2023

PERFORMANCE EVALUATION OF TERAPIXEL RENDERING IN CLOUD COMPUTING

210409183

AKHIL RAJ RAJAN (PGT)

Contents

List of Tabes

List of Figures

1. INTRODUCTION

Tera pixel images provide stakeholders with a convenient method of presenting information sets, enabling users to interactively navigate big data at various scales. The main challenge is how to achieve the supercomputer scale resources needed to create a genuine terapixel visualisation of Newcastle upon Tyne and its environmental data as collected by the Newcastle urban observatory.

The main objective of the report is to conduct a performance analysis of Terapixel rendering on a cloud-based supercomputer which process intensive visualisation application using TeraScope data set.

It is feasible to produce a high-quality terapixel visualisation by the path tracing renderer using public IaaS cloud GPU nodes. The Tera pixel image support interactive browsing of the city and data can be accessed across a wide range of team client devices.

TeraScope data set is generated from application checkpoints and system metric output from the production of terraced pixel images. With the help of these data, performance evaluation can be carried out using exploratory Data analysis by focusing on the below main areas.

1. Time consumed for each event type.
2. Analysis based on hostname, task Id regarding the GPU resources and the properties.
3. study showing how the GPU properties and time taken for rendering related to each tiles or task id.

In this report, these areas will be analysed critically, and the information obtained is interpreted to generate useful insights from the data.

CRISP-DM methodology is followed for exploratory analysis and answering the main questions.

2. METHODOLOGY

BUSINESS UNDERSTANDING

The purpose of analysing the data is to help stakeholders make the appropriate judgment for smooth and effective visualisation of Newcastle city by optimizing computational resources. The information derived from this raw data will help the technical teams (cloud engineers, architect etc) to design the system there by satisfying the requirement of business. The following questions will help to generate sensible conclusion from the data through exploratory analysis.

1. Which task Id took the highest time as well as lowest time for Total rendering, Rendering, tiling, uploading and saving config processes and their corresponding rendered image coordinates?
2. Which Event Name consumes more run time?
3. Is there any relationship between the variables in the GPU table? If so, how are they related?
4. Can any particular statistical model be fitted to any variables related to GPU performance matrix? If so, describe the model.
5. What are the host names which consumed maximum and minimum time for the rendering process?
6. What are the maximum and minimum values of computational resource used, temperature and power consumption of the GPU and identify the corresponding virtual machines (hostname)?

7. Which virtual machine processed most image rendering tasks? explain this with the help of a histogram.
8. How many image coordinates are associated with each level?
9. How are the tiles of the image and total rendering times related?
10. Explain the GPU properties using suitable graphs on the basis of hostnames and Task Id?
11. How are the GPU properties related to the tile properties of the rendered image?

DATA UNDERSTANDING

The provided data shows the performance timing of the render applications as well the performance of the GPU card, conveying the details of which part of the image rendering in each task while performing a run using 1024 GPU nodes.

1. Application-checkpoints

Field Names	Data Types	Description	Example
Time stamp	Time stamp	Shows the time for a particular event	2018-11-08T07:42:29.845Z
hostname	String	Host name of the virtual machine	0d56a730076643d585f77e00d2d8521a00000N
eventName	String	Name of the event occurring within the rendering application	Render
eventType	String	indicate whether the process starts or stops	START
jobId	String	ID of the Azure Batch job	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705
taskId	String	ID of the Azure Batch task	0002afb5-d05e-4da9-bd53-7b6dc19ea6d4

Table 1 Application- check point overview

- Under each task Id there are five processes namely saving config, Render, Tiling and uploading. Total render denotes the sum total of all this process. In conclusion, the data set shows the start time and stop time of each process for a particular taskId , Hostname and Job id.
- Primary keys for this table are task Id, event name and event Type
- Foreign keys are task Id and Hostname.

2. GPU

Field Name	Data Type	Description	example
Time stamp	Time stamp	Recoded Time	2018-11-08T08:27:10.424Z
hostname	String	Host name of the virtual machine	db871cd77a544e13bc791a64a0c8ed50000003
gpuSerial	String	The serial number of the physical GPU card	323217056464
gpuUUID	String	The unique system Id assigned by the Azure system to the GPU unit	GPU-2d4eed64-4ca8-f12c-24bc-28f036493ea2
powerDrawWatt	Number	Power draw of the GPU in watts	24.5
gpuTempC	Number	Temparature of the GPU in celcius	44
gpuUtilPerc	Number (%)	Percentage utilization of the GPU memory	88
gpuMemUtilPerc	Number (%)	Percentage utilization of the GPU cores	43

Table 2 GPU overview

- The table mainly shows the quantity of system resources used by each hostname for a particular time. In addition, it also indicates the temperature and power utilization of the core for each time.
- Primary key is GPUSrial/GPUUUID and the foreign key is the hostname.

- For each Hostname, GPUSrial and GPUUUID are unique.

3. Task-x-y

field name	Data type	Description	example
taskId	String	Id of the Azure Batch task	00004e77-304c-4fbd-88a1-1346ef947567
jobId	String	Id of the Azure Batch job	1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705
x	Number	Rendered image Y axis	116
y	Number	Rendered image X axis	118
level	Number	It represents the zooming level	12

Table 3 Task-x-y table overview

- This table shows the x, y coordinate of the part of rendered image under each tasked
- Primary Keys is task Id and foreign keys are task Id and job Id
- There are three levels of image rendering based on the zoom feature they are 4,8 and 12. Each level falls under 3 separate job Id.

Job Id	level
1024-lvl12-7e026be3-5fd0-48ee-b7d1-abd61f747705	12
1024-lvl4-90b0c947-dcfc-4eea-a1ee-efe843b698df	4
1024-lvl8-5ad819e1-fbf2-42e0-8f16-a3baca825a63	8

Table 4 Job ID vs Level

Overall, there are 3 distinct job Id, 1024 unique Host Name and 65793 distinct task Id.

DATA PREPARATION

1. New column is added to the Application-checkpoints table to indicate the time taken for each event Name. It is named “delta_dttm”.
2. The original field name Timestamp is renamed to dttm to avoid confusion with the data type
3. The field delta_dttm is converted to timestamp/float (seconds) depending on the requirement.
4. There are 2470 duplicates in the Application-checkpoints table. They were removed before processing the data.
5. There are also 9 duplicates in the GPU table. Here, the duplicates are eliminated using suitable queries.
6. The same pre-processing steps for the Application-checkpoint table are used in GPU as well like renaming the column ‘Timestamp’ to ‘dttm’ and changing the data type to timestamp/float (seconds) based on the requirements.
7. For the Task-x-y table there are no duplicates. So, renaming of the column was done.

MODELLING

Exploratory analysis is implemented on these data sets to generate useful information which could represent the performance evaluation of the process and system. In this report, as per the objectives, significant questions will be framed, and they are answered with corresponding tables or figures.

For carrying out the analysis, Python programming language is used and since it is difficult to analyse the data using python framework, panda SQL module is installed for easy analysis using MySQL query language.

The script for the analysis is generated through google colab which offers a markdown feature to create a pdf containing the script as well as the text for efficient communication and for reproducibility.

