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.

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

RESULTS

Handy Analysis Functions

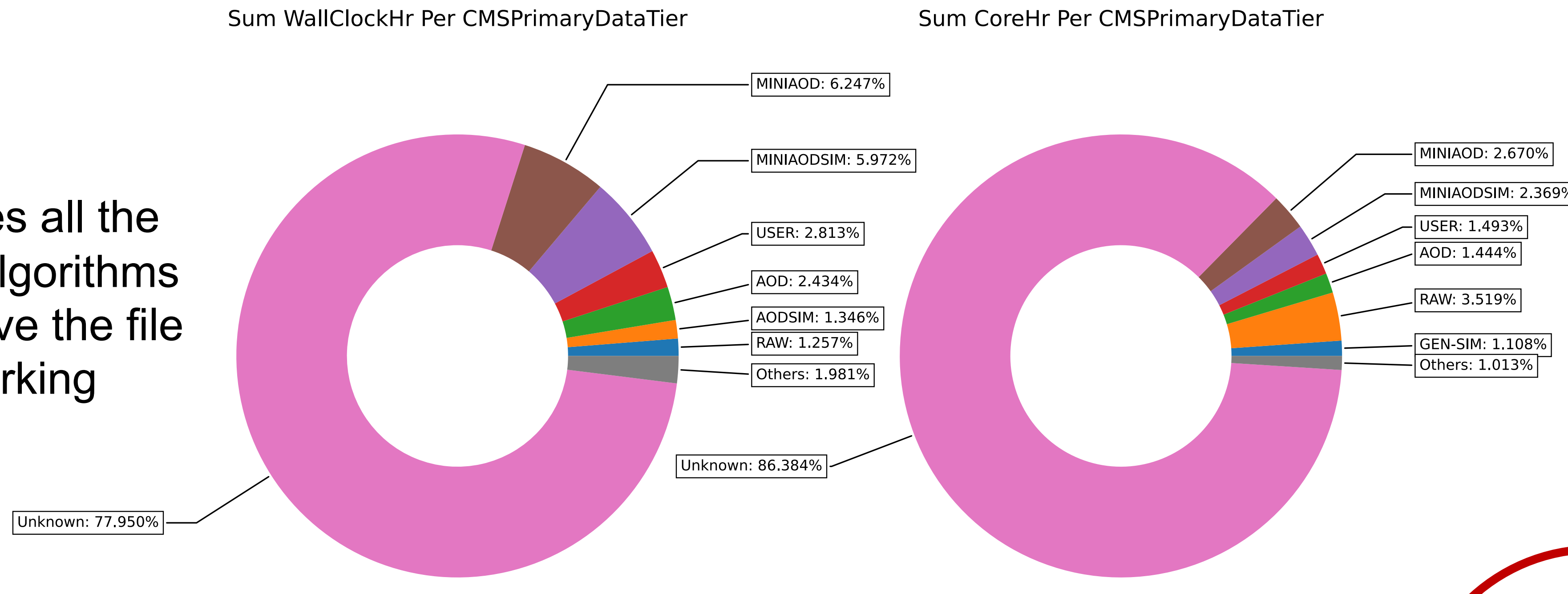
A simple "utils" Python code that does all the visualization and tedious repetitive algorithms for you. To use it, the user simply save the file in same directory as their current working directory.

```
├─ 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
```



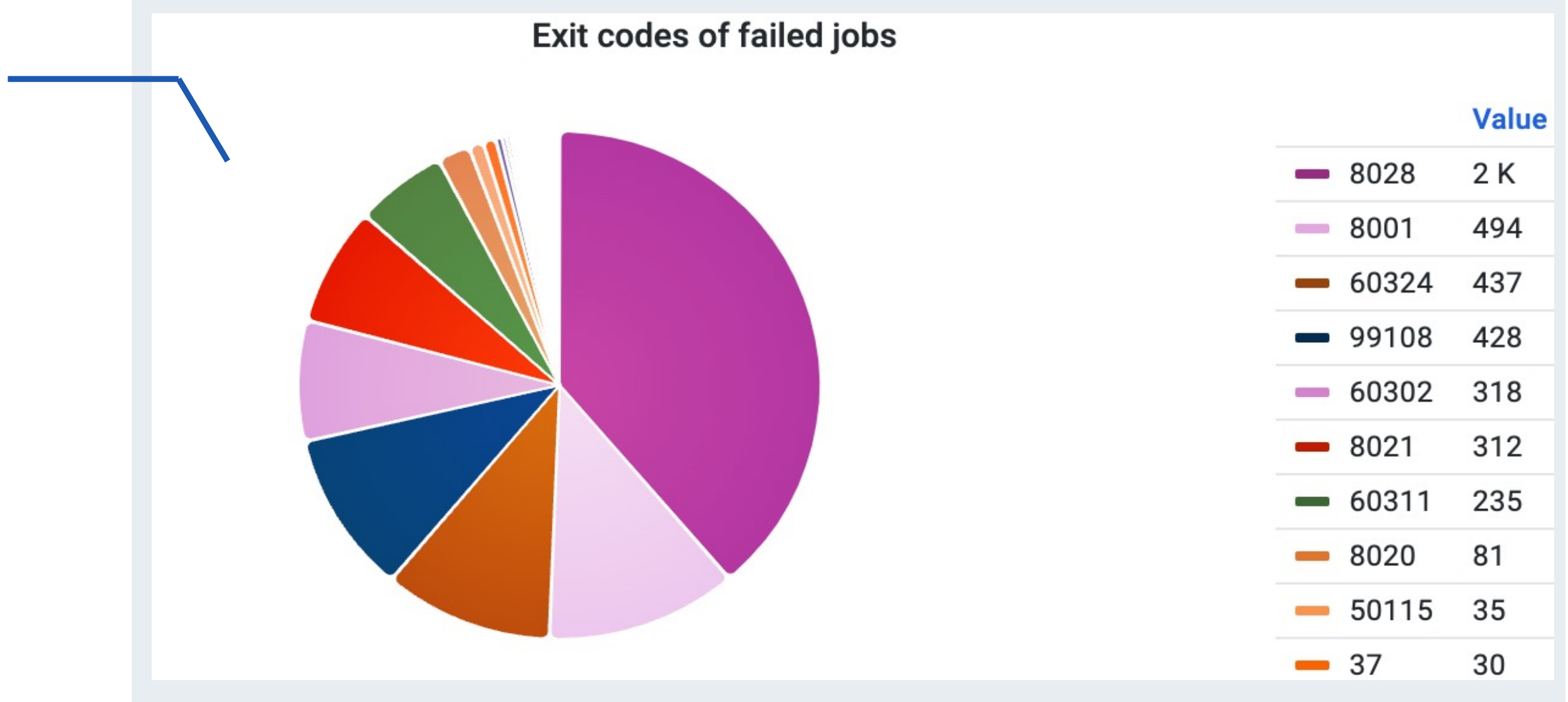
CRAB Analytics Guide and Examples

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.

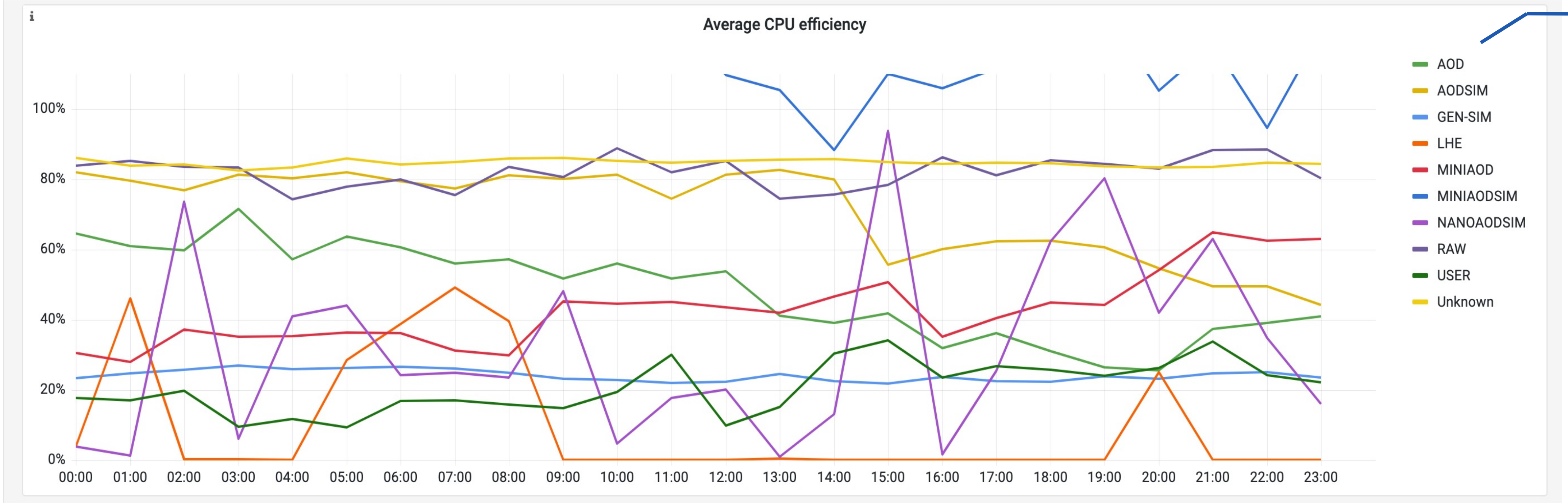
CRAB Monitoring Dashboard

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.

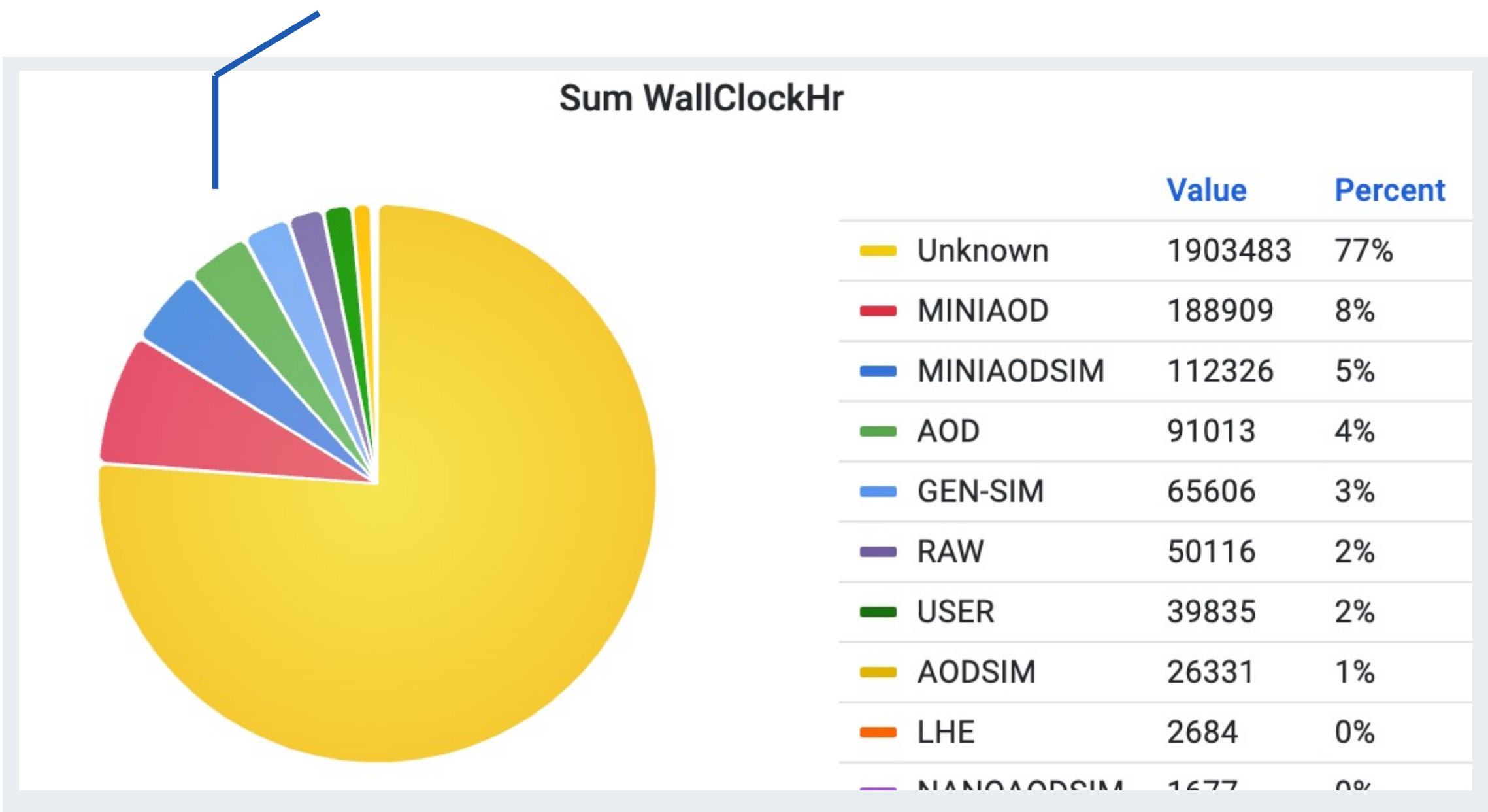
Monitoring exit code of the failed jobs to understand whether the failed jobs reflect human error or site error.



	Value
8028	2 K
8001	494
60324	437
99108	428
60302	318
8021	312
60311	235
8020	81
50115	35
37	30



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



	Value	Percent
Unknown	1903483	77%
MINIAOD	188909	8%
MINIAODSIM	112326	5%
AOD	91013	4%
GEN-SIM	65606	3%
RAW	50116	2%
USER	39835	2%
AODSIM	26331	1%
LHE	2684	0%
NANOAODSIM	1677	0%

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*