b8035526-report

Antreas Kasiotis

25/12/2021

Project Understanding

Background information

This is a project that concerns an effort that was made to develop a supercomputer architecture for scalable terapixel visualization by using the public cloud. The novelty of this work was that it proves the feasibility of producing a high quality terapixel visualization using a path tracing renderer in under a day by using public IaaS cloud GPU nodes.

Project Definition (*need for the project*)

For my extended project I will be investigating multiple datasets that were collected about the function of the system and try to find ways to address some of the issues that it is facing. The goal of this investigation will primarily be to delve into the matters of system and hardware performance and operation. Systems such as the TeraScope visualization require the use of systems that are far larger and more complex than the ordinary computer system. Therefore, an educated and well structured performance evaluation of cloud supercomputing systems is crucial in promoting the development, selection, introduction and effective utilization of a computer and it is of out-most importance to users, manufacturers and research institutions (ASC Community, 2018). To effectively approach this investigation I will take a closer look at the data to better understand the specific issues that I can address. However, it is already clear that the main performance indicators that could be inferred from the dataset are those that concern both the system's software but also those about the actual physical system components.

Data mining Objectives and Goals (*Problem definition*)

To better compartmentalize the problems that I want to address I have chosen to look into four very specific questions about performance that can be inferred from the dataset through data mining. The questions that I will be investigating as part of this project inculde the following:

1. Which event types dominate task runtimes?
2. What is the interplay between GPU temperature and performance?
3. Can we quantify the variation in computation requirements for particular tiles?
4. Can we identify particular GPU cards (based on their serial numbers) whose performance differs to other cards? (i.e. perpetually slow cards).

Solution elicitation and work plan

In order to ensure the productive and thorough investigation of the aforementioned question I have decided to carry out some due diligence prior to my investigation. The preliminary work that will be conducted includes; assessing and exploring the data towards its integrity and reliability in order to only keep what's most useful, engineering ways to explore the datasets in such a way that allows me to answer my questions by separating, merging and constructing new data out of the pre-existing ones.

In summary, after exploring and choosing my data, I will clean that data, construct new records of it, merge it with other pre-existing data and reformat it wherever necessary so that it is ready for analysis. The above described methodologies of handling a data mining project such as this one have primarily been inspired by the CRISP-DM methodology which sets out a very clear plan for data science project life-cycles that minimize the room for error and inconsistencies (IBM, 2018).

Solution Implementation (*What I did*)

Data Exploration

To better understand the task at hand I went ahead a took a look inside the datasets to see what kind of information I had to work with. In total I was given three datasets.

1st Dataset

The first one held information about the application checkpoint events throughout the execution of the render job.

```
## tibble [660,400 x 6] (S3: tbl_df/tbl/data.frame)
## $ timestamp: chr [1:660400] "2018-11-08T07:41:55.921Z"
##    "2018-11-08T07:42:29.842Z" ...
## $ hostname : chr [1:660400] "0d56a730076643d585f77e00d2d8521a00000N"
##    "0d56a730076643d585f77e00d2d8521a00000N" ...
## $ eventName: chr [1:660400] "Tiling" "Saving Config" ...
## $ eventType: chr [1:660400] "STOP" "START" ...
## $ jobId : chr [1:660400] "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705"
##    "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705" ...
## $ taskId : chr [1:660400] "b47f0263-ba1c-48a7-8d29-4bf021b72043"
##    "20fb9fcf-a927-4a4b-a64c-70258b66b42d" ...
```

As we can see, this datasets only has character variables. We can clearly see that the information held in this dataset corresponds to a time recorded operation of the rendering software system. A timestamp is also apparent for each operation however it looks like it might need to be reformatted into an actual datetime variable for processing.

2nd Dataset

The second one contained metrics that were output regarding the status of the GPU on the virtual machine.

```
## tibble [1,543,681 x 8] (S3: tbl_df/tbl/data.frame)
## $ timestamp : chr [1:1543681] "2018-11-08T08:27:10.314Z"
##    "2018-11-08T08:27:10.192Z" ...
## $ hostname : chr [1:1543681] "8b6a0eebc87b4cb2b0539e81075191b900001C"
```

```
##    "d8241877cd994572b46c861e5d144c85000000" ...
## $ gpuSerial : num [1:1543681] 3.23e+11 3.24e+11 ...
## $ gpuUUID : chr [1:1543681] "GPU-1d1602dc-f615-a7c7-ab53-fb4a7a479534"
##    "GPU-04a2dea7-f4f1-12d0-b94d-996446746e6f" ...
## $ powerDrawWatt : num [1:1543681] 132 117 ...
## $ gpuTempC : int [1:1543681] 48 40 45 38 41 ...
## $ gpuUtilPerc : int [1:1543681] 92 92 91 90 90 ...
## $ gpuMemUtilPerc: int [1:1543681] 53 48 44 43 47 ...
```