ASSUMPTIONS

1. Each event starts and stops consecutively which means there is no time gap between succeedi ng events. (In actual case there is a minute time gap)
2. Only one task Id is executed at a single point in time of a virtual machine.
   In some hostname multiple events is executed in the same timeframe). So, it is difficult to fin d the GPU properties for a particular task Id.

METHOD FOR REPRESENTING PERFORMANCE PARAMETER

To evaluate the performance of the virtual machine or GPU, the time taken for a particular activity by each unit is calculated. Then, the assumption is the performance is higher for the machines which take less time to complete the task.

The table _ can be used for finding the time taken for each evetime under each task id we can use the following method.

dttm	eventName	eventType	taskId	dttm_delta
07:45:14	Render	START	000993b6-fc88-489d-a4ca-0a44fd800bd3	00:00:39
07:45:53	Render	STOP	000993b6-fc88-489d-a4ca-0a44fd800bd3	
07:45:14	Saving Config	START	000993b6-fc88-489d-a4ca-0a44fd800bd3	00:00:00
07:45:14	Saving Config	STOP	000993b6-fc88-489d-a4ca-0a44fd800bd3	
07:45:53	Tiling	START	000993b6-fc88-489d-a4ca-0a44fd800bd3	00:00:01
07:45:54	Tiling	STOP	000993b6-fc88-489d-a4ca-0a44fd800bd3	
07:45:14	TotalRender	START	000993b6-fc88-489d-a4ca-0a44fd800bd3	00:00:41
07:45:55	TotalRender	STOP	000993b6-fc88-489d-a4ca-0a44fd800bd3	
07:45:53	Uploading	START	000993b6-fc88-489d-a4ca-0a44fd800bd3	00:00:01
07:45:54	Uploading	STOP	000993b6-fc88-489d-a4ca-0a44fd800bd3	

Table 5 Table derived from Application-checkpoint

For example, in the above table, the time taken for the Render process is calculated by subtracting the time corresponding to the start event type from the stop event type that is:

07:45:53-07:45:14 = 00:00:39

From the table, we can understand that the eventType saving config took less time compared to the other process and if we extend this method across different hostnames and compare the time taken for a particular task with similar complexities, the performance of the virtual machine can be analysed.

TABLES JOINING CONDITION

1. The tables Application-checkpoints and GPU can be joined using the Hostname as a foreign key to analysing machine performance, GPU temperature, GPU power and system resources.
2. The tables Application-checkpoints and Task-x-y can be joined using job Id and task Id which will help to explore the coordinate analysis and level-wise analysis.
3. Application-checkpoint (a) and GPU (b) can be joined by hostname, and the timestamp of the GPU table should be in between the start timestamp and end timestamp of the Application-checkpoint table where:
   start timestamp column is obtained using even type as START for a particular task Id and event name as Total Render. similarly, end timestamp column is obtained using even type as **STOP for a particular task Id and event name as Total Render
   The query condition is:
   (a.start timestamp ≤ b.timestamp ≤ a.end timestamp and a.hostname=b.hostname) . The result can be joined with the last table using taskId column.

3. RESULTS AND DISCUSSION

The Questions we are going answer through to Data exploration, methods of finding the answer are explained in this session through suitable data visualisation.

1. *Which task Id took the highest time as well as lowest time for Total rendering, Rendering, tiling, uploading and saving config processes and their corresponding rendered image coordinates?*
   Note the minimum time taken denote improved performance

Event Name	Task Id	x	y	level	Time taken (in seconds)	Comments
Render	0849dfbf-51a2-43d3-b0e4-bfa11f830010	30	21	12	22.58	Minimum
	a95d501e-d5d5-4fb4-9119-98120bf6f4d5	91	105	12	81.51	Maximum
Saving config	5140e07a-71fb-4b6c-ad80-c0695b5a626e	13	14	12	0	Minimum
	59ac7676-f371-4eee-aa67-5f7c7daf40dc	174	41	12	0.46	Maximum
Tiling	02029980-be9c-401f-b7ff-2313fa2a495b	41	0	12	0.68	Minimum
	910066f8-7f62-46ff-bab5-7dc2bdf4aadd	166	89	12	1.26	Maximum
Uploading	37ebe851-9042-49e3-9e81-6443603a98ab	20	31	12	0.72	Minimum
	83064f91-5a19-4526-8673-38ab28dd3ab7	14	1	8	43.52	Maximum
Total Render	bb205a5e-251e-4349-b8b0-3402a57e357e	2	32	12	23.37	Minimum
	ef15022d-f816-4434-b41e-709cb996bc08	3	7	8	93.7	Maximum

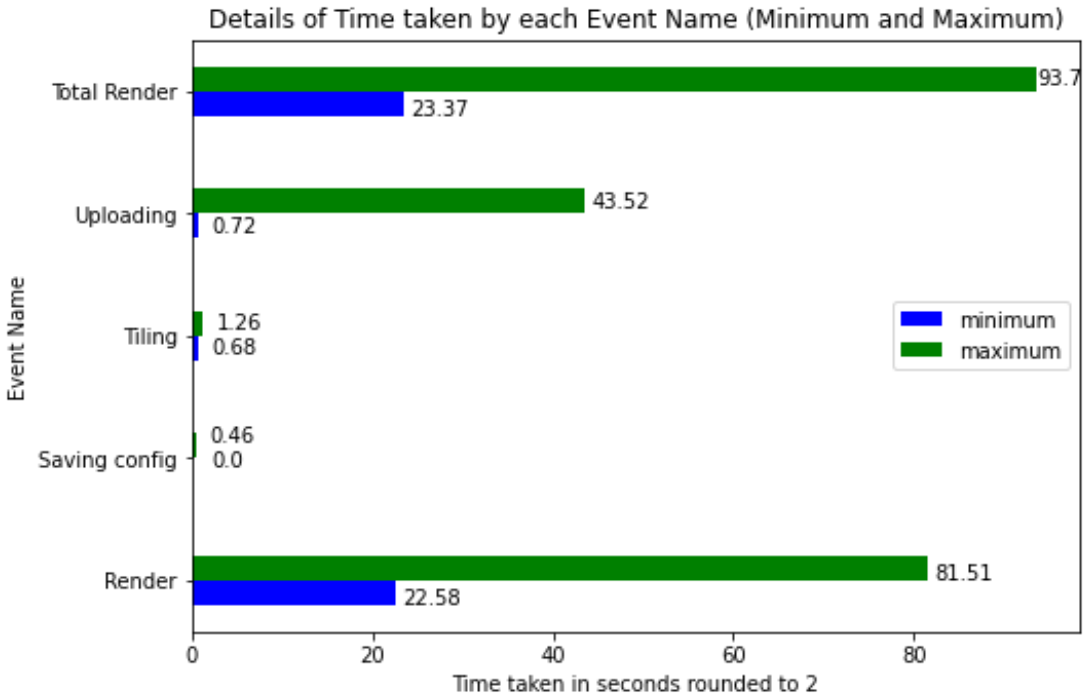

Figure 1 Details of Time taken by each event

Comments: The task Id **ef15022d-f816-4434-b41e-709cb996bc08** took the highest time (93.7 seconds) for Total Rendering of the image coordinates **(x, y) = (3,7**) under level 8 and the task Id **bb205a5e-251e-4349-b8b0-3402a57e357e** took the lowest time (23.57 seconds) for Total Rendering the image coordinates **(x, y) = (2,32)** under level 12.

