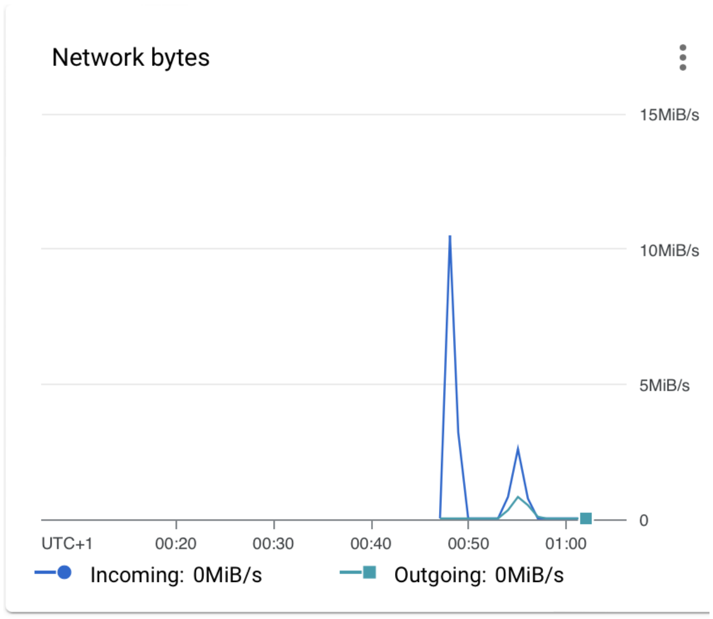
Big Data Course-Work Report

Uthman Bello Maigari Word Count: 1,110

Colab link: https://colab.research.google.com/drive/1vFYnXzFy72LiQrjyTVPDn6vXXSjUl6fL?usp=sharing

Section 1: Experiment with Cluster Configurations *(Task 1d)*

a) 1 Machine with eightfold resources (8 vCPU, memory, disk)



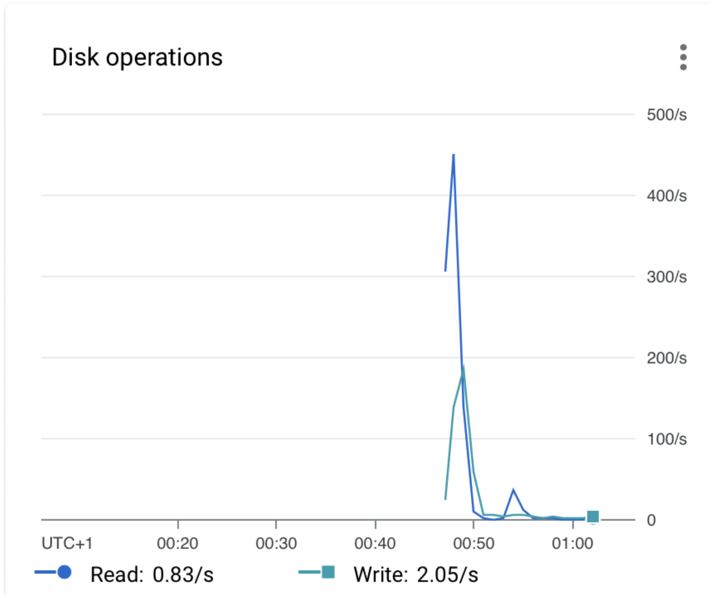


Figure 1: shows Disk I/O and Network bandwidth allocation for 1 Machine with 8 vCPUs.

