**Backend Performance Plotting Tool – Perf stats & PMUv3**

Table of Contents

[Frontend and Backend Performance Analysis Tool Flowchart 1](#_Toc169827647)

[Scope of Documentation 1](#_Toc169827648)

[Installation & Setup 1](#_Toc169827649)

[Usage 2](#_Toc169827650)

# **Frontend and Backend Performance Analysis Tool Flowchart**

A diagram of a computer process

Description automatically generated

# **Scope of Documentation**

In this documentation, we will discuss steps/usage to run Backend Plotting scripts which help in generating performance comparsion 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 usingPandas 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.
* 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

Go to Path: cd (YOUR\_HOME\_DIR)/Performance\_Analysis\_Backend/



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

1. PMUv3\_Backend

* 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.  
    
  USAGE:
* 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.

A screen shot of a computer

Description automatically generated

After editing the yaml file, run the python script.   
  
Usage pops up when you execute the script once in the traditional way.

(Documentation WIP..!)