# data-preperation-ammar-150454388

January 25, 2019

Table of Contents

## 0.1 Introduction

This report summaries the Data Preparation stage of the CRISP-DM cycle for this project. In particular, this report covers the background of the data and how it was modified for further exploratory analysis. This report covers the methodology of data preparation not the detailed technical aspects - those are covered in the Sphinx documentation.

## 0.2 Data Background

As stated in the Business Understanding Stage Report, the provided dataset consists of results from running three jobs of a terapixel render of the city of Newcastle upon Tyne in three levels (4, 8, 12) of terapixel image with 1024 GPU nodes over a cloud service. The data is recorded in three csv files:

- gpu.csv: GPU status
- application-checkpoints.csv: checkpoint events throughout render job
- task-x-y.csv: Co-ordinates of image parts being rendered in task

## 0.3 Dataset Description

### 0.3.1 Fields

- start_time (textual): Timestamp for tuple in 'YYYY-MM-DD H:M:S.f' format for start of event
- stop_time (textual): Timestamp for tuple in 'YYYY-MM-DD H:M:S.f' format for end of event
- hostname (textual): Unique system ID assigned to the Machine GPU runs from by Microsoft Azure

- gpuUUID (textual): Unique GPU ID assigned to the Machine GPU runs from by Microsoft Azure
- powerDrawWatt (numerical): Average power draw of system (Watts) for event
- gpuTempC (numerical): The Average GPU temperature (Celsius) for event
- gpuUtilPerc (numerical): The average GPU utilisation % (0-100) for event
- gpuMemUtilPerc (numerical): The average GPU memory usage % (0-100) for event
- eventName (textual): Name of current event occurring in the rendering process. Possible values:
  - TotalRender: The whole task itself
  - Render: Image tile being rendered
  - Saving Config: Configuration overhead
  - Tiling: Tile postprocessing
  - Uploading: Output uploading to Azure Blob Storage
- x (numerical): X coordinate of tile being rendered
- y (numerical): Y coordinate of tile being rendered
- level (numerical): Visualisation level (zoom) within the terapixel map (4, 8, 12)

### 0.3.2 Tables Preprocessing

Table processing was handled by src.data.make_datasetű script and tested by src.tests.test_make_dataset, both scripts are documented by the Sphinx documentation, hence this part of the report will avoid detailed technical explanations.

**GPU Dataset Cleaning** To clean the GPU dataset the unneeded GPUSerial field is dropped (GPUs can be identified by gpuUUID). Moreover, the timestamp was converted to the datetime format for later merges.

**Checkpoints and Tasks Datasets Merge** The Checkpoints and tasks datasets are joined in a left join in checkpoints to tasks direction via the shared jobId and taskId fields.

**Merged Checkpoints and Tasks Dataset Cleaning** The taskId and jobId fields are now not needed since the merge was already done and hence dropped since there is no need to identify the tasks and jobs with them in this analysis - the analysis will be concentrating on the events and types of tasks instead. Moreover, the timestamp was converted to datetime format for later merges.

**Merged Checkpoints and Tasks Dataset Merge with GPU Dataset** To merge the two datasets, first the timestamps and event types (start/stop) of the checkpoints and tasks dataset are used to form the start and stop times of events, which are now combined to one row (start/stop rows used to be separate). This updated dataset is then merged with the GPU dataset through a sql left join with hostname, but with a restriction that ensures it is between the start and stop time recorded before according to its timestamp. The dataset is then grouped by task (x, y and level) and hostname to find averages for GPU statistics (power draw, utilisation and temperatures).

## 0.4 Summary

To summarise, the Data Preparation stage in this project forms the final dataset for exploratory analysis using cleaning, merging and grouping to combine the raw data. These methods are carried in the src.data.make_dataset script and tested by src.tests.test_make_dataset script covered in the Sphinx documentation.