Task d **83064f91-5a19-4526-8673-38ab28dd3ab7** took the highest time (43.52 seconds) for Uploading the image coordinates **(x, y) = (14,1**) under level 8 and the task Id **37ebe851-9042-49e3-9e81-6443603a98ab** took the lowest time (0.72 seconds) for Uploading the image coordinates **(x, y) = (20,31)** under level 12.

The task Id **910066f8-7f62-46ff-bab5-7dc2bdf4aadd** took the highest time (1.26 seconds) for Tiling the image coordinates **(x, y) = (166,89**) under level 12 and task Id **02029980-be9c-401f-b7ff-2313fa2a495b** took the lowest time (0.68 seconds) for Tiling the image coordinates **(x, y) =(41,0)** under level 12.

The task Id **59ac7676-f371-4eee-aa67-5f7c7daf40dc** took the highest time (0.46 seconds) for Saving configuration of the image coordinates **(x, y) = (174, 41**) under level 12 and task Id **5140e07a-71fb-4b6c-ad80-c0695b5a626e** took the lowest time (0.002 seconds) for Saving configuration the image coordinates **(x, y) = (13,14)** under level 12.

The task Id **a95d501e-d5d5-4fb4-9119-98120bf6f4d5** took the highest time (81.51 seconds) for Rendering the image coordinates **(x, y) = (91,105**) under level 12 and task Id **0849dfbf-51a2-43d3-b0e4-bfa11f830010** took the lowest time (22.58 seconds) for Rendering the image coordinates **(x, y) = (30,21)** under level 12.

Most importantly the total time taken for Total rendering the image is 48 Minutes and 45 Seconds.

2. *Which Event Name consumes more run time?*

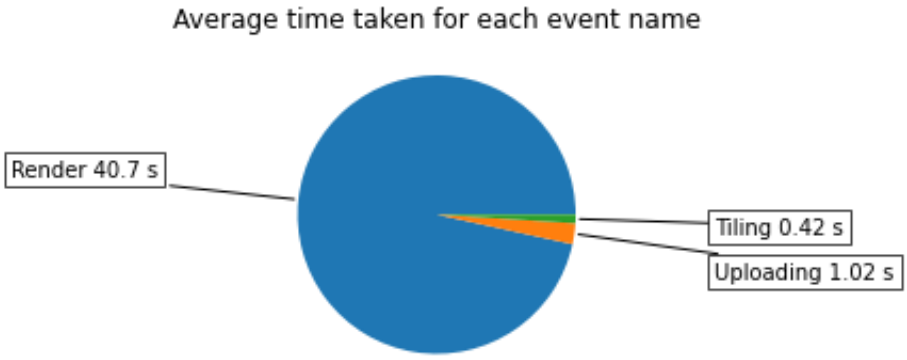

Figure 2 Pie chart showing the average time taken for each job name

Comments: The above bar plot shows that Rendering consumes more run time whereas saving config process consumed the least run time. Total Render will always consume more time as it is the sum that includes all four processes.

3. *Is there any relationship between the variables in the GPU table? If so, how are they related?*

For answering this question, the averages of the numerical variables are used after grouping based on Hostname.

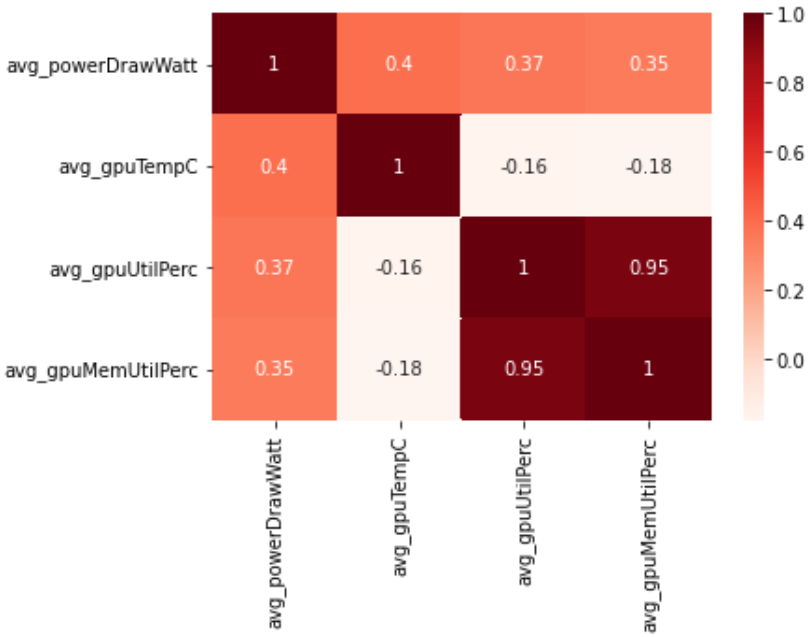

Figure 3 Heat map of Variables

Comments: The heat plot shows there is a strong relationship between the average GPU utilization percentage and average memory utilization percentage. There are no other significant relationships that can be identified from the heat plot.

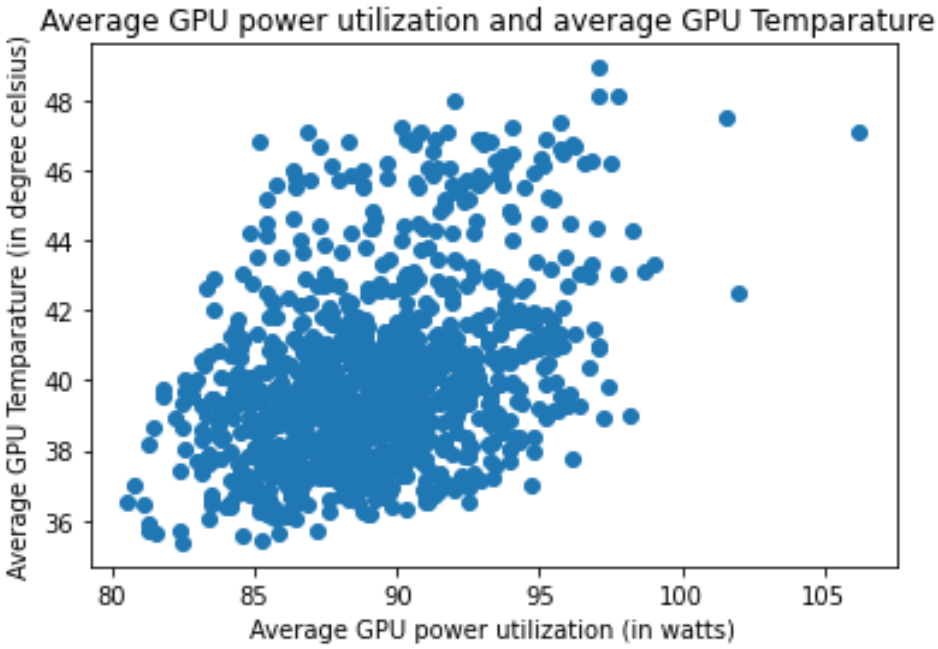

Figure 4 GPU power Utilization vs GPU temperature

Comments: The above figure shows that there is a slight increase in temperature as the power consumption increases.

