Cloud Computing

Performance evaluation of Terapixel rendering

Coursework CSC8634

Arushi Nautiyal
STUDENT ID: 220503835

1. Introduction

The objective of this report is to evaluate the performance of Terapixel rendering in cloud super computing. The motive is to address the issue of how to deliver the supercomputer scale resources needed to compute a realistic terapixel visualization of the city of Newcastle upon Tyne and its environmental data as captured by the Newcastle Urban Observatory. We're provided with different processes and their time stamps, GPU details, and task details. We'll further analyse these datasets in order to assess speed and accuracy of the super computing, we'll try to know more about the processes involved and their impact on different GPUs.

2. Data Understanding

It is crucial to understand what the data is about in order to draw conclusions from them. Three datasets are provided here, this data was produced using inputs from system metrics and application checkpoints during the creation of terapixel. Below is a glimpse on the same:

Dataset 1: Application checkpoint

Columns	Data type	Values
Timestamp /*timestamp*	Timestamp	Eg. 2018-11-08 07:53:47.398
Hostname /*hostname*	String	Eg. 4ad946d4435c42dabb5073531ea4f315000001
Event Name /*eventName*	String	TotalRender, Render, Saving Config, Tiling, Uploading
Event Type /*eventType*	String	Start, Stop
Job ID /*jobId*	String	Eg. 1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705
Task ID /*taskId*	String	Eg. 20fb9fcf-a927-4a4b-a64c-70258b66b42d

Table1. Application checkpoint data summary

Dataset2: GPU

Columns	Data type	Values
Timestamp /*timestamp*	Timestamp	Eg. 2018-11-08 07:53:47.398
Hostname /*hostname*	String	Eg. 4ad946d4435c42dabb5073531ea4f315000001
GPU serial number /*gpuSerial*	integer	Eg. 323217055910
GPU system ID /*gpuUUID*	String	Eg. GPU-1d1602dc-f615-a7c7-ab53-fb4a7a479534
Power Drawn /*powerDrawWatt*	Float	Different decimal values, eg., 131.55
GPU Temperature /*gpuTempC*	Float	Different integer values, eg., 48
GPU Utilization %/ gpuUtilPerc	Float	Eg. 92,90,91
GPU Memory Utilization %/ gpuMemUtilPerc	Float	Eg. 53, 48,47

Table1. GPU table data summary

Dataset 3: Task-x-y

Columns	Data type	Values

Job ID / jobId	String	Eg. 1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705
Task ID / taskId	String	Eg. 20fb9fcf-a927-4a4b-a64c-70258b66b42d
X coordinate of image/ x	Int	10
Y coordinate of image/ y	Int	125
Level/ level	Int	6

Table3. Task-x-y Data Summary

These three datasets are to be used to derive conclusions and answer few questions like:

- What is the time taken by different event?
- Which events take more time and which takes the least?
- If there any relationship among different GPU metrics?
- Which GPUs draw most power on average?
- Which GPUs take more time in completing the events?

3. Exploration Data Analysis

The provided data is used to explore about the processes and perform a thorough analysis over different features. This includes both graphical and numerical results in order to break down the problem statement and give a comprehensive result.

- **Different Events and Time Taken**

The 4 events namely Saving config, Render, tiling and Uploading constitute the total render process. Each of these events take a certain amount of time to complete. This time is calculated for each event and then an average of these duration is calculated:

Event	Average Time Taken (sec)
Render	41.2279
Uploading	1.39364
Tiling	0.97320
Saving Config	0.00247

Table 4: Average Time taken

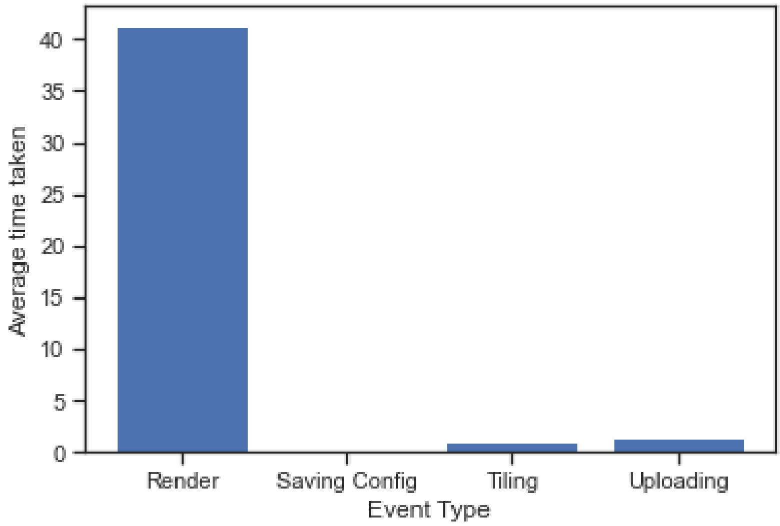

Fig1. Average time taken by Events

Time taken by render is significantly high and time taken by saving config is significantly low.