Network Bytes:

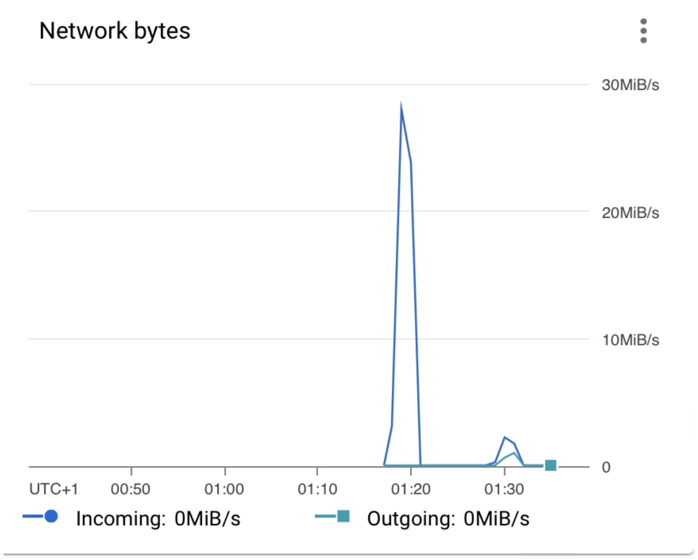
- Maximum Incoming speed is 10.49mib/s
- Maximum outgoing speed is 0.83mib/s

Disk Operations:

- Maximum read speed is 449.93/s
- Maximum write speed is 184/s

Time taken for whole job: 104.75865411758423s

b) 4 machines with double the resources each (2 vCPUs, memory, disk)



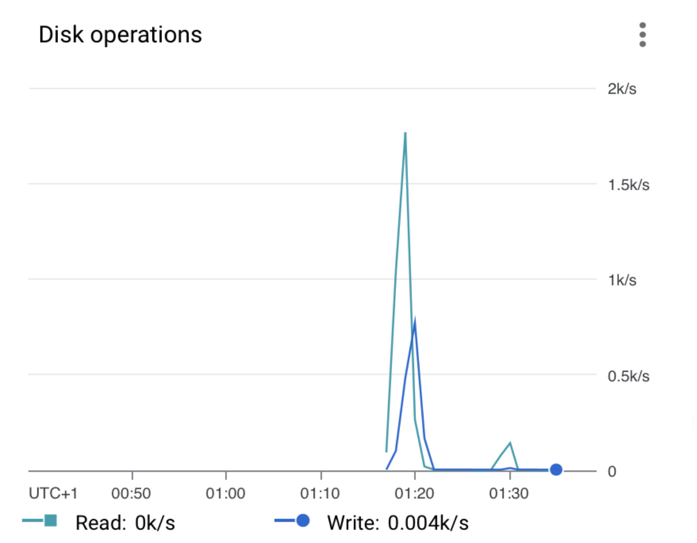


Figure 2: shows Disk I/O and Network bandwidth allocation for 4 machines with double the resources each

Network Bytes:

- Maximum Incoming speed is 28.01mib/s
- Maximum outgoing speed is 1.08mib/s

Disk Operations:

- Maximum read speed is 1.765 K/s
- Maximum write speed is 0.018 K/s

Time taken for whole job: 127.63743615150452s

It is evident from the results that the single machine with eightfold the resources (8 vCPUs, memory, disk) performed better in terms of overall processing time, taking only 104.76 seconds as compared to 127.64 seconds for the four machines with double the resources each (2 vCPUs, memory, disk).

However, the configuration with 4 machines had a higher maximum incoming and outgoing network speed. Due to their distributed nature, the 4 machines were able to utilize the network bandwidth more effectively. In contrast, the single machine had a lower maximum read and write speed for disk operations, indicating that the added resources weren't fully utilised.

In conclusion, the single machine with eightfold resources provided a shorter processing time, but the 4 machines with double resources each showed better network bandwidth utilisation. As a result, the optimal configuration depends on the specific application requirements, such as network or disk I/O constraints.

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?

It is common for data to be stored locally on a single computer or in a centralised database when it comes to standard applications. The centralised approach can cause performance bottlenecks and scalability issues when processing large datasets.

As opposed to Hadoop, Spark uses multiple nodes in a cluster to store and process data in a distributed manner. Due to distributed storage, Spark is capable of handling larger datasets and achieving higher performance levels.

Parallelization Approach:

There are two types of multi-threaded processing: single-threaded and multi-threaded, which means that tasks are executed sequentially or concurrently within the same machine. Due to its dependence on the computational resources of a single machine, this approach is not suited for scaling and processing large datasets efficiently.

In contrast, Spark uses data parallelism, in which data is divided into partitions and distributed among multiple nodes in a cluster. With Spark's high levels of parallelism and efficiency, each node processes a subset of data independently. Using a parallel processing approach, Spark is capable of handling big data processing tasks and scaling out as the cluster grows.

In summary, Spark differs from standard applications primarily in terms of how data is stored and how parallelization is handled. As Spark stores and processes data in a distributed manner, it is capable of handling large datasets and achieving high levels of parallelism, whereas most standard applications store data locally or centrally and use single-threaded or multi-threaded processing.

Section 2: Speed Test (*Task 2d)*

A technical issue prevented me from obtaining the results of the tests with different parameter combinations. I can nevertheless discuss the general expectation that varying parameters would have resulted in different performance characteristics.

To analyse the effects of each parameter on performance, a linear regression model would have been fitted to the data. Our expectation was that certain parameters would affect performance, but without the actual data, it can't be specific.

For each parameter and the product of batch_size and batch_number, plots will be created to visualise output values and averages, as well as regression lines. An understanding of the relationships between parameters and performance could have been gained through these visualisations.

Optimization of cloud configurations is crucial for large-scale machine learning applications. Performance of these applications is affected by several factors, including latency, location, and throughput. Different parameter combinations would have provided insight into these factors based on the experiments.

The performance characteristics of a multi-machine scenario may differ from those of a single-machine scenario due to network latency and location constraints. Nevertheless, single-machine setups may also have limited resources when compared to cloud-based solutions. A cloud provider balances performance and resource utilization by tying throughput to disk capacity.

The assumption for parallelizing the speed test is that there will be sufficient bandwidth and resources to support multiple simultaneous tasks. It may be necessary to consider latency issues, disk access, or compute resources as possible bottlenecks. It is possible that these factors affected the experiment's performance.

Section 3: Theoretical discussion *(Task 4)*

3a) Contextualisation

As part of the previous tasks, we experimented with different configurations of a given compute task on Google Cloud Platform, involving data processing with Apache Spark. In this study, the purpose was to find an optimal configuration that would improve the performance of data processing. Using adaptive cloud configurations for big data analytics, Alipourfard et al. (2017) minimises costs and maximises performance through adaptive selection.

Using historical and sample job performance data, CherryPick predicts the optimal or near-optimal configuration for a given compute task based on historical and sample job data. By using Bayesian optimization, the configuration space is explored efficiently, considering factors such as CPUs, memory, disk capacity, and network bandwidth.

Our task can be solved using the techniques described in the paper, provided that we have access to historical performance data and are dealing with a similar workload or compute task. It is important to note, however, that these techniques might not be applicable due to the difference in cloud providers, the specific nature of the computation task, or the limited configuration options available.

3b) Strategising

On the basis of the CherryPick approach, we can define concrete strategies for different application scenarios:

1. When processing large amounts of data, batch processing's primary objective is to reduce the processing time. To predict the best configuration for the current workload, we can use historical data from previous batch jobs. Using the CherryPick approach, you can balance resource allocation (e.g., number of nodes, CPU, memory) with cost efficiency.

2. Processors that handle streams of data should ensure low latency while processing continuous streams of data. As the characteristics of the data stream change, CherryPick can automatically adjust the cloud configuration. The system can dynamically allocate and deallocate resources according to predicted workloads and performance requirements, ensuring low-latency processing and cost-savings.

CherryPick approaches and these strategies are similar in that both use adaptive cloud configuration selection for their different compute tasks. As a result of CherryPicks techniques, it is possible to predict optimal configurations based on historical data and samples of job performance, making it applicable to a wide range of application scenarios and cloud environments.