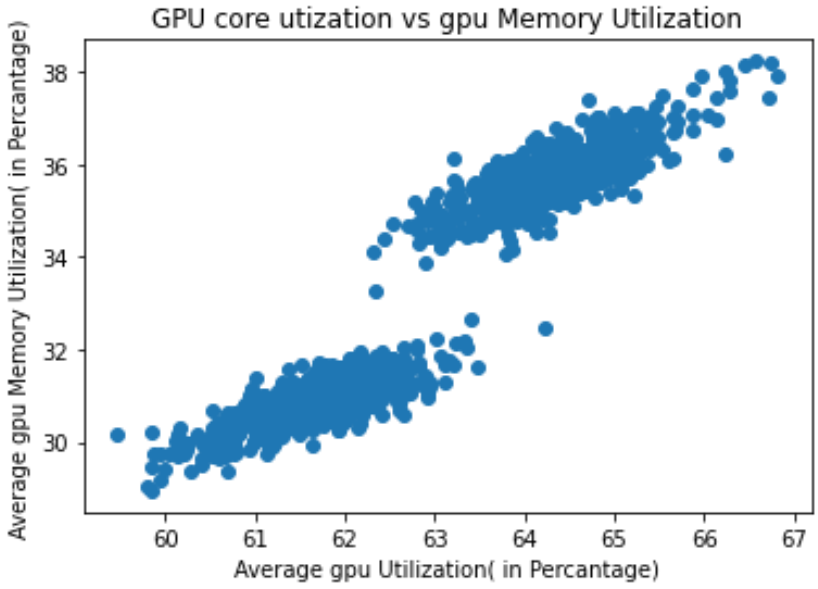

Figure 5 GPU core utilization vs GPU Memory Utilization

Comments: The scatter plot shows that as the core utilization increases memory utilization also increases.

4. *Can any particular statistical model be fitted to any variables related to GPU performance matrix? If so, describe the model.*

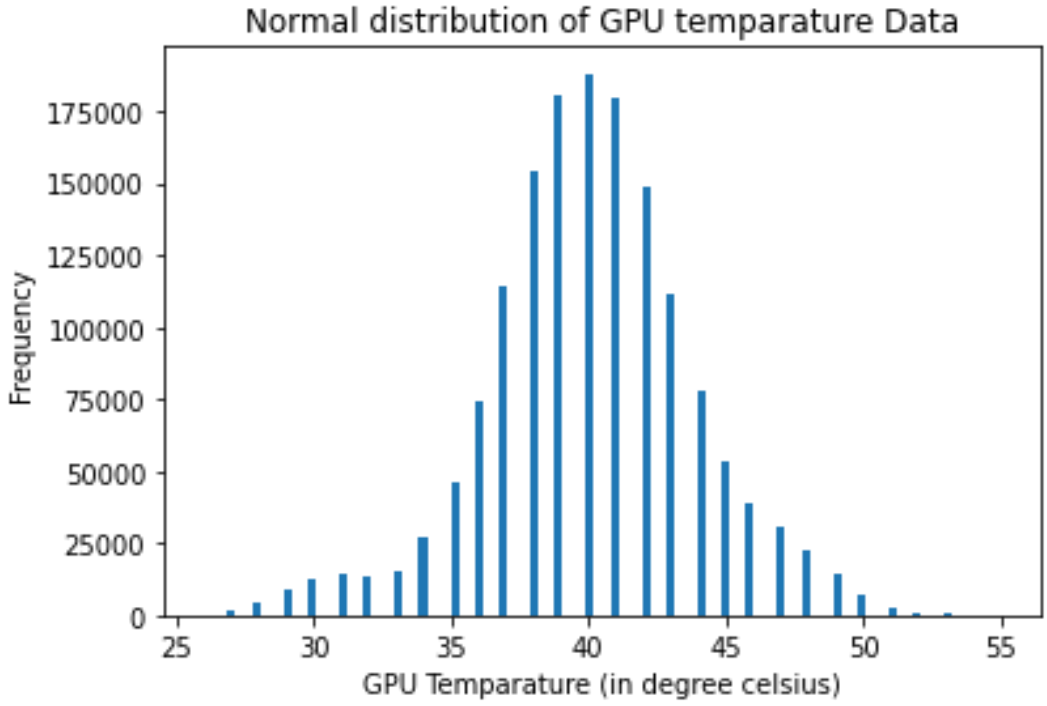

Figure 6 Histogram of GPU Temperature

Comments: The Histogram reveals that the temperature is distributed normally in the data. The mean temperature is 40 degrees Celsius and Standard Deviation is 3.8. (Include a table in the appendix)

5. *What are the host names which consumed maximum and minimum time for the rendering process?*

hostname	Time
dcc19f48bb3445a28338db3a8f002e9c00000S	47.03878
0d56a730076643d585f77e00d2d8521a00001B	47.01344
e7adc42d28814e518e9601ac2329c51300000D	46.99317

Table 6 Bottom 3 host names which took least Rendering Time

hostname	Time
35bd84d72aca403b8129a7d652cc275000000N	38.82447
265232c5f6814768aeefa66a7bec6ff600000W	38.66786
8b6a0eebc87b4cb2b0539e81075191b900000D	38.62792

Table 7 Top 3 host names which took highest Rendering Time

Comments: The hostname 8b6a0eebc87b4cb2b0539e81075191b900000D took minimum time to render the image whereas the hostname dcc19f48bb3445a28338db3a8f002e9c00000S took maximum rendering time.

6. *What are the maximum and minimum values of computational resource used, temperature and power consumption of the GPU and identify the corresponding virtual machines (hostname)?*

GPU Properties	hostname	Values	
power consumption(W)	0d56a730076643d585f77e00d2d8521a00000Q	80.51031312	Minimum
	a77ef58b13ad4c01b769dac8409af3f800000D	106.2474617	Maximum
Temperature(C)	b9a1fa7ae2f74eb68f25f607980f97d700001C	35.37175217	Minimum
	cd44f5819eba427a816e7ce648adceb200000H	48.92671552	Maximum
GPU Usage (%)	d8241877cd994572b46c861e5d144c8500000V	59.46461949	Minimum
	b9a1fa7ae2f74eb68f25f607980f97d7000005	66.8254497	Maximum
Memory Usage (%)	265232c5f6814768aeefa66a7bec6ff600000W	28.95460614	Minimum
	0d56a730076643d585f77e00d2d8521a00001B	38.23117921	Maximum

Table 8 GPU properties (Minimum and Maximum)

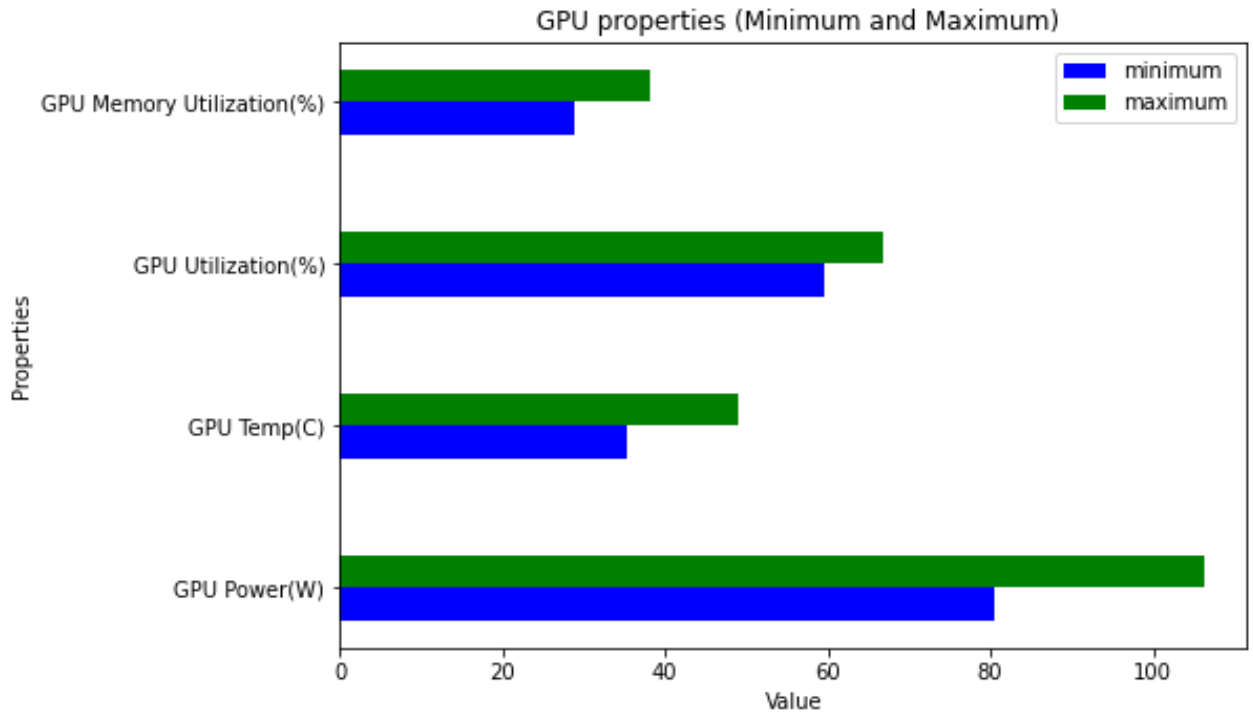

Figure 7 GPU properties (Minimum and Maximum)

Comments: The above-shown table as well as the figure together shows the details of virtual machines which use maximum as well as minimum computational resources and physical properties of GPU.

7. *Which virtual machine processed most image rendering tasks? explain this with the help of a histogram.*

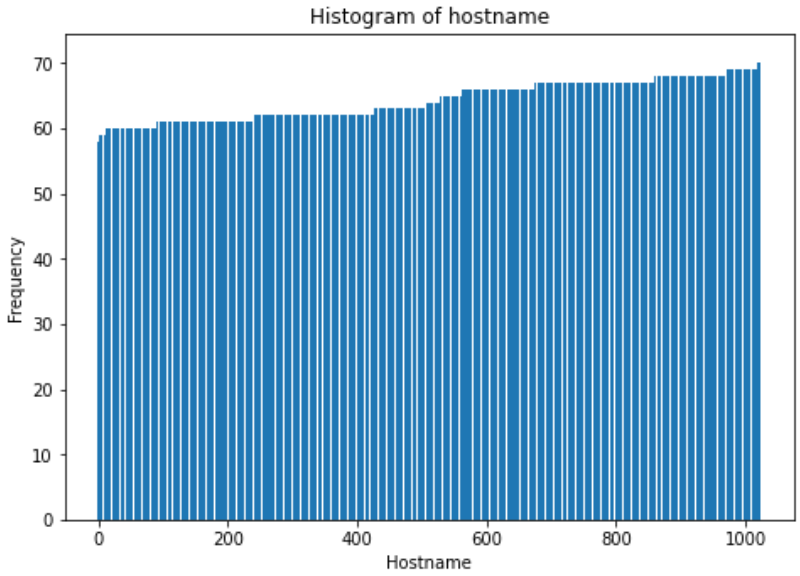

Figure 8 Histogram of Host Name

Comments: The histogram shows that a minimum of 58 image coordinates were processed by each virtual machine and the maximum image processing was done by the hostname with Id e7adc42d28814e518e9601ac2329c51300001D which is 71 image coordinates. The average number of task Id is 64.21 for one hostname with a standard deviation of 2.97.

8. *How many image coordinates are associated with each level?*

Comments: The highest count is for level 12 which is 65536 and the level with the lowest count is 1 for level 4.

level	Frequency
4	1
8	256
12	65536

Table 9 Level vs Frequency

9. *How are the tiles of the image and total rendering times related?*

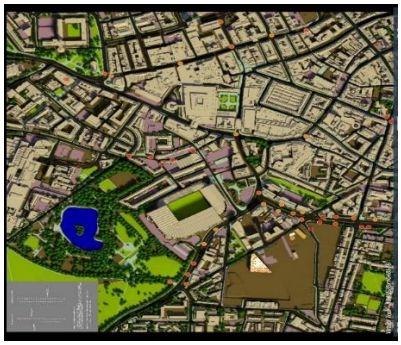

Figure 9 Original image

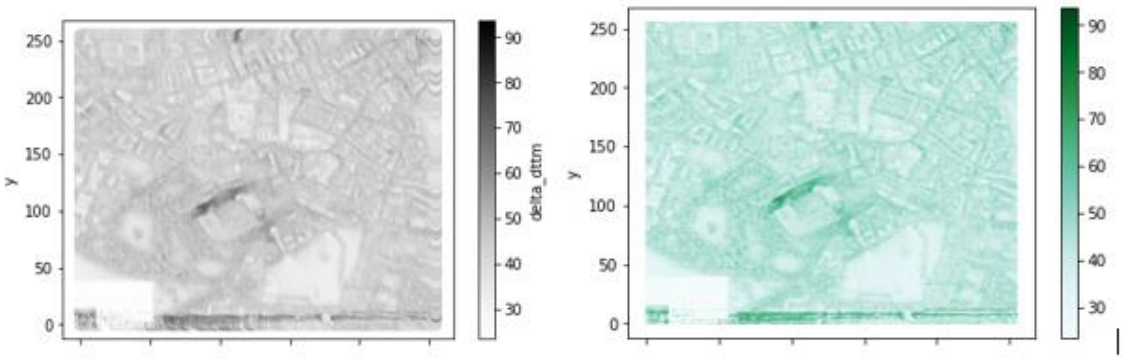

Figure 10 plot generated using tile coordinates and time taken for total rendering

Comments:

Figure 10 shows the plot generated using the data from the data set Application-check points and Task-x-y. Here, x and y coordinates are the corresponding x and y coordinates from the dataset and colour represent the time taken for the total rendering process of each tile. Subsequently, the scatter plot shows some similarities with the original image which concludes that the tile time taken for processing depends on the tile/pixel values of the original image. The dark-shaded region indicates that the process took a long time to execute for the particular tiles. This indicates that colour depth affects the Rendering time. That is as colour depth increases, power, GPU utilization and memory Utilization also increases.

10. Explain the GPU properties using suitable graphs on the basis of hostnames and Task Id?

Properties	hostname	taskId	values	comments
powerDrawWatt	04dc4e9647154250beeee51b866b071500000V	e4c83dfd-c1c2-4805-a8cb-6cf64b01904c	36.94	Minimum
	2ecb9d8d51bc457aac88073f6da05461000005	26c9de8f-d54d-4bda-b02e-f17d38dbbda3	144.28	Maximum
gpuTempC	6139a35676de44d6b61ec247f0ed865700000F	2744c60b-abea-47fe-a0eb-0be3d1fd4b5b	29.57	Minimum
	cd44f5819eba427a816e7ce648adceb200000H	f32bc56e-118e-4f9a-8fd9-49b0ecca2525	52.42	Maximum
gpuUtilPerc	db871cd77a544e13bc791a64a0c8ed5000000C	54f9f5e7-b737-49f3-8221-787a2b8145ad	17.12	Minimum
	4a79b6d2616049edbf06c6aa58ab426a000003	25b410b5-f5ef-4a2f-8b21-29175bca35fc	87.03	Maximum
gpuMemUtilPerc	2ecb9d8d51bc457aac88073f6da0546100000P	14ed2dea-1470-4455-9267-592e06e58a23	6.72	Minimum
	4c72fae95b9147189a0559269a6953ff00000T	4f13081c-5c45-43d0-b744-05caaf5377e2	53.08	Maximum

Table 10 GPU properties extreme values by hostname and task Id

Task Id	x	y	level
e4c83dfd-c1c2-4805-a8cb-6cf64b01904c	51	3	12
26c9de8f-d54d-4bda-b02e-f17d38dbbda3	113	111	12
2744c60b-abea-47fe-a0eb-0be3d1fd4b5b	26	7	12
f32bc56e-118e-4f9a-8fd9-49b0ecca2525	207	152	12
54f9f5e7-b737-49f3-8221-787a2b8145ad	51	0	12
25b410b5-f5ef-4a2f-8b21-29175bca35fc	92	106	12
14ed2dea-1470-4455-9267-592e06e58a23	22	6	12
4f13081c-5c45-43d0-b744-05caaf5377e2	238	208	12

Table 11 table containing x and y values of the task ids referenced to table 10

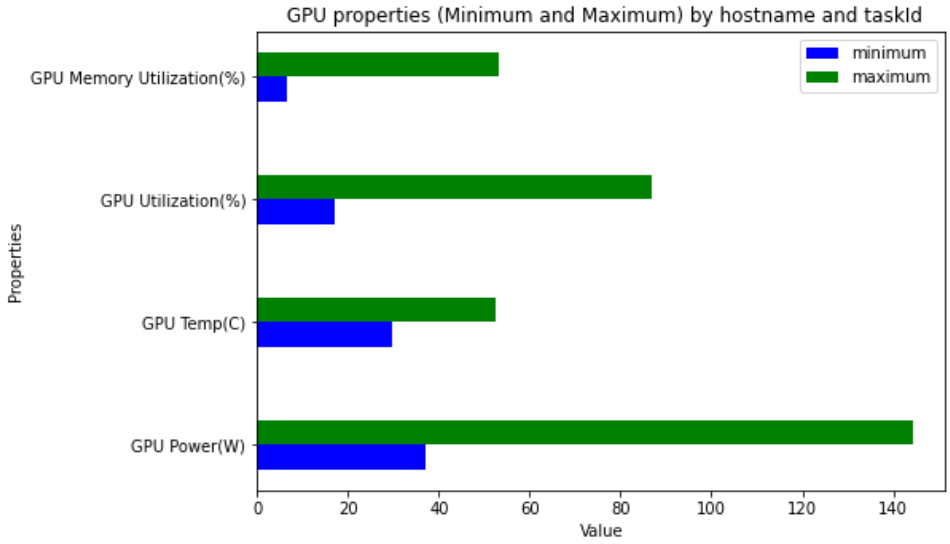

Figure 11 GPU properties by Host Name and Task Id

Comments:

- Figure 11 and the above two tables show that the task Id 26c9de8f-d54d-4bda-b02e-f17d38dbbda3 (113,11) while running in the hostname 2ecb9d8d51bc457aac88073f6da05461000005 and the task Id e4c83dfd-c1c2-4805-a8cb-6cf64b01904c (51,3) executed under 04dc4e9647154250beeee51b866b071500000V consumed maximum and minimum power of 114.28 and 36.94 respectively.
- 52.42 and 29.57 were the extreme temperatures that were recorded for the host Names cd44f5819eba427a816e7ce648adceb200000H and 6139a35676de44d6b61ec247f0ed865700000F while processing the ask Ids f32bc56e-118e-4f9a-8fd9-49b0ecca2525 and 2744c60b-abea-47fe-a0eb-0be3d1fd4b5b respectively.
- Task Ids 54f9f5e7-b737-49f3-8221-787a2b8145ad(51,0) and 25b410b5-f5ef-4a2f-8b21-29175bca35fc (92,106) which executed under 4a79b6d2616049edbf06c6aa58ab426a000003 and db871cd77a544e13bc791a64a0c8ed5000000C respectively showed the GPU utilization percentage of 87.03 and 17.12 which is also the maximum and minimum utilization percentage.
- The maximum and minimum GPU memory were utilized by the task ids 4f13081c-5c45-43d0-b744-05caaf5377e2(238,208) and 14ed2dea-1470-4455-9267-592e06e58a23(22,6) while running in the host Names 4c72fae95b9147189a0559269a6953ff00000T and 2ecb9d8d51bc457aac88073f6da0546100000P respectively. They are 53.08 and 6.78.