- **Relationship between GPU metrics and recorded temperature**

  Below pair plots are created to find a correlation between the metrics recorded in GPU table. The first pair plot is a combined plot for all hostnames, second pair plot is for a particular hostname "0745914f4de046078517041d70b22fe7000005".

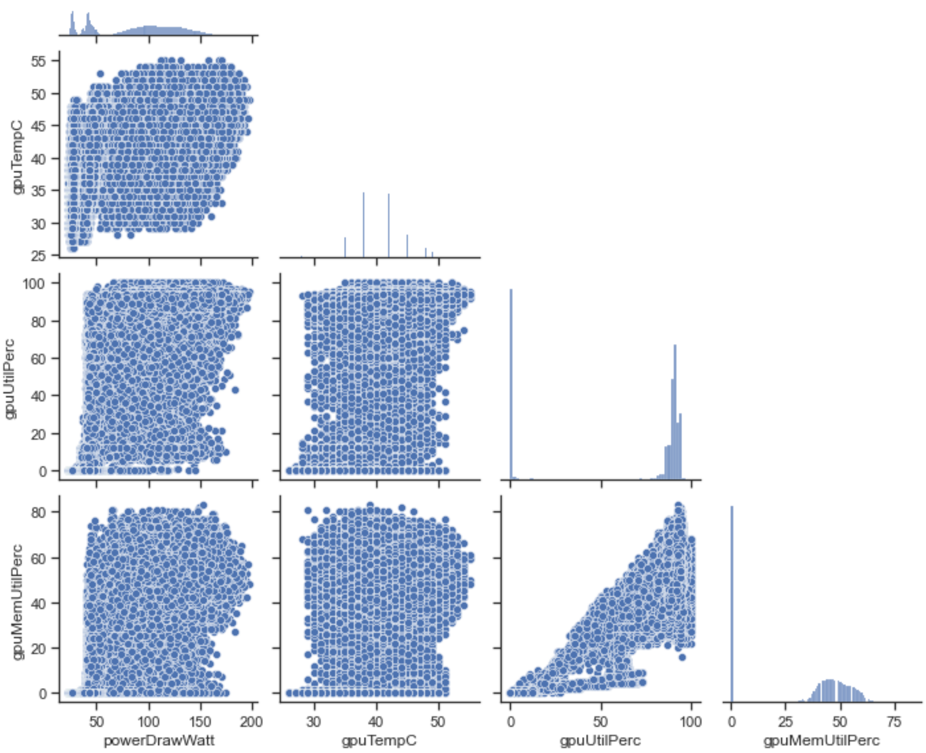

Fig2. Pair plot for all hostnames

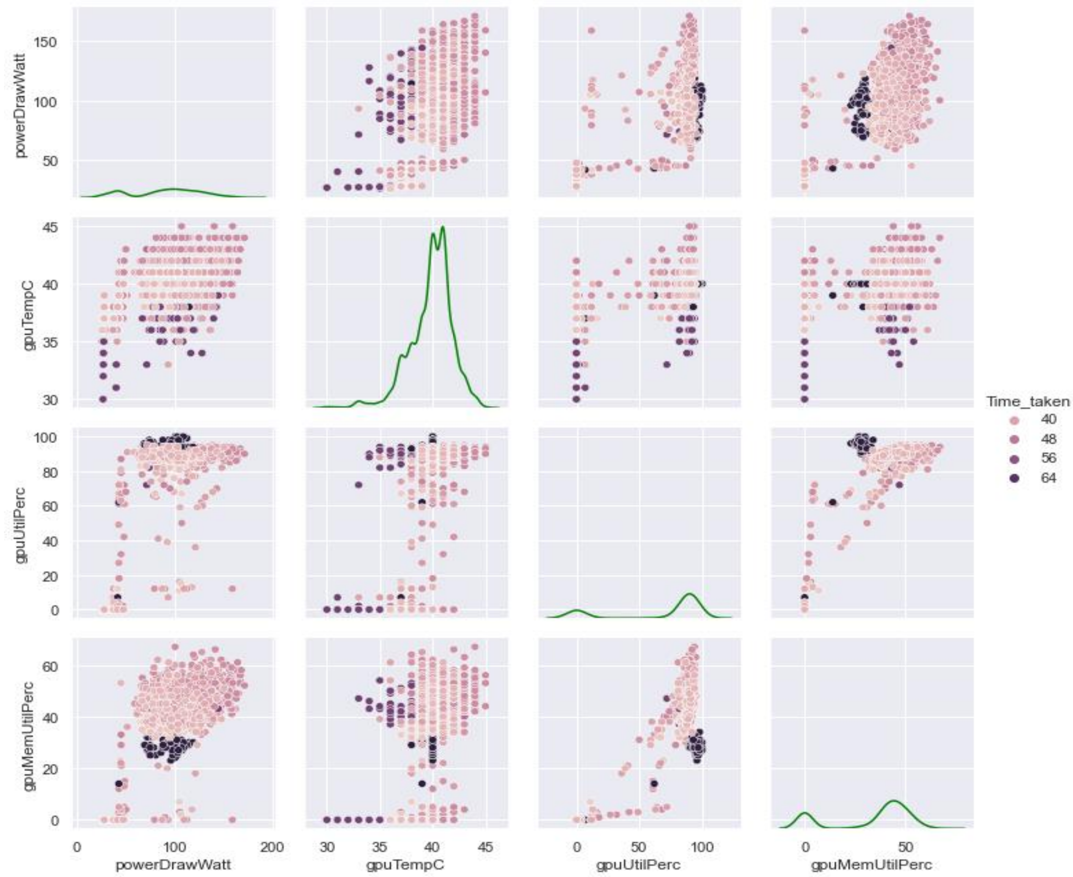

Fig3. Pair plot for a particular hostname with time taken for total render details

These 2 plots look very similar and depict a linear relationship between GPU utilization and GPU memory utilization. It is also worth noting in second plot that the points that represent higher time taken in dark colours are very clustered at one place, showing that higher time taking event leads to similar GPU metrics.

- **GPUs that drew most average power**

From the table GPU, average power drawn is calculated based on different GPU/hostname:

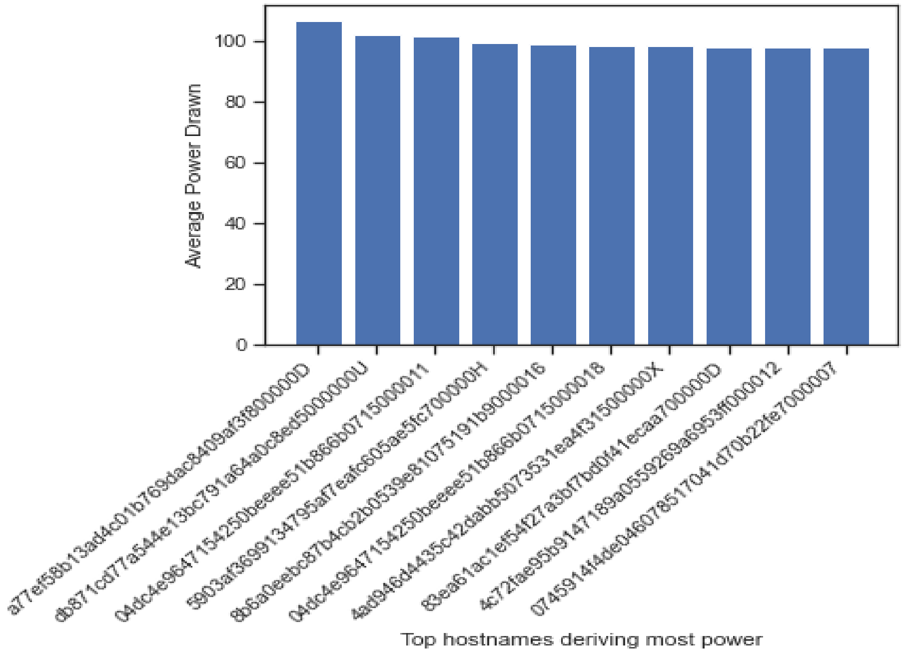

Fig4. Highest Power drawing GPUs/hostnames

Hostname	Average Power Drawn
a77ef58b13ad4c01b769dac8409af3f800000D	106.247462
db871cd77a544e13bc791a64a0c8ed5000000U	101.974324
04dc4e9647154250beeee51b866b0715000011	101.549633
5903af3699134795af7eafc605ae5fc700000H	99.057575

Table5. Average Power drawn by top 4 GPUs/hostname

- **Different events showing relationship between power drawn and GPU temperature**

GPU dataset and application checkpoint dataset are merged based on top 10 power drawing hostname and timestamp to find out average power drawn by each event. Top 10 hostnames are selected since GPU data is itself a very large dataset, and merging it with checkpoint data would create a dataset which is manyfold in size, hence considering any 10 hostnames with a special characteristics (here, most power drawn) seems like a good evaluation and would create a window to the larger picture.

