CERN

Summer Student Programme 2022

รายงานการเข้าร่วมโครงการนักศึกษาภาคฤดูร้อนเซิร์น

ระหว่างวันที่ 7 มิถุนายน - 26 สิงหาคม 2565

SIIT
Sirindhorn International Institute of Technology

นางสาวณัฐชยา ภูมีคำ

ภาควิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีนานาชาติสิรินธร มหาวิทยาลัยธรรมศาสตร์

Table of Content

Acknowledgement

I would like to express my deepest gratitude towards Her Royal Highness Princess Maha Chakri Sirindhorn for selecting me as one of the 2022 CERN Summer Student.

I would like to also express my gratitude towards CERN and Synchrotron Light Research Institute for giving me the opportunity to work in such a unique and challenging environment.

This project would not be completed without the sincere help and guidance from my supervisors, especially Diego Ciangottini, and the CRAB team members. Additionally, thank to Dr.Norraphat Srimanobhas for his guidance and support throughout the program.

Chapter 1

CERN Summer Student Program 2022

1.1 CERN Summer Student Program 2022

The CERN Summer Student Program offers an exclusive and one-of-a-kind experience for the students pursuing bachelor's or master's degrees in either physics, computing, engineering, or mathematics to contribute to day-to-day work researching and experimenting with the professional teams at CERN in Geneva, Switzerland. The program lasts for 8-12 weeks in the summer period between June to August. Apart from the scientific and researching experience, the participants also get to experience working with people from around the world with different background and culture but having same interest which is the unique experience that can be found only at CERN. The students have opportunities to listen to the lectures from leading scientists and researchers. Before leaving, CERN also offers the opportunity for the students to present their projects both by poster and by technical presentation.

Fig.1 CERN Summer Students 2022

1.2 Visits

I had a chance to visit Data Centre (DC) & the Antiproton Decelerator (AD). The data center is the core of CERN computing infrastructure The 450 000 processor cores and 10 000 servers run 24/7. Over 90% of the resources for computing in the Data Centre are provided through a private cloud based on OpenStack, an open-source project to deliver a massively scalable cloud operating system. We also went to see LINAC4 which is Linear accelerator 4 (Linac4). It is designed to boost negative hydrogen ions to high energies. It became the source of proton beams for the Large Hadron Collider (LHC) in 2020.

Fig.2 CERN Data Centre and the LINAC4 building

Another visit was at the Synchrocyclotron and the ATLAS visitor center. The Synchrocyclotron was the first CERN's accelerator which came into operation in 1957 and was closed in the 1999 after 33 years of providing beams for CERN's experiments in particle and nuclear physics. For the ATLAS visitor center, the students were led into a room and watched a 3D footage of the assembling of ATLAS. ATLAS is the largest general-purpose particle detector experiment at the Large Hadron Collider, a particle accelerator at CERN in Switzerland. We also had a chance to see the control room with many researchers working in there.

Fig.3 Synchrocyclotron visit

Fig.4 ATLAS operating room

1.3 Lectures

The lectures conducted by CERN in the period of June 27th, 2022, to July 29th, 2022. These lectures started at 9.15 in the morning, 50 minutes for one lecture, and three lectures per day. Between each lecture, the students got a short 10–15-minute coffee break to ask interesting questions and relax before new topic. This year's lectures were all conducted online via Zoom, but the students also booked a room to all watch the lectures together. Some of the lecturers were also at CERN and they encouraged the students to visit their office for deeper questions and discussions. Most of the topics were physics-related and it is not an exaggeration to say that they were all high-level. As for myself, I am a computer engineering student, I attend the lectures but could not understand everything that was taught. Nonetheless, the lectures were interesting and new to me.

The lecture topics are as follow:

1. Particle World by David Tong
2. Foundation of Statistics by Glen Cowan
3. Particle Accelerators and Beam Dynamics by Michaela Schaumann
4. From Raw Data to Physics Results by Paul James Laycock
5. Theoretical Concepts in Particle Physics by Tim Cohen
6. Detectors by Werner Riegler
7. Nuclear Physics at CERN by Stephan Malbrunot
8. Electronic, DAQ, and Triggers by Emilio Meschi
9. The Standard Model by Christophe Grojean
10. Making Predictions at Hadron Colliders by Mike Saymour
11. Introduction to Cosmology by Daniel Baumann
12. Astroparticle Physics by Dr.Bradley Kavanagh
13. Accelerator Technology Challenges (Part 1: Magnet Superconductivity) by Helene Felice

14. Antimatter in the Lab by Jack Devlin
15. Experimental Physics at Hadron Colliders by Marumi Kado
16. Accelerator Technology Challenges (Part 2: RF Superconductivity) by Walter Venturini Delsolaro
17. Flavor Physics by Mark Richard James Williams
18. Physics and Medical Applications by Manuela Cirilli
19. Accelerator Technology Challenges (Part 3: Accelerator Operation and Design Challenges) by Francesc Salvat Pujol
20. Heavy Ion by Francesca Bellini
21. Beyond the Standard Model by Tevong You
22. What is String Theory by Timo Stephan Weigand
23. Future High-Energy Collider Projects by Barbara Dalena
24. Experimental Physics at Lepton Colliders by Frank Simon

1.4 Presentation

CERN offers multiple ways for students to present their work to fellow Summer Students and other researchers at CERN. The presentations and their details are as follow:

1.4.1 Oral Presentation

The oral presentation was organized at 222/R-001 on the 3rd to 5th of August. The presenters were offered the chance to give a lecture and present their projects to fellow students and colleagues. There were 30 slots for presenters. The talk lasted for 10 minutes and then followed by a 5-minute Q&A session. The presentations were also recorded and made available online. The presenter names will be published in the CERN Annual Report

in the next year as having been part of the Summer Student Lecture Program.

1.4.2 Poster Presentation

The poster presentation was organized on the 28th of July at the 61/1-201 hallway. There were 30 places available in a first come, first served manner. The presenters were given the chance to present their projects in the bigger context by visualizing it through the poster in the size of A1 or A0. On that day, everyone can walk around to see all the posted posters, the presenters stood by their posters to answer all the questions when asked by the passerby.

1.4.3 CMS Group Meeting

This is similar to the oral presentation except that the audience is the members of CMS rather than fellow students.

I had the opportunity to present in the poster session. It was the most interesting presentation I have done in many years as the audience were knowledgeable friends and researchers who shared similar passion about physics and computer engineering.

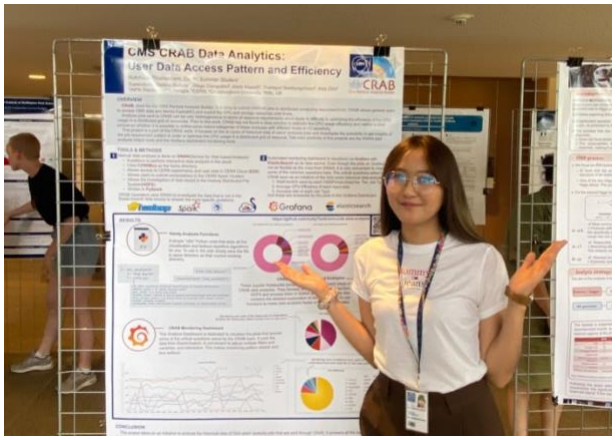


Fig.5 Presenting the poster

CMS CRAB Data Analytics:

User Data Access Pattern and Efficiency

Nutchaya Phumekham, CERN Summer Student

Supervisors: Stefano Belforte[1], Diego Ciangottini[2], Dario Mapelli[3], Thanayut Seethongchuen[4], Katy Ellis[5]

[1]INFN Trieste, [2]INFN Perugia, [3]CERN, [4]Chulalongkorn University, [5]RAL, UK

OVERVIEW

CRAB, short for the CMS Remote Analysis Builder, is a utility to submit CMSSW jobs to distributed computing resources(Grid). CRAB allows general users to access CMS data and Monte-Carlo(MC) and exploit the CPU and storage resources over there.

Analysis jobs sent to CRAB can be very heterogeneous in terms of resource requirements which leads to difficulty in optimizing the efficiency of the CPU usage in a distributed grid of resources. Prior to this work, CRAB has not found a clear solution to estimate the CPU efficiency and neither a clear picture on whether it is possible to define some macro-categories of these analyses with different levels of I/O sensitivity.

This project is a part of the CRAB work. It focuses on the analysis of historical data of users' analyses jobs and investigate the possibility to get insights of the job requirement pattern in order to optimize the CPU usage in a distributed grid of resource. Two main products of this projects are the SWAN data analysis helper tools and the Grafana dashboard monitoring tools.

TOOLS & METHODS

1 Manual data analysis is done on **SWAN**(Service for Web based Analysis)

- A platform to perform interactive data analysis in the cloud
- Uses **CERNBox** as the home directory
- Allows access to CERN experiments and user data in CERN Cloud (**EOS**)
- Allows users to submit computations to the CERN Spark Clusters
- Allows the investigation of data stored in the Hadoop Distributed File System(**HDFS**).
- Written in **PySpark**

CRAB Operators uses SWAN to investigate the data that is not in the ElasticSearch data source to answer the more specific questions.