Inside this datase we have multiple numeric metrics that represent some parameters of the system's hardware operation such as temperature, power usage and resource utilization. Additionaly we also see information about the GPU, the hosting computer and just like before, the timestamp.

3rd Dataset

Lastly, the third one contained the x and y co-ordinates of the tiles of the images that ware being rendered for each task.

```
## tibble [65,793 x 5] (S3: tbl_df/tbl/data.frame)
## $ taskId: chr [1:65793] "00004e77-304c-4fbd-88a1-1346ef947567"
##    "0002afb5-d05e-4da9-bd53-7b6dc19ea6d4" ...
## $ jobId : chr [1:65793] "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705"
##    "1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705" ...
## $ x : int [1:65793] 116 142 142 235 171 ...
## $ y : int [1:65793] 178 190 86 11 53 ...
## $ level : int [1:65793] 12 12 12 12 12 ...
```

This is a slightly different dataset. This dataset holds information about the task (rendering) of a visualization. It specifies the the level at which the visualization was produced as well as the x and y axis of the image being rendered. This essentially gives us a deeper look at how each task (vizualization) was being rendered, pixel by pixel for a total of 256 by 256 resolution.

Data Selection

Since not all of the columns from the datasets are going to be used, it is clucial that I start by selecting the data that I will be working with and then later on inspecting their quality in order to make appropriate pre-processing to ensure they are ready for analysis.

Question about Event runtimes

This question concerns the runtimes of events. Therefore, it seems that we have to only work with only one dataset, the application.checkpoint. Within this dataset I will more specifically have to work with the columns that have information about the event types, their status (start/stop) and the timestamps. These three columns will be suffucient for me to extract the event runtimes.

Question about GPU temp and performance interplay

To extract the interplay between the GPU temperature and performance I will again only have to work with one dataset, the gpu dataset. The particular columns of this dataset that I will be using include the gpuTemp, and the two GPU performance metrics about the memory and cores utilization percentages.

Question about computational requirements of tiles

To find the variation in computational requirements for particular tiles I will have to use the all.event.times and task.x.y datasets by merging them to find the corresponding render runtime of each tile of the render process. More specifically I will be using the columns of event type and timestamp from the application.checkpoint dataset and the columns of x and y coordinates from the tiles.x.y dataset that correspond to each tile of the visualization.

Question about GPU performance

For finding the performance of the different gpu cards I will have to merge the datasets of gpu and application.checkpoint from which I will work with columns of event name, timestamp and gpuSerial.

Data Preparation

A typical data preparation process includes cleaning the data and then merging or constructing new data to be used in the analysis later on.

Question about Event runtimes

To extract the runtimes for each particular event type I had to construct a new column that counted the time between the start and finish of each individual event type. To achieve that I separated the dataset into five new datasets where each new dataset only held information about a specific task. This allowed me to more easily separated concerns and manage my data pre-processing. In each of these five datasets I took the columns of the timestamp and event status (start/stop) and used them to created two new columns of start time and stop time for each individual row. This was done so that I would be easier to then extract the time difference between the two. From there I created a new column with the time it took each event to run and merged all of my five datasets back together.

Question about GPU temp and performance interplay

To answer this questions I took all of the uniques temperatures that were recorded and for each temperature values I found the corresponding core and memory utilization percentages which I averaged for every temperature value. This allowed me to create a new set of data that had the average core utilization percentage and the average memory utilization percentage for every GPU temperature.

Question about computational requirements of tiles

To extract this information I merged my newly created dataset from my previous question that held the runtimes of all events with the dataset about tiles based on their common columns of taskId. I then narrowd down my new dataset to the rows that only represented the total render event types. This allowed me to match the runtimes of the whole render of the task to the corresponding x and y axes of that task.

