al. 2017. Cherrypick: adaptively unearthing the best cloud configurations for big data analytics.