Analysis and Performance evaluation of Terapixel rendering in (Super)Cloud Computing Data

220403421

Gowdham Ramesh

1. Introduction

The purpose of this study is to analyze the IoT environmental data collected by Newcastle Urban Observatory for the city of Newcastle Upon Tyne. The main objective is to evaluate and explore performance timings of render application and GPU card and in each task, the details of which part of the image was being rendered. Terapixel images are rendered using a scalable cloud-based visualization architecture. The Terapixel image, once created, allows for interactive exploration of the city and its data at a wide range of sensing scales ranging from the entire city to a separate desk in a room, and is accessible via a broad range of thin client devices. **CRISP-DM** (Cross- Industry Standard Process for Data Mining) model will be used in this data analysis. This project will be entirely dedicated to the **EDA** (Exploratory Data Analysis) process.

CRISP-DM Methodology to be followed:

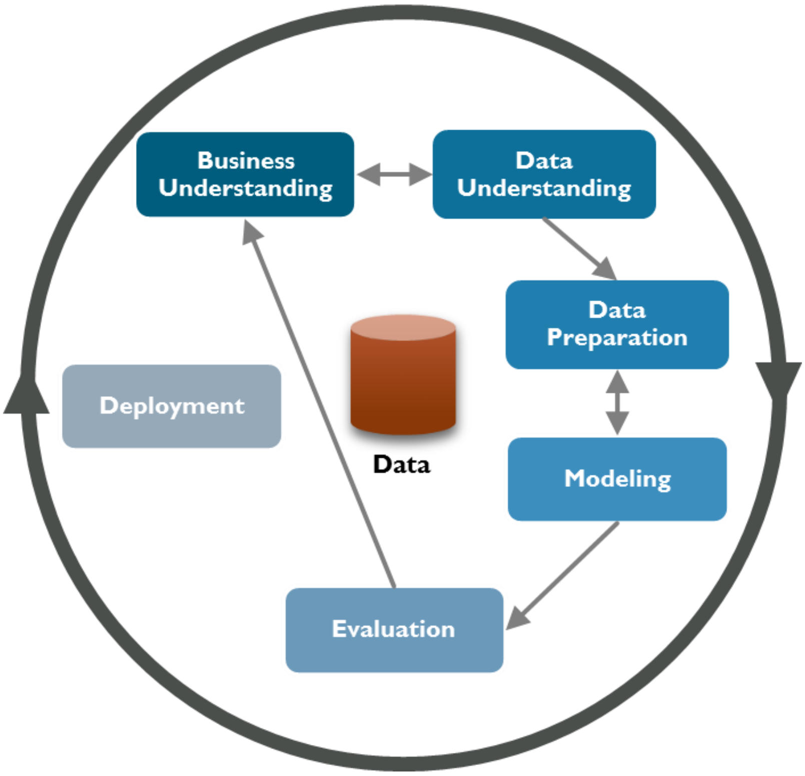


Fig 1

2. Data Exploration Planning and Analysis Requirement:

Tera scope terapixel data is subjected to preliminary analysis to better understand the data and provide information to business stakeholders. Based on the data set, this analysis of GPU cards and XY coordinates will aid in rendering Terapixel images in an efficient and effective manner.

2.1 Data Exploration Planning:

- Assessing the event types that dominate task runtimes.
- Discovering the relationship between GPU metrics.
- Identification of serial numbers of GPU with least performance.

- Interplay between GPU Temperature and Power Consumption.
- Variation in Memory Utilization Percentage of the GPU with GPU Temperature.
- Event Runtime for each co-ordinate of the Rendered Tile for level 8.

2.2 Analysis Plan and Requirement:

The analysis strategy for this report is to investigate the three data sets generated while different virtual machines render 3D images on 1024 GPU nodes during a run. This run is divided into three jobs to render the data visualization output, which show performance timing of the render application, performance of the GPU card, and details of which part of the image was rendered in each task. The requisite is to understand the data, then clean and preprocess the data before performing exploratory data analysis. This analysis will aid in the betterment of the rendering process.

2.3 Resources:

- Data sets ***application-checkpoint.csv, gpu.csv, task-x-y.csv***.
- Jupyter Notebook
- GitHub
- Microsoft Word

2.4 Impediment (Constraints):

- There could be insufficient accurate data to draw a convincing conclusion.
- The Timeframe in the datasets *gpu.csv* and *application checkpoints.csv* file varies slightly.
- Duplicate data in the data set may taint the outcome.
- The negative aspect of this analysis is the poor data quality.

3. Data Understanding:

3.1 Data Gathering and Description:

This analysis utilizes data from a dataset generated during a run with 1024 nodes. This analysis will be performed using three csv files *application-checkpoints.csv, gpu.csv, task-x-y.csv*. Below is the first three sets of data from the tables.

Table **application-checkpoints.csv:**

timestamp	**hostname**	**eventName**	**eventType**	**jobId**	**taskId**

2018-11-08T07:41:55.921Z	0d56a730076643d585f77e00d2d8521a00000N	Tiling	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	b47f0263-ba1c-48a7-8d29-4bf021b72043
2018-11-08T07:42:29.842Z	0d56a730076643d585f77e00d2d8521a00000N	Saving Config	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	20fb9fcf-a927-4a4b-a64c-70258b66b42d
2018-11-08T07:42:29.845Z	0d56a730076643d585f77e00d2d8521a00000N	Saving Config	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	20fb9fcf-a927-4a4b-a64c-70258b66b42d

- **timestamp** - Event start and stop time
- **hostname** - The virtual computer's hostname
- **eventName** - The name of the event that is taking place within the visualisation application.
  - **TotalRender -** is the entire task
  - **Render -** is when the image tile is being rendered
  - **Saving Config -** It’s just a metric for configuration overhead.
  - **Tiling** - is where post processing of the rendered tile is taking place
  - **Uploading -** is where the output from post processing is uploaded to Azure Blob Storage
- **eventType** – Start and Stop of the events.
- **jobID** – ID of the Azure batch job.
- **taskID -** The Azure batch task's identification number.

Table **gpu.csv:**

timestamp	**Hostname**	**gpuSerial**	**gpuUUID**	**powerDrawWatt**	**gpuTempC**	**gpuUtilPerc**	**gpuMemUtilPerc**
2018-11-08T08:27:10.314Z	8b6a0eebc87b4cb2b0539e81075191b900001C	3.23217E+11	GPU-1d1602dc-f615-a7c7-ab53-fb4a7a479534	131.55	48	92	53
2018-11-08T08:27:10.192Z	d8241877cd994572b46c861e5d144c85000000	3.23617E+11	GPU-04a2dea7-f4f1-12d0-b94d-996446746e6f	117.03	40	92	48

2018-11-08T08:27:10.842Z	db871cd77a544e13bc791a64a0c8ed50000006	3.23217E+11	GPU-f4597939-a0b4-e78a-2436-12dbab9a350f	121.64	45	91	44

- **gpuSerial -** The physical GPU card's serial number.
- **gpuUUID -** The unique system id assigned to the GPU unit by the Azure system.
- **powerDrawWatt -** The GPU's power consumption in watts.
- **gpuTempC -** GPU temperature in degrees Celsius.
- **gpuUtilPerc -** % utilization of the GPU Core (s).
- **gpuMemUtilPerc -** The percentage of GPU memory used.