Question about GPU performance

Since I needed to bring the gpuSerial column into the dataset with the runtimes I had a problem. Simply merging the two datasets was not possible as there was not a one to one relationship between them. However after exploring the data in two sets I realised that column hostname and gpuSerial do have a one to one relationship so I created a table of each hostname with its corresponding gpu. Next I matched the hostnames of my new dataset and application.checkpoint runtimes set and was able to import the gpuSerial column inside it.

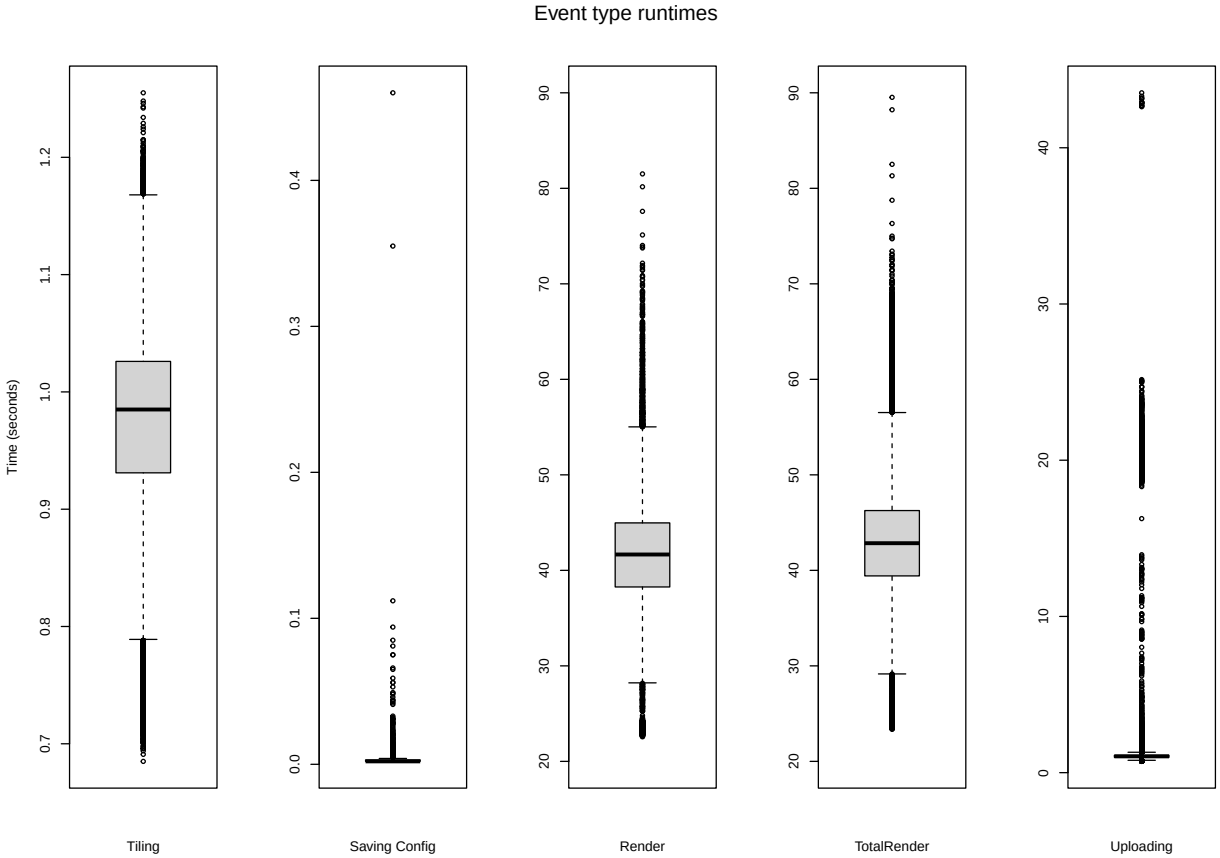
Data Analysis and results

Introduction

With all of the data preparation now complete, it is time to move to the next stage of this project, namely data analysis. My analysis will be solely based off of my initial data mining goals. The promary focus of this investigation was to look into the performance of the terapixel visualization system in both the physical and digital aspects of it. The exact questions I decided to work with can be found in the "Data mining Objectives and Goals" section of this report along with a more detailed explanation of which data sets were used and why. The following sections will include explanations of the processes that were followed for my analysis and the results that were produced as part of it, as well as the cases were the analysis was not particularly fruitful. The structure of the analysis will be a step by step walk through of the work that was done for each of the questions that I set.