Assumption: Since saving config takes less than a milli second to complete, it can be neglected from the analysis. This is specially because GPU data is recorded for different timestamps, and since saving config completes in fraction of seconds, no GPU data is recorded for saving config. <u>Hence, from now on, we're considering Render, tiling and uploading, and total render as the total event.</u>

Temperature and power drawn by different events in plotted below as a KDE plot:

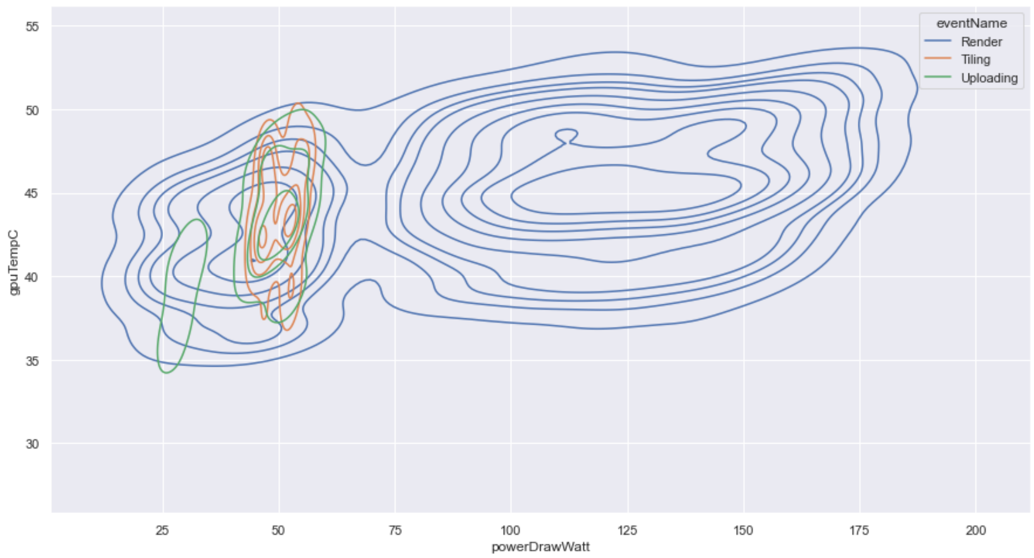

Fig5: GPU temperature Vs Power drawn by different events

This graph depicts that render shows an elaborate and slightly linear relation between power drawn and GPU temperature and these values are higher for only render. Tiling and uploading are confined to smaller space, depicting low power drawn but comparatively higher temperatures.

- **Average Power drawn by Each Event**

As mentioned earlier, GPU dataset has no data for the timestamp of Saving config, the latter would not be present for this conclusion. Secondly, this is for top 10 GPUs drawing most power.

