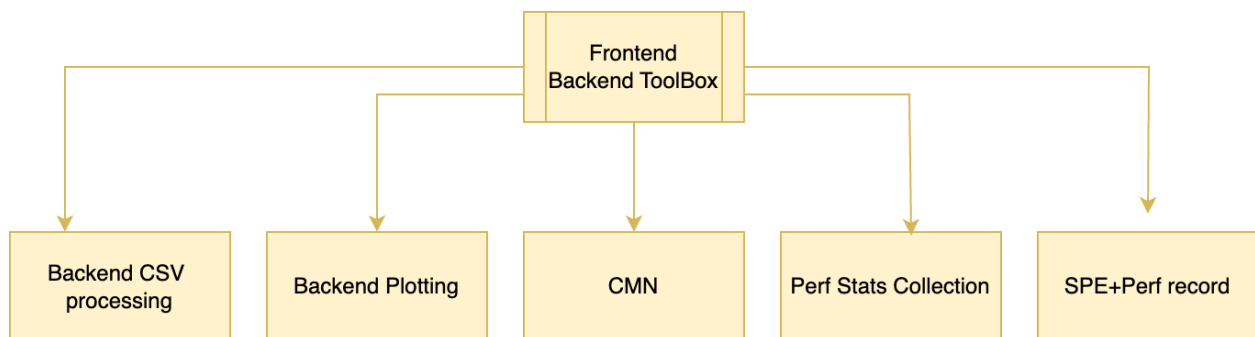


Backend Performance Plotting Tool – Perf stats & PMUv3

Table of Contents

FRONTEND AND BACKEND PERFORMANCE ANALYSIS TOOL FLOWCHART	1
SCOPE OF DOCUMENTATION	1
INSTALLATION & SETUP	1
USAGE FOR PMUV3_BACKEND	2
USAGE FOR PERF_STATS_BACKEND	6

Frontend and Backend Performance Analysis Tool Flowchart



Scope of Documentation

In this documentation, we will discuss steps/usage to run **Backend Plotting scripts** which help in generating performance comparison bar graphs for better analysis of PMU events and Metrics.

Installation & Setup

git clone https://github.com/GayathriNarayana19/Performance_Analysis_Backend

Request access if not given already to gayathrinarayana.yegnanrayanan@arm.com

Note:

- ❑ **Please install python dependencies/libraries like matplotlib, pandas etc.** For example, if you are using Pandas 2.0.3, I recommend upgrading Matplotlib to at least version 3.4.0 or higher as a lesser version would not be compatible with Pandas 2.0.3.
- ❑ `pip install <missing_package>` or `pip3 install <missing_package>` depending on your pip version. Eg: (`pip install PyPDF2` is required for viewing the final report in PDF format and `pip install pillow` is required too)
- ❑ This tool was tested and works clean on Python 3.8.10. You can either stick to the same version or resolve dependency or package issues should they arise.

Usage for PMUV3_Backend

Go to Path: `cd (YOUR_HOME_DIR)/Performance_Analysis_Backend/`

```
→ Performance_Analysis_Backend git:(main) ls
LICENSE                PMUv3_Backend          Perf_Stats_Backend
```

As you can see in the above snapshot, you have two subdirectories

PMUv3_Backend usage

- This can be used for generating the **Comparison graphs between different cores** or a general comparison between different CSVs you provide as input.
- **EXAMPLE :** The comparison can be between anything. For example, one can have three scenarios comparing N1, G2, G3 performance. Each processor N1, G2, G3 would have 15 CSVs since the PMUv3_plugin has 15 groupings/bundles of PMU events. Now the intent is to compare the performance numbers of every PMU event in all the 15 bundles between N1, G2 and G3. Through this one can see how performance dips or varies across generations of processors and find hotspots.

STEP 1:

- Open **config.yaml** file. Replace the **existing paths**, and **output filename** according to your scenario and context.
- **output_dir** is the key where you give the path to store the graph outputs. Note: You can give any path and the directory gets created on its own.
- **Scenario** represents the legend or the attributes you are comparing. (In above example N1,G2,G3 is the scenario and **Context** is usually the title of the graph.

```

base_dirs:
  - path: '/home/ubuntu/backend_plot_test_csvs/N1/'
    output_file: 'n1_du_metrics.csv'
  - path: '/home/ubuntu/backend_plot_test_csvs/G2/'
    output_file: 'g2_du_metrics.csv'
  - path: '/home/ubuntu/backend_plot_test_csvs/G3/'
    output_file: 'g3_du_metrics.csv'
output_dir: '/home/ubuntu/test_plotting/'
base_filename: 'bundle{}.csv'
num_bundles: 15
scenarios:
  - 'N1: 3GHz'
  - 'G2: 2.5GHz'
  - 'G3: 2.6GHz'
context: 'DUHIGH'

#####DO NOT MODIFY BELOW THIS LINE#####
kpi_metrics:

```

STEP 2: After editing the yaml file, run the python script.

```
python3 pmuv3_plotting.py -config config.yaml
```

Usage pops up when you execute the script once in the traditional way as

`python3 pmuv3_plotting.py` Refer snapshot below and you can use that to run the above command.

```

root@altra1p-sm-gpu-02:/home/ubuntu# python3 pmuv3_plotting.py
error: the following arguments are required: -config
usage: pmuv3_plotting.py [-h] -config CONFIG

Process CSV files and generate plots.

optional arguments:
  -h, --help            show this help message and exit
  -config CONFIG        Path to the YAML configuration file. Example: -config config.yaml

Examples:
  pmuv3_plotting.py -config config.yaml

Examples:
  pmuv3_plotting.py -config config.yaml

```

Screenshot of executing the python script and expected terminal output.

```
root@altraip-sm-gpu-02:/home/ubuntu# python3 pmuv3_plotting.py -config config.yaml
pmuv3_plotting.py:293: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 100
pmuv3_plotting.py:297: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 1000
pmuv3_plotting.py:300: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce')
pmuv3_plotting.py:293: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 100
pmuv3_plotting.py:128: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 100
pmuv3_plotting.py:214: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

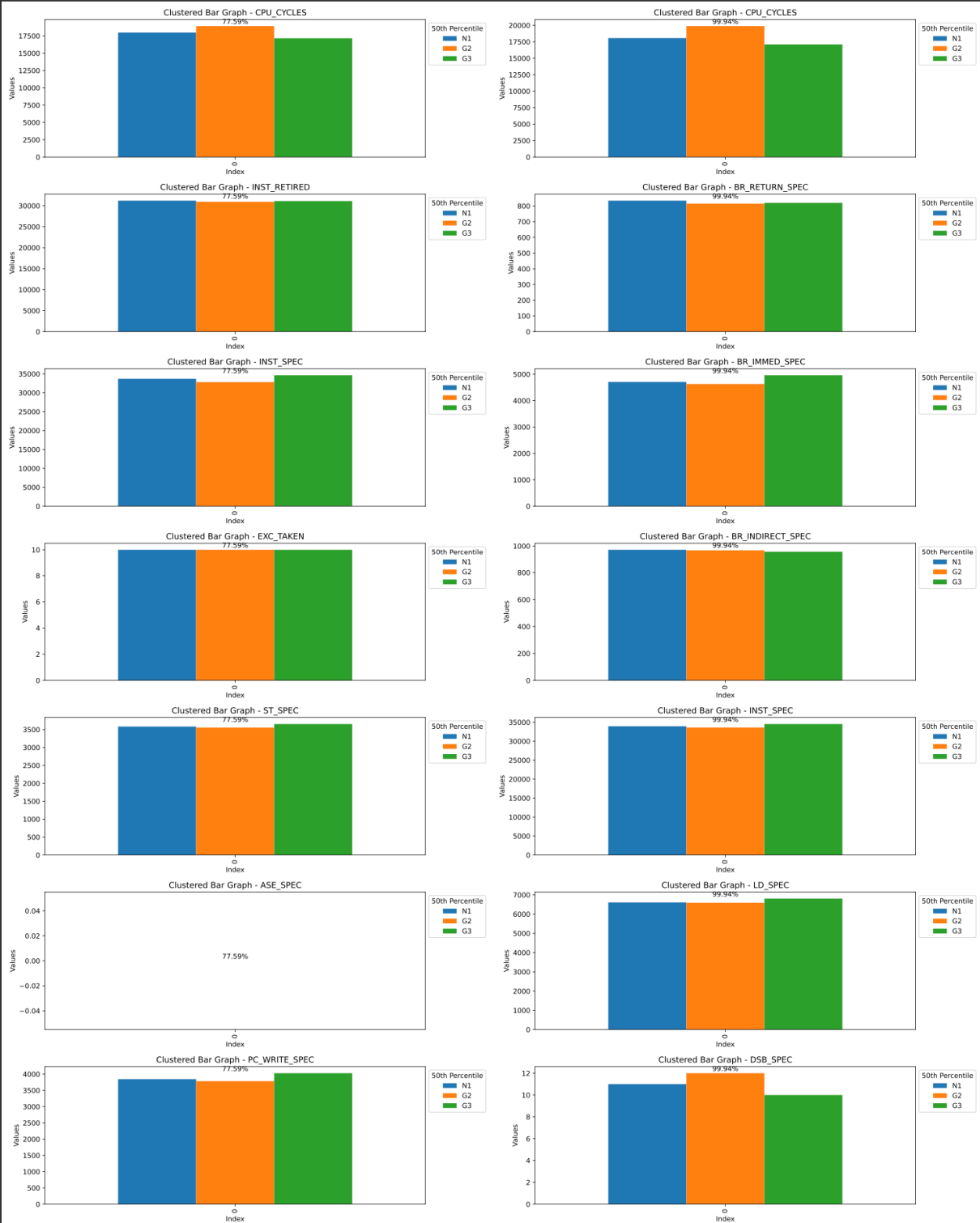
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 100
```

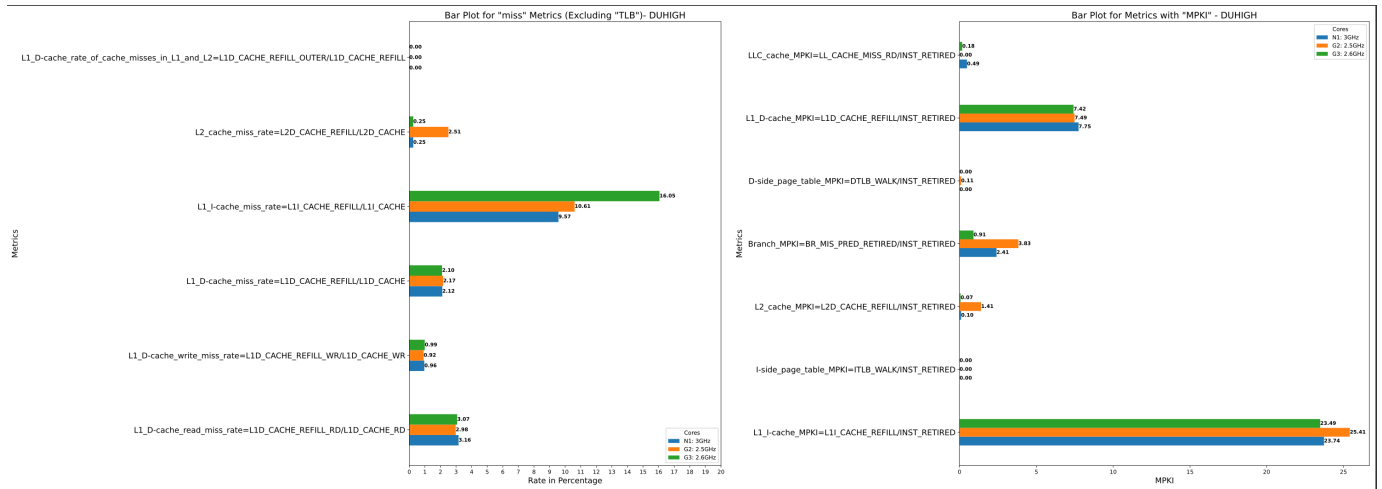
As you can see, in the output directory /home/ubuntu/plotting, there is a merged_output.pdf which will give a comprehensive report of all the KPI, PMU events and metrics in the form of different bundles.

```
root@altraip-sm-gpu-02:/home/ubuntu# cd plotting/
root@altraip-sm-gpu-02:/home/ubuntu/plotting# ls
1m.pdf  2t.pdf  3.pdf  bundle0.pdf  bundle11.pdf  bundle13.pdf  bundle1.pdf  bundle3.pdf  bundle5.pdf  bundle7.pdf  bundle9.pdf  g2_du_metrics.csv  n1_du_metrics.csv
1.pdf   2t.pdf  4.pdf  bundle10.pdf  bundle12.pdf  bundle14.pdf  bundle2.pdf  bundle4.pdf  bundle6.pdf  bundle8.pdf  g2_du_metrics.csv  merged_output.pdf
root@altraip-sm-gpu-02:/home/ubuntu/plotting# ls -lrth
total 912K
-rw-r--r-- 1 root root 19K Jun 21 17:43 bundle0.pdf
-rw-r--r-- 1 root root 19K Jun 21 17:43 bundle1.pdf
-rw-r--r-- 1 root root 18K Jun 21 17:43 bundle2.pdf
-rw-r--r-- 1 root root 28K Jun 21 17:43 bundle3.pdf
-rw-r--r-- 1 root root 18K Jun 21 17:43 bundle4.pdf
-rw-r--r-- 1 root root 18K Jun 21 17:43 bundle5.pdf
-rw-r--r-- 1 root root 20K Jun 21 17:43 bundle6.pdf
-rw-r--r-- 1 root root 19K Jun 21 17:43 bundle7.pdf
-rw-r--r-- 1 root root 21K Jun 21 17:43 bundle8.pdf
-rw-r--r-- 1 root root 20K Jun 21 17:43 bundle9.pdf
-rw-r--r-- 1 root root 19K Jun 21 17:43 bundle10.pdf
-rw-r--r-- 1 root root 21K Jun 21 17:43 bundle11.pdf
-rw-r--r-- 1 root root 20K Jun 21 17:43 bundle12.pdf
-rw-r--r-- 1 root root 21K Jun 21 17:43 bundle13.pdf
-rw-r--r-- 1 root root 20K Jun 21 17:43 bundle14.pdf
-rw-r--r-- 1 root root 5.0K Jun 21 17:43 n1_du_metrics.csv
-rw-r--r-- 1 root root 5.1K Jun 21 17:43 g2_du_metrics.csv
-rw-r--r-- 1 root root 5.0K Jun 21 17:43 g3_du_metrics.csv
-rw-r--r-- 1 root root 28K Jun 21 17:43 1.pdf
-rw-r--r-- 1 root root 24K Jun 21 17:43 2.pdf
-rw-r--r-- 1 root root 22K Jun 21 17:43 3.pdf
-rw-r--r-- 1 root root 23K Jun 21 17:43 4.pdf
-rw-r--r-- 1 root root 26K Jun 21 17:43 1m.pdf
-rw-r--r-- 1 root root 23K Jun 21 17:43 2t.pdf
-rw-r--r-- 1 root root 421K Jun 21 17:43 merged_output.pdf
```

OUTPUTS

Only some Sample outputs are displayed below. However, this backend tool captures about 15 groups of PMU events ie, covering approximately 70 events and 38 KPIs all at one shot and your final merged_output.pdf will have around 21 pages.





Usage for Perf_stats_Backend

Usage:

When you execute the script as

`python3 plotting_perf_stat.py`, the tool instructs you to give the correct command with usage

```
usage: plotting_perf_stat.py --csv /path/to/csv/CSV1 /path/to/csv/CSV2 -o /dir/path_for_output_plots/ -s Scenario_for_CSV1 Scenario_for_CSV2 -c CSV1_CSV2_COMPARISON
Enter 'python3 plotting_perf_stat.py -h' to know the description for arguments -o, -s and -c
plotting_perf_stat.py: error: Please provide the path to the CSV file using --csv_file or --csv
```

Example: Correct usage is mentioned below. Once correct arguments are passed, in the output directory that you mention, the plots get created and save into a PDF automatically.

```
root@altraip-hp-03:/home/ubuntu# python3 plotting_perf_stat.py --csv /home/ubuntu/core_Split/CPU1.csv /home/ubuntu/core_Split/CPU2.csv -o /home/ubuntu/result_plots -s
core1 core2 -c Comparison_core1_core2
/home/ubuntu/plotting_perf_stat.py:1: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

import pandas as pd
Namespace(csv_files=['/home/ubuntu/core_Split/CPU1.csv', '/home/ubuntu/core_Split/CPU2.csv'], op_dir='/home/ubuntu/result_plots', scenario=['core1', 'core2'], context
='Comparison_core1_core2')
/home/ubuntu/plotting_perf_stat.py:197: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 100
/home/ubuntu/plotting_perf_stat.py:388: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough to accommodate all axes decorati
ons.
plt.tight_layout()
/home/ubuntu/plotting_perf_stat.py:281: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

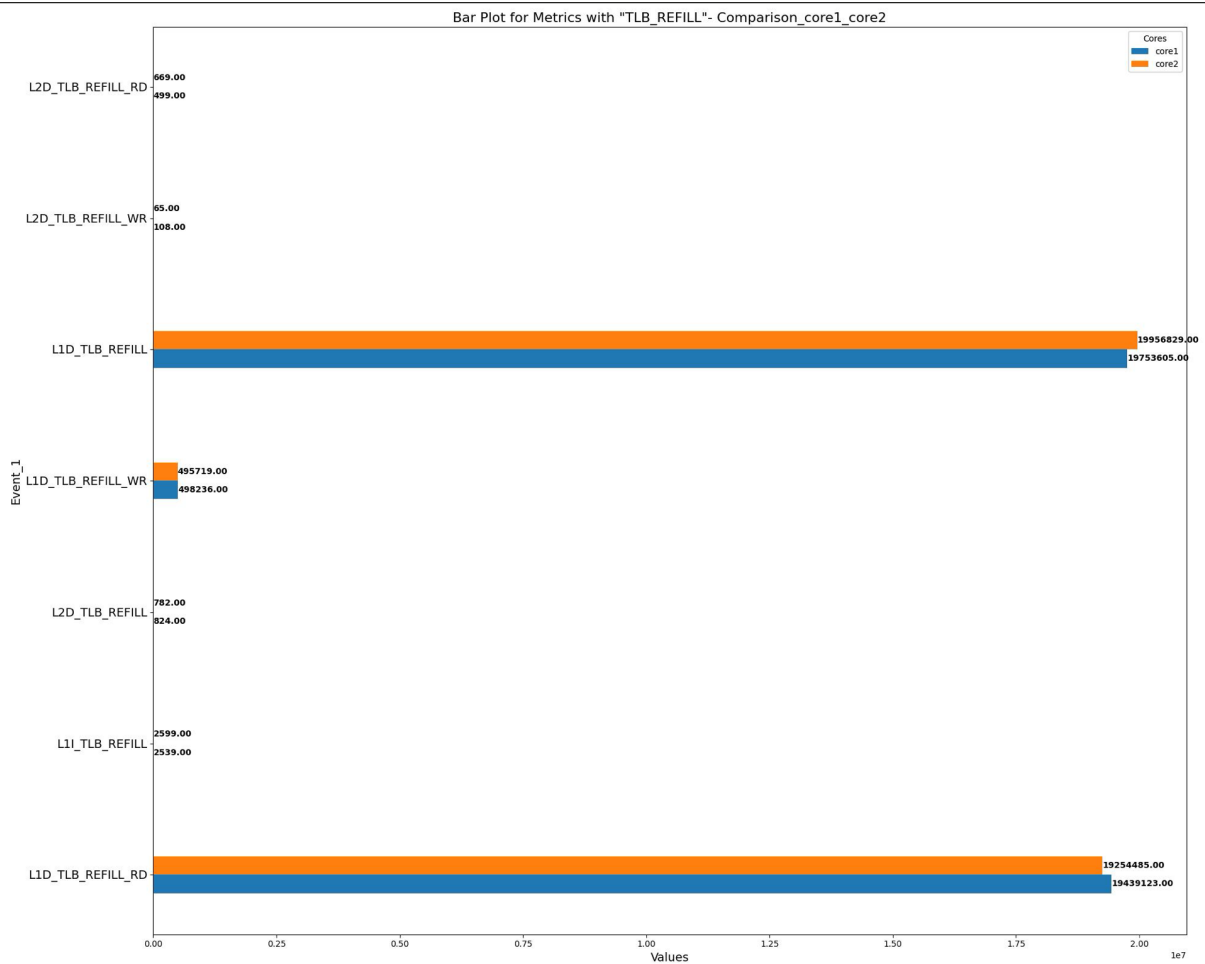
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df['Event_1/Event_2'] = pd.to_numeric(df['Event_1/Event_2'], errors='coerce') * 1000
/home/ubuntu/plotting_perf_stat.py:284: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

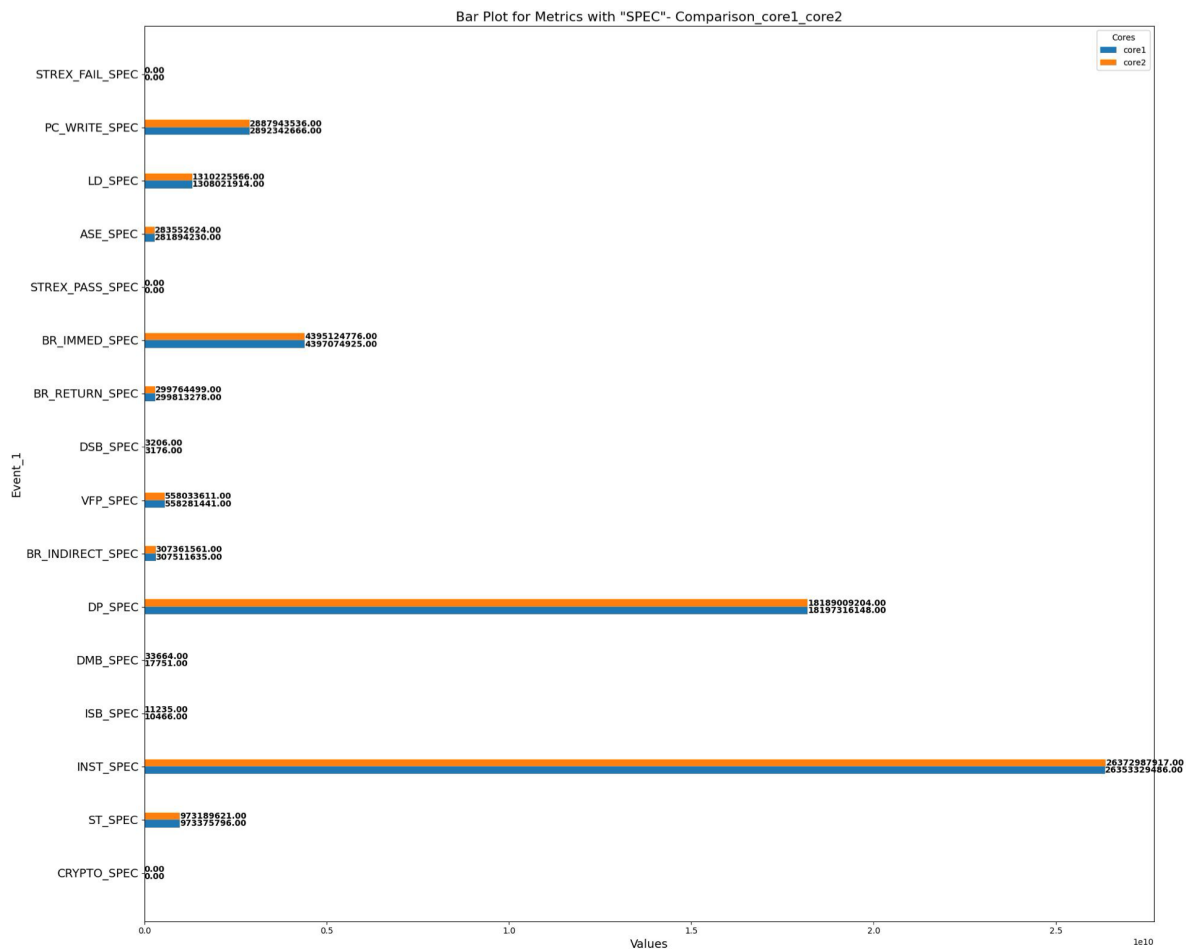
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
df.drop_duplicates(subset=['Name_1'], keep='first', inplace=True)
Successfully created the PDF '/home/ubuntu/result_plots/merged_output.pdf'
root@altraip-hp-03:/home/ubuntu# cd result_plots/
root@altraip-hp-03:/home/ubuntu/result_plots# ls -lrth
total 3.7M
-rw-r--r-- 1 root root 99K Jun 24 06:31 1.png
-rw-r--r-- 1 root root 90K Jun 24 06:31 2.png
-rw-r--r-- 1 root root 49K Jun 24 06:31 3.png
-rw-r--r-- 1 root root 95K Jun 24 06:31 4.png
-rw-r--r-- 1 root root 52K Jun 24 06:31 5.png
-rw-r--r-- 1 root root 52K Jun 24 06:31 6.png
-rw-r--r-- 1 root root 163K Jun 24 06:31 8.png
-rw-r--r-- 1 root root 70K Jun 24 06:31 9.png
-rw-r--r-- 1 root root 56K Jun 24 06:31 10.png
-rw-r--r-- 1 root root 119K Jun 24 06:31 1m.png
-rw-r--r-- 1 root root 88K Jun 24 06:31 1t.png
-rw-r--r-- 1 root root 75K Jun 24 06:31 11.png
-rw-r--r-- 1 root root 79K Jun 24 06:31 12.png
-rw-r--r-- 1 root root 52K Jun 24 06:31 13.png
-rw-r--r-- 1 root root 68K Jun 24 06:31 14.png
-rw-r--r-- 1 root root 142K Jun 24 06:31 15.png
-rw-r--r-- 1 root root 55K Jun 24 06:31 16.png
-rw-r--r-- 1 root root 45K Jun 24 06:31 17.png
-rw-r--r-- 1 root root 55K Jun 24 06:31 18.png
-rw-r--r-- 1 root root 53K Jun 24 06:31 19.png
-rw-r--r-- 1 root root 52K Jun 24 06:31 20.png
-rw-r--r-- 1 root root 2.1M Jun 24 06:31 merged_output.pdf

```

Merged_output.pdf will have the plots like below. (Attaching only some)





Note : Incase, the bars are touching the border meaning the values are higher than the current x axis limit, in the code, below lines can be adjusted accordingly in the relevant function.

```
ax.set_xlim(0, 40) # xlim for horizontal chart
ax.xaxis.set_major_locator(plt.MultipleLocator(2.5))
```

If the metrics or KPI formulae are cut off in the left corner of y axis, that can be fixed by modifying or adding appropriate lines similar to

```
if key_string == 'miss':
    fig, ax = plt.subplots(figsize=(20, 16))
```

Don't modify the exact line in the current code. This is only an example as to how to customize.