Question about Event runtimes

To conduct EDA on the runtimes of event types I extracted the time it took to run each event and separated them into the five types of events. The resulting plot that can be seen below shows us the quantile ranges of runtimes for each event type.



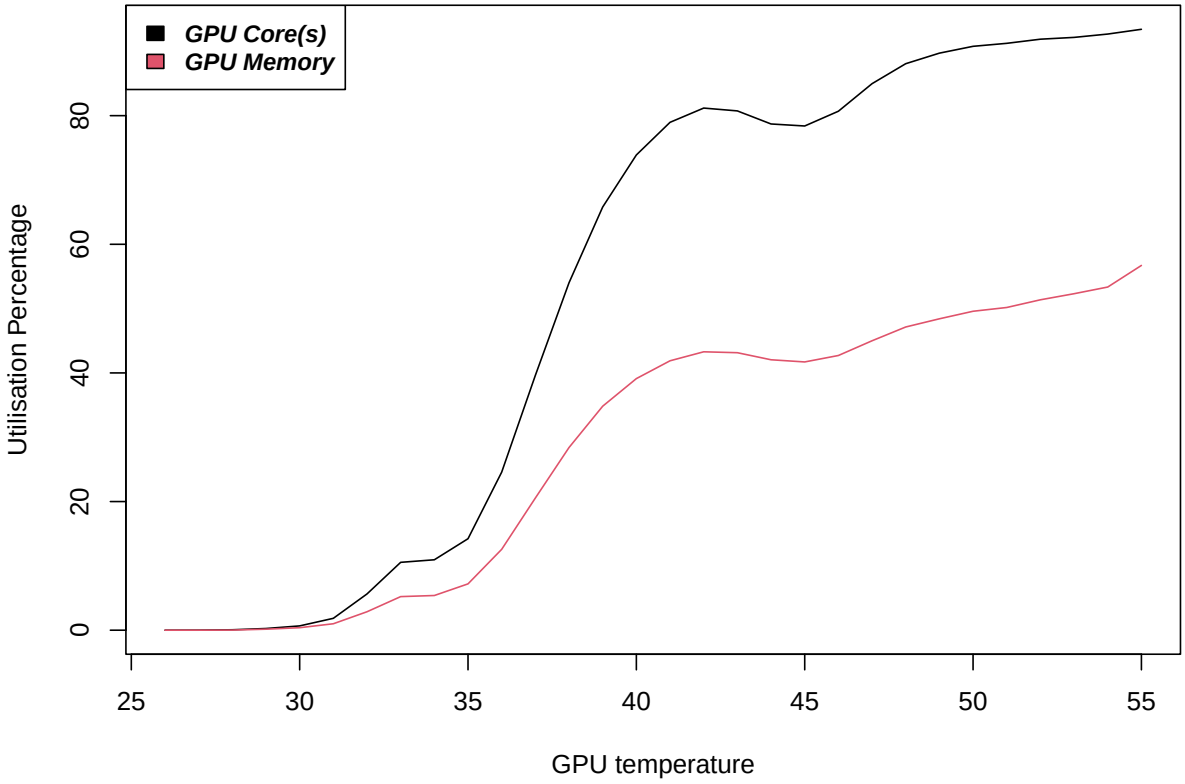

```
## [1] "The number of tasks for which these five events were run is:  65793"
```

Since event "TotalRender" is the collective sum of all other event types it should not be compared. Therefore by looking at the y axis of the plots we can tell that the event type "Render" was by far the most time

consuming with an average runtime of just above 40 seconds. Whereas all other event types have average runtimes of less than 2 seconds. This is almost 20 times difference in runtime. The event type "tilling" is quite equally spread out in terms of its quantiles and outliers. In contrast, when we look at "Saving Config" we see that although all runs were relatively quick, a few of these events took tremendously longer to run than the average. The "uploading" event type is even more uneven, with a large amount of outliers being a lot slower than the average, even taking more than 40 seconds. This indicated that uploading the rendered tile can often run into issues that cause delays.