Table **task-x-y.csv:**

taskId	**jobId**	**x**	**Y**	**level**
00004e77-304c-4fbd-88a1-1346ef947567	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	116	178	12
0002afb5-d05e-4da9-bd53-7b6dc19ea6d4	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	142	190	12
0003c380-4db9-49fb-8e1c-6f8ae466ad85	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	142	86	12

- **x -** The X co-ordinate of the image tile.
- **y -** The Y co-ordinate of the image tile.
- **level** – Zoomable level.

4. Data Preparation:

Following Data Understanding, the data is cleaned, the necessary datatypes are changed, and the data is preprocessed in preparation for further data analysis.

4.1 Data Pre-processing:

- All three csv files are read: **application-checkpoints.csv, gpu.csv,** and **task-x-y.csv.**

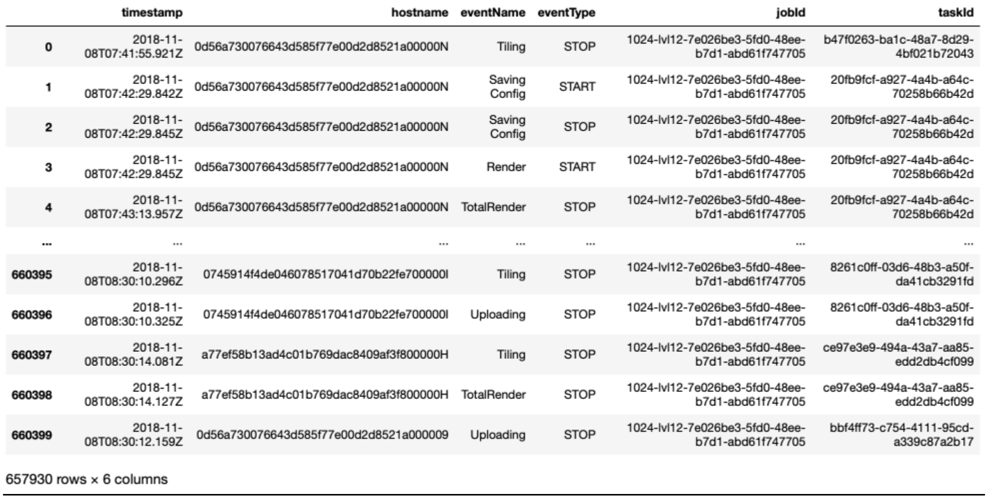
4.1.1 Data Cleaning:

- It is observed that all the three datasets contain no null values.
- The total number of duplicate records observed in the appcheckpoint, gpu, and xy tables is 2470,9, and 0.

- As part of data cleaning, duplicate records are removed from all three tables.

Below are the tables after data Cleaning:

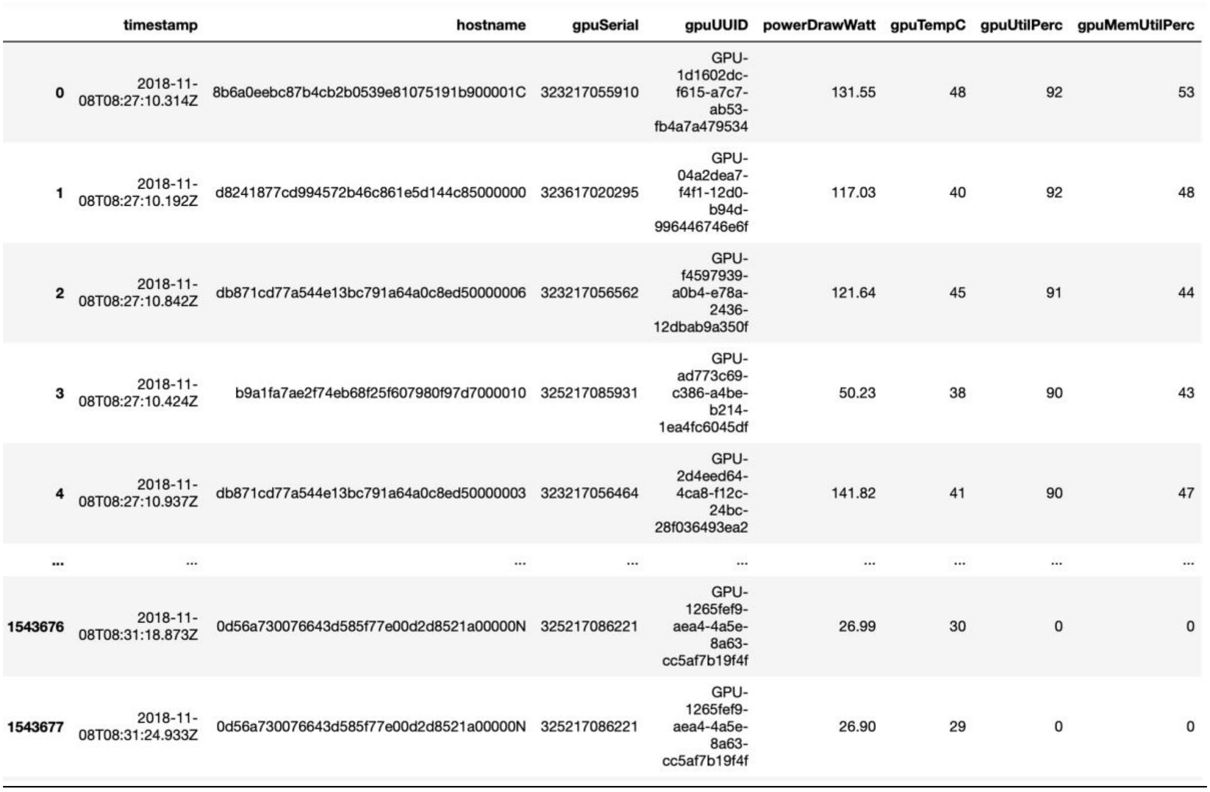
Table appcheck



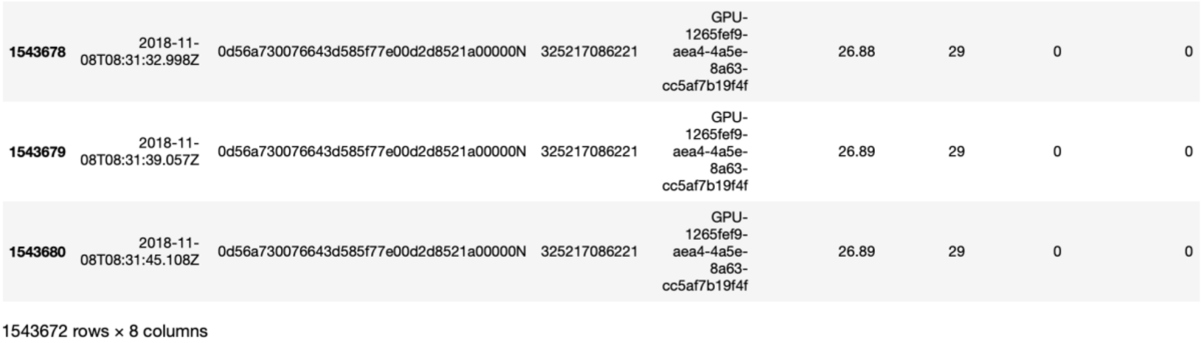
	timestamp	hostname	eventName	eventType	jobId	taskId
0	2018-11-08T07:41:55.921Z	0d56a730076643d585f77e00d2d8521a00000N	Tiling	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	b47f0263-ba1c-48a7-8d29-4bf021b72043
1	2018-11-08T07:42:29.842Z	0d56a730076643d585f77e00d2d8521a00000N	Saving Config	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	20fb9fcf-a927-4a4b-a64c-70258b66b42d
2	2018-11-08T07:42:29.845Z	0d56a730076643d585f77e00d2d8521a00000N	Saving Config	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	20fb9fcf-a927-4a4b-a64c-70258b66b42d
3	2018-11-08T07:42:29.845Z	0d56a730076643d585f77e00d2d8521a00000N	Render	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	20fb9fcf-a927-4a4b-a64c-70258b66b42d
4	2018-11-08T07:43:13.957Z	0d56a730076643d585f77e00d2d8521a00000N	TotalRender	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	20fb9fcf-a927-4a4b-a64c-70258b66b42d
...	...	...	...	...	...	...
660395	2018-11-08T08:30:10.296Z	0745914f4de046078517041d70b22fe700000I	Tiling	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	8261c0ff-03d6-48b3-a50f-da41cb3291fd
660396	2018-11-08T08:30:10.325Z	0745914f4de046078517041d70b22fe700000I	Uploading	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	8261c0ff-03d6-48b3-a50f-da41cb3291fd
660397	2018-11-08T08:30:14.081Z	a77ef58b13ad4c01b769dac8409af3f800000H	Tiling	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	ce97e3e9-494a-43a7-aa85-edd2db4cf099
660398	2018-11-08T08:30:14.127Z	a77ef58b13ad4c01b769dac8409af3f800000H	TotalRender	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	ce97e3e9-494a-43a7-aa85-edd2db4cf099
660399	2018-11-08T08:30:12.159Z	0d56a730076643d585f77e00d2d8521a000009	Uploading	STOP	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	bbf4ff73-c754-4111-95cd-a339c87a2b17

657930 rows × 6 columns

gpufinal



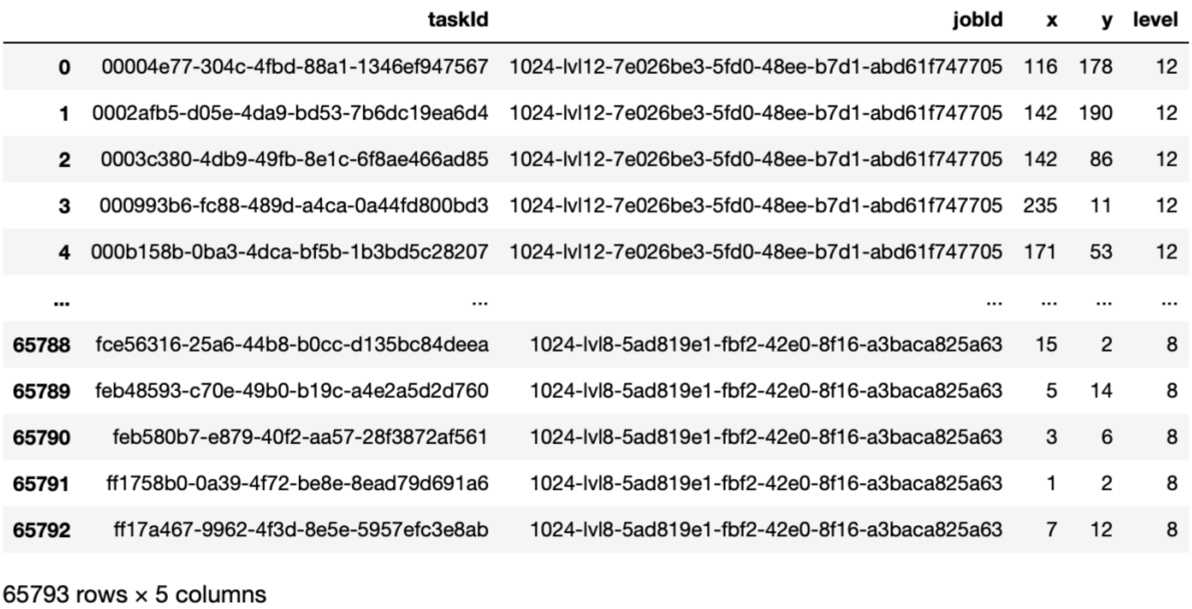
	timestamp	hostname	gpuSerial	gpuUUID	powerDrawWatt	gpuTempC	gpuUtilPerc	gpuMemUtilPerc
0	2018-11-08T08:27:10.314Z	8b6a0eebc87b4cb2b0539e81075191b900001C	323217055910	GPU-1d1602dc-f615-a7c7-ab53-fb4a7a479534	131.55	48	92	53
1	2018-11-08T08:27:10.192Z	d8241877cd994572b46c861e5d144c85000000	323617020295	GPU-04a2dea7-f4f1-12d0-b94d-996446746e6f	117.03	40	92	48
2	2018-11-08T08:27:10.842Z	db871cd77a544e13bc791a64a0c8ed50000006	323217056562	GPU-f4597939-a0b4-e78a-2436-12dbab9a350f	121.64	45	91	44
3	2018-11-08T08:27:10.424Z	b9a1fa7ae2f74eb68f25f607980f97d7000010	325217085931	GPU-ad773c69-c386-a4be-b214-1ea4fc6045df	50.23	38	90	43
4	2018-11-08T08:27:10.937Z	db871cd77a544e13bc791a64a0c8ed50000003	323217056464	GPU-2d4eed64-4ca8-f12c-24bc-28f036493ea2	141.82	41	90	47
...	...	...	...	...	...	...	...	...
1543676	2018-11-08T08:31:18.873Z	0d56a730076643d585f77e00d2d8521a00000N	325217086221	GPU-1265fef9-aea4-4a5e-8a63-cc5af7b19f4f	26.99	30	0	0
1543677	2018-11-08T08:31:24.933Z	0d56a730076643d585f77e00d2d8521a00000N	325217086221	GPU-1265fef9-aea4-4a5e-8a63-cc5af7b19f4f	26.90	29	0	0



1543678	2018-11-08T08:31:32.998Z	0d56a730076643d585f77e00d2d8521a00000N	325217086221	GPU-1265fef9-aea4-4a5e-8a63-cc5af7b19f4f	26.88	29	0	0
1543679	2018-11-08T08:31:39.057Z	0d56a730076643d585f77e00d2d8521a00000N	325217086221	GPU-1265fef9-aea4-4a5e-8a63-cc5af7b19f4f	26.89	29	0	0
1543680	2018-11-08T08:31:45.108Z	0d56a730076643d585f77e00d2d8521a00000N	325217086221	GPU-1265fef9-aea4-4a5e-8a63-cc5af7b19f4f	26.89	29	0	0