Event Name	Average Power Drawn
Render	108.204
Tiling	50.078
Uploading	45.704

Table6. Average power by each event

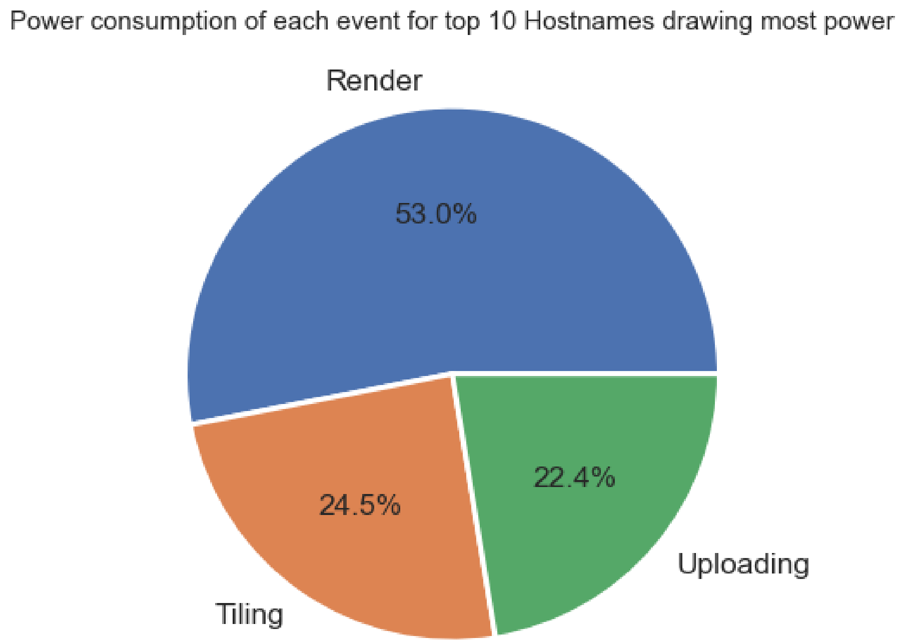

Fig6. Percentage Power drawn by different events

From this pie chart, its clear that Render takes most of the power, which is reasonable given the time that the process takes.

- **Temperature vs GPU for different events**

  Firstly, average temperature is calculated for different events against the GPUs to see which GPUs and which event takes up highest temperature.

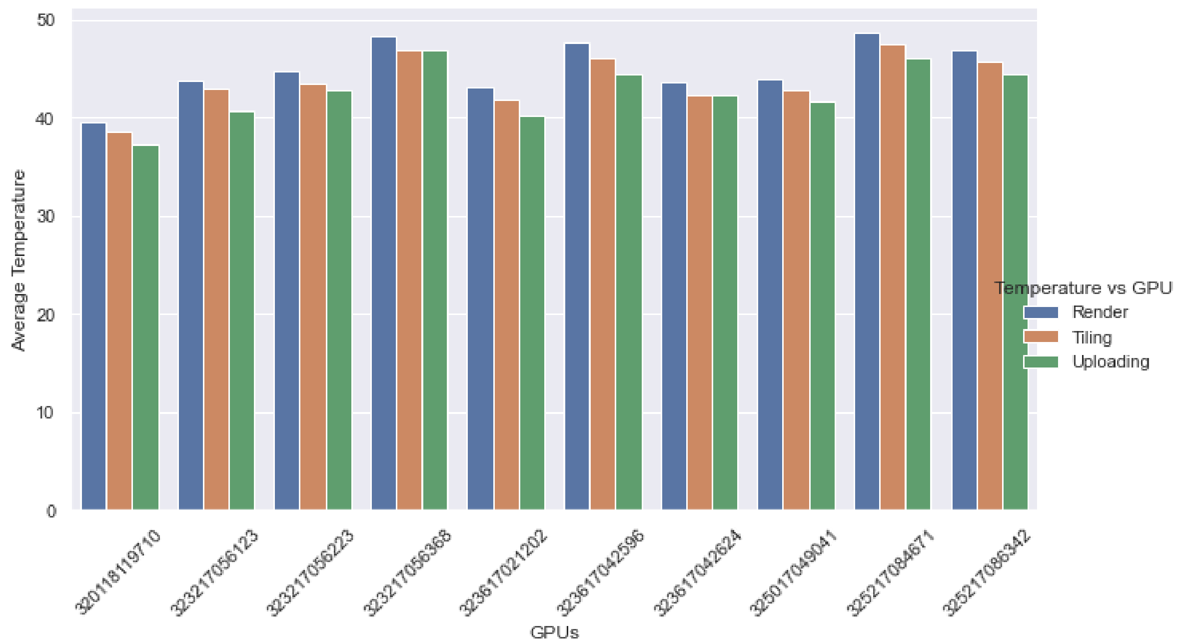

Fig7. Average Temperature V/s GPUs

It is clear here that for a given GPU, the event Render recorded the highest temperature, followed by tiling and then uploading. Also, we can say that the ninth GPU with serial no. 325217084671 has highest temperature, followed by fourth GPU with serial no. 323217056368. These GPUs would also draw more power hence these can be considered to be taken care of.

4. Conclusion

The EDA gave us an insight on what the processes are like, which event is more dominating and requires more resources, which GPUs require more attention given their temperature and power used. This analysis can be extended into the task dataset and extract information about the image processing as well. This can play a vital role in improving the efficiency and finding a pattern in the data so that future values can be predicted in advance and the organisation are ready for it beforehand.

5. Discussions

The analysis performed here is a subset of analysis that could be performed on this dataset. Although a lot of EDA is performed here, more details can be extracted like why render took so much time, how the data for saving config be captured in GPU dataset, is there a way to optimise render time duration and the power used during the event. This can help businesses in operation and creating products that are useful in these situations for better efficiency and accuracy.