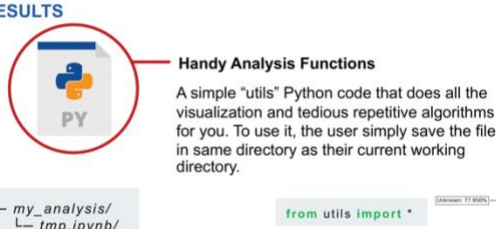
2 Automated monitoring dashboard is visualized via **Grafana** with **ElasticSearch** as its data source. Even though the plots on Grafana are not as flexible as the ones from SWAN, it is very convenient to monitor some of the common questions here. The critical questions asked by the CRAB team as an initiative of the Grid users historical data analysis are:

1. WallClockHr used by each CMSPrimaryDataTier, Tier, Job Type
2. Average CPU Efficiency of each input data
3. Success rate of each Job Type

And these are answered by the plots in this Grafana Dashboard.

RESULTS

https://github.com/nutty7fold/cern-crab-data-analysis



Handy Analysis Functions

A simple "utils" Python code that does all the visualization and tedious repetitive algorithms for you. To use it, the user simply save the file in same directory as their current working directory.

```
├─ my_analysis/
   └─ tmp.ipynb/
   └─ utils.py/
```

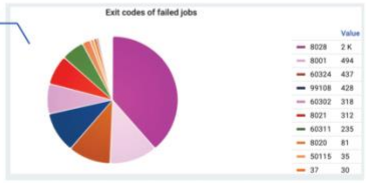
```
from utils import *
_donut(dictlist, "tmp_analysis")
```

```
_to_dict: converts PySpark DataFrame to Python Dictionary
_donut: plots 1 or many donuts chart
_pie: plots 1 or many pie chart
_better_label: return a list of labels concatenated with percentage
_line_graph: plots 1 or many lines in a graph with mean values
_table: creates a table
_exitcode_info: translates number exitcode to meaningful string
```

CRAB Analytics Guide and Examples

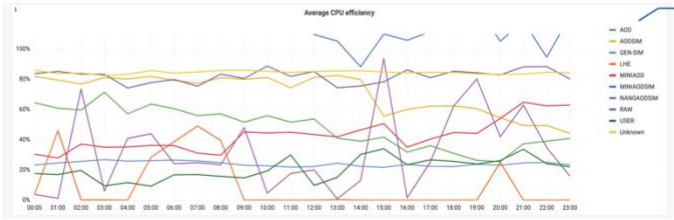
These Jupyter Notebooks contains explanations of each steps of CRAB data analytics. They focuses on how to read raw data from HDFS and process them in SWAN environment. They also contains the detailed explanation of the "utils" ready-to-use functions to make data analytics faster and less complicated.

Monitoring exit code of the failed jobs to understand whether the failed jobs reflect human error or site error.

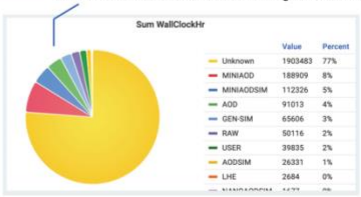


CRAB Monitoring Dashboard

This Grafana Dashboard is dedicated to visualize the plots that answer some of the critical questions asked by the CRAB team. It uses the data from ElasticSearch, is convenient to adjust multiple filters and variables, and interactive. This makes monitoring pattern clearer and less tedious.



Monitoring sum of wallclock hour and CPU efficiency to understand how each data tier or site uses grid resources.



CONCLUSION

This project takes on an initiative to analyze the historical data of Grid users' analysis jobs that are sent through CRAB. It answers all the important questions asked by the CRAB Team and shows that there are rooms to improve the CPU usage and prevent job failure. Furthermore, it provides tools to make answering future questions, data analysis, and investigation more convenient.

https://www.linkedin.com/in/nutchaya-phumekham-90a8551bb/ *https://github.com/nutty7fold*

Fig.6 My Poster

1.5 Other Activities

1.5.1 Summer Student Welcome Drink

This is a session to officially welcome all the summer students. There were drinks and snacks served. There was nothing specific by the hosts, the students were just networking and getting to know each other more.

Chapter 2

Student Project

2.1 Project Topic

Data Analytics of User Historical Data Access Patterns and CERN Grid Resource Efficiencies

2.2 Project Brief Details

This project is a part of the CRAB work. It focuses on the analysis of historical data of users' analyses jobs and investigate the possibility to get insights of the job requirement pattern to optimize the CPU usage in a distributed grid of resource. This project takes on an initiative to provide tools for future CRAB data analysis while also provide the prove that the data are worth discovering pattern in depths.

2.3 Technical Report

Introduction

This project is a part of the CERN CMS Remote Analysis Builder (CRAB) team. CRAB, short for the CMS Remote Analysis Builder, is a utility to submit CMSSW jobs to distributed computing resources (Grid). CRAB allows general users to access CMS data and Monte-Carlo (MC) and exploit the CPU and storage resources over there.

Analysis jobs sent to CRAB can be very heterogeneous in terms of resource requirements which leads to difficulty in optimizing the efficiency of the CPU usage in a distributed grid of resources. Prior to this work, CRAB has not found a clear solution to estimate the CPU efficiency and neither a clear picture on whether it is possible to define some macro-categories of these analyses with different levels of I/O sensitivity.

This project is a part of the CRAB work. It focuses on the analysis of historical data of users' analyses jobs and investigate the possibility to get insights of the job requirement pattern to optimize the CPU usage in a distributed grid of resource.

Tools and Methodology

Tools

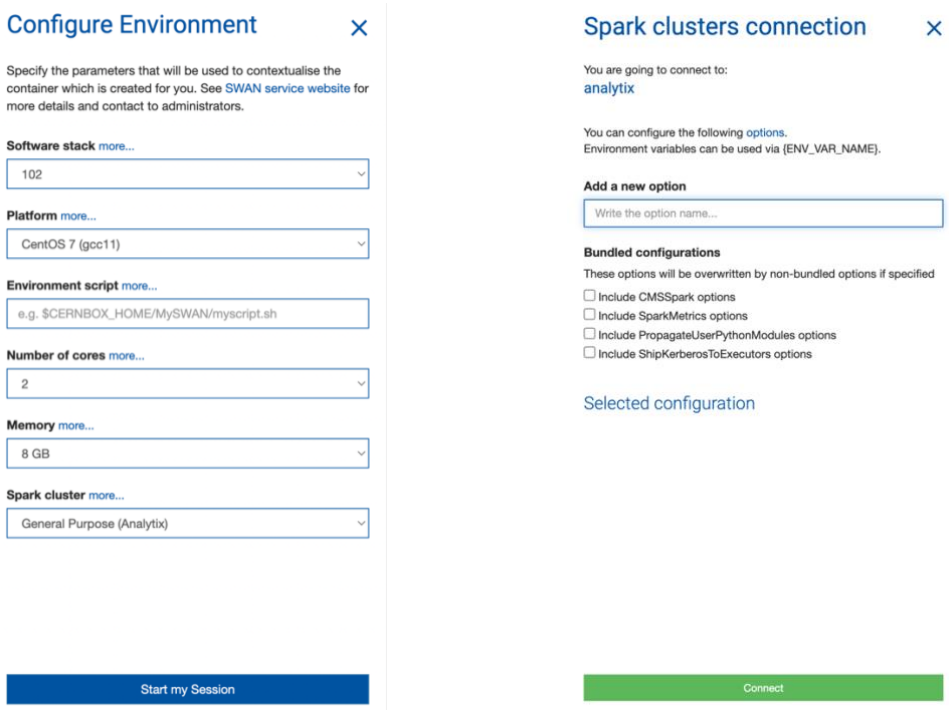
Manual data analysis is done on SWAN (Service for Web based Analysis), which is a platform to perform interactive data analysis in the cloud. SWAN uses CERNBox as its home directory. It allows access to CERN experiments and user data in CERN Cloud (EOS). Furthermore, it allows users to submit computations to the CERN Spark Clusters and investigate the data stored in the Hadoop Distributed File System (HDFS). Programming is done using PySpark. PySpark is the collaboration of Apache Spark and Python. Apache Spark is an open-source cluster-computing framework, built around speed, ease of use, and streaming analytics whereas Python is a general-purpose, high-level programming language.

Automated monitoring dashboard is visualized via Grafana with ElasticSearch as its data source. Even though the plots on Grafana are not as flexible as the ones from SWAN, it is very convenient to monitor some of the common questions here.

Methodologies

General method to do CRAB users' historical data analytics via SWAN are as follows:

1. Connect to the spark cluster and set the configuration. In this project, the cluster used is the CERN General Purpose (Analytix) cluster with the default software stack 102 and default configuration.