1543672 rows × 8 columns

xyfinal



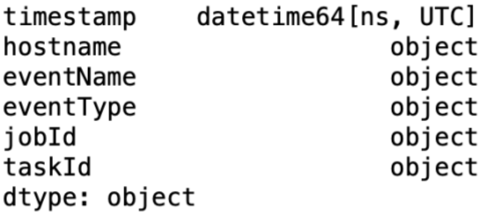
	taskId	jobId	x	y	level
0	00004e77-304c-4fbd-88a1-1346ef947567	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	116	178	12
1	0002afb5-d05e-4da9-bd53-7b6dc19ea6d4	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	142	190	12
2	0003c380-4db9-49fb-8e1c-6f8ae466ad85	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	142	86	12
3	000993b6-fc88-489d-a4ca-0a44fd800bd3	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	235	11	12
4	000b158b-0ba3-4dca-bf5b-1b3bd5c28207	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	171	53	12
...	...	...	...	...	...
65788	fce56316-25a6-44b8-b0cc-d135bc84deea	1024-lvl8-5ad819e1-fbf2-42e0-8f16-a3baca825a63	15	2	8
65789	feb48593-c70e-49b0-b19c-a4e2a5d2d760	1024-lvl8-5ad819e1-fbf2-42e0-8f16-a3baca825a63	5	14	8
65790	feb580b7-e879-40f2-aa57-28f3872af561	1024-lvl8-5ad819e1-fbf2-42e0-8f16-a3baca825a63	3	6	8
65791	ff1758b0-0a39-4f72-be8e-8ead79d691a6	1024-lvl8-5ad819e1-fbf2-42e0-8f16-a3baca825a63	1	2	8
65792	ff17a467-9962-4f3d-8e5e-5957efc3e8ab	1024-lvl8-5ad819e1-fbf2-42e0-8f16-a3baca825a63	7	12	8

65793 rows × 5 columns

4.1.2 Preprocessing:

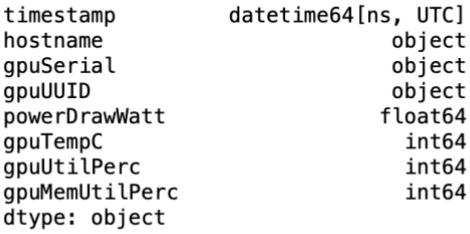
- To ensure clarity, the Time Stamp format in appcheckand gpu is changed from object to DateTime64.

Appcheck data frame datatype



```
timestamp    datetime64[ns, UTC]
hostname                  object
eventName                 object
eventType                 object
jobId                     object
taskId                    object
dtype: object
```

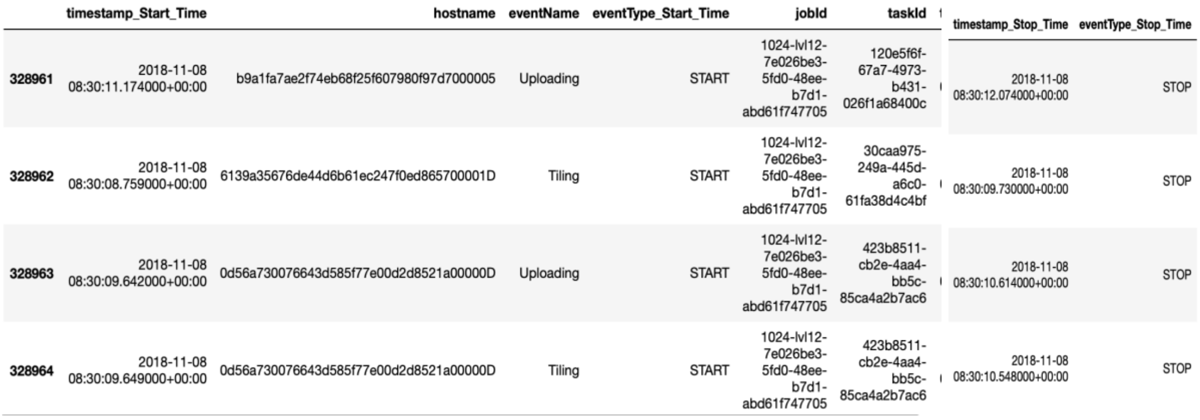
gpufinal data frame datatype



```
timestamp         datetime64[ns, UTC]
hostname                       object
gpuSerial                      object
gpuUUID                        object
powerDrawWatt                 float64
gpuTempC                        int64
gpuUtilPerc                     int64
gpuMemUtilPerc                  int64
dtype: object
```

- The gpuSerial column in the gpufinal table is converted from numeric to object format in order to perform additional analysis.
- The start and stop times is calculated by using the eventType from appcheck table.

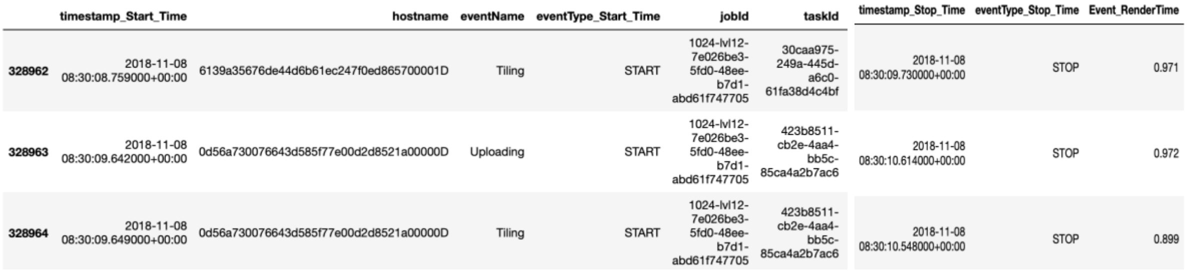
Below table shows the start and stop time calculated and displayed for last four rows of appcheck data frame:



	timestamp_Start_Time	hostname	eventName	eventType_Start_Time	jobId	taskId	timestamp_Stop_Time	eventType_Stop_Time
328961	2018-11-08 08:30:11.174000+00:00	b9a1fa7ae2f74eb68f25f607980f97d7000005	Uploading	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	120e5f6f-67a7-4973-b431-026f1a68400c	2018-11-08 08:30:12.074000+00:00	STOP
328962	2018-11-08 08:30:08.759000+00:00	6139a35676de44d6b61ec247f0ed865700001D	Tiling	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	30caa975-249a-445d-a6c0-61fa38d4c4bf	2018-11-08 08:30:09.730000+00:00	STOP
328963	2018-11-08 08:30:09.642000+00:00	0d56a730076643d585f77e00d2d8521a00000D	Uploading	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	423b8511-cb2e-4aa4-bb5c-85ca4a2b7ac6	2018-11-08 08:30:10.614000+00:00	STOP
328964	2018-11-08 08:30:09.649000+00:00	0d56a730076643d585f77e00d2d8521a00000D	Tiling	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	423b8511-cb2e-4aa4-bb5c-85ca4a2b7ac6	2018-11-08 08:30:10.548000+00:00	STOP

- Event Render Time is calculated based on the variation between start and stop EventType times.

Below table shows the Event Render Time computed:



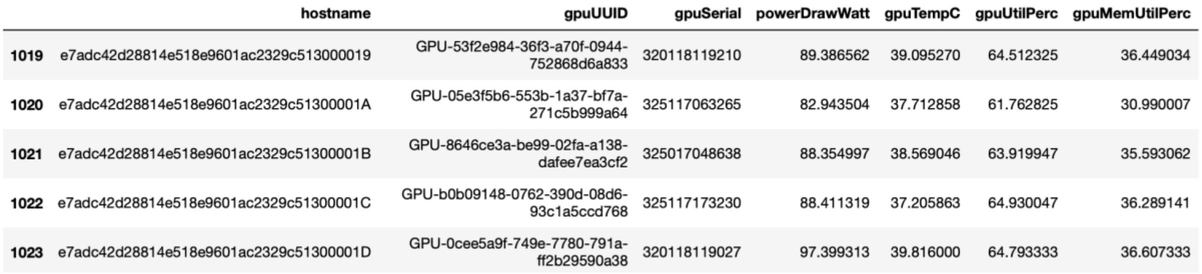
	timestamp_Start_Time	hostname	eventName	eventType_Start_Time	jobId	taskId	timestamp_Stop_Time	eventType_Stop_Time	Event_RenderTime
328962	2018-11-08 08:30:08.759000+00:00	6139a35676de44d6b61ec247f0ed865700001D	Tiling	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	30caa975-249a-445d-a6c0-61fa38d4c4bf	2018-11-08 08:30:09.730000+00:00	STOP	0.971
328963	2018-11-08 08:30:09.642000+00:00	0d56a730076643d585f77e00d2d8521a00000D	Uploading	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	423b8511-cb2e-4aa4-bb5c-85ca4a2b7ac6	2018-11-08 08:30:10.614000+00:00	STOP	0.972
328964	2018-11-08 08:30:09.649000+00:00	0d56a730076643d585f77e00d2d8521a00000D	Tiling	START	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	423b8511-cb2e-4aa4-bb5c-85ca4a2b7ac6	2018-11-08 08:30:10.548000+00:00	STOP	0.899

- For further interpretation, Event Render Time of appcheck table is sorted by Hostname.

