● Google Summer of Code Program links interesting open sourced problem statement to capable students.

● One of the most prestigious competitions in the world. Every project receives funding and global recognition.

● CERN-HSF is the open sourced umbrella organization participating in GSoC every year.
In 2019, CERN-HSF has 38 projects with 30 organizations. It is hosting about 33 interns out of ~1100 interns selected all over the world.

The complex problem statement of building a AWAKE analysis library was developed by my mentors Spencer Gessner and Marlene Turner.

Out of 7,555 proposals submitted across 103 countries, 45 proposals were submitted on the problem statement of AWAKE out of which my proposal was accepted.

As part of internship I’ll be working with Spencer and Marlene from roughly May-September to deploy a stable python based analysis tool for AWAKE.
PREVIOUS WORK

- Previously, pyTimber library and AWAKE Analysis Tools were used for performing data logging and analysis on data discovered by AWAKE.
- pyAWAKE would be a new library inspired from pyTimber and AWAKE_ANALYSIS_TOOLS to create a new database engine as well as provide tools for data analysis.

```python
import pytimber

# Setting up the database
ldb = pytimber.LoggingDB()

t1 = '2018-06-11 00:00:00.000'
t2 = '2018-06-12 00:00:00.000'
d = ldb.get('%AWAKE%', t1, t2)

print(d)
```
Replace this line with your local AWAKE_ANALYSIS_TOOLS directory:
```
import cutParser
from bunchRotationStatus import bunchRotationStatus
import returnGoodData as rgd
from returnPlasmaDensity import returnPlasmaDensity
import frame_analysis as fa
from custom_cmap import custom_cmap
```

Here we specify the data we want to load with an external file. Alternatively, we can generate this file from within the notebook.
```
input_file = 'input_delay_scan.txt'
input_path = os.environ['AAT']+'/examples/'
InputParsed = cutParser.inputParser(input_path+input_file)
file_list, file_bool = InputParsed()
```

```
print(file_list)
```
OBJECTIVES

- To read about 12TB of data created during 2017-18 experimental run and create a faster and simpler AWAKE database.
- To supply API for Indexing as well as Searching images or datasets.
- Provide capability to visualize one or many datasets.
- Using NumPy, SciPy and Matplotlib for data analysis and visualization.
- Porting existing analysis done by Awake Analysis tools.
- Encapsulate in a library structure and provide example notebooks.
AWAKE DATABASE STRUCTURE

12 TB HDF FILES

AWAKE DATABASE

220K 2016

200K 2017

120K 2018

25K each 01-12

2K HDF FILE each 01-31

2K CSV FILE each

HDF to CSV Mapping is one-to-one

YEAR

MONTH

DAY
INDEXING PROCESS

- In first attempt, we used a multi-threaded python program to create CSVs.
- For 12TB data, it would have taken 30 days to index which was not feasible.
- In second attempt, we used CERN-IT team’s help in deploying the program on SPARK technology under SWAN service.
- With the help of head of SPARK division Mr. Prasanth Kothuri and Mr. Piotr Mrowczynski, we ported the code as well as fine tuned the SPARK to accommodate the program.
- Using utmost 256 nodes, The whole process now runs in 3 hours ideally.
- The indexing for year 2017-18 is done and testing has begun.
SEARCHING PROCESS AND VISUALIZING

1. Enter Dataset value and/or comment value with Timestamp Range
   - <1 sec
   - Looks for substrings in caches and loads all related datasets

2. User selects dataset and loads all CSV with dataset value in timestamp
   - 25 ms/file

3. User gives range of datasets to load
   - 2 sec/file

4. Datasets loaded can be visualized in movies and graph
   - <1 sec
SEARCHING PROCESS AND VISUALIZING

dataset = search_ptr.search(dir_home, dir_csv_filepath, dir_csv_central_filepath, "streak data xmpp", ",", "2017-06-06 21:00:00", "2017-06-06 23:21:46")
visualize_ptr.displayMovie(dataset)

0. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageTime
1. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageHeight
2. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageProjection
3. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageData
4. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageTimeValues
5. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageTrigLevel
6. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageInfo
7. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageMcpGain
8. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageExposureTime
9. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageTimeRange
10. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageWidth
11. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageBinning
12. /AwakeEventData/XMPP-STREAK/StreakImage/streakImageMode
13. /AwakeEventData/XMPP-STREAK/StreakImage

Enter which dataset to load (0,14)

Checking if 186 files have dataset value and are in timestamp range
0. 2017-06-06 21:00:24+02:00 1496775624135000000_53_461.csv
1. 2017-06-06 21:01:07+02:00 1496775667335000000_53_462.csv
2. 2017-06-06 21:01:50+02:00 1496775710535000000_53_463.csv
3. 2017-06-06 21:02:33+02:00 1496775753235000000_53_464.csv
STREAK DATASET VISUALIZED USING AWAKE ANALYSIS TOOLS AND pyAWAKE
TABLE OF CONTENTS

1. Origin Story
2. Google Summer of Code
3. Building A Database
4. Building A Database Engine And Visualizing Tools
5. Covariance Analysis and Future Scope
6. Experience and Important Links
Streak Covariance Analysis
THANK YOU