The below figure indicates the average of these four GPU parameters

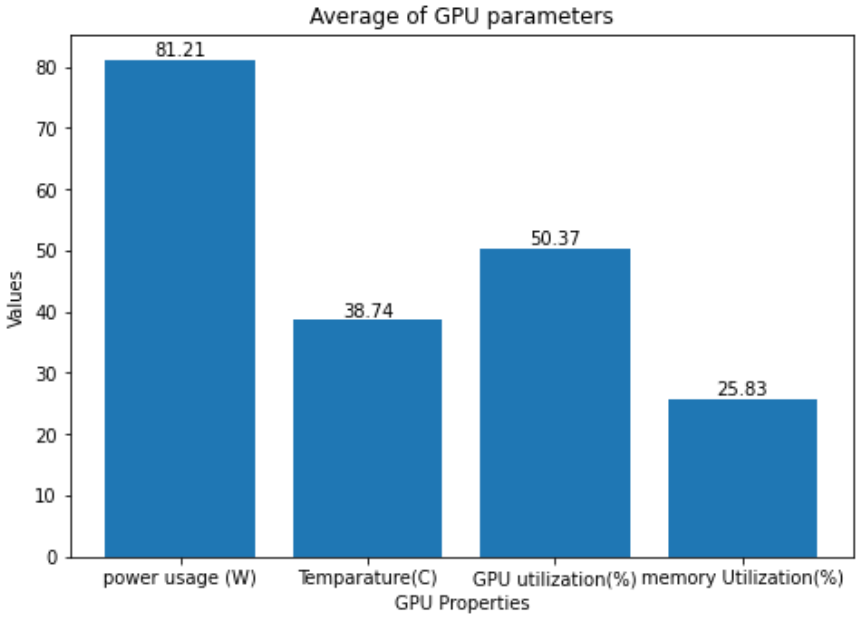

Figure 12 Average GPU Attributes

11. How are the GPU properties related to each tile of the rendered image?

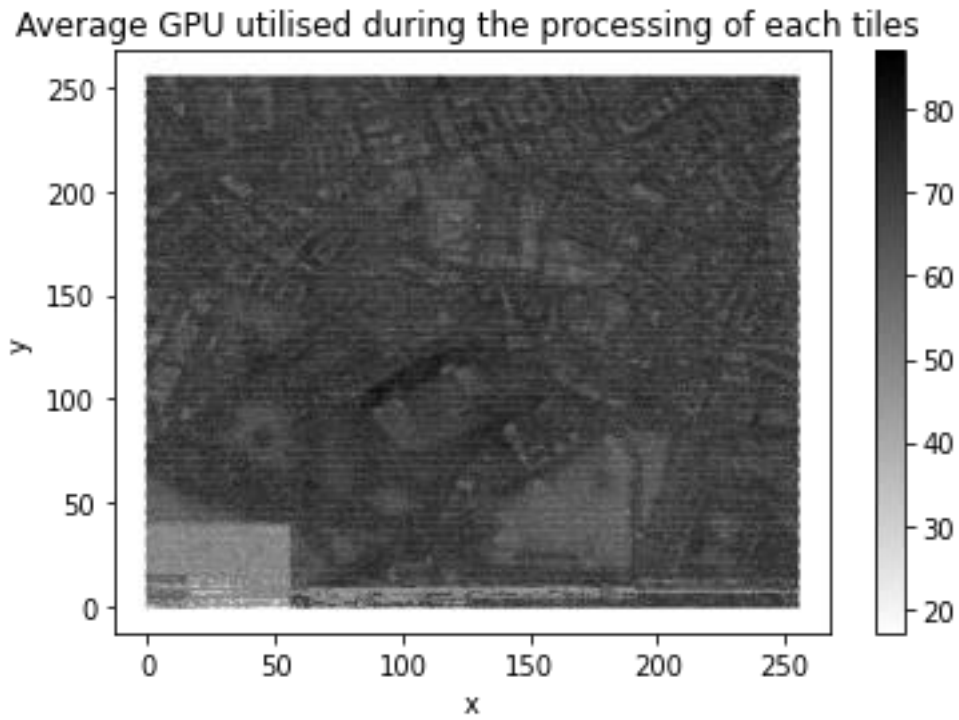

Figure 13 Mean GPU utilization (colour chart shows increase in GPU utilised (%))

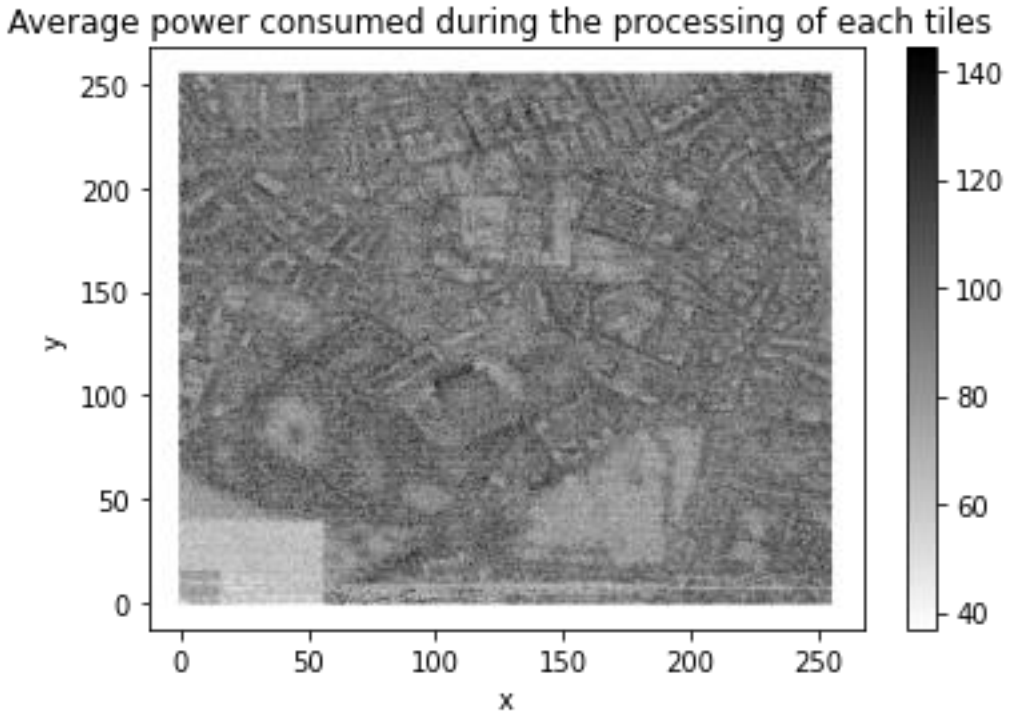

Figure 14 Mean Power consumed (colour chart shows increase in power consumed(W))

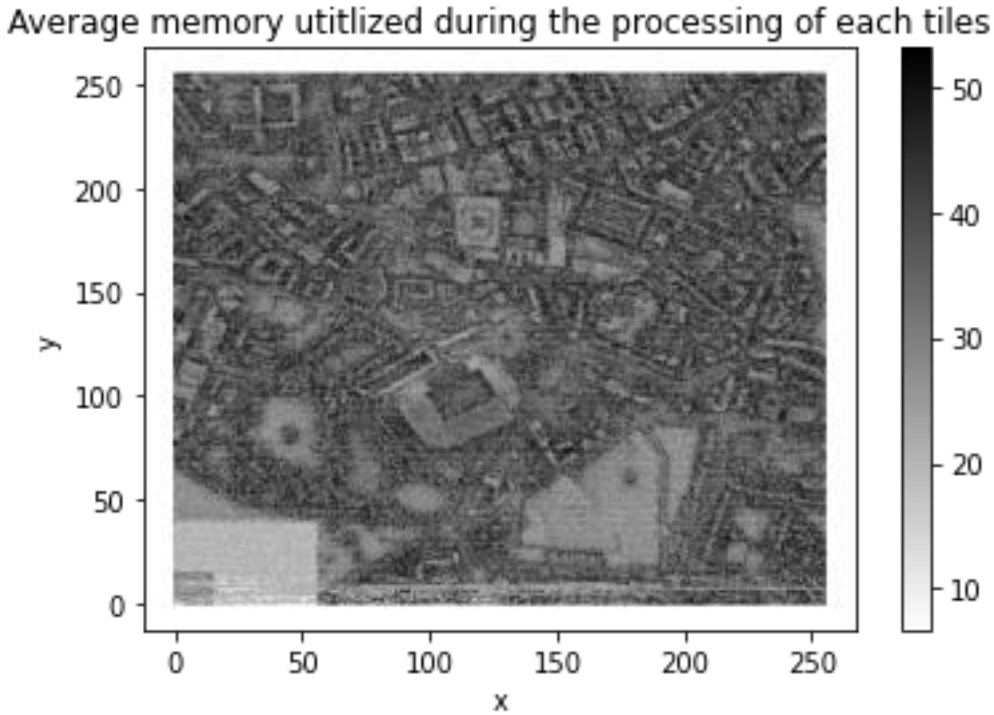

Figure 15 Mean GPU memory utilization (colour chart shows increase in memory utitlized(%))

Comments: The colour maps show that the tiles associated with the building and other structure took less power, GPU utilization and memory Utilization. This indicates that colur depth affect these properties. That is as colour depth increases, power, GPU utilization and memory Utilization also increases.