First Three Rows of table based on hostname and Event Render Time is displayed:

hostname	**Event_RenderTime**
04dc4e9647154250beeee51b866b0715000000	44.350327
04dc4e9647154250beeee51b866b0715000001	45.698051
04dc4e9647154250beeee51b866b0715000002	40.275421

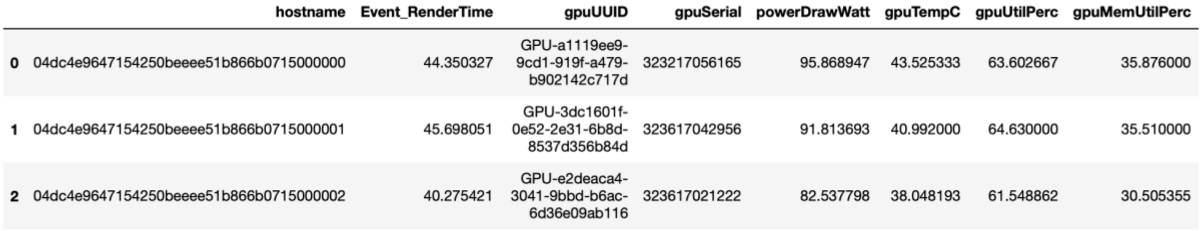
- Grouping gpufinal table based on hostname and other performance parameters:



	hostname	gpuUUID	gpuSerial	powerDrawWatt	gpuTempC	gpuUtilPerc	gpuMemUtilPerc
1019	e7adc42d28814e518e9601ac2329c513000019	GPU-53f2e984-36f3-a70f-0944-752868d6a833	320118119210	89.386562	39.095270	64.512325	36.449034
1020	e7adc42d28814e518e9601ac2329c51300001A	GPU-05e3f5b6-553b-1a37-bf7a-271c5b999a64	325117063265	82.943504	37.712858	61.762825	30.990007
1021	e7adc42d28814e518e9601ac2329c51300001B	GPU-8646ce3a-be99-02fa-a138-dafee7ea3cf2	325017048638	88.354997	38.569046	63.919947	35.593062
1022	e7adc42d28814e518e9601ac2329c51300001C	GPU-b0b09148-0762-390d-08d6-93c1a5ccd768	325117173230	88.411319	37.205863	64.930047	36.289141
1023	e7adc42d28814e518e9601ac2329c51300001D	GPU-0cee5a9f-749e-7780-791a-ff2b29590a38	320118119027	97.399313	39.816000	64.793333	36.607333

- A TP data frame is created by combining host PT and gpu PF based on the hostname.

TP dataframe:



	hostname	Event_RenderTime	gpuUUID	gpuSerial	powerDrawWatt	gpuTempC	gpuUtilPerc	gpuMemUtilPerc
0	04dc4e9647154250beeee51b866b0715000000	44.350327	GPU-a1119ee9-9cd1-919f-a479-b902142c717d	323217056165	95.868947	43.525333	63.602667	35.876000
1	04dc4e9647154250beeee51b866b0715000001	45.698051	GPU-3dc1601f-0e52-2e31-6b8d-8537d356b84d	323617042956	91.813693	40.992000	64.630000	35.510000
2	04dc4e9647154250beeee51b866b0715000002	40.275421	GPU-e2deaca4-3041-9bbd-b6ac-6d36e09ab116	323617021222	82.537798	38.048193	61.548862	30.505355

- xyfinal1 is made by combining xyfinal and appcheck_start_stop on the basis of Total Render (Entire task).
- The xy_level8 data frame is created for the level 8 image by grouping Event Render Time by x and y coordinates.

x	y	Event_RenderTime
0	0	43.853001
0	1	24.809999
0	2	27.111000
0	3	35.481998
0	4	43.898998

- The Level RT table is created for calculating Average Event Render Time for the level 4, 8, and 12.

level	Event_RenderTime
4	52.181999
8	48.45982
12	42.58176

The tables listed above will be used for additional data analysis.

5. Exploratory Data Analysis:

Exploratory Data Analysis is used to analyze data and envision models on Terapixel preprocessed data. This analysis will help to improve the image rendering process.

5.1 Data Exploration:

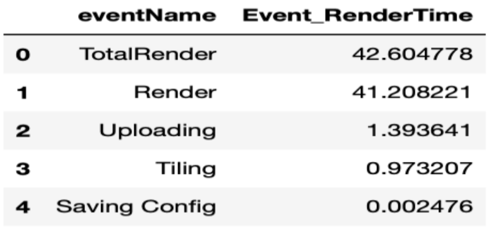
Data is explored based on the below scenarios

5.1.1 Assessing the event types that dominate task runtimes:

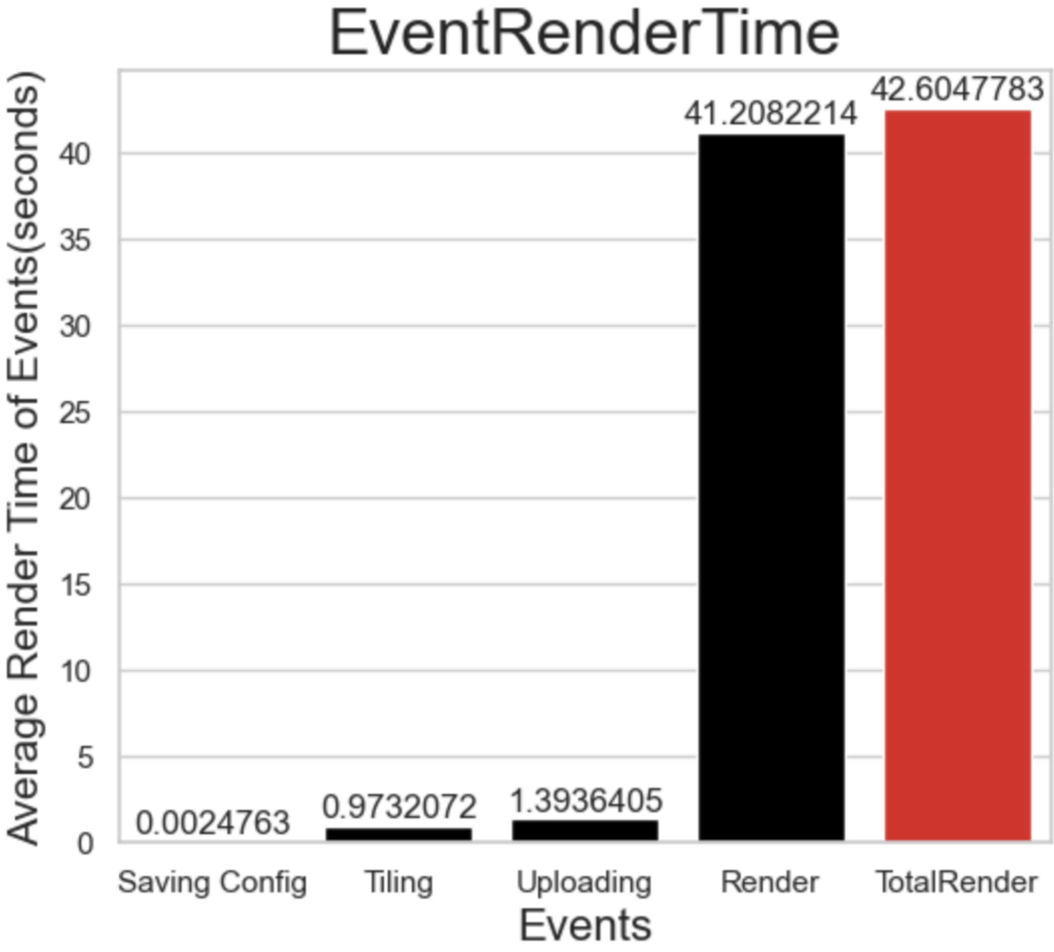
There are five events occurring within the visualization process:

- **Total Render -** is the entire task
- **Render -** is when the image tile is being rendered
- **Saving Config** - It’s just a metric for configuration overhead.
- **Tiling** - is where post processing of the rendered tile is taking place
- **Uploading -** is where the output from post processing is uploaded to Azure Blob Storage

The table below shows the average time taken by all events in seconds:

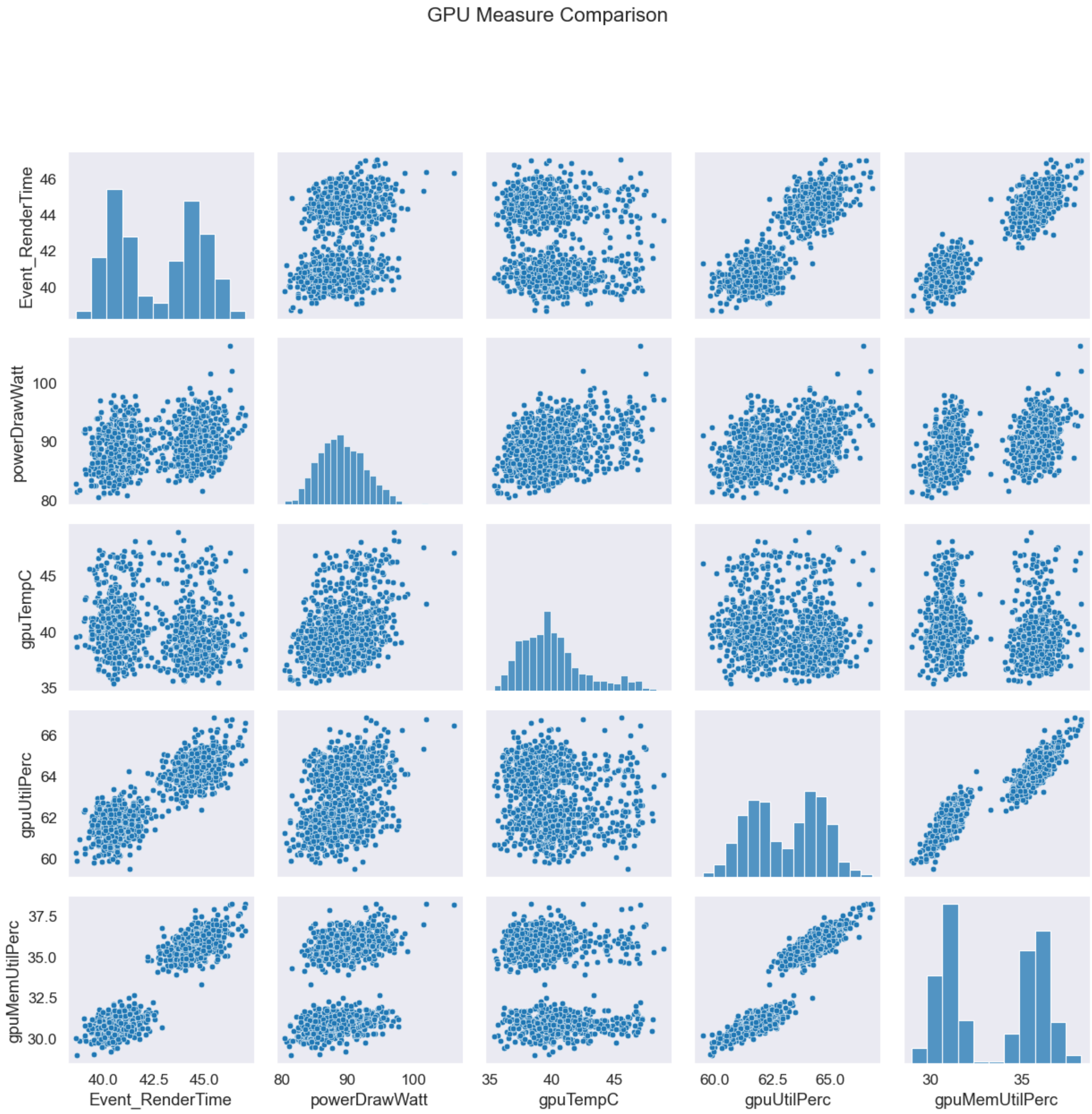


	eventName	Event_RenderTime
0	TotalRender	42.604778
1	Render	41.208221
2	Uploading	1.393641
3	Tiling	0.973207
4	Saving Config	0.002476



The above bar plot depicts the average render time for all events. The plot shows that **Total Render** takes **42.6047** seconds of render time, which is the average time required to render the entire task. The **Render** event takes the most time, with an average of **41.2082** seconds, followed by **Uploading** and **Tiling**, which take **1.3936** and **0.9732** seconds, respectively. **Saving Config** event takes only **0.00247** seconds, which is the shortest average time when compared to other events.

5.1.2 Correlation between GPU metrics:



The pair plot above depicts the correlation between all GPU metrics:

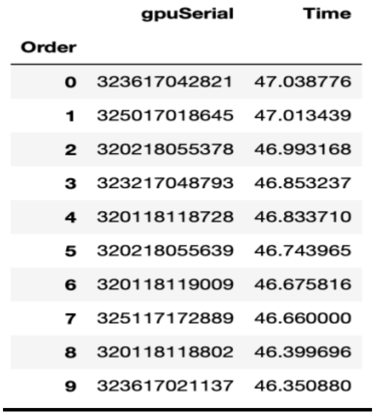
- Event_Render Time
- powerDrawWatt
- gpuTempC

- gpuUtilPerc
- gpuMemUtilPerc

GPU utility percentage and GPU memory utility percentage are found to correlate with each other, according to the plot, as GPU utilization keeps increasing with memory utilization. Furthermore, we can deduce that an increase in memory utilization can result in a long event render time. GPU power draw and temperature have the least significant relation. It's also worth noting that render time is inversely related to GPU temperature, so as Event Render time increases, so does GPU temperature.