2. Read raw data from HDFS as PySpark DataFrame

a. Set the desired base path that keeps the interested raw data. In this project, the base path used is *'/project/monitoring/archive/condor/raw/metric'*

```
_DEFAULT_HDFS_FOLDER = "/project/monitoring/archive/condor/raw/metric"
```

b. Set the desired time range. Note the time range is defined by the function *datetime(yyyy, m, d)*.

```
start_date = datetime(2022,1,1)
end_date = datetime(2022,1,2)
```

c. Get the candidate files that contain the data from the time range. It is important to understand that raw data is stored 1 file per 1 day. Ideally all data rows in the file should be from the same date but that appears to not be the case. It is found that data rows from multiple dates are saved in the same file thus the *get_candidate_files()* function is necessary. The *days* variable shown below is set to 3, meaning to look through the span of 1 week, but setting it to 1 is also done in other projects. This is a trade-off between runtime and accuracy.

```
def get_candidate_files(start_date, end_date, spark, base):
    st_date = start_date - timedelta(days=3)
    ed_date = end_date + timedelta(days=3)
    days = (ed_date - st_date).days
    sc = spark.sparkContext
    candidate_files = [
        f"{base}/{(st_date + timedelta(days=i)).strftime('%Y/%m/%d')}"
        for i in range(0, days)
    ]
    FileSystem = sc._gateway.jvm.org.apache.hadoop.fs.FileSystem
    URI = sc._gateway.jvm.java.net.URI
    Path = sc._gateway.jvm.org.apache.hadoop.fs.Path
    fs = FileSystem.get(URI("hdfs:///"), sc._jsc.hadoopConfiguration())
    candidate_files = [url for url in candidate_files if fs.globStatus(Path(url))]
    return candidate_files
```

Set the desire schema. The raw data saved in the HDFS has almost 700 columns and usually we only need to study a few columns at a time. Hence setting the desired schema before reading the raw data. The function shown below is the example of desired columns.

```
def _get_schema():
    return StructType(
        [
            StructField(
                "data",
                StructType(
                    [
                        StructField("RecordTime", LongType(), nullable=False),
                        StructField("DESIRED_CMSDataset", StringType(), nullable=True),
                        StructField("GlobalJobId", StringType(), nullable=False),
                        StructField("CMS_SubmissionTool", StringType(), nullable=True),
                        StructField("CRAB_DataBlock", StringType(), nullable=True),
                        StructField("CMSPrimaryDataTier", StringType(), nullable=True),
                        StructField("CRAB_Workflow", StringType(), nullable=True)
                    ]
                ),
            ),
        ]
    )
```

a. Read raw data from HDFS. The filter should more generic if this raw DataFrame will be used multiple times in the further analysis. Notice the use of *_DEFAULT_HDFS_FOLDER*, which is the base path set on (a.), *get_candidate_files()* function, and the *schema* variable obtained from the function *_get_schema()*.

```
raw_df = (
    spark.read.option("basePath", _DEFAULT_HDFS_FOLDER)
    .json(
        utils.get_candidate_files(start_date, end_date, spark, _DEFAULT_HDFS_FOLDER),
        schema=schema,
    ).select("data.*")
    .filter(
        f"""CMS_SubmissionTool == 'CRAB'
      AND CMSPrimaryDataTier != 'Unknown'
      AND CRAB_DataBlock IS NOT NULL
      AND RecordTime >= {start_date.timestamp() * 1000}
      AND RecordTime < {end_date.timestamp() * 1000}
      """
    )
    .drop_duplicates(["GlobalJobId"])
)
```

3. Plot graphs and charts to see the pattern of the data.
   Plots and graphs in this project are created with the help of Matplotlib library.
4. Manipulate the DataFrame by, for example, adding more columns to it or joining multiple DataFrame together.
5. Save the useful DataFrame to the analyst's HDFS space for future work. The code below is for saving; writing, the DataFrame to HDFS as a parquet file.

```
output_df.write.parquet("hdfs://analytix/cms/users/nphumekh/crab_2022.parquet")
```

Results and discussion

This project aims to answer a set of questions set by the CRAB team and deliver some tools according to their requests. These are the pioneer questions for the team to start the investigation of users' historical data and CPU efficiency. The main questions are as follows:

1. What is wall clock time spent by each CMS data tier and each job type?
2. What is the average CPU efficiency of each input data type in the time series function?
3. What is the success rate of the Analysis job type?
4. Which are the most used datasets in the last 6 month? How much CPU time was spent on those? How big are those? How many users/tasks hit each dataset?
5. What is the "CRAB active dataset size" and datablock as a function of time?
6. Request: A function capable of doing the Active dataset size calculation by having to input only the time range

The products of this project that answer the above questions are in the form of Python Notebook. The Notebooks are publicly shared on my Github repository named *https://github.com/nutty7fold/cern-crab-data-analysis*. Each Notebook is the tool that can be used to do further analysis and deeper investigation. There are 5 main products which are the customized analysis tools, the analysis and validation of condor raw data, data source schema and definitions, CMS datasets usage analysis, and CRAB Dataset and DataBlock size data source. Their details are shown below.

1. Customized Analysis Tools

   After a few weeks into this project, I noticed that there are many repetitions of tasks when analyzing the raw data. Thus, I created the below tools to help the project progress quicker and more convenient. These files are in this path *https://github.com/nutty7fold/cern-crab-data analysis/blob/main/crab_data_analysis_doc*.

   a. Utils.py

This is simple Python utils that contains all repeated analysis tasks that are re-written as functions and can be shared and used by others. To use the functions in this file, go to *'https://github.com/nutty7fold/cern-crab-data-analysis/blob/main/crab_data_analysis_doc/utils.py'*, save the file in the same directory as the current working notebook, then import it when needed. The details of the functions are as follows:

- _to_dict(df)
  This function takes in 1 parameter *df* of type PySpark DataFrame and returns a Python dictionary.

```
def _to_dict(df):
    rows = [list(row) for row in df.collect()]
    ar = np.array(rows)
    tmp_dict = {}
    for i, column in enumerate(df.columns):
        tmp_dict[column] = list(ar[:, i])
    return tmp_dict
```

- _better_labels(index, data)
  This function takes in 2 parameters: *index* of type list and *data* of type list, and returns a list *labels*. This function helps concatenate the index and its value in term of percentage. For example, index "MINIAOD" and its value "10,000" would become "MINIAOD: 12%".

```
def _better_label(index, data):
    labels = []
    for i in range(len(index)):
        percent = float(data[i])*100/sum(map(float, data))
        labels.append(index[i]+": %.3f"%percent+"%")
    return labels
```

- _other_fields(index, value, lessthan)
  This function takes in 3 parameters: *index* of type list, *value* of type list, and *lessthan* of type int, and returns a Python dictionary. In case of the data having many different indices but we are interested in only the few majorities, often we need to compute the top x indices and the others will be combined as the other field. This function helps to do just that by indicating the desired percentage that we would like to ignore as the *lessthan* parameter. For example, *index* = ["a", "b", "c"], *value* = [50, 49, 1], and *lessthan* = 2. This simply means to discards any value that is less than 2 percent of the entire list as other fields. Hence the result *tmp_dict* = {"index": ["a","b", "Others"], "data_percent": [50, 49, 1], "other_index": ["c"], "other_percent": ["1"]}

```
def _other_fields(index: list, value: list, lessthan: int):
    others = 0
    tmp_dict = {"index": [], "data_percent": [], "other_index": [],"other_percent": []}
    for i in range(len(index)):
        percent = float(value[i])*100/sum(map(float, value))
        if(percent<lessthan):
            others+=percent
            tmp_dict['other_index'].append(index[i])
            tmp_dict['other_percent'].append("%.3f" % percent)
        else:
            tmp_dict['index'].append(index[i])
            tmp_dict['data_percent'].append("%.3f" % percent)
    tmp_dict['index'].append("Others")
    tmp_dict['data_percent'].append("%.3f" % others)
    return tmp_dict
```

- _donut(dictlist, figname)
  This function takes in 2 parameters: *dictlist* of type list of Python dictionary and *figname* of type string and returns nothing. The input *dictlist* is a list of dictionaries of the needed data that you want to plot. The proper structure of *dictlist* is as follows:
  [{ "data field names": [name1, name2, ...], "data values": [value1, value2, ...], "plot name": "name"}, { "data field names": [name1, name2, ...], "data values": [value1, value2, ...], "plot name": "name"}, ...]

One element of the *dictlist* will be plotted as 1 donut chart. You can input multiple elements in case you want to plot them side by side. In case of plotting just 1 chart, do not forget to put it into a list form. It saves the figure as a PNG file to your current working directory according to the input *figname*. It also shows the plot.

- _pie(dictlist, figname)

```
def _donut(dictlist: list, figname: str):
    fig, ax = plt.subplots(nrows=1,ncols=len(dictlist), figsize=(10, 10), subplot_kw={'aspect': 'equal'})
    for i in range(len(dictlist)):
        values_lst = list(dictlist[i].values())
        if(len(dictlist)==1):
            wedges, texts = ax.pie(values_lst[1], wedgeprops={'width': 0.5}, startangle=0)
            ax.set_title(values_lst[2], y=1.08, fontsize=15)
            bbox_props = {'boxstyle': 'square,pad=0.3', 'fc': 'w', 'ec': 'k', 'lw': 0.72 }
            kw = {'arrowprops': {'arrowstyle': "-"},
                    'bbox': bbox_props, 'zorder': 0, 'va':"center"}
            for j, p in enumerate(wedges):
                ang = (p.theta2 - p.theta1)/2. + p.theta1
                y = np.sin(np.deg2rad(ang))
                x = np.cos(np.deg2rad(ang))
                horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
                connectionstyle = "angle,angleA=0,angleB={}".format(ang)
                kw["arrowprops"].update({"connectionstyle": connectionstyle})
                ax.annotate(values_lst[0][j], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                            horizontalalignment=horizontalalignment, **kw)
        else:
            wedges, texts = ax[i].pie(values_lst[1], wedgeprops={'width': 0.5}, startangle=0)
            ax[i].set_title(values_lst[2], y=1.08, fontsize=15)
            bbox_props = {'boxstyle': 'square,pad=0.3', 'fc': 'w', 'ec': 'k', 'lw': 0.72 }
            kw = {'arrowprops': {'arrowstyle': "-"},
                    'bbox': bbox_props, 'zorder': 0, 'va':"center"}
            for j, p in enumerate(wedges):
                ang = (p.theta2 - p.theta1)/2. + p.theta1
                y = np.sin(np.deg2rad(ang))
                x = np.cos(np.deg2rad(ang))
                horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
                connectionstyle = "angle,angleA=0,angleB={}".format(ang)
                kw["arrowprops"].update({"connectionstyle": connectionstyle})
                ax[i].annotate(values_lst[0][j], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                            horizontalalignment=horizontalalignment, **kw)

    plt.savefig(figname+".png")
    plt.subplots_adjust(left=0.5,
                        bottom=0.1,
                        right=2,
                        top=0.9,
                        wspace=0.4,
                        hspace=0.4)
    plt.show()
```