Question about GPU temp and performance interplay

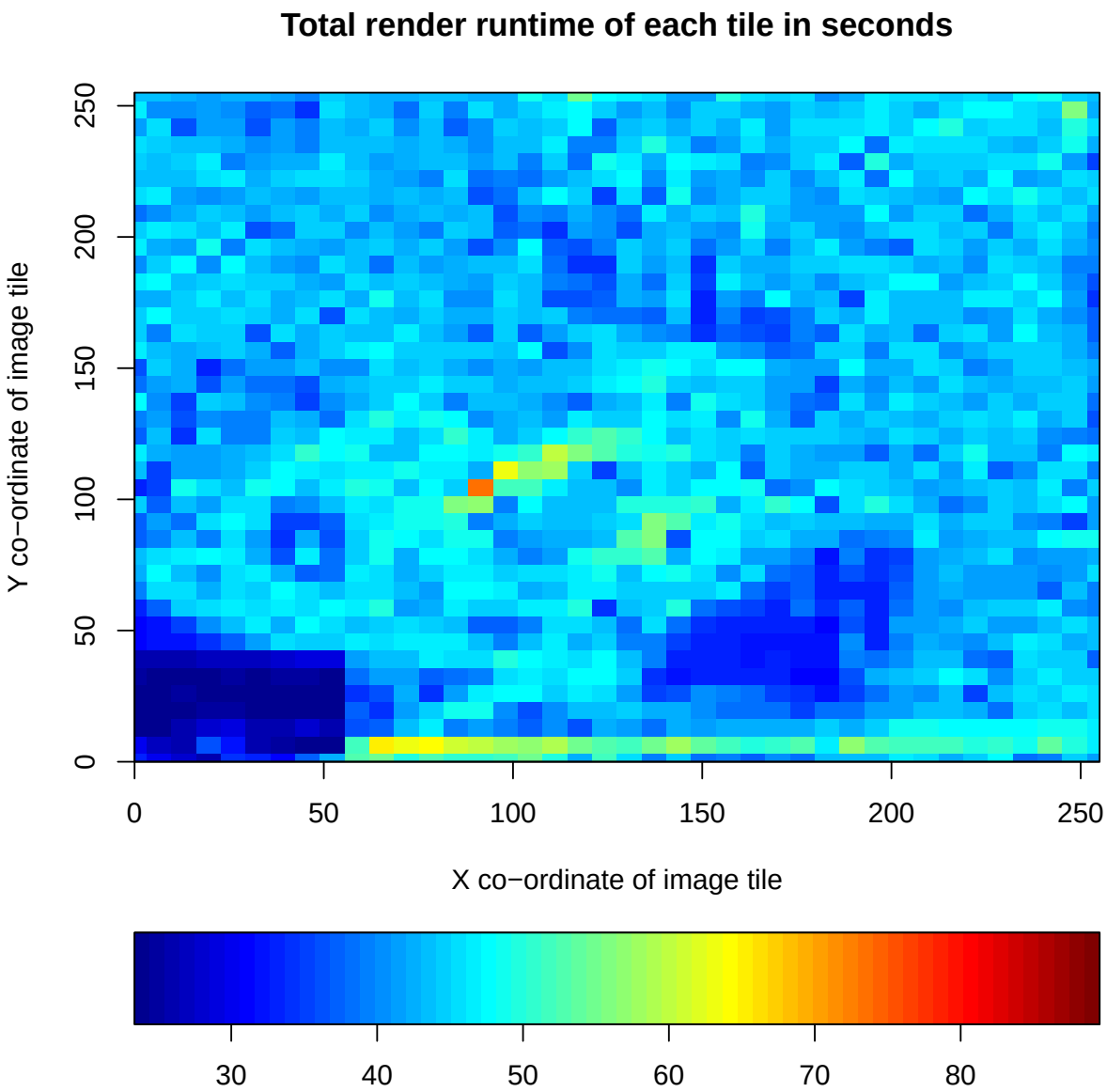
To understand the interplay between the GPU temperature and the performance of the system we will look at the two metrics that were tracked about the system's memory and cores utilization percentage against the corresponding temperature at that instance.



From the plot above we can tell that the average utilization percentage for both the system's cores and memory go up as the temperature increases. However at the temperature range between 42 to 45 degrees we observe that the utilization percentage goes down for both. This phenomenon could of course just simply be a system measure that gets initiated to prevent overheating such as the fans getting powered on. Additionally, we can see that once the memory reaches 40% from there onwards the GPU temperature continues increasing while the utilization begins to slow down. This is also the case with the cores of the GPU, where once they reach 80% their utilization percent slows down while their temperature keeps going up.