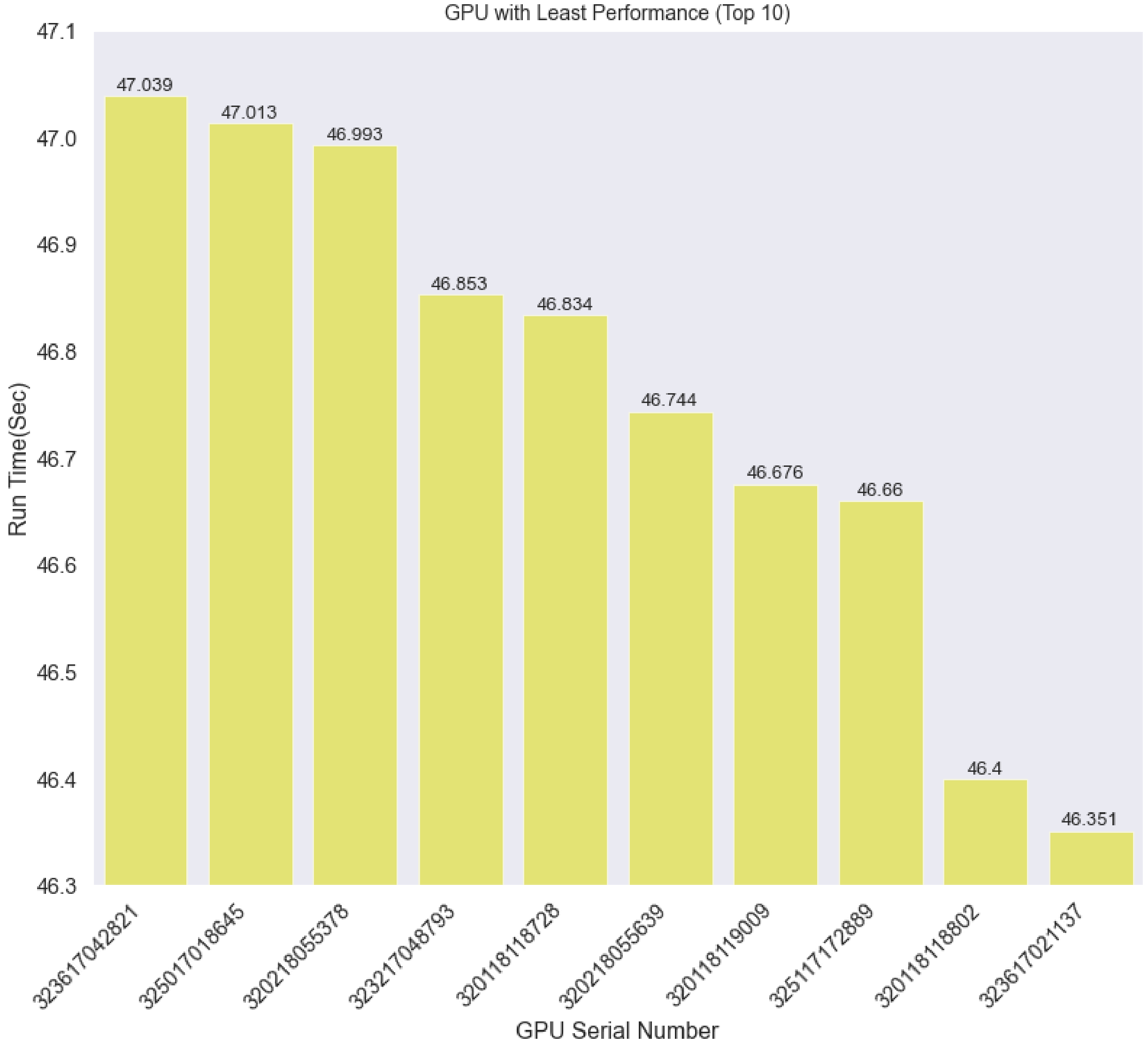
5.1.3 <u>Identification of serial numbers of GPU with least performance:</u>

The table that follows lists the unique identifiers of the list of top ten GPUs with the least performance.



	gpuSerial	Time
Order		
0	323617042821	47.038776
1	325017018645	47.013439
2	320218055378	46.993168
3	323217048793	46.853237
4	320118118728	46.833710
5	320218055639	46.743965
6	320118119009	46.675816
7	325117172889	46.660000
8	320118118802	46.399696
9	323617021137	46.350880

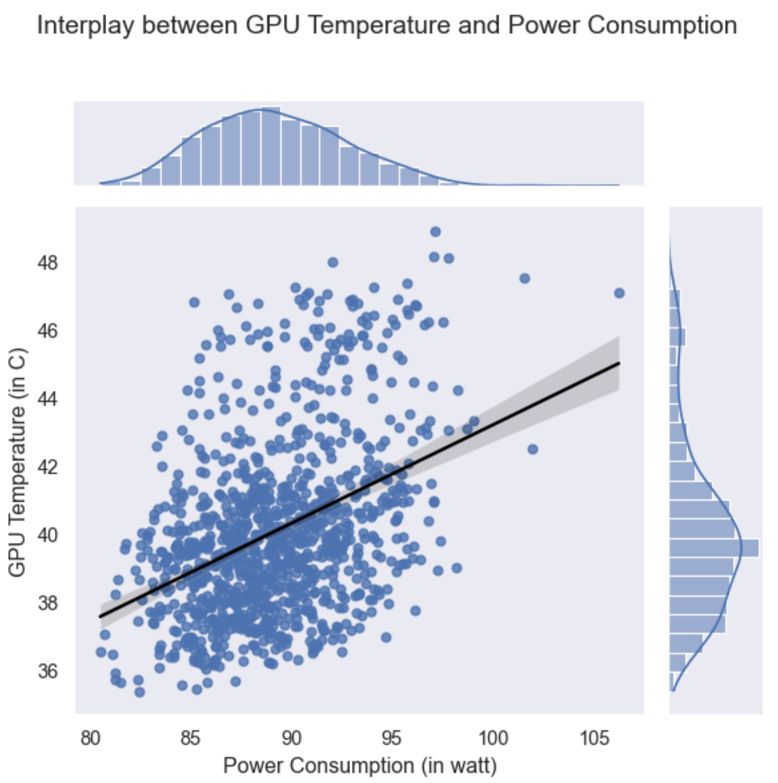
The bar plot below depicts GPU card runtime. The GPU card with serial number 2821 has the worst efficiency, with a mean render time of 47.038776 seconds, followed by the GPU card with serial number 8645, which has a render time of 47.013439 seconds. Leaving GPU card ending with serial number 5378 with a 0.020271 second delay in render time from GPU card ending with serial number 5378. Leaving GPU cards 8802 and 1137 with render times of 46.399696 and 46.350880 seconds, respectively, with slightly better performance than the other GPU cards.



5.1.4 <u>Interplay between GPU Temperature and Power Consumption:</u>

The below joint plot shows Interplay between GPU temperature and Power Consumption. It is inferred from the below plot that Power Consumption is directly proportional to GPU temperature as the Power Consumption increased the Temperature of the GPU increases linearly.

