Performance evaluation of Terapixel rendering in Cloud (Super)computing

Peiqiang Li - 200987503

- Which event types dominate task runtimes?

- What is the interplay between GPU temperature and performance?

- What is the interplay between increased power draw and render time?

  - Can we quantify the variation in computation requirements for particular tiles?

- Can we identify particular GPU cards (based on their serial numbers) whose performance differs to other cards? (i.e. - - perpetually slow cards).

- What can we learn about the efficiency of the task scheduling process?

- **TotalRender** is the entire task

- **Render** is when the image tile is is being rendered

- **Saving Config** is simply a measure of 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

Introduce

"Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services"[1].Cloud computing concentrates computing and storage at the heart of the service, and high-bandwidth connections are adopted to link high-performance machines, so that all these resources could be carefully managed. The user sends a computing request to the cloud computing service and obtains the final result[1]. In this project, Cloud (super)computing is used for high quality terapixel visualization. The visualization of trillions of pixels requires powerful computing resources. Therefore, cloud-based supercomputer resources provide greater flexibility for visualization.

Methodology

The structure of the project is built by **ProjectTemplate**. As we all know, the correctness and reproducibility of data science code is very important. Thus, an appropriate project template can not only improve the efficiency of development, but also make the code and documents more concise and orderly.

Git is used as a version control tool for this project. All development activities are being recorded using Git and stored in a private repository. Due to the size and privacy of dataset, the raw data will not be uploaded to the GitHub repository.

Rstudio is the development tool of this project, The project process adopts the data exploratory analysis steps of **CRISP-DM** model. And, the literate programming framework is used to record the results of data analysis and generate the final project report.

README file will be used to record the analysis steps of the project and the reproduction process of the analysis.

Business Understanding

The realistic terapixel visualization of the city of Newcastle upon Tyne is calculated by cloud-based supercomputer resources. The project detail see [link] (https://github.com/NewcastleDataScience/StudentProjects202122/blob/master/TeraScope/Summary.md).

```
## # A tibble: 6 x 6
##   timestamp              hostname      eventName   eventType jobId      taskId
##   <chr>                  <chr>         <chr>       <chr>     <chr>      <chr>
## 1 2018-11-08T07:41:55.921Z 0d56a730076~ Tiling      STOP      1024-lvl~ b47f0263~
## 2 2018-11-08T07:42:29.842Z 0d56a730076~ Saving Co~  START     1024-lvl~ 20fb9fcf~
## 3 2018-11-08T07:42:29.845Z 0d56a730076~ Saving Co~  STOP      1024-lvl~ 20fb9fcf~
## 4 2018-11-08T07:42:29.845Z 0d56a730076~ Render      START     1024-lvl~ 20fb9fcf~
## 5 2018-11-08T07:43:13.957Z 0d56a730076~ TotalRend~  STOP      1024-lvl~ 20fb9fcf~
## 6 2018-11-08T07:43:56.239Z 0d56a730076~ Render      STOP      1024-lvl~ 3dd4840c~
```

	Fields	Info
1	timestamp	Timestamp
2	hostname	Hostname of the virtual machine auto-assigned by the Azure batch system.
3	eventName	Name of the event occuring within the rendering application.
4	eventType	Possible Values: "START", "STOP"
5	jobId	ID of the Azure batch job.
6	taskId	ID of the Azure batch task.

Table 1: Table information of gpu.csv

|	Fields	Info
1	timestamp	Timestamp
2	hostname	Hostname of the virtual machine auto-assigned by the Azure batch system.
3	gpuSerial	The serial number of the physical GPU card.
4	gpuUUID	The unique system id assigned by the Azure system to the GPU unit.
5	powerDrawWatt	Power draw of the GPU in watts.
6	gpuTempC	Temperature of the GPU in Celsius
7	gpuUtilPerc	Percent utilisation of the GPU Core(s).
8	gpuMemUtilPerc	Percent utilisation of the GPU memory.

Table 2: Table information of gpu.csv

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

- **jobId**: Id of the Azure batch job.
- **taskId**: ID of the Azure batch task.
- **x**: X co-ordinate of the image tile being rendered.
- **y**: Y co-ordinate of the image tile being rendered.
- **level**: The visualisation created is a zoomable "google maps style" map. In total we create 12 levels. Level 1 is zoomed right out and level 12 is zoomed right in. You will only see levels 4, 8 and 12 in the data as the intermediate level are derived in the tiling process.

Including Plots

You can also embed plots, for example:
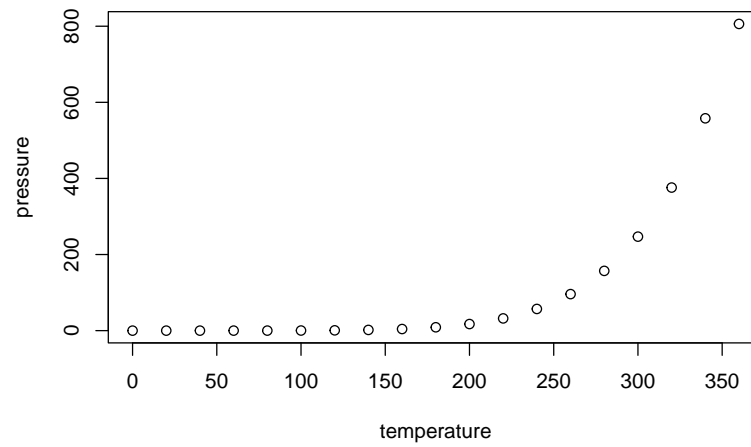
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. A view of cloud computing. Communications of the ACM, 53(4), pp.50-58.

[2] Hayes, B., 2008. Cloud computing.

[3] Christensen, R., 2006. Log-linear models and logistic regression. Springer Science & Business Media.

[4] McLeod, A.I. and Xu, C., 2010. bestglm: Best subset GLM. URL http://CRAN. R-project. org/package= bestglm.

[5] McNeish, D.M., 2015. Using lasso for predictor selection and to assuage overfitting: A method long overlooked in behavioral sciences. Multivariate Behavioral Research, 50(5), pp.471-484.