Performance evaluation of Terapixel rendering in Cloud (Super)computing

Dimitrios Poulimenos - 200291237

Semester 1 - 2023/24

Introduction

Cloud computing is everywhere nowadays, and a need for the visualization of real-time data as well. By collecting the gathered data from Newcastle Urban Observatory, Newcastle University trying to create a system which presents those data in more detail. More specifically, by processing these data about different places in Newcastle City and its surroundings, Newcastle University has created a system that processes the data and creates super-detailed pictures called terapixels. Their solution involves using cloud computing, and also they built a system that can scale up or down based on how much power they need at any given time. The main goals of their work were to set up a powerful computer system in the cloud, make a detailed 3D visualization of the city that gets updated every day, and test how well cloud computing works for this kind of detailed visualization. Moreover, they showed that it is possible to create these high-quality images quickly using cloud-based graphics processing units (GPUs).

1. Business Understanding (*need for the Project*)

Understanding the business is the most crucial step to begin with. The project addresses the need for efficient and scalable visualization of terapixel images for the city of Newcastle upon Tyne, leveraging cloud-based resources. The increasing volume of data captured by the Newcastle Urban Observatory demands a computational infrastructure that can handle the complexity of rendering large-scale visualizations. Therefore, conducting a thorough and systematic evaluation of the performance of cloud supercomputing systems becomes crucial for guiding the development, selection, and effective utilization of computing resources (Holliman, Nicolas S. et al., 2019). Subsequently, the establishment of solid objectives and success criteria will enable this report to achieve its goal. Finally, in this report, I am going to explore the capabilities of the hardware and its limitations providing a clear answer for every objective established below.

1.1 Objectives

Here is where I am addressing the objectives of my analysis in order to have a more comprehensive picture of what this analysis tries to answer. The questions that I will investigate in this project are the following:

Objective 1

- Which event types dominate task runtimes?

Objective 2

- What is the interplay between GPU temperature and performance and which event places the greatest stress on the system?

Objective 3

- Are there specific GPU cards that consistently show better or worse performance based on their serial numbers?

By following the above order I make sure that my analysis builds upon foundational insights, leading to a comprehensive understanding of GPU behavior in the context of terapixel image rendering.

1.2 Success Criteria

By defining the objectives of the analysis it is time to outline the success criteria about these objectives.

Success criteria for objective 1

- Visualize task runtimes for each event type, using appropriate graphical representations to identify the distribution and central tendency of runtimes.
- Quantify and compare the median or mean runtimes of different event types, highlighting those with consistently higher runtimes.

Success criteria for objective 2

- Establish a clear correlation or relationship between GPU temperature and performance metrics, considering factors such as GPU utilization and power draw. Visualization should highlight trends and patterns in their interplay.
- Quantify system stress during different events by analyzing GPU utilization, power draw, and other relevant metrics. Identify the specific event that consistently exhibits the highest combination of GPU temperature and performance metrics, indicating the event that places the greatest stress on the system.

Success criteria for objective 3

- Profile GPU performance based on serial numbers to compare key metrics.
- Identify specific GPU cards that consistently exhibit superior or inferior performance.
- Detect GPU cards with perpetually slower performance by comparing performance metrics.

1.3 Project Justification and Work Plan

To ensure a precise and efficient investigation into the aforementioned queries, I have chosen to adhere to a structured methodology. Initially, I will conduct a thorough examination of the three data files. Subsequently, I will meticulously clean the data, eliminating any empty cells, resolving inconsistencies, and eliminating duplicates. Following the cleansing process, I intend to optimize the dataset for enhanced productivity. This involves extracting essential information and merging relevant data to construct a refined dataset according to the investigation questions, primed for in-depth analysis.

It is worth to mention that the above described methodologies and the analysis presented in this report is inspired and follows the Cross Industry Standard Process for Data Mining (CRISP-DM)(Chapman, P. (2000)), a comprehensive framework that facilitates data-driven decision-making. To derive meaningful insights, this report conducts one cycle of the CRISP-DM model.

2. Data Understanding (*What I did - Part 1*)

In this phase of the Crisp-dm cycle starting with the data, we should be able to understand the shape, the size, and the values included in them in order for us to get a more comprehensive overview of the relationship between the data and our research questions. As I mentioned above, there are three data files from which we will derive our answers.

2.1 Data Colection

Initially, the data were gathered from the Newcastle Urban Observatory and included data from Newcastle City and surrounding areas. These data have once again been given to Newcastle University in order to create the cloud computing system that I am analyzing in this report. Therefore, Newcastle University gave me access to the data to conduct my analysis. The data are structured in three CSV files, namely application-checkpoints.csv, gpu.csv, and task-x-y.csv.

2.2 Exploring the data

Now to the exploration part of this phase I will display the structure of each data file in order to decide which files I am going to use for the next phase which is data preparation.

1. application-checkpoints.csv

```
## 'data.frame':    660400 obs. of  6 variables:
## $ timestamp: POSIXct, format: "2018-11-08 07:41:55" "2018-11-08 07:42:29" ...
## $ hostname : chr "0d56a730076643d585f77e00d2d8521a00000N"
##    "0d56a730076643d585f77e00d2d8521a00000N" ...
## $ eventName: chr "Tiling" "Saving Config" ...
## $ eventType: chr "STOP" "START" ...
## $ jobId : chr "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705"
##    "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705" ...
## $ taskId : chr "b47f0263-ba1c-48a7-8d29-4bf021b72043"
##    "20fb9fcf-a927-4a4b-a64c-70258b66b42d" ...
```

This dataset appears to contain details about rendering events, including timestamps, hostnames, event names, event types, and associated job and task IDs. In order to fulfill the success criteria of my first objective, this dataset provides me everything I need to to fulfill the success criteria of it.

2. gpu.csv

```
## 'data.frame':    1543681 obs. of  9 variables:
## $ timestamp : POSIXct, format: "2018-11-08 08:27:10" "2018-11-08 08:27:10" ...
## $ hostname : chr "8b6a0eebc87b4cb2b0539e81075191b900001C"
##    "d8241877cd994572b46c861e5d144c85000000" ...
## $ gpuSerial : num 3.23e+11 3.24e+11 ...
## $ gpuUUID : chr "GPU-1d1602dc-f615-a7c7-ab53-fb4a7a479534"
##    "GPU-04a2dea7-f4f1-12d0-b94d-996446746e6f" ...
## $ powerDrawWatt : num 132 117 ...
## $ gpuTempC : int 48 40 45 38 41 ...
## $ gpuUtilPerc : int 92 92 91 90 90 ...
## $ gpuMemUtilPerc: int 53 48 44 43 47 ...
## $ Performance : num 191 164 ...
```

This dataset contains information about GPU metrics, including timestamps, GPU temperature, and other relevant details, and therefore in combination with the first dataset provides the essential information needed to investigate the interplay between GPU temperature and performance and detect which event stresses the system the most. On the other hand, it also provides the information I need to address my third objective.

3. task-x-y.csv

```
## 'data.frame':    65793 obs. of  5 variables:
## $ taskId: chr "00004e77-304c-4fbd-88a1-1346ef947567"
##    "0002afb5-d05e-4da9-bd53-7b6dc19ea6d4" ...
## $ jobId : chr "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705"
##    "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705" ...
## $ x : int 116 142 142 235 171 ...
## $ y : int 178 190 86 11 53 ...
## $ level : int 12 12 12 12 12 ...
```

While the last dataset at hand doesn't contribute directly to my current objectives, it contains valuable information that could prove beneficial for addressing potential future objectives. Therefore, although it won't be included in the current analysis, its content holds promise for addressing different goals down the line.

Overall, the initial two datasets will serve as the primary sources for subsequent preparation and analysis. These datasets include the necessary information needed to address all the established objectives in my analysis.

3. Data Preparation (*What I did - Part 2*)

The initial phase of the analysis, centered around preparing the chosen datasets, stands as a pivotal step encompassing crucial tasks such as data cleaning, exploration, transformation, and feature engineering. In the next part, I will explain in detail what I did to get the datasets ready for exploratory data analysis.

3.1 Data Wragnling

In order to start working with both datasets first I had to change the format of the "timestamp" column. This will ensure a column "write something here". The next step was to start working with the data based on the objective that I had to answer.

Objective 1

Initially, to address the first objective of determining the dominant event in task runtimes, I have created a function. This function selectively filters rows based on the specified event name and type. It subsequently constructs a dataframe, incorporating both 'START' and 'STOP' eventType entries, organized by 'taskId'. The newly introduced 'Runtime' column captures the time difference between the start and stop of each task. Moreover, given the absence of missing data I had to remove duplicates in order to ensure data integrity. And finally, for enhanced clarity, I have changed the name of certain columns, and also I have combined the results and I merged these five distinct dataframes into a unified final dataframe to achieve the desired result.

Objective 2

For the second objective, focusing on the relationship between temperature and system performance, and identifying the event exerting the most stress on the system, I adopted a different approach. Initially, I renamed the timestamp column to "time_start" in order to join the "gpu" dataset and the final dataset from the previous step. This step facilitated the extraction of GPU performance and temperature values corresponding to each event. To explore the overall interplay between performance and temperature, I filtered

unique temperature values and computed the mean of gpuUtilPerc and gpuMemUtilPerc values, generating a dedicated dataframe for these aggregated values. Similarly, I pursued this process for each specific event, resulting in distinct datasets tailored for individual event analyses.

Objective 3

To achieve my last objective of identifying the fastest and slowest GPUs based on their serial numbers, I refined my approach, particularly in light of the second objective related to GPU performance. Combining key metrics such as "gpuUtilPerc," "gpuMemUtilPerc," and "powerDrawWatt," I introduced a unified metric, "performance," calculated as follows: Performance = ((gpuUtilPerc * powerDrawWatt) + (gpuMemUtilPerc * powerDrawWatt) / 100). This comprehensive metric incorporated multiple aspects of performance, providing a holistic basis for comparing GPUs by serial number.

Subsequently, I created a new dataframe containing the mean performance values for each unique serial number. This mean performance dataframe served as the foundation for identifying the fastest and slowest GPUs. By extracting the top and bottom three serial numbers based on their mean performance, I obtained two distinct dataframes. Finally, I have transformed the timestamps of these two new dataframes into minutes and grouped the data to their minute and gpuSerial for visualisation purposes. These dataframes encapsulated all relevant metrics for each GPU in their respective categories (fast/slow). All the above steps enabled me to get the most out of the performance of the GPU cards based on their serial numbers in order to answer my question.

4. Modelling

After carefully formatting the data to align with the defined objectives, the current phase of the CRISP-DM model involves conducting Exploratory Data Analysis (EDA). The main objective is to integrate the data seamlessly with the success criteria linked to our goals. In the following section, I will delve into a comprehensive examination of the data, aiming to extract valuable insights that significantly contribute to our overarching objectives.

4.1 Exploratory Data Analysis

Which event types dominate task runtimes?

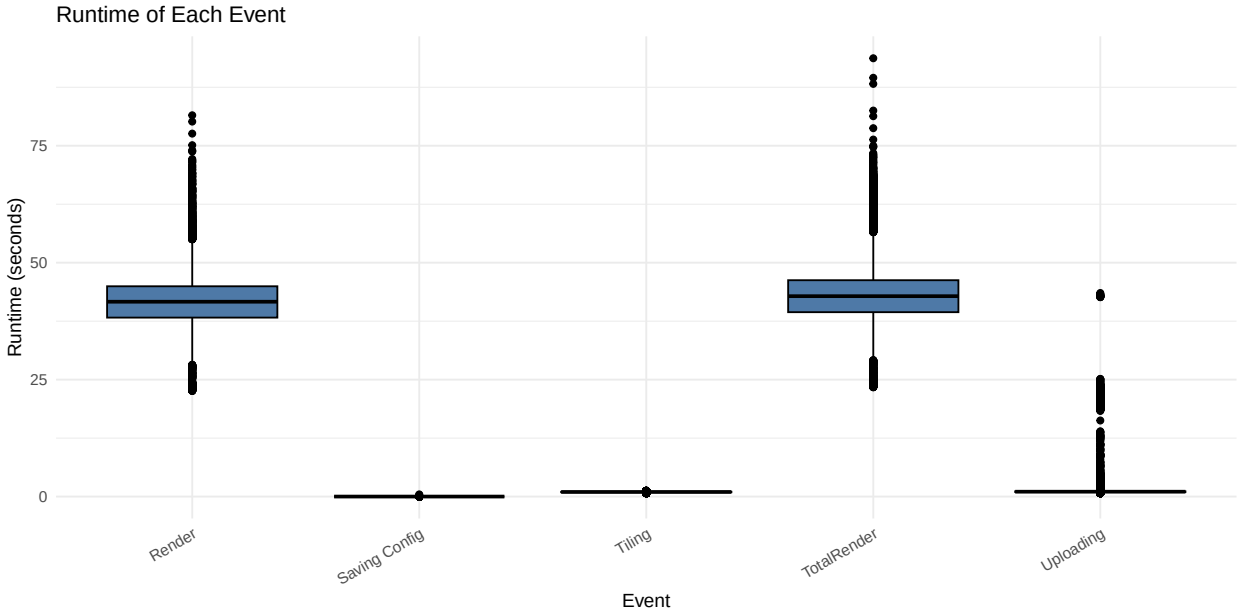
To address this inquiry, a boxplot has been generated to visualize the distribution of runtimes per event and identify potential outliers. The analysis reveals that the predominant event is "totalRender," representing the cumulative runtime of all events. Subsequently, the "Render" event emerges as the second most time-consuming, surpassing 40 seconds in duration. These observations are succinctly summarized below:

Summary Statistics for Runtimes:

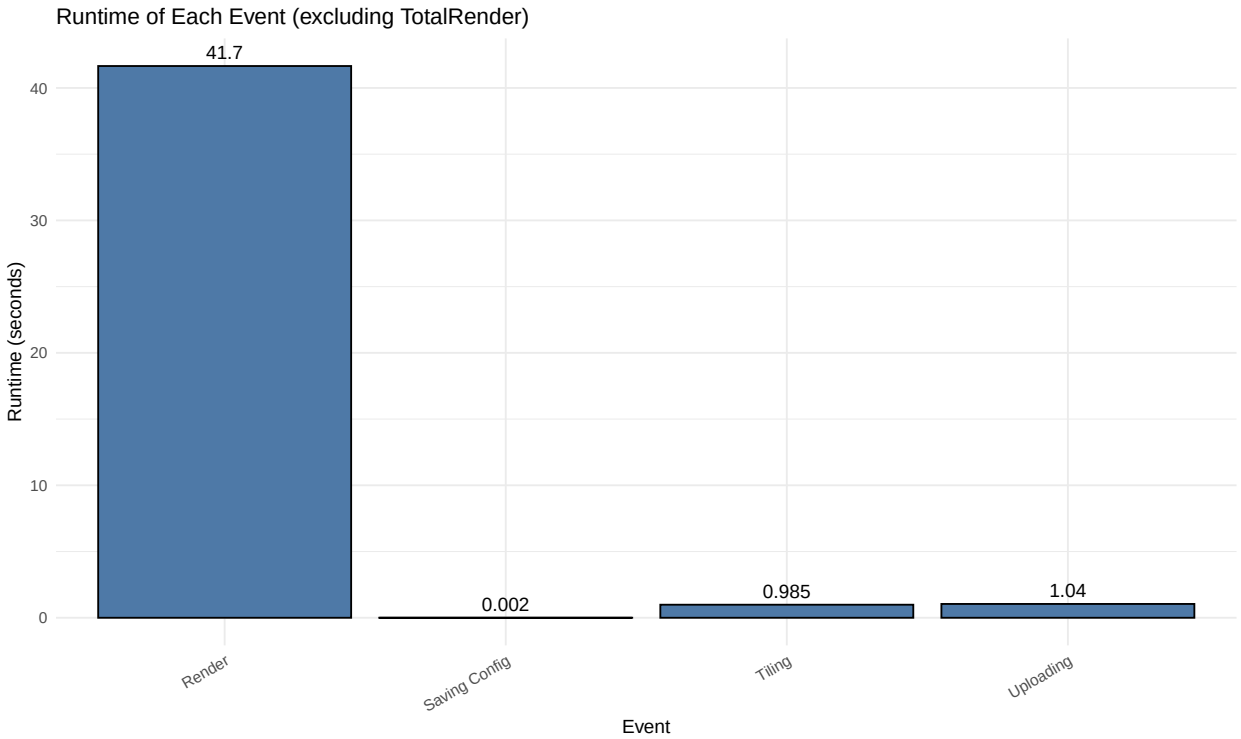
```
## Mean: 17.236

## Median: 1.066

## Standard Deviation: 20.572
```

Quantiles:

```
## Q1 (25th percentile): 0.902

## Q2 (50th percentile - Median): 1.066

## Q3 (75th percentile): 40.736
```

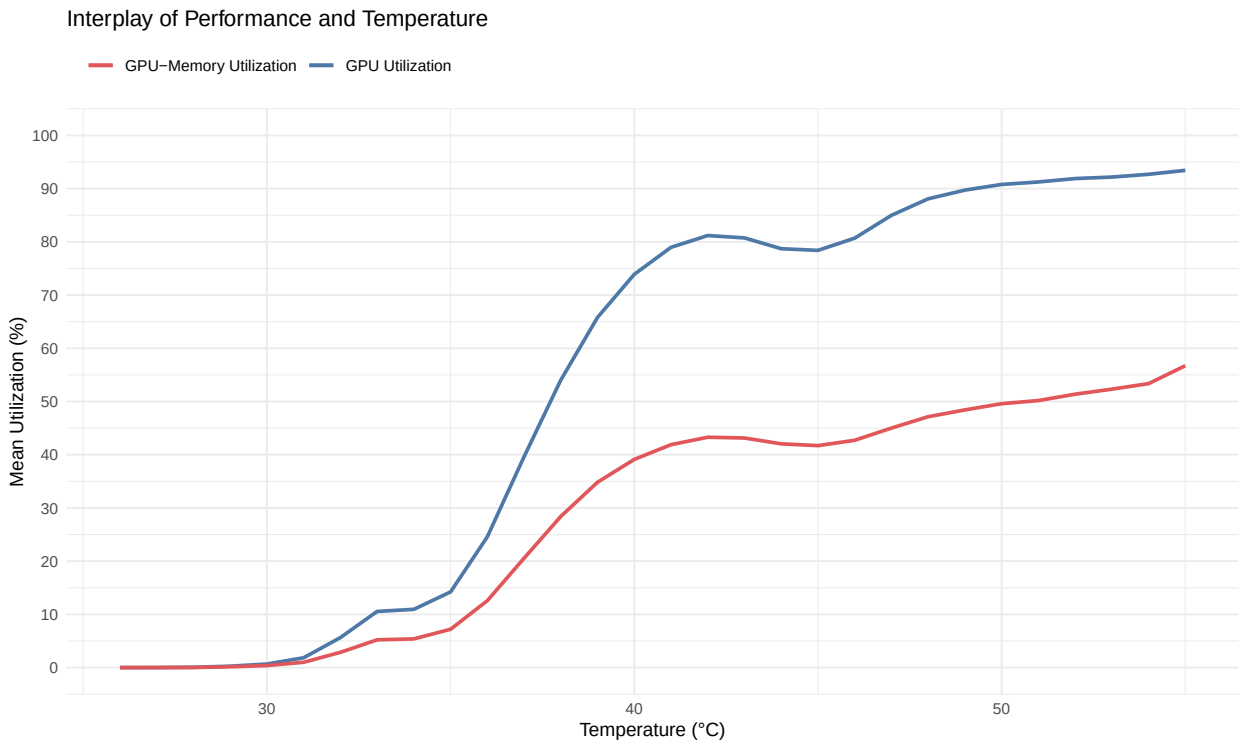


Recognizing that the "totalRender" event does not provide a solution to our objective, I have generated a barplot, excluding this event to unveil the authentic dominant event concerning runtimes. As indicated in the preceding boxplot, the "Render" event emerges as the foremost contributor to task runtimes. This observation is succinctly conveyed as follows:



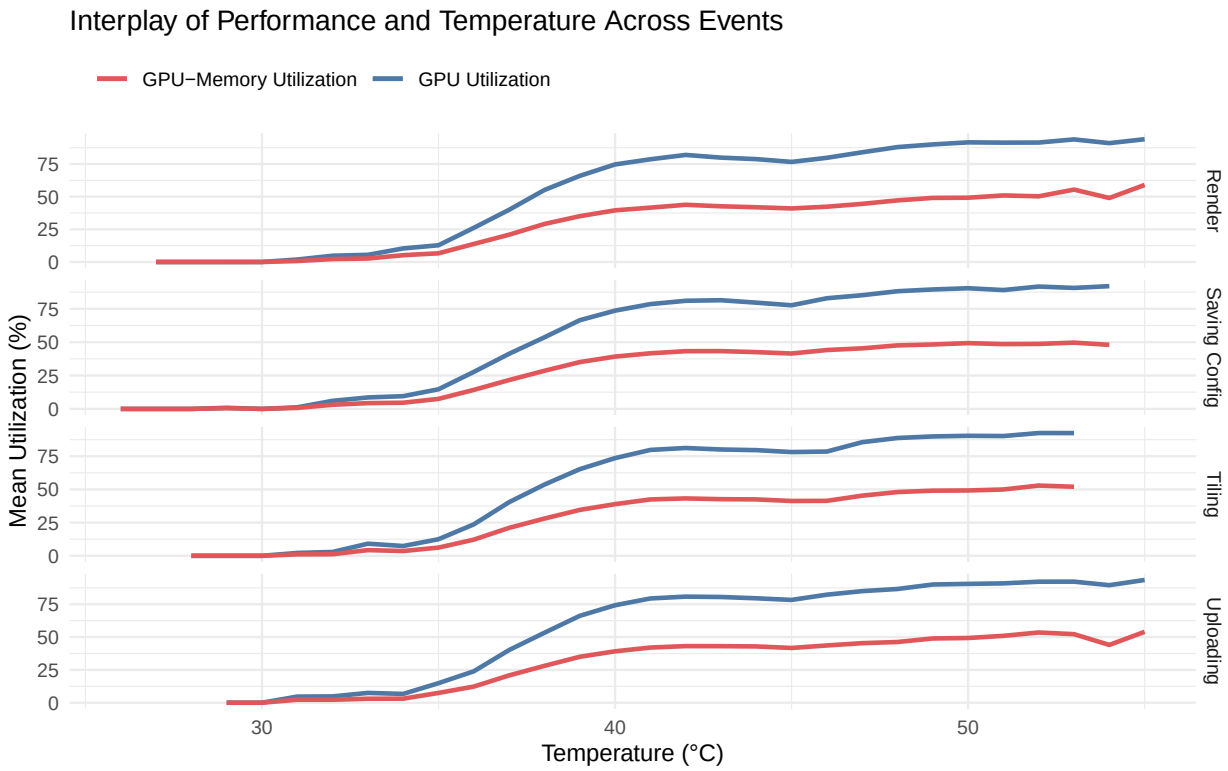
What is the interplay between GPU temperature and performance and which event places the greatest stress on the system?

To address this objective, I have chosen to employ a line plot to illustrate the relationship between performance and temperature. The graphical representation indicates a notable trend wherein the performance of GPU cards exhibits an upward trajectory when the temperature surpasses 30°C. The most pronounced increase in performance is observed within the temperature range of 35 to 40°C, sustaining a steady level until the temperature reaches 45°C. Subsequently, there is a discernible resurgence in performance, peaking again as the temperature ascends to 55°C.



We need to acknowledge the mechanisms that limit the performance from increasing between 40 and 45°C. Potential reasons for this plateau may include the optimal operating conditions within this temperature range, efficient thermal regulation mechanisms, stability in processing demands, potential constraints imposed by other system components (such as power or cooling limitations), and architectural characteristics inherent to the GPU design.

Regarding the second subquestion for this objective, I have generated individual line plots for each event. These plots reveal a consistent behavior across events concerning performance, with noteworthy distinctions observed in the Render and Uploading events. Both events demonstrate an increased power requirement as temperatures rise, with the Render event exhibiting slightly higher power demands. Consequently, it is evident that the Render event imposes the greatest stress on the system, as illustrated in the subsequent visualization:



Are there specific GPU cards that consistently show better or worse performance based on their serial numbers?

In order to address our last objective I had to create two visuals. First I had to identify which are the top three GPU cards based on their serial numbers and which are the worst three according to their mean performance in the timeframe the dataset provided.

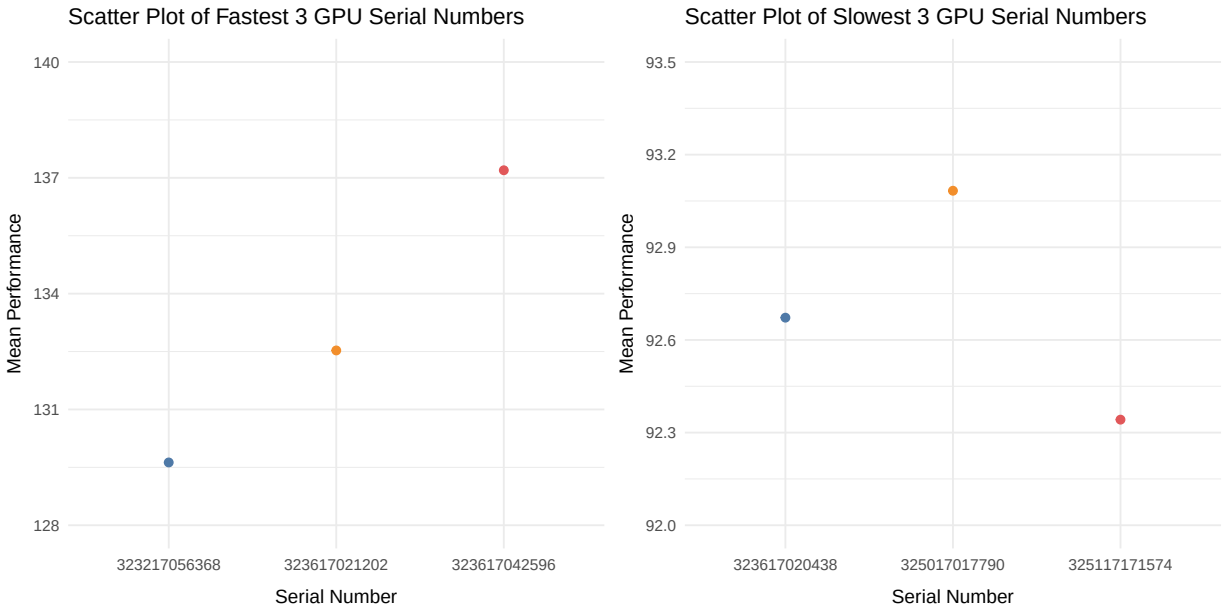
The second visual was on how the three top and worst serial numbers performed throughout this timeframe to indicate the consistency of their performance and answer the objective's question. I also considered the standard deviations of each serial number to prove the consistency of their performance.

Top 3 GPU serial numbers:

```
## # A tibble: 3 x 3
##      gpuSerial mean_Performance sd_Performance
##          <dbl>            <dbl>          <dbl>
## 1 323217056368             130.           89.3
## 2 323617021202             133.           87.6
## 3 323617042596             137.           91.8
```

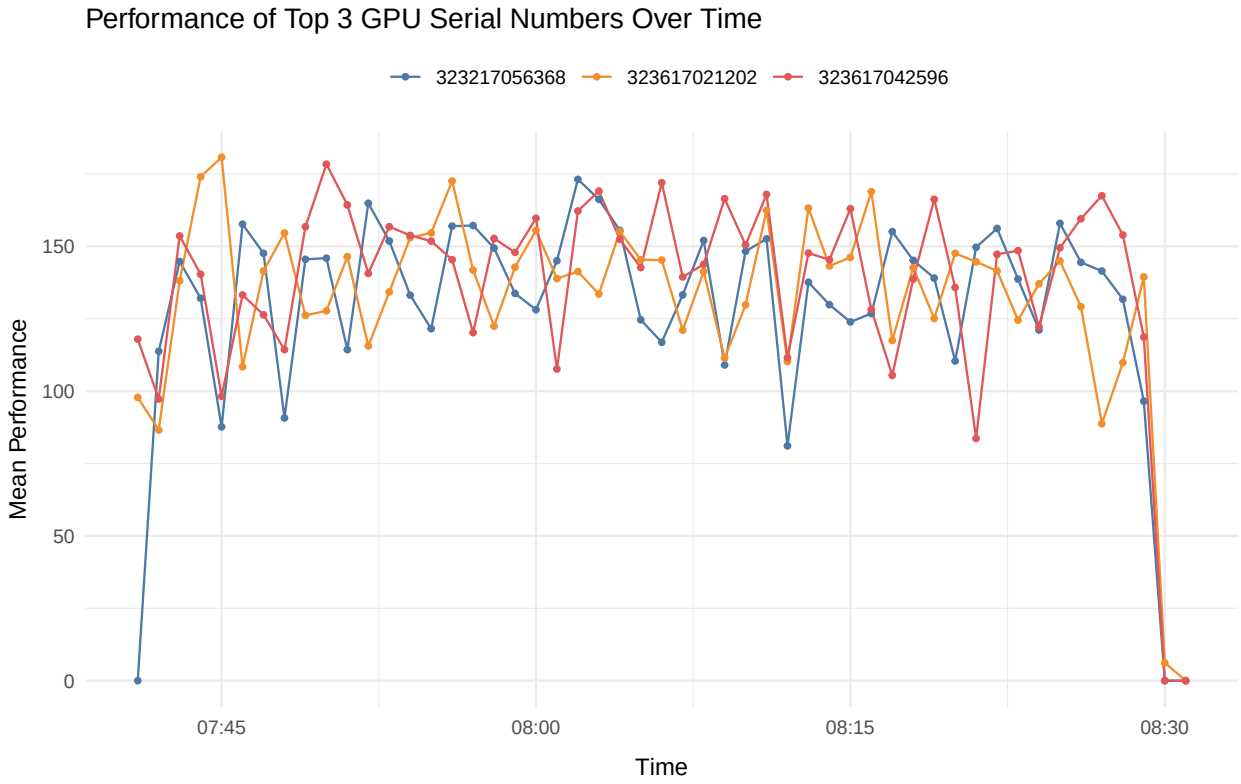
Worst 3 GPU serial numbers:

```
## # A tibble: 3 x 3
##      gpuSerial mean_Performance sd_Performance
##          <dbl>            <dbl>          <dbl>
## 1 323617020438             92.7           71.7
## 2 325017017790             93.1           70.0
## 3 325117171574             92.3           68.4
```

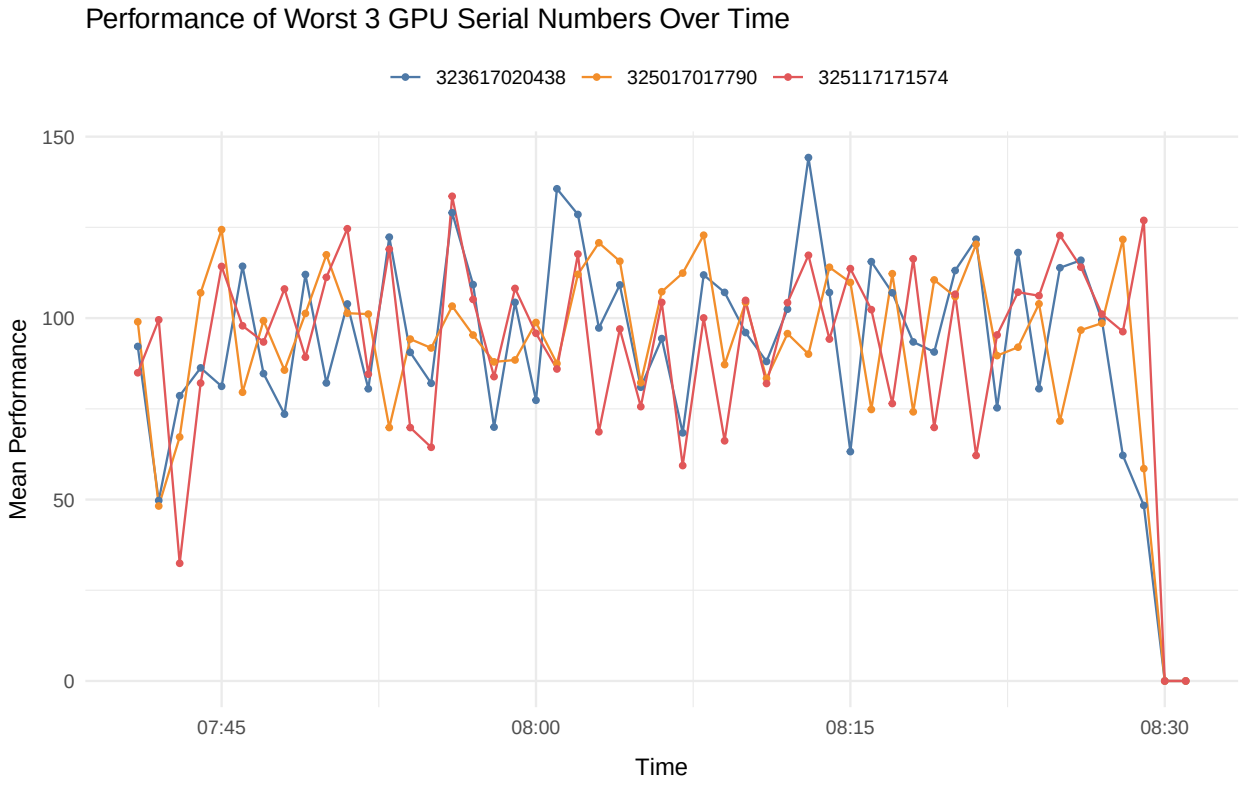


As can be seen from the above scatter plots, based on their mean performance the top three GPU cards are "323617042596", "323617021202", and "323217056368". The best overall mean performance belongs to the "323617042596" serial number with a metric a bit higher than 137.

On the other hand, the worst three GPU cards based on their serial number are the "3251171711574", "323617020438", and "325017017790" with the lowest mean performance of the serial number "3251171711574" being the slowest of the three with a metric value a bit above 92.3.



Analyzing the mean performance of the top three GPU serial numbers during the given timeframe reveals that the serial number "323617042596" registers the highest mean performance. However, when considering both mean performance and standard deviation, "323617021202" emerges as the superior choice. This is attributed to its impressive combination of high mean performance coupled with the lowest standard deviation among the three. Such a profile suggests that "323617021202" not only consistently delivers high performance but also maintains greater stability compared to the others.



Examining the mean performance of the worst three serial numbers throughout the same timeframe reveals a consistent trend that aligns with the mean performance values and also the standard deviation ones. Notably, the serial number "3251171711574" stands out for its consistently lower mean performance. Additionally, it exhibits the smallest standard deviation compared to the other two, indicating a pattern of stable yet subpar performance. This consistency in lower performance levels positions "3251171711574" as the most predictably underperforming GPU, characterized by its reliably low and stable performance metrics.

5. Evaluation

In approaching the evaluation phase of the CRISP-DM methodology, it is crucial to concentrate on three distinct aspects:

Objectives & Success Criteria

Firstly, the analysis should be assessed in terms of its alignment with the predetermined objectives and success criteria. The objectives outlined in the analysis report have been partially met, with some room for variation in approach. Specifically, while addressing the second objective, I utilized the data as presented in the GPU dataset. In contrast, for the third objective, I adopted a more innovative strategy by creating a new column that involves various provided metrics. This divergence in methods between objectives introduces a potential inconsistency in the analysis, a factor that could be perceived as a drawback in a report.

Nevertheless, the approach taken for the third objective is noteworthy. It integrates all the available metrics to establish a more comprehensive performance indicator. This holistic measure is particularly effective in addressing the third objective, as it encompasses a wider range of data points, thereby offering a more robust and insightful assessment of performance. Regarding the success criteria, I am confident that my approach aligns well with them, especially in the context of providing definitive answers to all the initial objectives as outlined at the beginning of the analysis.

CRISP-DM

Secondly, an evaluation of the implementation of the CRISP-DM methodology in this analysis is essential. The adapted methodology clearly outlines all the necessary steps for conducting a thorough analysis tailored to a specific business problem. However, it's important to highlight a key aspect of the approach: the entire analysis was structured within a single CRISP-DM cycle, encompassing all objectives.

An alternative approach could have been to structure the analysis across three separate CRISP-DM cycles, with each cycle devoted exclusively to addressing one of the objectives. This method might have allowed for a more focused and in-depth exploration of each objective. In concluding the evaluation of the methodology, I can affirm that the CRISP-DM framework was rigorously followed as outlined in the guiding literature, with all its steps being correctly and thoroughly implemented.

Code & Plots

Finally, our attention turns to assessing the interpretability and effectiveness of the code and plots created for presenting the results. While the code implementation has room for improvement, particularly in reducing redundancy, it effectively meets the project's needs. Additionally, there's potential for a more cohesive strategy in plot design and visualization. However, it's important to recognize that the code and plots effectively cover all the necessary aspects to address the objectives.

Notably, this analysis involved experimenting with various visualization techniques, and the chosen final results stand out as the most effective. This exploration and selection process highlights a commitment to finding the most suitable methods for data presentation, ensuring that the final visualizations are both informative and aligned with the project's goals.

6. Future Work

In terms of future work, it will be vital to explore various other aspects of the GPU cards provided, such as their architecture, memory capacity, and power efficiency. These factors are integral for a comprehensive evaluation of the cloud system's performance. Additionally, the existing analysis suggests the potential need for hardware upgrades to enhance system efficiency.

Notably, the observed correlation between GPU utilization and temperature rise merits attention. At a certain point, GPU utilization plateaus even as temperatures continue to increase. This phenomenon could indicate thermal throttling, where the GPU downclocks to prevent overheating, leading to a consistent utilization level despite rising temperatures. Addressing this could involve enhancing cooling solutions or optimizing workload distribution to maintain optimal performance while mitigating thermal challenges.Moreover, it's important to highlight the necessity for specialized GPU cards tailored to specific tasks. For instance, during rendering events, the system experiences considerable strain. To effectively manage such demanding processes, it would be advantageous to deploy additional GPU cards. This targeted approach not only alleviates the burden on the system but also optimizes performance for specific, resource-intensive events.

7. Self Reflection

Reflecting on my work and the journey through this analysis presented its own set of challenges. To effectively address the objectives, a thorough comprehension of the business context was crucial. Additionally, working with varied datasets posed difficulties, particularly in merging and manipulating different values or in creating new datasets tailored to my objectives.

This journey was both enlightening and somewhat familiar, as it paralleled the type of analysis and methodology I had engaged with in a previous university module. I gained new insights into cloud computing development and its evaluation from a data scientist's perspective, enhancing my understanding of the necessity for such implementations. Despite moments of disappointment when certain attempts didn't yield immediate success, persistent trial and error enabled me to successfully complete the task. This entire experience instilled in me a sense of accomplishment regarding my work, and I am enthusiastic about applying these newly acquired techniques and methodologies in future projects.