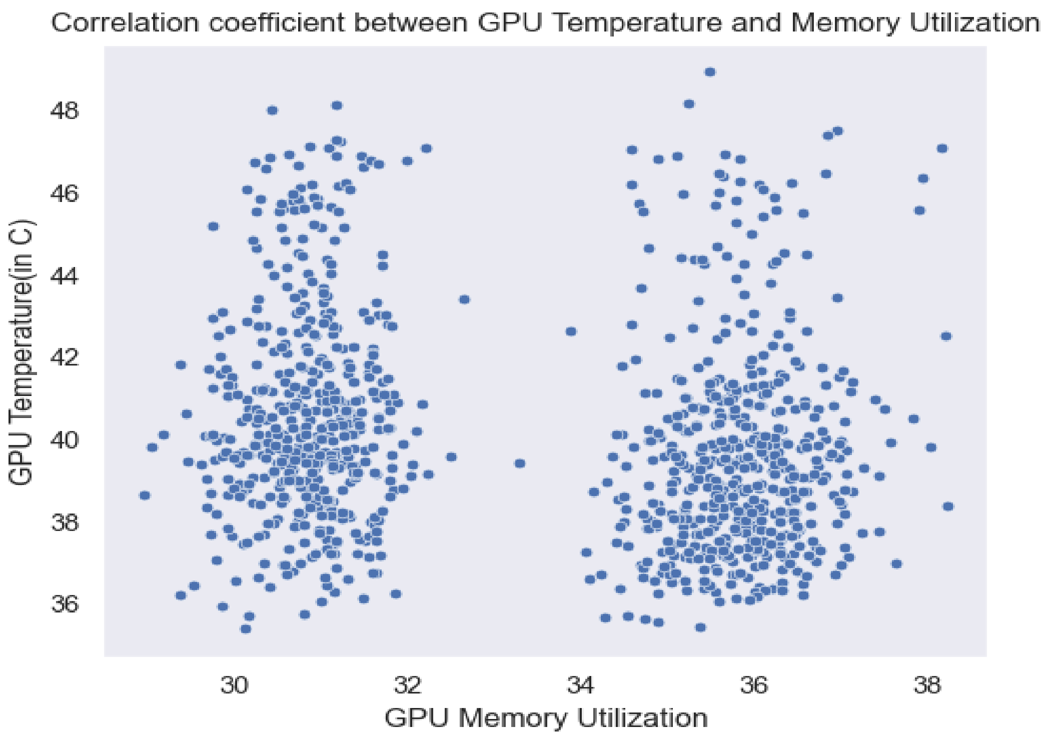
The Correlation coefficient between GPU Temperature and Power Consumption = **0.4**



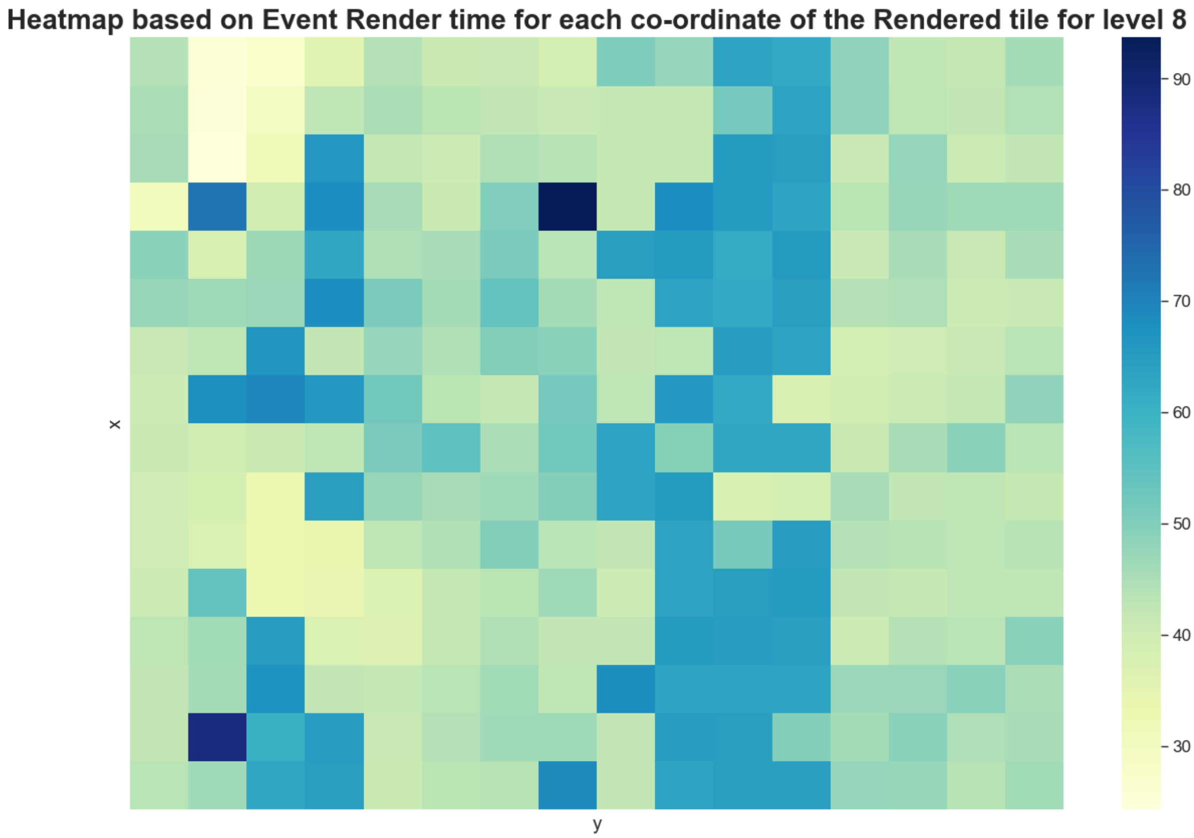
5.1.5 Variation in Memory Utilization Percentage of the GPU with GPU Temperature:

The scatter plot below depicts the variation in GPU Memory Utilization Percentage with temperature. It depicts two clusters that formed in the same temperature range. When the Memory Utilization Percentage ranges from 28 to 32 and 34 to 37, the GPU temperature varies. There is no linearity between the two. There is no significant relationship between GPU Memory Utilization Percentage and temperature because they are independent of one another.

The Correlation Coefficient between GPU Temperature and Memory Utilization = **-0.18**



5.1.6 Event Render time for each co-ordinate of the Rendered Tile (Level 8):



The heatmap for Event Render Time for each coordinate of the rendered tile for level 8 is shown above. Shades of green to blue represent tiles with steadily rising event render time, while shades of green represent tiles with average event render time and shades of yellow tiles represent areas with very little time taken for rendering.

6. Conclusion:

The following are the conclusions of exploratory data analysis:

- When the event types that dominate task runtimes are examined, it is discovered that the Render event takes the maximum. Reducing the time taken for render event can improve the efficiency in rendering.
- Analysis shows GPU utility percentage and GPU memory utility percentage are linearly proportional to each other.
- The GPU card with serial number 2821 has the lowest efficiency and requires the most render time. Improving GPU performance can aid in reducing rendering time.
- Power consumption is directly proportional to GPU temperature; as power consumption increases, so does GPU temperature.
- There is no significant relationship between GPU Memory Utilization Percentage and temperature because they are independent of one another.
- There is no significant relationship between GPU Memory Utilization Percentage and temperature because they are independent of one another.
- Heatmap pattern for Event Render Time for level 8 helps to focus on Render Time and improve the efficiency in rendering the image.

The observations made above can help and provide insights into ways to improve render time and efficiency in rendering images.

7. References:

- http://www.decisivefacts.nl/2016/08/crisp-dm-process-for-data-mining/?lang=en
- https://github.com/NewcastleDataScience/StudentProjects202223/blob/master/TeraScope/Summary.md
- cdcadcsdcasxas

8. Git Log: