

# 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.

*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.

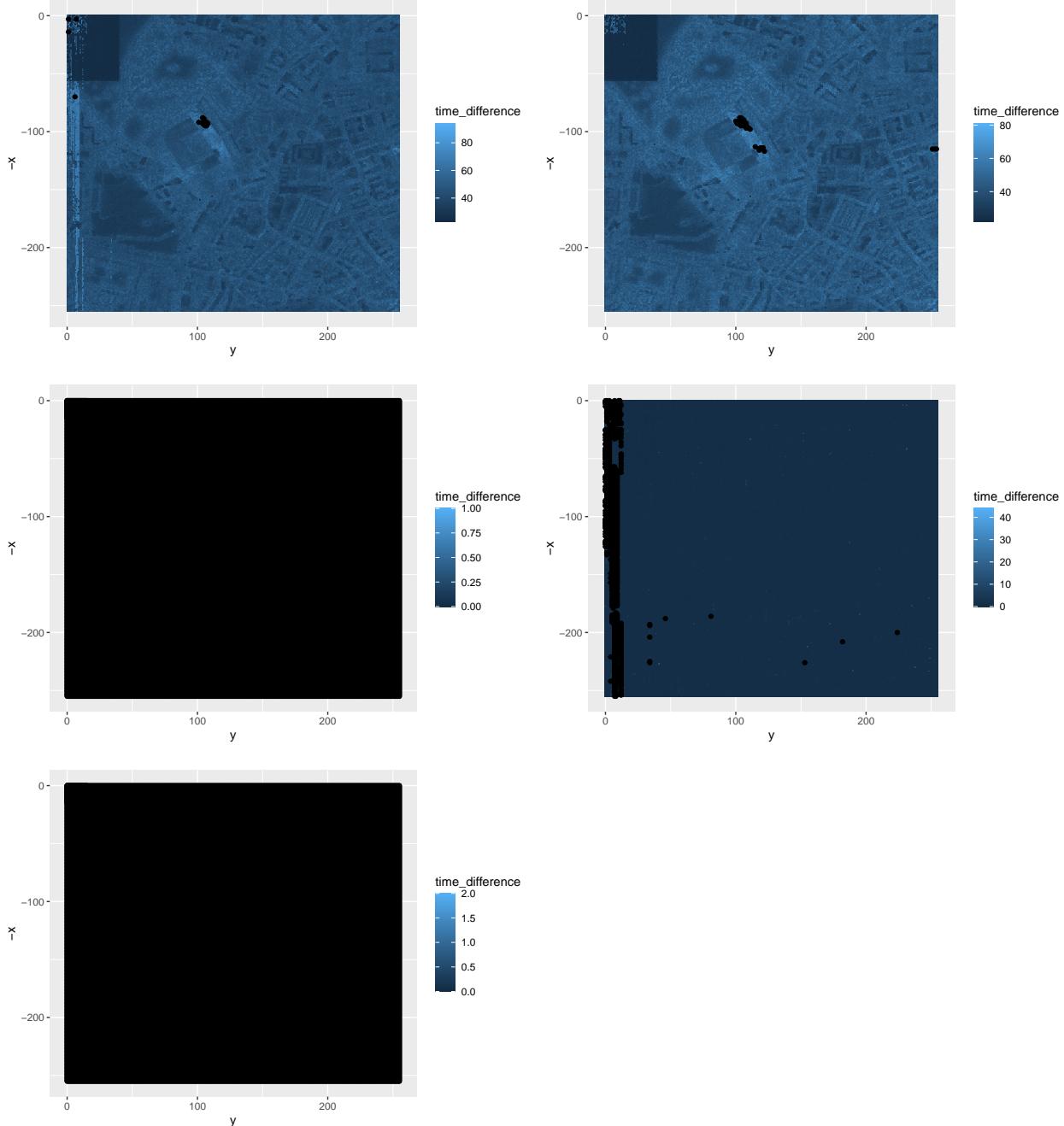
## Data Pre-processing

- 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.

## Graphical Analysis

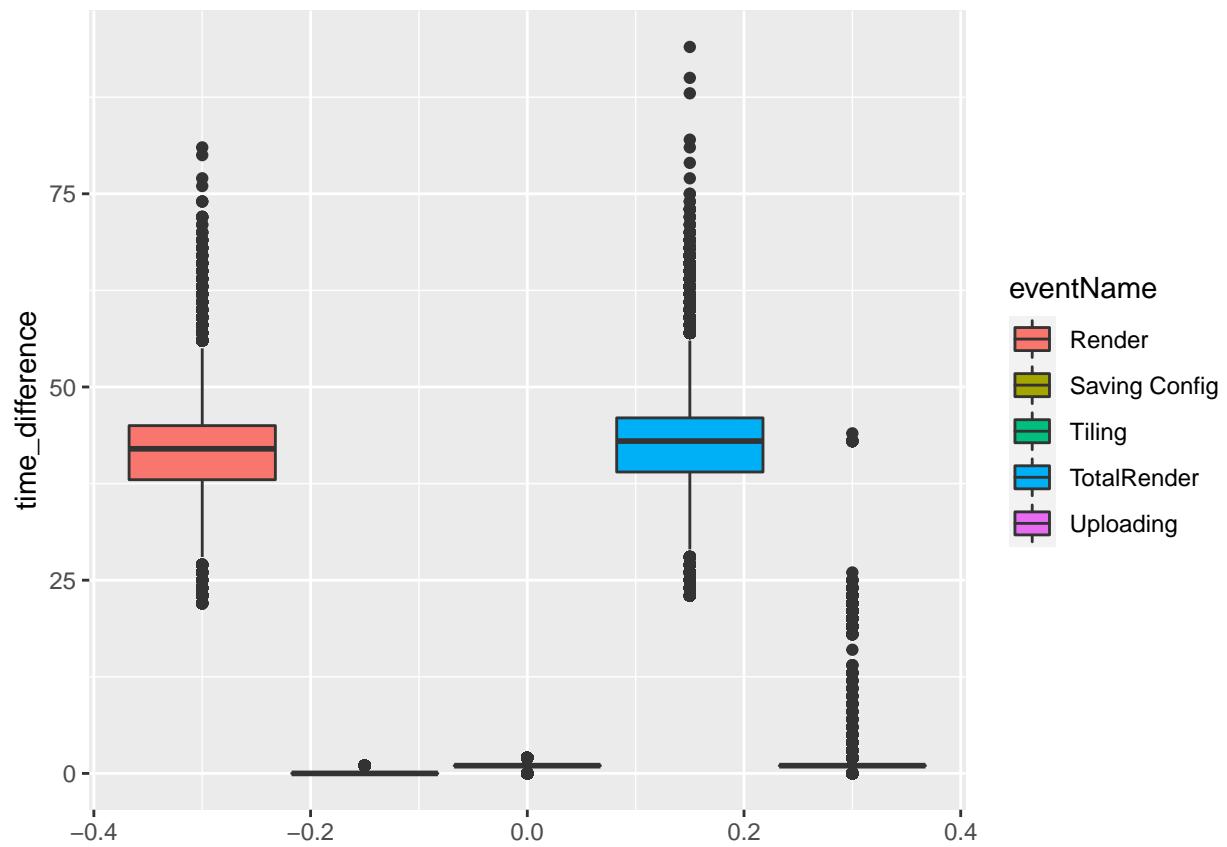
The graphical analysis is made with Heat maps, line graphs, points, bar plots and box plots which helps to visualize and interpret the data. The **ggplot2** library is used for this graphical analysis.

### Graphical representation 1 :

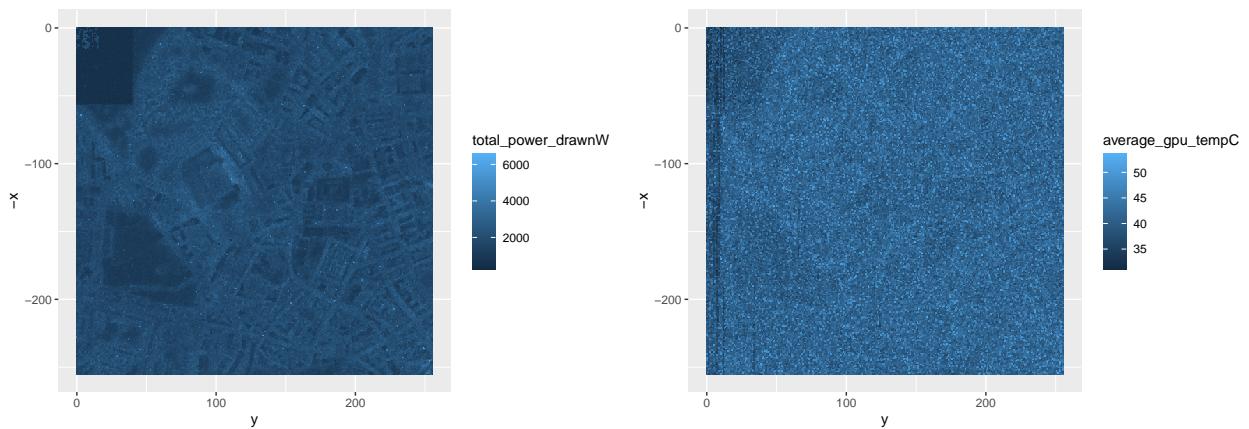


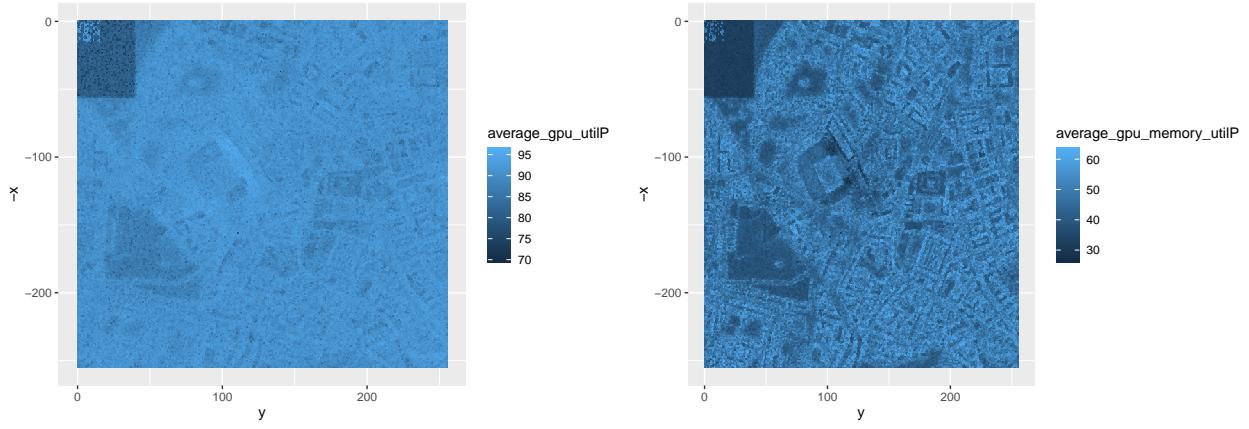
- This

### Graphical representation 2 :

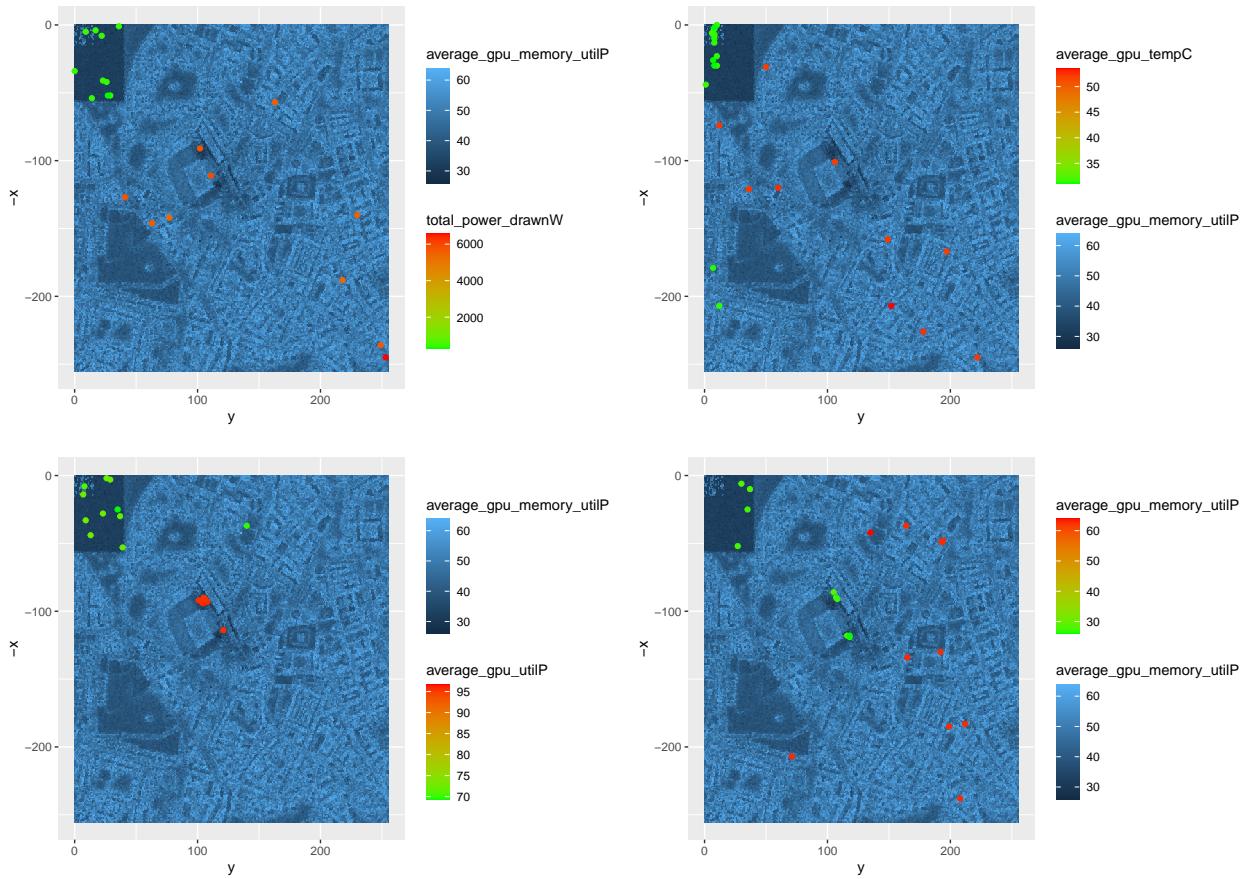


**Graphical representation 3 :**

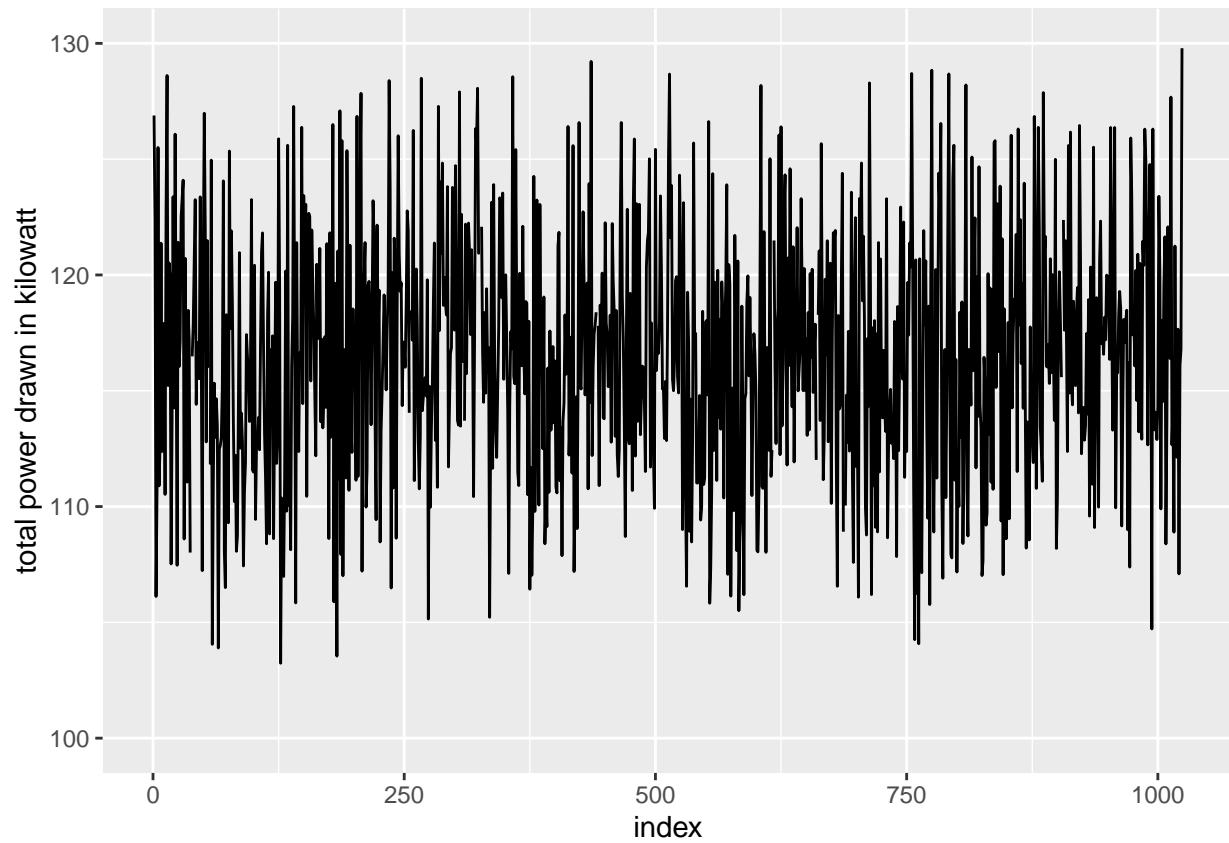




**Graphical representation 4 :**



**Graphical representation 5 :**



**Graphical representation 6 :**

