

Analysis of Terapixel Visualization

Siddharthan Saravanan

25/01/2021

Introduction

Terapixel images offer an intuitive, accessible way to present information sets to stakeholders, allowing viewers to interactively browse big data across multiple scales. The challenge that was addressed is how to deliver the SuperComputer scale resources which are needed to compute a realistic Terapixel image and the data was captured by the Newcastle Urban Observatory. The Terapixel image obtained from the data can be viewed through this link, <http://terapixel.wasabi.1024.s3.eu-central-1.wasabisys.com/vtour/index.html>.

This scalable architecture was created for cloud based visualization where it is possible to deploy and pay for only as needed. The Terapixel image is created in such a way that, it can be viewed in all sorts of devices irrespective of its computational power. This analysis report is to find out the outliers using different GPU's which can be analysed for the improvement of the performance of the Super Computer in generating the image.

Insights of Data

The dataset which is taken for the analysis was created from application checkpoint and system metric output from the production of the Terapixel image. It particularly consists of three datasets which are application-Checkpoints, GPU and Task-x-y. Each dataset has different aspects which are used for the analysis to improve the performance of generating the Terapixel image. The **Hostname** and **TaskId** are the two variables which plays an important role that interconnects between these three datasets. There are totally 1024 GPU nodes and each are encoded as **Hostname**, **GPU Serial** and **gpuUUID**.

Application-Checkpoints dataset:

- The Application-Checkpoints dataset consists of six variables that are Timestamp - the time recorded based on each task that has been occurred, Hostname - Hostname of the virtual machine auto-assigned by the Azure batch system, EventName - Name of each event occurring while rendering the application, EventType - Type of each event names, jobId - ID of the Azure batch job and TaskId - ID of the Azure batch task.
- There are totally five different EventNames which includes TotalRender - the entire task, Render - the image tile is being rendered, Saving Config - Measure of a configuration, Tiling - Post processing of the rendered tile is taking place and Uploading - The output from post processing is uploaded to Azure storage.
- Each TaskId is linked with the EventNames and each TaskId consists of all the five EventNames.

GPU dataset:

- The GPU dataset consists of eight variables that are Timestamp - the time recorded based on the GPU utilization of each hosts, Hostname - Hostname of the virtual machine auto-assigned by the Azure batch system, GPU Serial - The serial number of the physical GPU card, gpuUUID - The unique system id assigned by the Azure system to the GPU unit, powerDrawWatt - Power draw of the GPU in watts, gpuTempC - Temperature of the GPU utilized in Celsius, gpuUtilPerc - Percent utilisation of the GPU Core(s), gpuMemUtilPerc - Percent utilisation of the GPU memory.
- The GPU variables such as power drawn, Temperature, GPY utilization and GPU memory utilization are observed for every two second once each task takes place for rendering the Terapixel image.

Task-x-y dataset:

- The Task-x-y dataset consists of five variables that are TaskId - ID of the Azure batch task, jobId - ID of the Azure batch job, x - X co-ordinate of the image tile being rendered, y - Y co-ordinate of the image tile being rendered and level - The visualisation created is a zoomable “google maps style” map.
- In total the image was rendered with the creation of 12 levels of zoom, while for analysing purpose the dataset consists of only 3 levels that are levels 4,8 and 12 where 12 being the zoomed right in level.
- The X and Y variables given represent the co-ordinates of the image being rendered which corresponds to the city of Newcastle Upon Tyne.

Analysis

Before proceeding with the analysis, the data is inspected to verify whether the structure of the data is good for analysis and whether the data has duplicate observations in some of the responses. If the data contains duplicate observations then these should be removed as part of the data wrangling process.

- From the given source of input it is stated that totally there are 65,793 tasks which are being rendered for getting the Terapixel image.
- Whereas, we could find the total observations in the Application-checkpoints dataset consists of 6,60,400, as each task processes five EventNames each and these EventNames are processed based on exactly two EventTypes, the total Observations should be 6,57,930.
- This clearly states there might be duplicates in the dataset and these duplicates are removed in both the Application-Checkpoints dataset and GPU dataset for further analysis to take place.

Data Pre-processing

- The analysis is technical carried out mainly based on the data handling that has been done during the pre-processing process.
- After removing the duplicates from the dataset, all the three datasets are combined respectively into one Master dataset based on the **TaskId**, **Hostname** and **Timestamp** variables.
- Further to this, the TimeStamp variable is converted completed into seconds which could be easy for the analysis to take place. This process is done by splitting up of the Timestamp variable based on the unknown attributes and converted the Hours, minutes and seconds of the variable to Seconds completely.

- While combining the three datasets as a single Master data there might be *NA* values in the variables, since there wont be any match of the Timestamp observed in the GPU dataset with the Checkpoints dataset as GPU data is observed and noted at every two seconds as the GPU node was being Utilized.
- Hence these are handled by using the Last occurrence function which replaces the *NA* value of a variable with the last occurred value of that particular variable.
- To calculate the Time taken for each TaskId to complete a particular EventName, a new function Pivot Wider is used which converts the rows to columns based on the given criteria and thus helps in identifying the time taken for the process to complete.
- While calculating the summarized GPU variables, we can assume that if the GPU utilization is less than 10 then that particular GPU is not in use and hence, for the graphical analysis new columns have been added which consists of Total Power consumed, Average utilization Temperature, Average GPU utilization and Average GPU memory utilization for each TaskId and it was grouped together based on the TaskId's whose GPU utilization is more than 10.
- After these pre-processing steps, the final Master data is obtained which consists of 2201382 observations with 23 variables.

```

##           timestamp taskId                      hostname
## 1 2018-11-08T07:41:27.242Z <NA> 04dc4e9647154250beeee51b866b0715000000
## 2 2018-11-08T07:41:29.259Z <NA> 04dc4e9647154250beeee51b866b0715000000
## 3 2018-11-08T07:41:31.285Z <NA> 04dc4e9647154250beeee51b866b0715000000
## 4 2018-11-08T07:41:33.301Z <NA> 04dc4e9647154250beeee51b866b0715000000
## 5 2018-11-08T07:41:35.322Z <NA> 04dc4e9647154250beeee51b866b0715000000
## 6 2018-11-08T07:41:37.338Z <NA> 04dc4e9647154250beeee51b866b0715000000
##   eventName eventType seconds x y jobId level    gpuSerial
## 1      <NA>     <NA>  27687 NA NA <NA>    NA 323217056165
## 2      <NA>     <NA>  27689 NA NA <NA>    NA 323217056165
## 3      <NA>     <NA>  27691 NA NA <NA>    NA 323217056165
## 4      <NA>     <NA>  27693 NA NA <NA>    NA 323217056165
## 5      <NA>     <NA>  27695 NA NA <NA>    NA 323217056165
## 6      <NA>     <NA>  27697 NA NA <NA>    NA 323217056165
##           gpuUUID powerDrawWatt gpuTempC gpuUtilPerc
## 1 GPU-a1119ee9-9cd1-919f-a479-b902142c717d      25.94     32      0
## 2 GPU-a1119ee9-9cd1-919f-a479-b902142c717d      25.84     32      0
## 3 GPU-a1119ee9-9cd1-919f-a479-b902142c717d      25.84     32      0
## 4 GPU-a1119ee9-9cd1-919f-a479-b902142c717d      25.94     32      0
## 5 GPU-a1119ee9-9cd1-919f-a479-b902142c717d      25.84     32      0
## 6 GPU-a1119ee9-9cd1-919f-a479-b902142c717d      25.93     32      0
##   gpuMemUtilPerc START STOP time_difference total_power_drawnW
## 1          0    NA    NA            NA            NA
## 2          0    NA    NA            NA            NA
## 3          0    NA    NA            NA            NA
## 4          0    NA    NA            NA            NA
## 5          0    NA    NA            NA            NA
## 6          0    NA    NA            NA            NA
##   average_gpu_tempC average_gpu_utilP average_gpu_memory_utilP
## 1             NA            NA            NA
## 2             NA            NA            NA
## 3             NA            NA            NA
## 4             NA            NA            NA
## 5             NA            NA            NA
## 6             NA            NA            NA

```

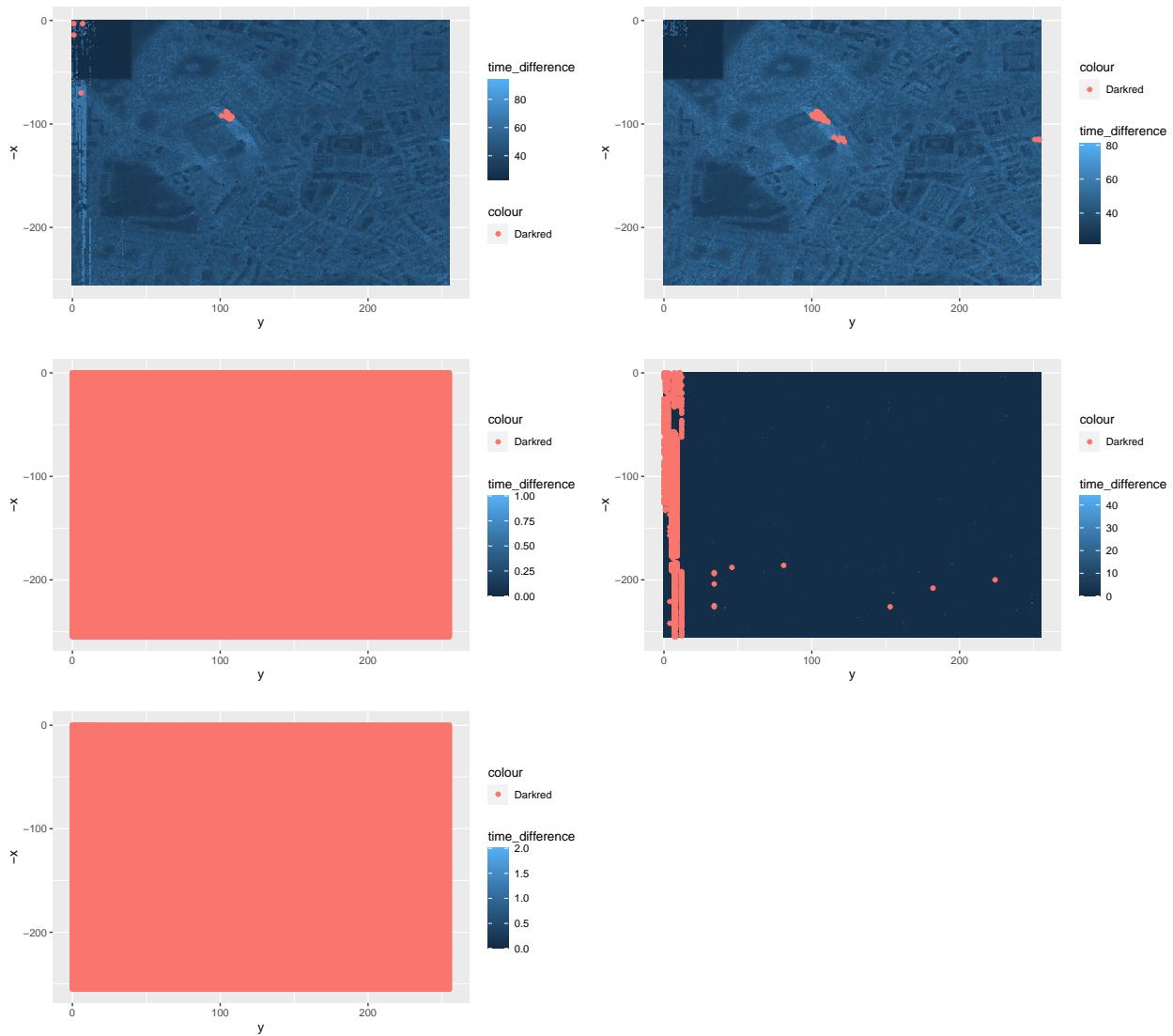
- The above displayed dataset is the first few observations of the final Master dataset.
- Further to this, a new dataset is generated which consists of Total Power consumed, Average utilization Temperature, Average GPU utilization and Average GPU memory utilization for each hostname and it was grouped together based on the hostnames whose GPU utilization is more than 10.

Graphical Analysis

The graphical analysis is made with Heat maps, line graphs, points, bar plots and box plots which helps to visualize and interpret the data. The **ggplot2** library is used for this graphical analysis.

Analysis based on Event types

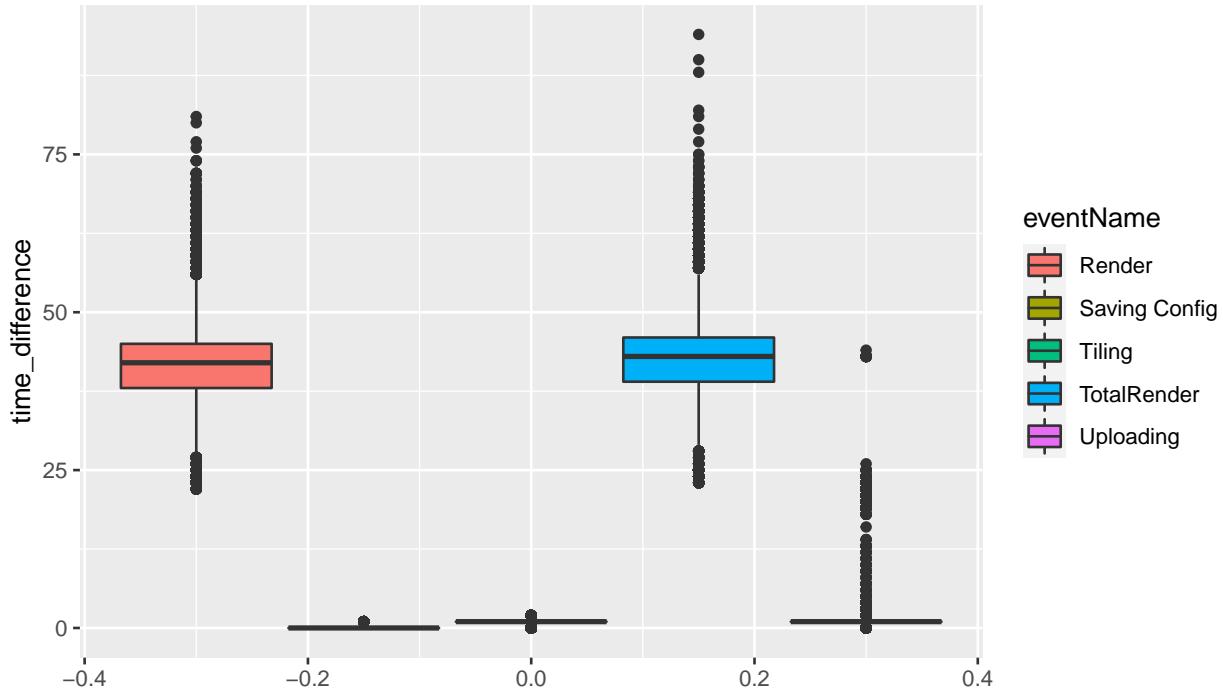
Graphical representation 1 :



- This graphical analysis is made using the Heat Map to represent time taken for each tasks to complete its process by plotting over the x and y co-ordinates of the Terapixel image.

- The Red points in the graph represent the regions where the highest time is taken for the tasks to complete the particular event process.
- From the plots, it is observed that both the Saving Config and Tiling events have taken similar time for completing the process for each taskId's.
- The Uploading event plot shows that majority of the regions are taking similar time for completing the process, whereas the regions which is filled with Red points seems to take higher time for completing the process for each TaskId's.
- The Rendering event takes more time for completing each task and hence while plotting it with the X and Y co-ordinates we are able to generate a look a like plot of the original Terapixel image, Where the Red points shows the region which takes higher time for completing this event.
- The total Render event is combination of all the events that takes place for completing the tasks. Hence, it also plots the graph similar to the Original Terapixel image.

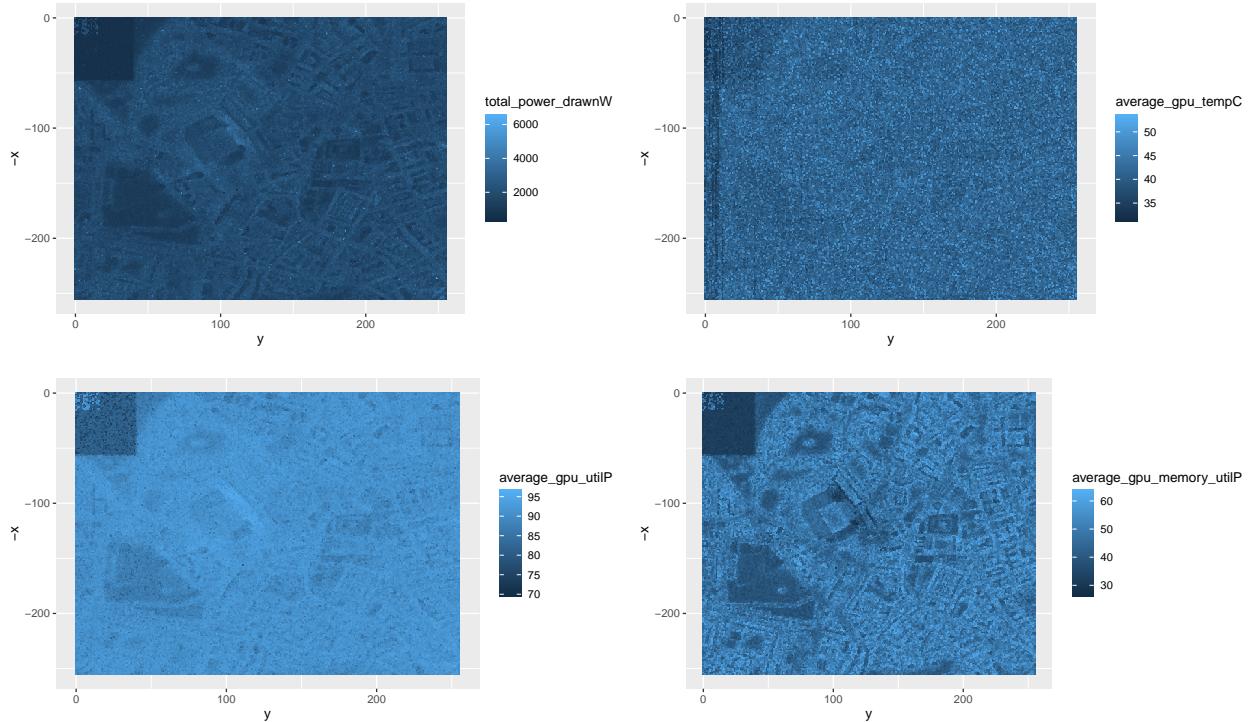
Graphical representation 2 :



- The above analysis is made using the Box plot to represent the outliers of each EventNames based on the time difference, when the particular event has completed its process.
- This plot clearly justifies the findings of the first graphical representation that the *Render* event dominates the total run time of generating the Terapixel image.
- In this plot, we could see that there are some outliers for the *Uploading* Event. From the findings from the first graphical representation and significantly with this box plot, we could observe that there are some tasks which takes excess time for completing this Uploading process.
- In-order to explore this finding, we need further information of the size of the pixel that is being uploaded by those specific tasks which could be able to find the trend for this activity.

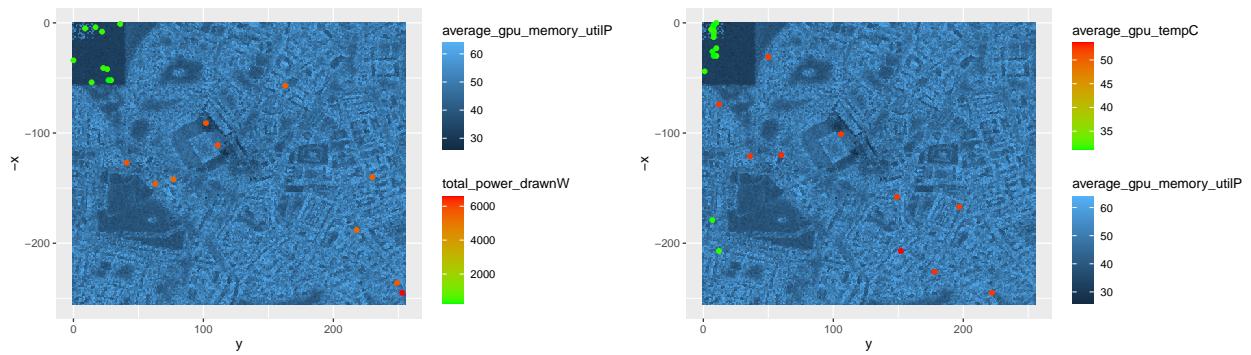
Analysis based on GPU Performance

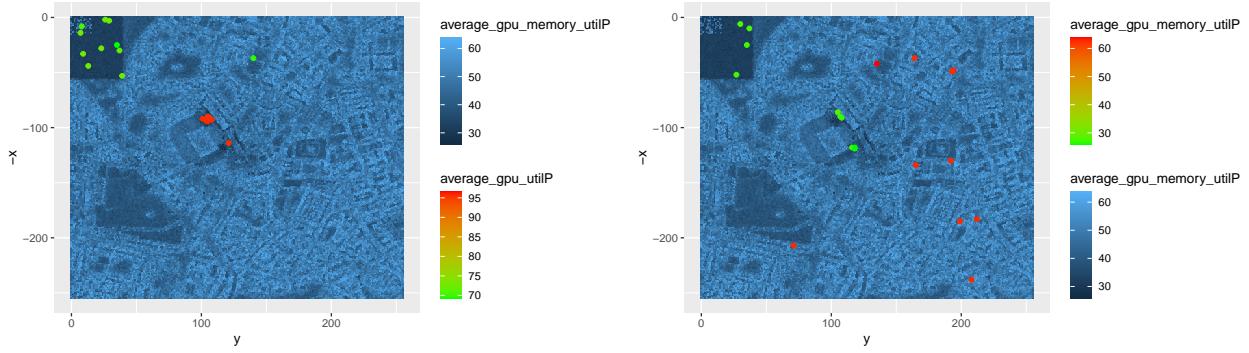
Graphical representation 3 - Based on TaskId :



- The above generated plots represent the GPU performance analysis which is made using the Heat Map by plotting over the x and y co-ordinates of the Terapixel image based on the TaskId.
- The GPU performances are observed through four variables as PowerDrawn, Gpu Temperature, GPU utilization and GPU Memory utilization.
- This graph is generated by finding the summarized values of these variables with respect to each TaskId, which was earlier explained in the Data Pre-processing step.
- There is not much information out of this plot, but this plot similarly makes the original Terapixel image.

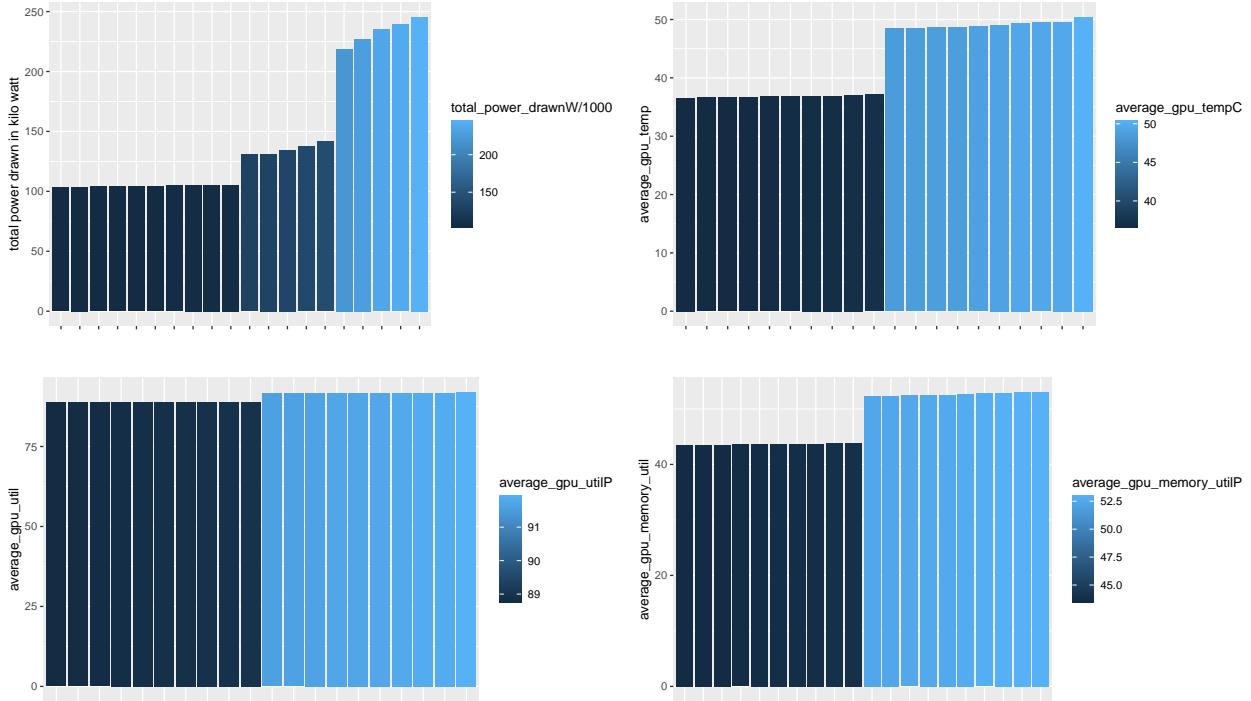
Graphical representation 4 - Based on TaskId :





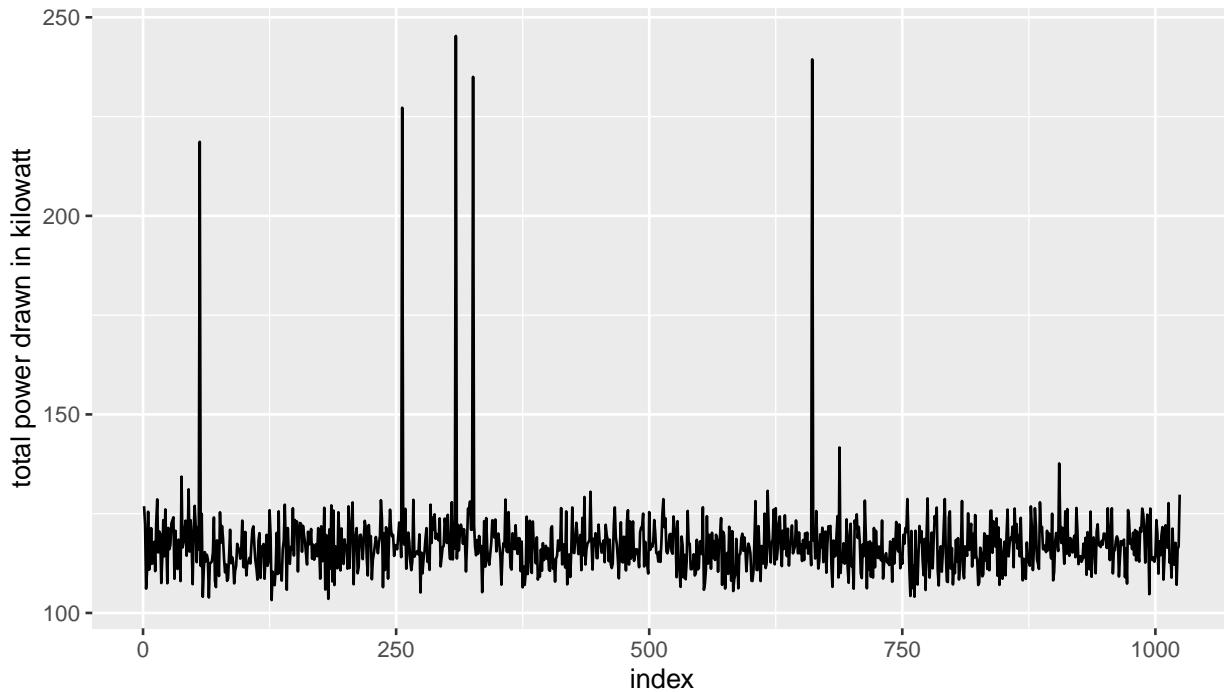
- This graphical representation is based on the outliers of each GPU variables with respect to the average GPU memory utilization based on the performance of each tasks.
- From these four plots its visible that the tasks which is being run on the top left corner of the Heat map takes less computational power overall.
- From the first three plots of this representation, we could see the centre region of the map has used some high computational power while rendering the tasks.
- The last plot shows that the centre region of the map uses less GPU memory utilization suggesting that the tasks which are performed in that particular region will either be the one that uses high computational power orelse the otherway.
- This outlier representation is made by extracting the top and bottom 10 observations of each GPU variables.

Graphical representation 5 - Based on Hostname :



- The bar graphs represents the most and least GPU variable usages while rendering process takes place based on the different hostnames.
- From the first plot we could interpret that out of the top 10 hostnames, there are five hostnames which are consuming high power which suggests either the GPU's are not performing good or else the pixel that needs to be rendered is of huge size.
- The second plot of this representation shows the average temperature that was utilized for rendering the image using the observed hostnames. Its evident that the temperature lies in the range of 35% to 50% during the whole rendering process.
- Further to this, the GPU utilization and GPU memory utilization is more or less in near range for both the most and least used GPU nodes.

Graphical representation 6 - Based on Hostname :



- This Line graph is generated based on creating an index value to the hostname with respect to the Total power consumption while rendering the Terapixel image.
- From this representation, we could observe that there are some hostnames which consumes high power while rendering the Terapixel image.
- These outliers of high usage might be based on poor performance of the GPU or may be based on size of the pixel that needs to be rendered.
- To withstand and find outliers of these graphical representations, further analysis is made by finding out the high power consuming taskId's along with the hostnames so that we could find whether those particular GPU nodes have poor performance or the rendering time of a particular pixel is of huge size.

Business Solution

- The analysis is mainly carried out in order to improve the Performance of the SuperComputer for generating the Terapixel image.
- From the obtained graphical representations, it is evident that some of the GPU nodes use high power for rendering the Terapixel image and the Temperature of those nodes is equally high.
- To improve the performance of the SuperComputer we could suggest by replacing those high power consuming GPU nodes with respect to the less power consuming GPU nodes.
- Along with this GPU temperature also plays a key role in which by replacing those high temperature consuming GPU nodes with less temperature consuming GPU nodes will significantly improve the performance of the SuperComputer.

Reproducibility

- Project Template is used to pre-process the data, compute the necessary functions mentioned and then generate the report through the Rmarkdown.
- If there is a new file which is required to generate the report, it should be inserted in the data directory which consists of any of the following three variables (*taskId*, *timestamp* or *hostname*) in order to analyse the performance of the super computer.
- For the purpose of analysis the datasets are combined as a single dataframe and hence if there is any news files which should be reported, it needs to be manually update the pre-process file to combine the data which would be in the munge sub directory.
- The analysis of the report is not completely reproducible as some manual interventions are required to change the variable names, binding up of the datasets as a single dataframe when there appears a new datafile.
- The data file given is of huge size and hence while processing the files are stored in the Cache folder which can be used for multiple runs for reproducing the analysis.
- Apart from this, the report will be automatically generated when it is made to run from the R markdown report file which is in the Report sub directory.

Conclusion

After the brief analysis of the observations available, rendering a Terapixel image computes high energy as it is made to be compact for smaller devices also. In order to improve the Performance of the Super Computer for generating such kind of a Terapixel image, it is recommended that replacing the GPU nodes which consumes only less power and less temperature for performing the rendering process could be the ideal solution.