4. EVALUATION

The demand for scaling supercomputer resources can be assessed and confirmed by this EDA analysis. Furthermore, the thorough study offers some suggestions for further research and optimization of the cloud architecture.

According to the analysis's findings, the rendering process took significantly longer than the other 3 stages. Additionally, there is a minimal correlation between the GPU's power consumption and temperature whereas the GPU utilisation and memory utilisation both exhibit a strong linear relationship.

It's interesting to note that the measured temperature is normally distributed, with a mean and expected value of 40 degrees Celsius. In addition, the statistical analysis shows that the host names process an average of 62.41 task IDs. Additionally, the average GPU performance is 25.83% (GPU memory utilisation), 50.37% (GPU utilisation), 81.21 W (power consumption), and 38.74 C (temperature) (GPU temperature). Besides that, the scatter plot created by plotting GPU memory utilisation, Total Render time, GPU utilisation, and power consumption produced a similar representation of the image, indicating a strong correlation between the image's pixel/tile values and these properties in the XY plane, where x denotes the tile's x coordinate and y denotes the tile's y coordinate. In other words, the colour depth affects the rendering parameters. Lastly, It took around 48 minutes and 45 seconds to render the image completely.

The outcome can be used to improve rendering performance and it guides our decision-making over whether to adopt a different cloud architecture or add more virtual machines. The exploratory data analysis aids in determining whether or not additional system resources are required.

In summary, by examining the data produced during the rendering process, this exploratory analysis offers helpful information that aids in the assessment of the supercomputer.

5. DATA QUALITY CHALLENGES

1. There are several duplicates in the data sets Application-checkpoints and GPU. This might affect in future if the data obtained is very large and complex.

2. It was difficult to join the tables Application-checkpoints and GPU based on hostname and timestamps alone as it was an ambiguous method. It would be better if there are well-defined keys to join the tables for analysis.

3. The time taken for a particular task cannot be used as a measure to compare the performance as the processing difficulties vary over the task Id. So, a new GPU attribute is required for effectively calculating the performance of the GPU for comparing it with other GPUS.

4. New attribute is needed to check whether the process is using significant resources or not related to the total capacity of the supercomputer. Hence, such an attribute representing the machine's capacity is needed.

5. In a virtual computer, only one task ID is ever active at once.

Multiple events are run simultaneously on various host names. Therefore, it is challenging to locate the GPU attributes for a specific job Id.

6. FUTURE SCOPE

1. The visual interaction can be improved by generating 3D visuals from the images, for that multiple images from different perspectives can be used to create 3D models which will pack more information in addition to better user interaction.

2. For improving the analysis and optimization, a suitable parameter which should represent the performance of the GPU can be introduced rather than using the time difference as a performance factor.

3. Additionally, more information can be gathered and incorporated with visual data like real-time local weather conditions, more efficient route map planning, local transportation details etc to make people's life much easier

4. Machine learning algorithms can be used to predict virtual cloud performance in future so that proactive measures can be taken to allocate resources. Additionally, it can also be used to predict changes over time based on the data collected.

5. It would be better if the user has the option to view the different images taken in the past to compare the development of the city.

7. REFLECTION

This project was a better opportunity to learn how cloud architecture helps to render complex images by utilizing the resources of the cloud-based supercomputer. I could improve my skills especially in querying the data for exploratory analysis. I really enjoyed working on this project especially when I found that the data was able to recreate the original image from numbers.

The project helped me to learn different virtualisation techniques used for processing high-quality image data by reducing overall costs. The project demanded several skills like SQL, Python, and problem-solving skills through deep analysis of the problem. Moreover, the project needed intensive research and offered a chance to explore useful internet content for this specific project. Apart from this, guidance from the university and our module leader helped me to achieve greater results. Besides, I was able to use Git for maintaining consistent code, version control and cross-platform transfer. Furthermore, I tried my best in following the best project writing style, and functionalities of the Markdown feature of the python notebook.

However, I think utilizing the machine learning aspects to predict some of the variables associated with the GPU properties and event names will give more insights into future events. Apart from it, for verifying whether the supercomputer can execute the task efficiently, we need more data related to the capacity of the supercomputer so that, we could predict the need for using another unit or not since proactive measures will help in reducing unnecessary bottlenecks. In addition, time as a measure of performance cannot be considered as it depends on many other factors. So the data representing performance should also be incorporated. Moreover, joining the table was not that easy as I had to deal with ambiguous values.

I could also find that using R language will reduce the code complexities as it packed more inbuilt features which will give attractive visualisation with only fewer lines of code. Also, using SQL API in python takes more time to execute than in R. However, if we are using panda's inbuilt functions the code is executed quickly, at the same time querying using pandas is difficult to code. So, getting a proper trade-off between these two gives efficient results, that is for complex queries, we can use SQL API and for simple requirements, panda's functionalities will be helpful. In addition, integrating

machine learning models can be considered for predicting variables which will help to estimate the performance of the machine. These factors will be considered for future expansion of the same or similar projects. Furthermore, in this analysis all GPU data for the hostname for every time frame is used, so in the next phase of analysis only data necessary for obtaining valid conclusions can be considered.

8. CONCLUSION

The main objective of the report is to analyse the performance of a cloud-based supercomputer which render a realistic tera-pixel image of the city of Newcastle upon Tyne. In addition, it provides interactive support for the city visualization to various stakeholders. Through this EDA analysis, the need for scaling supercomputer resources can be evaluated and verified. Moreover, the rigorous analysis provides some ideas about the areas of development and to perform optimization in the cloud architecture.

The dataset for the analysis is generated while processing the image shown in figure 9. There are total

65793 tiles in which each tile is linked to a particular task Id. Each tile is associated with x coordinates and y coordinates values. There are mainly 4 tasks namely saving the configuration, Tiling, Rendering and uploading. The time at which these tasks start and stop are given in the application checkpoints table. The hostname is the virtual machine which processes these tasks. There is a total of 1024 virtual machines which render the full-size image. One hostname can execute many task Ids and the GPU properties of these hostnames while executing a particular task Id can be obtained from the GPU table.

The results of the analysis show that the Rendering process took a significant amount of time compared to the other 3 processes. Also, there is slight relation between the GPU temperature and the power it consumed. At the same time, a strong linear relationship can be found between the GPU utilization and memory utilization. Interestingly, the temperature recorded is distributed normally with the mean and expected value of 40 degrees Celsius. Besides that, the statistical analysis indicates an average number of 62.41 task IDs is processed by the host names. Also, The mean GPU parameters are 25.83 % (GPU memory Utilization), 50.37 % GPU utilization, 81.21 W Power consumption, and 38.74 degree Celsius (GPU temperature). Moreover, the scatter plot obtained by plotting GPU memory utilization, Total Render time, GPU utilization and power consumption generated a similar representation of the image which implies that there is a strong relation between the pixel/ tile values of the image and these properties in the XY plane, where x denote tile x coordinate and y denote tile y coordinate. Lastly, It took around 48 minutes and 45 seconds to render the image completely.

The result obtained can be used to optimize the rendering performance and it helps us to decide whether to choose different cloud architecture or to increase the number of virtual machines. The exploratory data analysis also helps to decide if more system resource is needed or not.

In conclusion, this exploratory analysis provides useful information which helps in the evaluation of the supercomputer by analysing the data generated during the rendering process.