Question about computational requirements of tiles

There are two ways to look at the computational requirements of tiles, one is by looking at the amount of time it took to render it and the other is by measuring the core and memory utilization percentage. However, due to the nature of the data that I was working with I decided to go with the first option which is the runtime. Below we can see a plot that I produced which displays the runtime of each tile/task in the form of a heatmap where each pixel represents a tile. The color scale below that goes from 30 to 80 or from dark blue to dark red represents the amount of time it took to fully render a given tile.

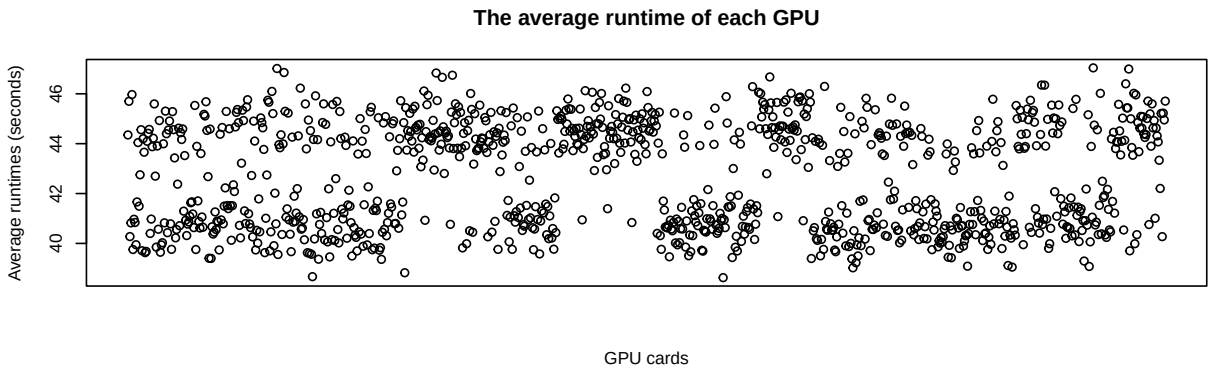


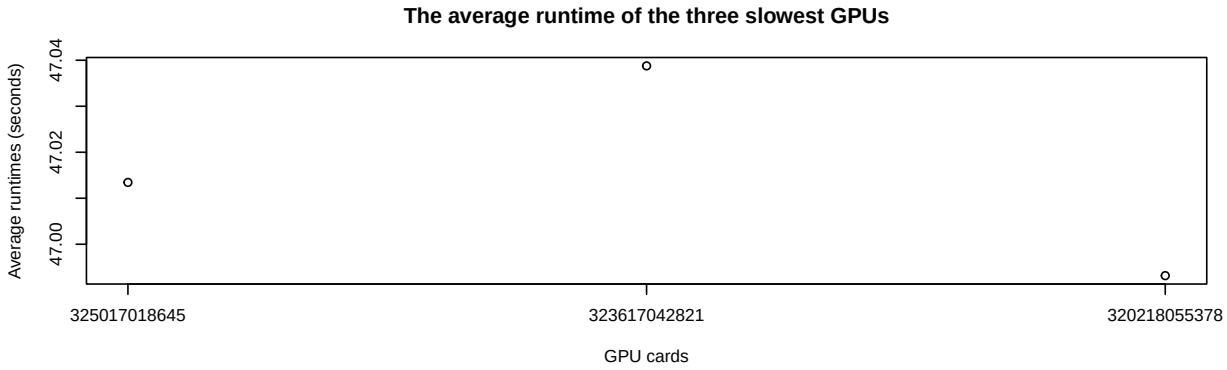
By looking at the orange and yellow pixels of the plot we can see the tiles that took the most time to render. It is clear that the lower edge had some slow tiles, as well as the top right corner. We also see that the tiles around the 100,100 coordinated were also quite time consuming. The light blue pixels, which took approximately 50 seconds to render can be found all over the plot and only really concentrated around the 50,75 coordinates. The fastest tiles to be rendered, the dark blue, were found to be those around the left bottom corner and also at the 175,50 coordinates, as well as some other less concentrated areas all around

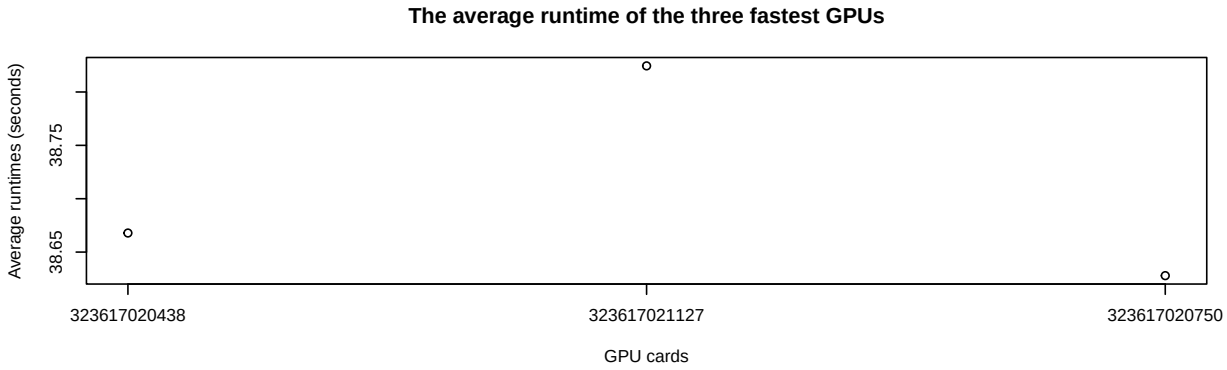
the plot.

Question about GPU card performance

To visualize the GPU card difference I used the pre-calculated runtimes and separated them according to the GPU card that produced them. Thereafter, I averaged these runtimes to get an estimate of the average runtime of a GPU to totally render a tile. The resulting plot can be seen below.





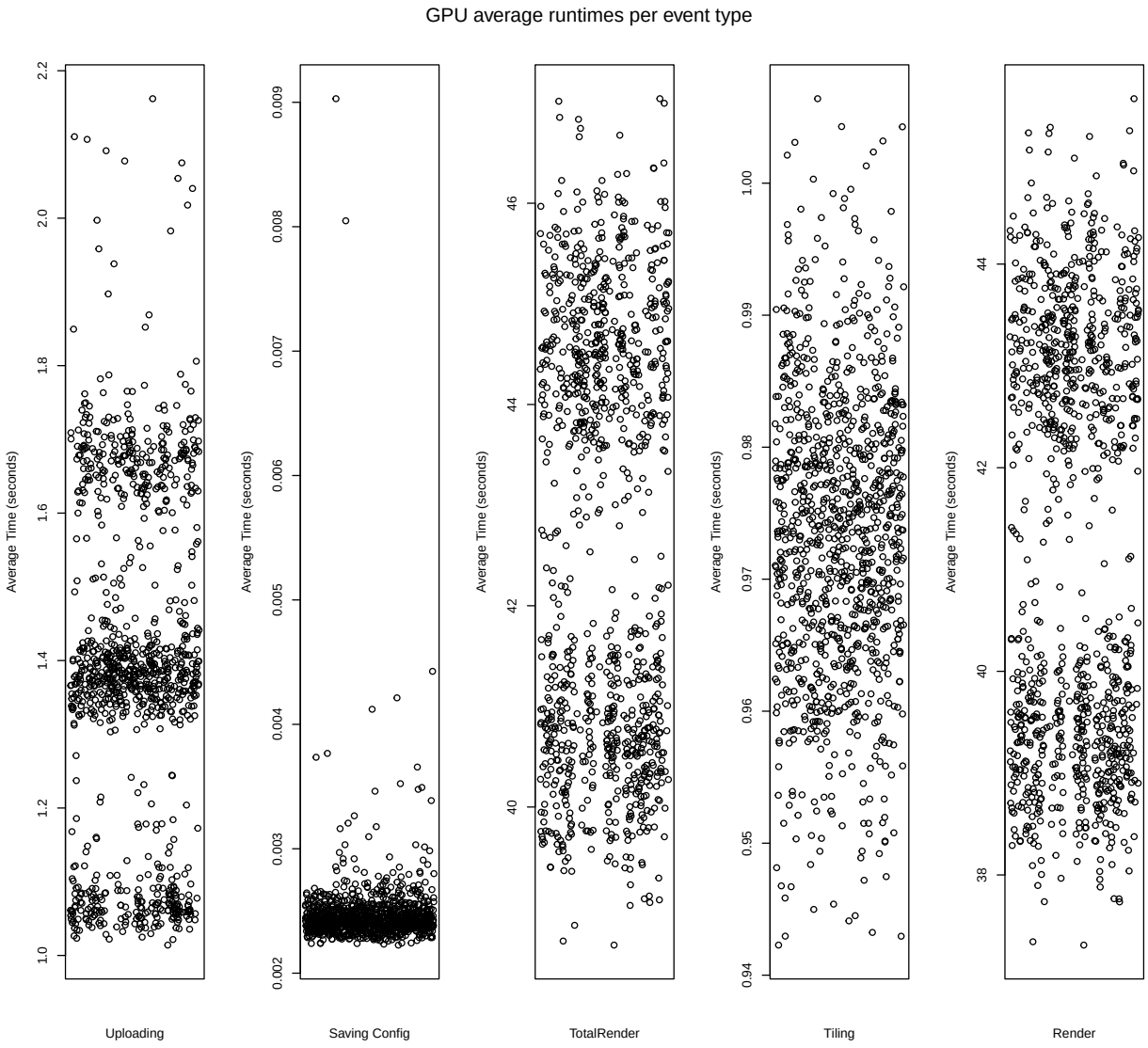



```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   38.63   40.69   42.98   42.70   44.60   47.04
```

I the first plot we see all of the Graphics cards and their estimated runtimes. The runtimes were found to range from 38.63 to 47.04 seconds. The plot clearly indicates that there are two quite distinct clusters of

GPU performance. One cluster is faster and is closer to a runtime of 41 seconds, while the other cluster is slower and is closer to a runtime of 45 seconds. Although the difference is only 4 seconds, when we are working with hundreds of thousands of tasks it can cause some problems. In the other two plots we can see the serial numbers of 3 GPUs with the fastest runtimes and the 3 with the slowest performance.

To further inspect the performance of the GPUs I want to look into their runtime average across different event types. Below we can see a plot of all the GPUs and their average runtime in completing different events.



When looking at the runtimes of the GPUs for each event type individually we can make some interesting observations. The type of "TotalRender" we have seen earlier so it can be used as reference. The "Render" event type is very similar to the "TotalRender". However, when we take a look at the "Uploading" event type we see that instead of two clusters, now we have three. This could suggest that there are three categories of GPUs that handle uploading information differently and that one of these categories performs better than the rest in this regard. With this event type we also observe some relatively high outliers which would suggest that a few of the GPUs are particularly bad at handling the uploading event. When it comes to the tiling event we see that all of the GPUs perform quite well with minimal difference amongst them. For the Saving configuration event type it seems that most GPUs perform very similarly with the exception of a handful of GPUs that perform relatively slower. However, the amount of time that it takes any of the

GPUs to save a config is quite insignificant as it very small when compared to the runtimes of other types of events.

Conclusion

Some very interesting findings were made from this investigation. All of our initially set questions have been answered successfully and there was even room for further exploration in many of them. Through the data pre-processing and exploratory data analysis that was carried out I was able to find the event types that dominate runtimes, the interplay between GPU temperatures and performance, I found the variation in computational requirements for particular tiles and I was also able to evaluate the performance of various GPU cards against multiple tasks. My work plan was executed as expected, with all parts of this project commencing smoothly and providing me with the required resources to move on from stage to stage.

Future Work

From the findings that I have gathered about the GPUs I realize that further research could be conducted on the specific details of each of these GPU cards to find what were the characteristics of the good and bad performing cards. Such an analysis could prove to be very valuable in understanding the correct equipment that can be used in cloud supercomputers such as this one.

Another interesting topic that could be further research is the interplay between the GPU temperature and its performance levels. After seeing that as the temperature goes up, the performance increase slows down we could investigate as to why this is happening and try to look into ways to change this effect if possible in order to maximize the performance of the GPUs.

Self-Reflection

This investigation project has been really enjoyable for me. Although this project was quite similar to the one in my previous module, I still learned many new thing because this time I was working with data that concerned something that I previously new nothing about, therefore a clear domain understanding was necessary to acquire before commencing on with the exploration and analysis of the data. In the technical aspect of the project it was interesting to see how correlated each dataset was with the others. A personal milestone that I reached was to delve into the cross-dataset data exploration by merging and combining data or even creating new ones out of many datasets at the same time even when the relationships of the variables being merges were not a one to one relationship. This was something that I had not done in the past and although it was very challenging, I am very happy that I chose to do it because it offered me invaluable knowledge about the possibilities that exist in manipulating data and extracting from it information that to the untrained eye may seem as non-existent. In the future I would like to test myself even further, without fearing about limitations that I might face or how ambitious my plan is. Working on a data science project throughout its lifecyle is not hard per say, but rather it is a matter of ingenuity and creativity.