This function takes in 2 parameters: *dictlist* of type list of Python dictionary and *figname* of type string and returns

nothing. *dictlist* is a list of dictionaries of the needed data that you want to plot. The proper structure of *dictlist* is as follows: [{ "data field names": [name1, name2, ...], "data values": [value1, value2, ...], "plot name": "name"}, { "data field names": [name1, name2, ...], "data values": [value1, value2, ...], "plot name": "name"}, ...]
Note that one element of the *dictlist* will be plotted as 1 pie chart. You can input multiple elements in case you want to plot them side by side. In case of plotting just 1 chart, do not forget to put it into a list form. It saves the figure as a PNG file to your current working directory according to the input *figname*. It also shows the plot.

```
def _pie(dictlist: list, figname: str):
    fig, ax = plt.subplots(nrows=1,ncols=len(dictlist), figsize=(10, 10), subplot_kw={'aspect': 'equal'})
    for i in range(len(dictlist)):
        values_lst = list(dictlist[i].values())
        if(len(dictlist)==1):
            ax.pie(values_lst[1], labels=values_lst[0], autopct='%1.1f%%',
            shadow=False, startangle=90)
            ax.axis('equal')
            ax.set_title(values_lst[2], fontsize=15)
        else:
            ax[i].pie(values_lst[1], labels=values_lst[0], autopct='%1.1f%%',
                shadow=False, startangle=90)
            ax[i].axis('equal')
            ax[i].set_title(values_lst[2], fontsize=15)

    plt.savefig(figname+".png")
    plt.subplots_adjust(left=0.5,
                        bottom=0.1,
                        right=2,
                        top=0.9,
                        wspace=0.4,
                        hspace=0.4)
    plt.show()
```

- _line_graph(xvalues, dictlist, figinfo, figname, show_mean)
  This function takes in 5 parameters: *xvalues* of type list, I *dictlist* of type list of Python dictionary, *figinfo* of type dict, *figname* of type string, and *show_mean* of type Boolean. It returns nothing. *xvalues* is a list of x-axis values. Note that plotting several y lines requires that the x-axis is in the same scale thus only input it 1 time is enough. *diclist* is a list of

dictionaries of the needed data that you want to plot. The proper structure of *dictlist* is as follows:
[{"y-axis1 values": [y1, y2, ...], "label1": "label", "color1": "color"}, {"y-axis2 values": [y1, y2, ...], "label2": "label", "color2": "color"}, ...]. *figinfo* is a dictionary containing the figure information. It can be declared as follows: *figinfo* = {"x_label": "this is x-axis","y_label": "this is y-axis","title": "this is plot"}. *figname* is a string to be saved into a PNG file. Lastly, *show_mean* is a Boolean to select whether you want to plot the mean values of each line.

```
def _line_graph(xvalues: list, dictlist: list, figinfo: dict, figname: str, show_mean: bool):
    fig, ax = plt.subplots(figsize=(10, 10))

    for i in range(len(dictlist)):
        values_lst = list(dictlist[i].values())
        ax.plot(xvalues, values_lst[0], color=values_lst[2], label=values_lst[1])
        if(show_mean):
            plt.hlines(mean(values_lst[0]), min(xvalues), max(xvalues), linestyles="dashed", colors=values_lst[2])
            ax.text(mean(xvalues),mean(values_lst[0]),'%f' % (mean(values_lst[0])))

    ax.set(xlabel=figinfo["x_label"], ylabel=figinfo["y_label"],
           title=figinfo["title"])
    ax.grid()
    plt.legend()
    plt.savefig(figname+".png")
    plt.show()
```

- short_datasetname(lst)
  This function takes in 1 parameter: *lst* of type list and returns a list *tmp*. Note that the datasets saved in the HDFS are structured as /xxxx/yyyy/zzzz and can be very long in length. This function simply cut the first two parts to not be longer that 15 in length but keeps the last part as it is the most important part of the dataset name. Example output is [“/SingleMuon/Run2018C-UL2018…/MINIAOD”, “QCD_HT700to1000…/RunIIAutumn18Mi…/MINIAODSIM”]

```
def short_datasetname(lst: list):
    tmp = []
    for name in lst:
        if((len(name.split("/")[1])<15) & (len(name.split("/")[2])<15)):
            tmp.append("/"+"/".join([name.split("/")[i] for i in range(1,4)]))
        elif((len(name.split("/")[2])>15) & (len(name.split("/")[1])<15)):
            tmp.append("/"+name.split("/")[1]+"/"+name.split("/")[2][0:15]+".../"+name.split("/")[3])
        elif((len(name.split("/")[1])>15) & (len(name.split("/")[2])<15)):
            tmp.append("/"+name.split("/")[1][0:15]+".../"+name.split("/")[2]+"/"+name.split("/")[3])
        else:
            tmp.append("/"+".../".join([name.split("/")[i] for i in range(1,3)])+".../"+name.split("/")[3])
    return tmp
```

- _exitcode_info(exitcode)
  This function takes 1 parameter: *exitcode* of type int and return a dictionary *exitcode_info*. As exit codes are in integer, it takes more effort to translate them into meaningful strings. Thus, this function is created to make translating them more convenient. Nevertheless, it needs maintenance as the exit codes have constant updates.

```
def _exitcode_info(exitcode: int):
    exitcode_info = {"ExitCode": exitcode, "Type": "", "Meaning": exitcode_dict.get(str(exitcode), "")}

    """ending range plus one as python range exclude the last number"""
    if exitcode in range(1, 512+1):
        exitcode_info['Type'] = "standard ones in Unix and indicate a CMSSW abort that the cmsRun did not catch as exception"
    elif exitcode in range(7000, 9000+1):
        exitcode_info['Type'] = "cmsRun (CMSSW) exit codes. These codes may depend on specific CMSSW version"
    elif exitcode in range(10000, 19999+1):
        exitcode_info['Type'] = "Failures related to the environment setup"
    elif exitcode in range(50000, 59999+1):
        exitcode_info['Type'] = "Failures related executable file"
    elif exitcode in range(60000, 69999+1):
        exitcode_info['Type'] = "Failures related staging-OUT"
    elif exitcode in range(70000, 79999+1):
        exitcode_info['Type'] = "Failures related only for WMAgent."
    elif exitcode in range(80000, 89999+1):
        exitcode_info['Type'] = "Failures related only for CRAB3"
    elif exitcode in range(90000, 99999+1):
        exitcode_info['Type'] = "Other problems which does not fit to any range before"

    return exitcode_info
```

- d_size_df(HDFS_DBS_FILES, HDFS_DBS_DATASETS)
  This function takes in 2 parameters: *HDFS_DBS_FILES* of type string with '/project/awg/cms/CMS_DBS3_PROD_GLOBAL/current/FILES/part-m-00000' as its default and *HDFS_DBS_DATASETS* of type string with '/project/awg/cms/CMS_DBS3_PROD_GLOBAL/current/DATASETS/part-m-00000' as its default and returns a

DataFrame with the following schema:

```
root
 |-- d_dataset: string (nullable = true)
 |-- Dataset_Size: double (nullable = true)
```

```
def d_size_df(HDFS_DBS_FILES='/project/awg/cms/CMS_DBS3_PROD_GLOBAL/current/FILES/part-m-00000',
              HDFS_DBS_DATASETS='/project/awg/cms/CMS_DBS3_PROD_GLOBAL/current/DATASETS/part-m-00000'):
    csvreader = spark.read.format('csv') \
        .option('nullValue', 'null') \
        .option('mode', 'FAILFAST')
    dbs_files = csvreader.schema(cms_schemas.schema_files()) \
            .load(HDFS_DBS_FILES) \
            .select(['f_file_size', 'f_dataset_id'])\
            .withColumnRenamed('f_dataset_id', 'DATASET_ID')
    dbs_datasets = csvreader.schema(cms_schemas.schema_datasets()) \
            .load(HDFS_DBS_DATASETS) \
            .select(['d_dataset_id', 'd_dataset']).withColumnRenamed('d_dataset_id', 'DATASET_ID')
    d_size_df = dbs_datasets.join(dbs_files, ['DATASET_ID'], how='left') \
            .groupby('d_dataset') \
            .agg(
                _sum('f_file_size').alias('Dataset_Size'))
    return d_size_df
```

- b_size_df(HDFS_DFS_BLOCKS)
  This function takes in 1 parameter: HDFS_DFS_BLOCKS of type string with '/project/awg/cms/CMS_DBS3_PROD_GLOBAL/current/BLOCKS/part-m-00000' as its default and returns a DataFrame with the following schema:

```
root
 |-- b_block_name: string (nullable = true)
 |-- Block_Size: double (nullable = true)
```

```
def b_size_df(HDFS_DFS_BLOCKS = '/project/awg/cms/CMS_DBS3_PROD_GLOBAL/current/BLOCKS/part-m-00000'):
    csvreader = spark.read.format('csv') \
        .option('nullValue', 'null') \
        .option('mode', 'FAILFAST')
    b_size_df = csvreader.schema(cms_schemas.schema_blocks()) \
        .load(HDFS_DFS_BLOCKS) \
        .select(['b_block_name', 'b_block_size']).withColumnRenamed('b_block_size', 'Block_Size')
    return b_size_df
```

b. exitcode_dict.py
This contains a dictionary of the exit codes and their meaning. It should be maintained and updated constantly. This file is used inside the above Utils.py so the only step is to save it in the same directory as the Utils.py. Go to 'https://github.com/nutty7fold/cern-crab-data-analysis/blob/main/crab_data_analysis_doc/ exitcode_dict.py' to save the file. A few lines of the exit code dictionary are shown below. The definitions are referenced from *https://twiki.cern.ch/twiki/bin/view/CMSPublic/JobExitCodes*

```
exitcode_dict = {"7000": "Exception from command line processing",
                 "7001": "Configuration File Not Found",
                 "7002": "Configuration File Read Error",
                 "8001": "Other CMS Exception",
                 "8002": "std::exception (other than bad_alloc)",
                 "8003": "Unknown Exception",
                 "8004": "std::bad_alloc (memory exhaustion)",
                 "8005": "Bad Exception Type (e.g throwing a string)",
```

c. Analysis_Guide.ipynb
This is a complete guide about how to read the raw data from HDFS using PySpark, basic query commands, and the detailed explanations and examples of the functions in Utils.py. Access this file here 'https://github.com/nutty7fold/cern-crab-data-analysis/blob/main/crab_data_analysis_doc/analysis_guide.ipynb'

d. Analysis_Example.ipynb
This is a re-write of the *final-brief-analysis-condor-raw-data.ipynb* with the purpose to showcase the usage of Utils.py ready-to-use functions. The Notebook show plots and queries of the following tasks:
   - Task1 - Sum of "WallClockHr" by "CMSPrimaryDataTier"
   - Task2 - Sum of "WallClockHr" by "Type" ['production', 'analysis']
   - Task3 - Sum of "WallClockHr" filter "Type"['analysis'] by "CRAB_DataBlock" ['MCFakeBlock', Else] (MC Prod vs Analysis)
   - Task4 - Average CPU Efficiency group by "RecordTime" each hour and "InputData" ['onsite', 'offsite']
   - Task5 - Success rate of the "Type" ['analysis']

2. Analysis and Validation of condor raw data
This is written in the Notebook named final-brief-analysis-condor-raw-data.ipynb. This notebook is created to validate the HDFS raw data available in the condor raw source, further understand the data columns, and investigate for appropriate filters. The file was replicated/ validated by comparing with

the data on Grafana with help of Dario Mapelli. The data used to validate are as follows:

a. Task1 - Sum of "WallClockHr" by "CMSPrimaryDataTier"
b. Task2 - Sum of "WallClockHr" by "Type" ['production', 'analysis']
c. Task3 - Sum of "WallClockHr" filter "Type"['analysis'] by "CRAB_DataBlock" ['MCFakeBlock', Else] (MC Prod vs Analysis)
d. Task4 - Average CPU Efficiency group by "RecordTime" each hour and "InputData" ['onsite', 'offsite']
e. Task5 - Success rate of the "Type" ['analysis']

3. Data source schema and definitions

This is the Git issue 7313 with the Notebook named draft-issue-7313.ipynb. This notebook defines the schema and the usage/ manipulation of each column of the data source to be stored in the database. It aims for minimum schema to answer the following questions:

"cpueff/walltime/failure rate of jobs group by input data tier/site/mc production/offsite vs onsite"

Unfortunately, the data source and schema were not used for further data analysis. However, there are some important definitions to note from this notebook.

a. Calculate the Average CPU Efficiency

Average CPU Efficeincy can be calculated according to the below equation. Also note that WallClockHr*RequestCpus = CoreHr and WallClockHr = RemoteWallClockTime / 3600.0. This is reference from *https://github.com/dmwm/cms-htcondor-es/blob/7fdcb7667b39081ddff98da26ad4e3ed33f9e244/src/htcondor_es/convert_to_json.py#L838*

```
def _cal_avg_cpu_eff(CpuTimeHr,WallClockHr,RequestCpus):
    return (CpuTimeHr)/(WallClockHr*RequestCpus)
```

b. Derive the success rate from the exit code

There are many different exit codes as seen from the exitcode_dixt.py but the success rate is calculated considering whether the task fail or succeed. To calculate this, it is important to

understand that when the exit code is 0, the task is considered succeeded, and when the exit code is other numbers, the task is considered failed.

```
def _manipulate_exitcode(ExitCode):
    if(ExitCode==0):
        return ('Success')
    elif (ExitCode!=0 & ExitCode.isNotNull()):
        return ('Fail')
    else:
        #when ExitCode.isNull()
        return ('Null Exit Code')
```

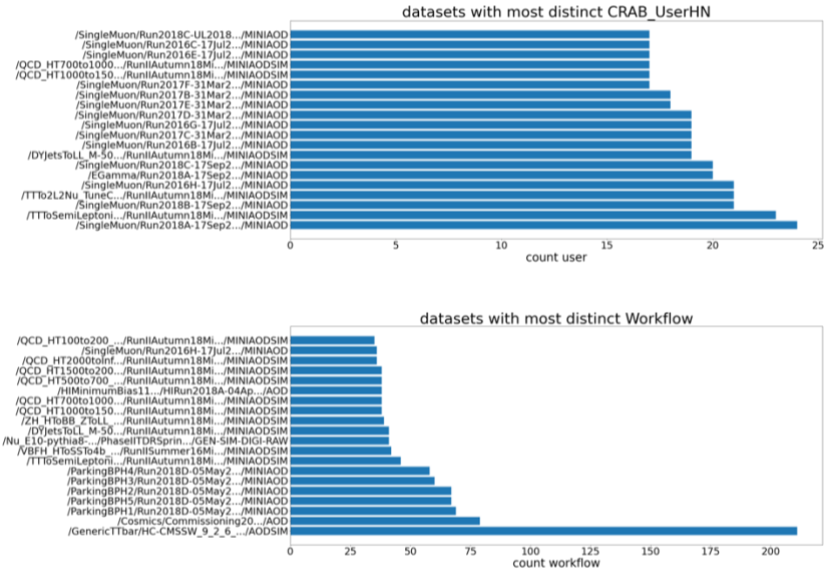
c. Determine the MC Production job

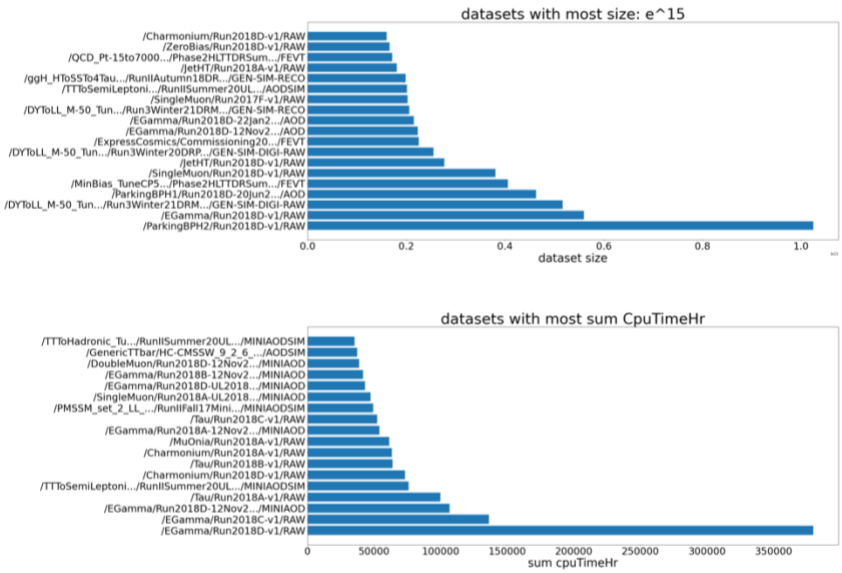
To determine whether the job is a MC Production job or not, check according to the code below.

```
def _is_mc_prod_job(CRAB_DataBlock, Type):
    if(CRAB_DataBlock=='MCFakeBlock' & Type=='analysis'):
        return True
    else:
        return False
```

4. CMS Datasets Usage Analysis

This work is written in the Notebook named Top_used_CMS_dataset.ipynb. This notebook studies the most used datasets determined by their distinct CRAB_UserHN and distinct workflow. This leads to finding the answer of which datasets has more frequent access and how long should those datasets be kept. This notebook also shows each dataset's CPU time hour and size. The results from the interested time range of 1 week between 05/01/22 - 05/08/22 is shown below.





5. CRAB Dataset and DataBlock Size Data Source
   This work is done in the Notebook named get_cms_dataset_block_size.ipynb. This notebook saves the active CRAB datasets and datablocks as files. Active datasets in this case refer to the datasets that appear in the Condor Raw path in that specific day. The data is saved one day per one file. Time range is from 1st of January 2022 to 30th of June 2022 (6 months). The data can be accessed via the following HDFS path: *'hdfs://analytix/cms/users/nphumekh/crab_dataset_datablock_size_2022_month_day.parquet'*. The schema of the content is shown below.

```
root
 |-- CMSPrimaryDataTier: string (nullable = true)
 |-- DESIRED_CMSDataset: string (nullable = true)
 |-- Dataset_Size: double (nullable = true)
 |-- CRAB_DataBlock: string (nullable = true)
 |-- Block_Size: double (nullable = true)
 |-- CRAB_Workflow: string (nullable = true)
 |-- First_Access: long (nullable = true)
 |-- Last_Access: long (nullable = true)
```

   One important note in the file is the *get_active_dataset_datablock_size_df* function. With this function, the loop for saving multiple files can happen. It takes in 2 parameters: *start_date* of type datetime and *end_date* of type datetime and returns a PySpark DataFrame with the above schema. The start and end date dates are used as the time range of each output DataFrame. Then this function joins the dataset size and datablock size with your desired raw DataFrame.

The loop is shown below. By defining the month and number of days in that month, the loop can work for all month without the need of any further modification.

```
#note: there is a use of utils.get_candidate_files
def get_active_dataset_datablock_size_df(start_date, end_date):
    raw_df = (
        spark.read.option("basePath", _DEFAULT_HDFS_FOLDER)
        .json(
            utils.get_candidate_files(start_date, end_date, spark, _DEFAULT_HDFS_FOLDER),
            schema=schema,
        ).select("data.*")
        .filter(
            f"""CMS_SubmissionTool == 'CRAB'
          AND CMSPrimaryDataTier != 'Unknown'
          AND CRAB_DataBlock IS NOT NULL
          AND RecordTime >= {start_date.timestamp() * 1000}
          AND RecordTime < {end_date.timestamp() * 1000}
          """
        )
        .drop_duplicates(["GlobalJobId"])
    )
    datasets = raw_df.select(['CMSPrimaryDataTier', 'DESIRED_CMSDataset']).drop_duplicates(['DESIRED_CMSDataset'])
    dateset_size = datasets.join(d_size_df, datasets.DESIRED_CMSDataset==d_size_df.d_dataset)\
                           .select(['CMSPrimaryDataTier', 'DESIRED_CMSDataset', 'Dataset_Size'])
    blocks = raw_df.withColumn("First_Access", _min('RecordTime').over(Window.partitionBy('CRAB_DataBlock')))\
            .withColumn("Last_Access", _max('RecordTime').over(Window.partitionBy('CRAB_DataBlock')))\
            .select(['DESIRED_CMSDataset', 'CRAB_DataBlock', 'CRAB_Workflow', 'First_Access', 'Last_Access'])\
            .drop_duplicates(['CRAB_DataBlock'])
    block_size = blocks.join(b_size_df, blocks.CRAB_DataBlock==b_size_df.b_block_name)\
                           .select(['DESIRED_CMSDataset', 'CRAB_DataBlock', 'Block_Size',
                                    'CRAB_Workflow', 'First_Access', 'Last_Access'])
    output_df = dateset_size.join(block_size, dateset_size.DESIRED_CMSDataset == block_size.DESIRED_CMSDataset)\
                .select([dateset_size.CMSPrimaryDataTier, dateset_size.DESIRED_CMSDataset, \
                         'Dataset_Size', 'CRAB_DataBlock', 'Block_Size', 'CRAB_Workflow', \
                         'First_Access', 'Last_Access'])\
                .orderBy(col('Dataset_Size').desc())
    return output_df
```

```
month = 6
days = 30
```

```
for day in range(1, days+1):
    if(day==days):
        start_date = datetime(2022, month, day)
        end_date = datetime(2022, month+1, 1)
    else:
        start_date = datetime(2022, month, day)
        end_date = datetime(2022, month, day+1)
    output_df = get_active_dataset_datablock_size_df(start_date, end_date)

    output_df.write.parquet("hdfs://analytix/cms/users/nphumekh/crab_dataset_datablock_size_2022_%s_%s.parquet" \
                            % (str(month), str(day)))
```

Results Summary

To sum up the results, both the analysis and validation of condor raw data and the Analysis_Example.ipynb answer the questions of the amount of wall clock time spent by each CMS data tier and each job type, the average CPU efficiency of each input data type in the time series function, and the success rate of the Analysis job type. Furthermore, the CMS datasets usage analysis answer the questions of the most used datasets in the last 6 month, the amount of CPU time spent on those, their sizes, and the number of users/tasks hit each dataset. Lastly, the CRAB dataset and datablock size data source answer the question of the "CRAB active dataset size" and datablock as a function of time and provide a function capable of doing the Active dataset size calculation by having to input only the time range named *get_active_dataset_datablock_size_df*. Most importantly, the data source saved as *hdfs://analytix/cms/users/nphumekh/crab_dataset_datablock_size_2022_month_day.parquet* can be used to do further investigation in the future.

Conclusion

This project takes on an initiative to analyze the historical data of Grid users' analysis jobs that are sent through CRAB. It provides answers to the important questions asked by the CRAB Team and shows that there are rooms to improve the CPU usage and prevent job failure. The tools I created can be used to do analysis of the wall time usage, CPU efficiency calculation, and the CRAB active datasets and datablocks. Many functions were created to reduce the work of repeated tasks thus allow more convenient and faster analysis. Lastly, a data source for the CRAB active dataset and datablock size were saved in the HDFS for future analysis and investigation.

Chapter 3
Diary

Monday, 6 June 2022

I arrived at the Geneva airport and Dr. Norraphat Srimanobhas welcomed us and showed us how to buy the tram and bus tickets. Luckily, I bought the prepaid card for cash spending of multiple foreign currencies so spending money and buying things here were not a hassle. After some traveling, we all arrived at the hostel in St. Genis then we noticed that there was nobody there. It turned out that it was a holiday and 2 of us needed to travel back to CERN to get the room keys. It was a fun experience on the very first day in France/ Switzerland.

Tuesday, 7 June 2022

All four of us woke up early as we had the first meeting in CERN with the coordinators and the first group of summer students. We went to CERN and had the first breakfast at R1 then walked around a little bit before going to the meeting room. The hosts gave a short introduction about the program then they let the students introduce themselves. The atmosphere was chill and welcoming, I realized that it will be a great summer this year.

Wednesday, 8 June 2022

Today one of the high school students was sick so all of us Thai summer students took him and the teacher to the CERN hospital. It was raining the whole day luckily; I bought the raincoat. I learned that the weather here changes rapidly, so it is important to check the weather before leaving the hostel.

Thursday, 9 June 2022

I had a Zoom meeting with my supervisor as he was still busy in Italy. He explained that the CRAB team members who are at CERN will be the ones to take care of my day-to-day tasks. In the afternoon, I went to visit the CERN Data Center and the LINAC4. The students were separated into 2 groups, so I had the chance to get to know new friends.

Friday, 10 June 2022

I finally got the office key from the locks and keys department so today I arrived at the office before anyone and was able to open the office by myself. There was not much for me to do accept following up with the access/ permission to the system and the HDFS cluster. One of my CRAB team members is also Thai, his name is Wa, he helped me with the requests.

Saturday, 11 June 2022
I chilled at the hostel. It was peaceful around where I live. Also, I borrowed a hair dryer from the hostel, and it was the last one. So, I and my other Thai friend shared that hair dryer. We needed to coordinate when which of us will take a shower first. Today I also sketched a picture of my favorite animation character, GrimmJow.

Sunday, 12 June 2022
I wanted to cook something, but I did not have any salt. None of the Thai friends has any salt as well. They only have Knore but I did not know how to cook with it. So, my food today has one flavor which was soy sauce.

Monday, 13 June 2022
Today Wa lectured me briefly about the CRAB work and system architecture. It was hot in the office, so I asked him if we have any air conditioner in the office. He told me that our office is the only office in the building that has the air conditioner, but it was old and would make loud noise when turned on. So, we would only turn it on when it is really hot like today.

Tuesday, 14 June 2022
I worked on the studying of the CRAB system as told by Wa. I also took note on Notion so that it can be shared with the team as well. In the afternoon, I went to visit the ATLAS visitor center and the Synchrocyclotron. I was excited to watch the 3D video of the assembling of the ATLAS. When we took a walk to ATLAS visitor center, there was construction going on and it was hot outside, so we had to be cautious.

Wednesday, 15 June 2022
I got the Analytix cluster access today, so I was able to start learning to use the SWAN environment. I ended up spending the whole day learning its functions and features. I learned the Spark SQL commands as well as the HDFS common commands. I also created my own directory in the CMS user's directory which means now I have my own space in the HDFS storage.

Thursday, 16 June 2022
Today I was working from the hostel. Wa sent me the CMS HTCondor Github repository. I needed to understand the backend coding more to move on with my project. It was quite challenging, but I had fun studying the code as I had an experience working with backend engineering prior to coming here.

Friday, 17 June 2022
Today I was studying the code like yesterday. There were many small things that could not be missed otherwise I would have a different interpretation of the code. I was contemplating whether I should ask for help but, in the end, I tried to focus more and I think I Google about two thousand times today.

Saturday, 18 June 2022
Today there was a gathering of the summer students by the lake. It was kind of a barbeque, bring-your-own-booze party. I went there with another Thai friend. I have never been to a party like this before. It turned out to be quite fun. I got to talk to people that I have never known before. Some of them also went to swim in the lake. It was a chill Saturday.

Sunday, 19 June 2022
Today I stayed at the hostel and did some work. I also drew a dress that was worn by Angelina Jolie.

Monday, 20 June 2022
I was able to finish my first SWAN Notebook about the daily average CPU efficiency group by each site. I faced a problem about the average CPU efficiency is higher than 100. But Wa was not free to discuss this with me, so I had to wait until tomorrow to solve this. I also did more study on the ClassAds slides given to me by my supervisor Diego.

Tuesday, 21 June 2022
Today Wa introduced me to the Monitoring Team Grafana page which contains many graphs and charts created from the aggregated data. This is my first time working with Grafana, so I took some time to get familiar with it. Both of my CRAB team members, Wa and Dario, were also not quite familiar with Grafana, so we took this opportunity learn more about it together.

Wednesday, 22 June 2022
I continued working on my SWAN Notebook about the daily average CPU efficiency. There were many complications about the correct ways to calculate the average CPU efficiency and as my supervisor did not have that much free time to discuss things with me, it took me more time to read things by myself.

Thursday, 23 June 2022
I had progress on the SWAN Notebook so Wa asked me to share the code on Github. I was curious whether the CRAB Team or CERN itself has its own Github but it turned out there was not any. So, I needed to share my progress on my personal

Github repository. I also started writing a Markdown file about the CRAB team as there was not any document for the newcomer to the CRAB team, so the team members decided to let me do it. It was another challenging but fun work as I really enjoy documenting things.

Friday, 24 June 2022
I worked more on studying how to use Grafana. More importantly, I looked through the existing charts and graphs to see which ones I could use as a guide of my work. I realized that even though creating a graph on Grafana sounds simple, it is in fact a difficult task that needs understanding of many components to be able to create an optimize and accurate chart.

Saturday, 25 June 2022
I went out to walk around the city. There was a place called Manor where I was told that it was a fancy shopping mall here in Geneva. But it was not as fancy as Siam Paragon. I think people here in Geneva are more about supporting local business, so the mall is not a popular thing here.

Sunday, 26 June 2022
I was also working a bit even though it was Sunday as I was faced with new information that got me really curious. I found charts and graphs on one of the MONIT Grafana pages that matches with what I needed to create. So, I was curious whether I should continue my work if the charts needed are already there. But sadly, nobody in the team answered my messages (understandable as it is weekend). At the end of the day, I stopped working as I realized I should wait until Monday.

Monday, 27 June 2022
Today I also arrived at the office before others, so I had some time so get warm coffee from the café down the corridor. I felt happy that my office is near the said café as I love having snack when working. Snacks and coffee make me think faster and focus on working. About the work, I had completed most of the tasks given to me, but there was this one query that made things complicated. It was about the grouping between the Analysis job and the MC Production job. Nobody in the team knew which filters to use, so I had to try different filters to investigate it with trials and errors. Luckily, Wa found a document about this problem in the afternoon.

Tuesday, 28 June 2022
The lecture started today. As I was not a physics student, I was just sitting in to listen to new jargons and in awe of how the lecturers were so knowledgeable and cool when they talk about physics. My other Thai friend who studies physics told me that even for him, the topic was challenging. We had a good laugh about it as well. In the

afternoon, I finished my first task about validating the raw data in the Condor Raw. And I started drafting the schema of the aggregated data to be stored in the HDFS and used as the data source in Grafana.

Wednesday, 29 June 2022
I asked my CRAB team member Dario to help check my plots. He compared them with the existing plots in Grafana and confirmed that the filter used are accurate. I was happy about it as I was new to this kind of work. It was challenging but fun for me.

Thursday, 30 June 2022
Today I continued my work on defining the schema and the usage/ manipulation of each column of the datasource to be stored in the database. The aim is for having minimum schema to answer the questions about the CPU efficiency, wall time hour, failure rate of jobs then grouping them by input data tier, site, MC production type, analysis type, offsite, and onsite job. I also got the delivery of my Swiss Pass card which is for unlimited access to all the public transportation in Zone 10. It is a necessary, in my opinion.

Friday, 1 July 2022
I was able to save the aggregated files into my space in HDFS. The files contain data one day per file of the month May. I was really happy about this data source. There was also a problem I faced today. After saving the files, I was supposed to investigate the files size to determine whether it will be possible to save many years' worth of files in the ElasticSearch. But I was not able to find a way to actually check the file size in HDFS. I went back to the hostel with many thoughts in mind.

Saturday, 2 July 2022
I left a frozen mushroom bag in the normal fridge instead of the freezer. Then the mushroom water melted and leaked through the fridge. Now my room smelled like frozen mushroom the entire day. It was the first time I did not enjoy staying in my room.

Sunday, 3 July 2022
I went out to do some shopping as I just bought a ticket to a Slipknot concert on the 1st of August, so I needed some cool clothing. I ended up buying a new skirt and a t-shirt from a shop in Bel Air.

Monday, 4 July 2022
Today after having lunch, it was raining cats and dogs. Even though I bought an umbrella, I did not want my shoes and trousers to get wet, so I ended up working in

the library for the rest of the day. I actually prefer working in the library because there are air conditioners in there 24/7. I think I work better when it is a bit cold. I also had a Zoom meeting with all the CRAB team.

Tuesday, 5 July 2022
There was a launch for LHC Run3 today. I watched the live in my office and did not go to the gathering as a big group of gathering is not my preference. The launch was exciting. Many cool and smart people working together to make this happen which inspires me so much. As for the work, apaart from my main work, Wa asked me to write another Markdown for the day-1 that will come to CRAB in the future.

Wednesday, 6 July 2022
Today was chill. I worked in the hostel. I also cooked myself hot ramyon, fried chicken drumsticks, and scrambled eggs.

Thursday, 7 July 2022
I finished the CRAB Day-1 guide today. Stefano also gave new comments about the analysis guide that I am working on as my main task. This is the first time I create such guide for other analyst that is not myself. He reminded me to be explicit and write in more details about what I created so that others can easily understand them. In the evening, all the Thai summer students, along with Wa and Dr.Phat went out for dinner together. We chose a Thai restaurant, and we thoroughly enjoyed the food.

Friday, 8 July 2022
Today there was bad news. One of the Thai summer students got Covid positive. He had to move from the hostel to the quarantine hotel. I also quarantine myself at the hostel even though my test was negative. I learned that people here treat Covid as a common flu, they say that I just must wear a mask when going out to buy food. But anyhow I still had some food in the fridge, so I stay in the hostel the whole day.

Saturday, 9 July 2022
I and two other Thai friends went out to a chill Bar. One of the friends was really good at finding nice place to drink. We wondered around Cornavin before finding the place. I believe this was the first time I see Geneva when the Sun is completely down. As the Sun sets at around 9-10 PM every day. Geneva without the Sun was even calmer and more peaceful than during the day.

Sunday, 10 July 2022
I went out to Bel Air and bought a chocolate from a well-known Switzerland chocolate shop called Laderach. They sold the chocolate by breaking it into small

pieces, weighted the pieces, then sold by actual price. I loved the strawberry flavor and the nutty flavor.

Monday, 11 July 2022
Today I was able to finish my Notebook on the analysis guide, analysis example, and the utils I created for CRAB. It was fun for me to develop functions that help visualizing and querying the raw data easier. I was encouraged by the team to write a detailed guide to my functions, so I had the passion for this work. I also had a fun session discussing about HTML and CSS with Wa. He was interested in renewing the UX/UI of the current CRAB document page.

Tuesday, 12 July 2022
I was working on the exit code translation. There was not any dictionary on the exit codes, so I decided to create a simple dictionary that translates the integer to meaningful string exit codes.

Wednesday, 13 July 2022
Today I and Thai summer students went to the Thai Embassy in Geneva. We had a short talk about what we are doing at CERN. We were offered Switzerland chocolate and it was super delicious. I had a great time explaining what I do and sharing experience to the people who are not in the field of science and computer. We were also promised that next time we meet, there will be nice Thai food waiting. So, I am looking forward to the next meeting.

Thursday, 14 July 2022
Today Wa taught me about the ElasticSearch index with some data, aggregated by hour. Even after more than a month here, there are still many things to learn about the system at CERN and the CRAB team also has many complicated data structure that needs time to be fully understood.
Friday, 15 July 2022
Today I did not feel so good. I think my health is not in its best condition, so I asked for a break from my team. They were understanding and allowed me to take a day off. I took a headache medicine and rested for the whole day. I also cooked myself hot ramyon as it is my comfort food, I always feel better after eating it.

Saturday, 16 July 2022
Today I and two other Thai Summer students woke up at 4AM because we were taking TPG train to Paris. We met with a friend in Paris and had lunch there. In the afternoon, we went to Versailles. It was huge and a lot more beautiful than I imagined. In the evening, we walked around Eiffel tower, took some pictures, then

had dinner there. Paris was not like what I had imagined but it sure had its charm. I would love to come back here again someday.

Sunday, 17 July 2022
We were still in Paris. We woke early to have breakfast Paris style. Unfortunately, it was almost the hottest day of the year. I almost ran out of breath walking around. But I did not give up as I had to see what's inside the famous Louvre regardless of how hot and long the queuing was. If I remember correctly, we walked around in Louvre for more than 4 hours. I think my feet will ache for a couple of days after this. Then we traveled back Geneva with full hearts.

Monday, 18 July 2022
After coming back to Paris, my team told me to work remote so that I can quarantine for a bit to make sure that I am Covid negative. So today I spent the whole day in my hostel. My other friends were sick when we were in Paris, but I did not have any symptoms nor any sickness. I had fun but also a bit tired from the traveling.

Tuesday, 19 July 2022
Today I started working on finding the size of CRAB active datasets. There were many complications. Mainly because nobody in the team has tried to query the datasets size before. I was faced with some misunderstanding as there were not any column name in the new path that I was told to navigate through. Also, I tried working with the lucene query in Grafana. In the afternoon, the CRAB team also had a meeting. I updated my work with the team members. It was always refreshing talking to these people.

Wednesday, 20 July 2022
I worked on finding dataset size. I was surprised that it was more complicated than I expected. I assumed that my team members noticed that I was quite down about not making fast progress on this task, so Wa asked if I and all Thai summer students are free for a chill dinner. Of course, I had to say yes.

Thursday, 21 July 2022
Today I went to the office as I was sure that I did not have any Covid symptoms. The weather was nicer, so I was happy. But I still had problems working on Grafana. Wa told me to consult Ceyhun but he was offline. I think I will have to be more patient.

Friday, 22 July 2022
There was not so much about work. But I had fun today as after work, all the Thai friends and one French friend all went out to have dinner together. We went to

James' Pub which is the most delicious Thai restaurant in Geneva. Today I got to tried eating Fresh shrimp with Thai seafood sauce in Geneva for the first time. I believe it was spicier than what I have in Thai. The owner of the restaurant really deserves a shout-out.

Saturday, 23 July 2022
Today I and another Thai friend went to Zermatt. We went there by the slow, scenic train. We met with my Thai university friend. It was beautiful.

Sunday, 24 July 2022
I stayed at the hostel the whole day because I was tired from sitting in the train all day yesterday.

Monday, 25 July 2022
Something funny happened today at work. I was again surprised that CERN did not use Grafana Enterprise. Which means that the feature that I wanted to use is not available. These days I have been preparing the poster session and I wanted to export Grafana chart as PDF so that I can have a better picture for my poster which. This feature is available in Grafana Enterprise, not the normal Grafana. So, I decided to just screenshot the graphs.

Wednesday, 27 July 2022
Today I was still working on my poster presentation as I will present it on this Thursday. There was also bad news that another Thai friend got Covid. I hope he get well soon so that we can have lunch together again. In the afternoon, I went to get my poster. As I saw the printed version, I think I laughed a bit because it was a lot bigger than expected. I felt great about it. I placed it in the office then went to have dinner with Wa and other Thai friends.

Thursday, 28 July 2022
Today I presented the poster session. There were many cool posters posted in the hallway. Many people came to walk around and talk casually about the poster. There were free coffee and snacks served as well. I was happy to be able to explain my work to other people. After the presentation, all the Thai friends, together with Dr. Phat went out to have dinner at an Italian place. The pizza was so big that we had to pack it back to the hostel.

Friday, 29 July 2022
After finishing my poster presentation, the team allow me to rest for a bit so today was a chill day. I also tried to cook Thai food, but my skill was not ready. It turned out not so edible, so I gave up and just cook some hot ramyon.

Saturday, 30 July 2022
I and the Thai friends went out for a drink because one of the friends will fly back to Thailand soon. We went separately and I got lost for a bit. Even after staying here for a while, I still could not remember all the roads around Manor.

Sunday, 31 July 2022
I chilled at the hostel and drew a cute character from my favorite animation called Hunter X Hunter.

Monday, 1 August 2022
After work, I went to Slipknot concert. I had a blast. Probably the best concert in my life.

Tuesday, 2 August 2022
Today I continued working where I left off. The CRAB team also held a meeting where I got to explain my progress.

Wednesday, 3 August 2022
I had a job interview today. It went quite well but I was really nervous. After the interview, I continued my work in the hostel. These past few days I have been having runny nose and itchy eyes. I think I am allergic to something so, I bought antihistamine medicine. Hoping to get better soon.

Friday, 5 August 2022
Today I encountered new keywords at work. It was about data block size. As I have been working on the dataset size, my supervisor told me to also include the data block size in my study as well. So, I have been trying to understand it. I worked in the library; Wa was kind enough to walk to the library to guide me about my confusion. In the evening, I and the Thai friends went to see Jet d'eau and there was a fair around there. It was like a small amusement park.

Monday, 8 August 2022
I asked for a leave as I had another job interview. It was scary this time. So, after the interview, I cooked myself hot ramyon to feel better about my life.

Wednesday, 10 August 2022
Today I presented another schema that I believe can be used for another data source. Wa helped me decide what to add more and what to remove. I was also working on my interview which will be done this Friday.

Friday, 12 August 2022
I had my final job interview today. After that, I continued working on documenting my work. This will my final task here at CERN. I hope I can finish it early because I still have to make a final report for CERN as well.

Saturday, 13 August 2022
I and the Thai friends went out have Korean Barbeque. I almost cried as it was the most delicious food I have had while staying in Geneva. But the price was not kind. I still have not gotten used to the food prices in Geneva.

Sunday, 14 August 2022
Today I and the other Thai friend went to Zurich. We did not plan anything. We just got on the train and walked around the city. We also went the zoo there. We also went boat pedaling in the lake. It was unexpected but I enjoyed today so much.

Monday, 15 August 2022
I worked on the small details of my final work. My other Thai friends were also working hard. I sat in the library but at a different desk this time. I did not bring my hoodie and the desk was directly under the air conditioner flow. After finished working, I made sure to take some medicine because I do not want to catch a cold again.

Tuesday, 16 August 2022
There was not nothing interesting about work today. But I was scolded in the canteen R1 today. I believe the chef was offended when I did not know the dish name when ordering. I tried to stay positive about it but after coming back to the hostel, I felt a bit sad by the situation. I think living abroad needs a strong mindset. I will work on that as well.

Wednesday, 17 August 2022
I got a job offer today. So, I was working in the library with a smile on my face the whole day. I met one of the Thai friends in the office as well. We were all working on finishing the project as soon as possible. But we did not forget to always grab hot coffee and talk about the weather and life in general. I also had a chat with a technical student who works in the office close to me. He offered me ice cream too.

Thursday, 18 August 2022
Today was so cold that I woke up with a mild fever. I must have forgotten to wear socks when going to sleep. I took paracetamol then prepared for a Zoom meeting. Then in the afternoon I treated myself with hot ramyon and 2 boiled eggs.

Friday, 19 August 2022
I almost completed the work at CERN. There were some minor details I needed to document, some that slipped my mind. I tried to be as complete as possible. The weather was even colder, I really hope I can stay healthy throughout the entire stay here. It is really important to be prepared for any kind of weather.

Saturday, 20 August 2022
I did not do anything much. I cooked some fried chicken and Thai rice. And walk around the city in the afternoon.

Sunday, 21 August 2022
I woke a bit late to enjoy the nice weather. Today I really wanted to have Korean barbeque, but I had to stop myself as it is really expensive here in Geneva. I found a place that looks like the food will be heavenly, but the price stops me. So, I just cooked myself some easy food.

Monday, 22 August 2022
I finished my work already and started working on the final report to submit to CERN. I was at the library the whole day. I love the peaceful environment where everyone just read their physics books or code something. Even though nobody talks to each other, I still feel encouraged by their presence.

Tuesday, 23 August 2022
The CRAB team meeting was canceled so I just walked around CERN and wrote my report. Most of the Summer students already left CERN. I and the other 2 Thai friends were the first group to arrive and almost the last to leave. So, it was quite lonely currently. But I did not feel like eating alone was a burden. I would feel like that before coming to CERN, but after spending quite some time here, I think I have gotten more mature about life.

Thursday, 25 August 2022
Today I and the other Thai friend went out to James' Pub with 2 of my CRAB team members, Wa and Dario. Dario drove us to the place. It was the first time for me to sit in a car since I arrived here. It was a different experience because they drive the opposite way from Thailand. Then I learned that even though there was not heavy traffic jam like in Thailand, using the car is still slower than public transportation. I believe it is the way Geneva design the road to encourage people to use more public transportation.

Friday, 26 August 2022
This was my last day as CERN Summer student. I woke up early to return my badge, bicycle, and say goodbye to my team members. I gifted them Thai inhalers and wrote them some notes. I submitted the final report and felt like 3 months was not long ago. I walked around CERN for bit. I think this will be one of the best memories for me for sure. Then I and the other Thai friend went to travel outside of Geneva. We even went on a boat trip for 4 hours. I had so much fun.

Saturday, 27 August 2022
Today I packed my back for going back to Thailand. I was worried that the weight will exceed the limit but surprisingly it did not. I also cooked myself some Thai rice and fried pork. One friend from CERN also came by to gift me a parting gift. I was not sure whether I am ready to go back to Thailand.

Sunday, 28 August 2022
I traveled to the airport with one Thai friend. We woke up early and took the bus. We saw France view for one last time then said sad goodbyes to these beautiful memories. Then we flown back safely to Thailand.

Monday, 29 August 2022
Arrived at Thailand.

Chapter 4
Author's Biography

Name Nutchaya Phumekham (ณัฐชยา ภูมีคำ)

Address 18/244 19th floor, Q House Condo Sathorn, Krung Thonburi Road, Kwang Klong Tonsai, Khet Klongsan, Bangkok 10600

Education Bachelor's degree in Computer Engineering from Sirindhorn International Institute of Technology, Thammasat University

Previous Work and Experience

1. Some of the author's previous programming-related projects can be found on *https://github.com/nutty7fold.*
2. A publication on "Human Activity Recognition Using an IoT-based Posture Corrector and Machine Learning" for the 2022 International Conference on Business and Industrial Research. The online publication can be found on the following website: *https://ieeexplore.ieee.org/document/9786396.*
3. The author worked as a part-time backend software engineer at Agoda during May 2021 – April 2022.

Interests

1. Machine Learning and Artificial Intelligence
2. Backend and Frontend Software Engineer
3. Data Storage Management
4. Data Analysis and Visualization
5. Business Intelligence


CERN


ไทยเซิร์น


Siam Photon

สวทช
NSTDA

NARIT

สนุก
TINT

สสวท

THEP
CENTER

กฟผ.

POWER OF INNOVATION
FOUNDATION

IRPC