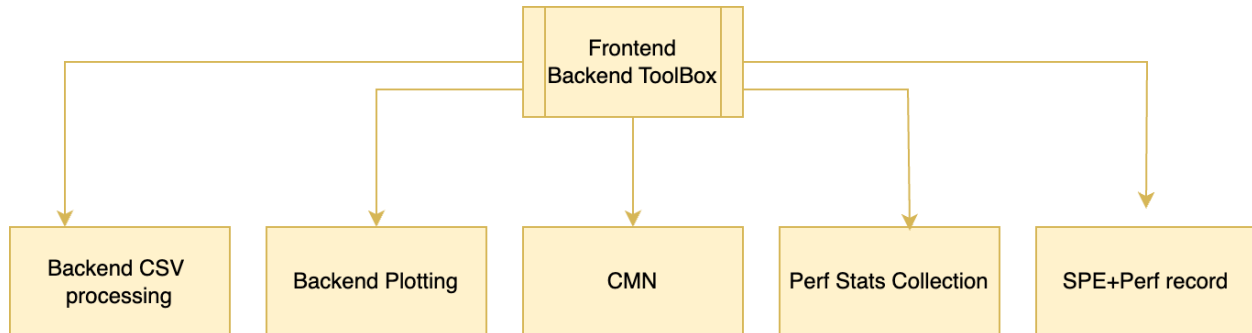


# Frontend and Backend Tool – Perf stats

## Table of Contents

FRONTEND AND BACKEND PERFORMANCE ANALYSIS TOOL FLOWCHART .....	1
SCOPE OF DOCUMENTATION .....	1
FRONTEND_BACKEND_TOOL INSTALLATION .....	1
PERF INSTALLATION .....	2
USAGE FOR FRONTEND PMU PERFORMANCE SCRIPT .....	2
USAGE FOR BACKEND SCRIPT - CSV GENERATION .....	3
CSV_GENERATION.PY .....	3
CSV_SPLIT.PY .....	4

## Frontend and Backend Performance Analysis Tool Flowchart



## Scope of Documentation

In this documentation, we will discuss steps/usage to run **frontend and Backend scripts** related to **perf stats event collection**. **Note: Please use Perf version 5.17 or higher** to avoid any register mapping errors.

Please find Perf Installation steps/references below if not done already.

## Frontend Backend Tool Installation

git clone [https://github.com/GayathriNarayana19/Frontend\\_Backend\\_Performance\\_Tool](https://github.com/GayathriNarayana19/Frontend_Backend_Performance_Tool)

Request access if not given already to [gayathrinarayana.yegnanrayanan@arm.com](mailto:gayathrinarayana.yegnanrayanan@arm.com)

## Perf Installation

<https://medium.com/@manas.marwah/building-perf-tool-fc838f084f71>

- In case of missing dependencies and error, use **apt-cache** and install relevant packages and resolve dependencies.
- Set this line in Makefile.config of linux/tools/perf `NO_LIBTRACEEVENT=1` and do a make
- Set environment variable.

```
export PATH="/home/ubuntu/linux/tools/perf:$PATH"
```

## Usage for Frontend PMU Performance script

Go to Path: `cd`

`(YOUR_HOME_DIR)/Frontend_Backend_Performance_Tool/Perf_Stats/`

The **ampere\_pmu\_parallel.sh** script has the ability to run parallelly on multiple cores saving time.

Usage pops up when you execute the script once in the traditional way. Here the `output_directory` is an argument where you store your results. The argument `cores` is where you would mention the range of cores you would like to monitor and `sleep` is for the sleep interval in which you would like to gather performance data.

Below snippet has different usage examples that you can refer.

```
root@arm:/home/ubuntu/Frontend_Backend_Performance_Tool/Perf_Stats#  
./ampere_pmu_parallel.sh
```

Usage: ./ampere\_pmu\_parallel.sh <output\_directory> <cores> <sleep>

Example:

For many cores, run like below

```
./ampere_pmu_parallel.sh /home/ubuntu/test_collect "37,33" 10
```

For single core, refer below usage

```
./ampere_pmu_parallel.sh /home/ubuntu/test_collect "49" 10
```

For **multiple consecutive cores**,

```
./ampere_pmu_parallel.sh /home/ubuntu/test_collect "30-41" 10
```

## **Usage for Backend Script - CSV Generation**

Go to Path:

```
cd (YOUR_HOME_DIR)/Frontend_Backend_Tool/Backend_CSV_processing
```

You will have two files -

1. csv\_generation.py
2. csv\_split.py

### **csv\_generation.py**

When you execute "python3 csv\_generation.py" usage pops up as follows. Follow the usage.

```
Usage: python3 csv_generation.py <directory_path_where_logs_are_present>  
<output_file_name.csv>
```

```
python3 csv_generation.py home/ubuntu/pmu_logs/ output_log.csv
```

In the second parameter, specify the directory path you mentioned while running the frontend script for storing the output results. This python script executes upon the directory containing the log files generated from the frontend script and creates a CSV file. You can name the CSV file name by mentioning it in the 3rd argument!

The output CSV contains the below fields and this can be used to analyse data using the graphing backend script. It contains performance data of all cores, events, KPIs along with the formula.

Core,Metrics,Name\_1,Event\_1,Name\_2,Event\_2,Event\_1/Event\_2,Graph\_Xlabel

```
1 Core,Metrics,Name_1,Event_1,Name_2,Event_2,Event_1/Event_2,Graph_Xlabel
2 CPU28,Front_end_stall_rate,STALL_FRONTEND,2749602401.0,CPU_CYCLES,23899858802.0,0.11584680524597519,Front_end_stall_rate=STALL_FRONTEND/CPU_CYCLES
3 CPU28,CRYPTO_instruction_rate_over_time,CRYPTO_SPEC,0.0,CPU_CYCLES,23936944014.0,0.0,CRYPTO_instruction_rate_over_time=CRYPTO_SPEC/CPU_CYCLES
4 CPU28,CRYPTO_instruction_rate_per_instructions,CRYPTO_SPEC,0.0,INST_SPEC,11470448490.0,0.0,CRYPTO_instruction_rate_per_instructions=CRYPTO_SPEC/INST_SPEC
5 CPU28,L1_D-cache_read_rate,L1D_CACHE_RD,2888831824.0,L1D_CACHE,4317126145.0,0.6691562226750731,L1_D-cache_read_rate=L1D_CACHE_RD/L1D_CACHE
6 CPU28,L1_data_TLB_misses_per_context_swap,L1D_TLB_REFILL,150877440.0,TTBR_WRITE_RETIRED,24.0,6286560.0,L1_data_TLB_misses_per_context_swap=L1D_TLB_REFILL/TTBR_WRITE_RETIRED
7 CPU28,SIMD_instruction_rate_per_instructions,ASE_SPEC,2613882.0,INST_SPEC,11513395699.0,0.00022782963298890436,SIMD_instruction_rate_per_instructions=ASE_SPEC/INST_SPEC
8 CPU28,BR_RETURN_instruction_rate_per_instructions,BR_RETURN_SPEC,252994420.0,INST_SPEC,11501578635.0,0.021996495266321327,BR_RETURN_instruction_rate_per_instructions=BR_RETURN_SPEC/INST_SPEC
9 CPU28,DMB_rate_per_instructions,DMB_SPEC,29843184.0,INST_SPEC,11598951151.0,0.0025729209142696563,DMB_rate_per_instructions=DMB_SPEC/INST_SPEC
10 CPU28,DSB_rate_over_time,DSB_SPEC,363315.0,CPU_CYCLES,24072763271.0,1.5092367914309143e-05,DSB_rate_over_time=DSB_SPEC/CPU_CYCLES
11 CPU28,Branch_misprediction_rate,BR_MIS_PRED_RETIRED,12983600.0,BR_RETIRED,1953980325.0,0.006644693313378168,Branch_misprediction_rate=BR_MIS_PRED_RETIRED/BR_RETIRED
12 CPU28,BR_INDIRECT_instruction_rate_over_time,BR_INDIRECT_SPEC,307325597.0,CPU_CYCLES,24123184705.0,0.012739843464213113,BR_INDIRECT_instruction_rate_over_time=BR_INDIRECT_SPEC/CPU_CYCLES
13 CPU28,BR_RETURN_instruction_rate_over_time,BR_RETURN_SPEC,252954155.0,CPU_CYCLES,23868774630.0,0.010597701763963574,BR_RETURN_instruction_rate_over_time=BR_RETURN_SPEC/CPU_CYCLES
14 CPU28,L1_D-cache_miss_rate,L1D_CACHE_REFILL,199423606.0,L1D_CACHE,4325812239.0,0.04610937379592465,L1_D-cache_miss_rate=L1D_CACHE_REFILL/L1D_CACHE
15 CPU28,MEM_access_write_rate,MEM_ACCESS_WR,1432709675.0,MEM_ACCESS,4320036450.0,0.331642959864378,MEM_access_write_rate=MEM_ACCESS_WR/MEM_ACCESS
16 CPU28,LD_instruction_rate_per_instructions,LD_SPEC,2955159344.0,INST_SPEC,11608351999.0,0.25457182417061197,LD_instruction_rate_per_instructions=LD_SPEC/INST_SPEC
17 CPU28,L1_D-cache_write_miss_rate,L1D_CACHE_REFILL_WR,82963096.0,L1D_CACHE_WR,1425002826.0,0.058219601032566654,L1_D-cache_write_miss_rate=L1D_CACHE_REFILL_WR/L1D_CACHE_REFILL
18 CPU28,L2_cache_miss_rate,L2D_CACHE_REFILL,219825440.0,L2D_CACHE,608476042.0,0.3612721369890846,L2_cache_miss_rate=L2D_CACHE_REFILL/L2D_CACHE
19 CPU28,D-side_page_table_MPKI,DTLB_WALK,47558785.0,INST_RETIRED,10743205203.0,0.0044268711340186805,D-side_page_table_MPKI=DTLB_WALK/INST_RETIRED
20 CPU28,VFP_instruction_rate_per_instructions,VFP_SPEC,4271945.0,INST_SPEC,11513835141.0,0.00037102711196444776,VFP_instruction_rate_per_instructions=VFP_SPEC/INST_SPEC
21 CPU28,L2_TLB_misses_per_context_swap,L2D_TLB_REFILL,52746502.0,TTBR_WRITE_RETIRED,8.0,6593312.75,L2_TLB_misses_per_context_swap=L2D_TLB_REFILL/TTBR_WRITE_RETIRED
22 CPU28,PC_WRITE_instruction_rate_over_time,PC_WRITE_SPEC,1250832590.0,CPU_CYCLES,23983194244.0,0.05215454527342317,PC_WRITE_instruction_rate_over_time=PC_WRITE_SPEC/CPU_CYCLES
23 CPU28,L1_data_TLB_miss_rate,L1D_TLB_REFILL,150281558.0,L1D_TLB,7070642356.0,0.02125430059016833,L1_data_TLB_miss_rate=L1D_TLB_REFILL/L1D_TLB
24 CPU28,L1_D-cache_rate_of_cache_misses_in_L1_and_L2,L1D_CACHE_REFILL_OUTER,74102021.0,L1D_CACHE_REFILL,199963825.0,0.3705771331389565,L1_D-cache_rate_of_cache_misses_in_L1_and_L2=L1D_CACHE_REFILL_OUTER/L1D_CACHE_REFILL
25 CPU28,Branch_MPKI,BR_MIS_PRED_RETIRED,13164675.0,INST_RETIRED,10794968847.0,0.0012195194989986988,Branch_MPKI=BR_MIS_PRED_RETIRED/INST_RETIRED
26 CPU28,LLC_cache_miss_rate,LL_CACHE_MISS_RD,291866778.0,LL_CACHE_RD,299126952.0,0.9757287868864455,LLC_cache_miss_rate=LL_CACHE_MISS_RD/LL_CACHE_RD
27 CPU28,L2_cache_write_rate,L2D_CACHE_WR,237417311.0,L2D_CACHE,615618969.0,0.3856562629732743,L2_cache_write_rate=L2D_CACHE_WR/L2D_CACHE
28 CPU28,MEM_access_read_rate,MEM_ACCESS_RD,2899416105.0,MEM_ACCESS,4339557611.0,0.6681363320653931,MEM_access_read_rate=MEM_ACCESS_RD/MEM_ACCESS
29 CPU28,L1_D-cache_write_rate,L1D_CACHE_WR,1430270066.0,L1D_CACHE,4323851648.0,0.3307861098013324,L1_D-cache_write_rate=L1D_CACHE_WR/L1D_CACHE
30 CPU28,ISB_rate_per_instructions,ISB_SPEC,557988.0,INST_SPEC,11584072682.0,4.850351831252451e-05,ISB_rate_per_instructions=ISB_SPEC/INST_SPEC
31 CPU28,Successful exclusive store access rate,STREX_PASS_SPEC,0.0,STREX_SPEC,0.0,NAN,Successful exclusive store access rate=STREX_PASS_SPEC/STREX_SPEC
```

## csv\_split.py

This script splits the CSV based on cores. This helps us to analyze cores individually.

```
Usage: python3 csv_split.py -csv <csv_file> -dir_name_for_csvs
<output_directory>
```

```
python3 csv_split.py -csv output_log.csv -dir_name_for_csvs
/home/ubuntu/split_cores
```

As you can see, if core 1 and core 2 were mentioned as cores while generating the CSV, output\_log.csv would contain all the core data together and the csv\_split.py would split the data based on cores as below.

```
[root@altra1p-hp-03:/home/ubuntu/split_cores# ls
CPU1.csv  CPU2.csv
```

```
vim CPU1.csv
```

```
1 Metrics,Name_1,Event_1,Name_2,Event_2,Event_1/Event_2,Graph_XLabel
2 L1_data_TLB_read_miss_rate,L1D_TLB_REFILL_RD,19439123.0,L1D_TLB_RD,1279698586.0,0.015190391872481135,L1_data_TLB_read_miss_rate=L1D_TLB_REFILL_RD/L1D_TLB_RD
3 L1_D-cache_read_rate,L1D_CACHE_RD,1257543375.0,L1D_CACHE,2213933852.0,0.5680130749452943,L1_D-cache_read_rate=L1D_CACHE_RD/L1D_CACHE
4 LLC_cache_MPKI,LL_CACHE_MISS_RD,1027690.0,INST_RETIRED,24507340761.0,4.1933966235758416e-05,LLC_cache_MPKI=LL_CACHE_MISS_RD/INST_RETIRED
5 D-side_page_table_walk_rate,DTLB_WALK,1207.0,L1D_TLB,2239299992.0,5.39087727554174e-07,D-side_page_table_walk_rate=DTLB_WALK/L1D_TLB
6 CRYPTO_instruction_rate_per_instructions,CRYPTO_SPEC,0.0,INST_SPEC,26351513572.0,0.0,CRYPTO_instruction_rate_per_instructions=CRYPTO_SPEC/INST_SPEC
7 ST_instruction_rate_per_instructions,ST_SPEC,973375796.0,INST_SPEC,26341816834.0,0.036951733516863615,ST_instruction_rate_per_instructions=ST_SPEC/INST_SPEC
8 Front_end_stall_rate,STALL_FRONTEND,306877491.0,CPU_CYCLES,12024133878.0,0.0255217959242353,Front_end_stall_rate=STALL_FRONTEND/CPU_CYCLES
9 Speculatively_executed_IPC,INST_SPEC,26353329486.0,CPU_CYCLES,12024122764.0,2.1917049587102837,Speculatively_executed_IPC=INST_SPEC/CPU_CYCLES
10 L1_instruction_TLB_miss_rate,L1I_TLB_REFILL,2539.0,L1I_TLB,6622374662.0,3.833972146832909e-07,L1_instruction_TLB_miss_rate=L1I_TLB_REFILL/L1I_TLB
11 L1_D-cache_write_miss_rate,L1D_CACHE_REFILL_WR,2431838.0,L1D_CACHE_WR,956897620.0,0.0025413774150676643,L1_D-cache_write_miss_rate=L1D_CACHE_REFILL_WR/L1D_CACHE_W
R
12 Branch_MPKI,BR_MIS_PRED_RETIRED,49533800.0,INST_RETIRED,24509143583.0,0.002021033490307575,Branch_MPKI=BR_MIS_PRED_RETIRED/INST_RETIRED
13 Back_end_stall_rate,STALL_BACKEND,2513413986.0,CPU_CYCLES,12024126585.0,0.2090308986879316,Back_end_stall_rate=STALL_BACKEND/CPU_CYCLES
14 L2_cache_read_rate,L2D_CACHE_RD,45030404.0,L2D_CACHE,89995480.0,0.5003629515615673,L2_cache_read_rate=L2D_CACHE_RD/L2D_CACHE
15 ISB_rate_over_time,ISB_SPEC,10466.0,CPU_CYCLES,12024126972.0,8.704166235412904e-07,ISB_rate_over_time=ISB_SPEC/CPU_CYCLES
16 L2_TLB_miss_rate,L2D_TLB_REFILL,824.0,L2D_TLB,19774587.0,4.1669643972842516e-05,L2_TLB_miss_rate=L2D_TLB_REFILL/L2D_TLB
17 DMB_rate_over_time,DMB_SPEC,17751.0,CPU_CYCLES,12024142037.0,1.476279962027978e-06,DMB_rate_over_time=DMB_SPEC/CPU_CYCLES
18 DP_instruction_rate_over_time,DP_SPEC,18197316148.0,CPU_CYCLES,12024129771.0,1.5133998463563323,DP_instruction_rate_over_time=DP_SPEC/CPU_CYCLES
19 Exception_rate_per_instructions,EXC_TAKEN,1017.0,INST_RETIRED,24517803401.0,4.148006178883545e-08,Exception_rate_per_instructions=EXC_TAKEN/INST_RETIRED
20 BR_INDIRECT_instruction_rate_over_time,BR_INDIRECT_SPEC,307511635.0,CPU_CYCLES,12024137056.0,0.025574528431256762,BR_INDIRECT_instruction_rate_over_time=BR_INDIRE
CT_SPEC/CPU_CYCLES
21 L1_I-cache_miss_rate,L1I_CACHE_REFILL,145279.0,L1I_CACHE,7771230971.0,1.8694464305866017e-05,L1_I-cache_miss_rate=L1I_CACHE_REFILL/L1I_CACHE
22 Branch_misprediction_rate,BR_MIS_PRED_RETIRED,49607962.0,BR_RETIRED,4524656622.0,0.0109639175178054,Branch_misprediction_rate=BR_MIS_PRED_RETIRED/BR_RETIRED
23 I-side_page_table_walk_rate,ITLB_WALK,137.0,L1I_TLB,6622619075.0,2.0686679763474092e-08,I-side_page_table_walk_rate=ITLB_WALK/L1I_TLB
24 L1_I-cache_MPKI,L1I_CACHE_REFILL,148486.0,INST_RETIRED,24480779072.0,6.065411544432077e-06,L1_I-cache_MPKI=L1I_CACHE_REFILL/INST_RETIRED
25 VFP_instruction_rate_per_instructions,VFP_SPEC,558281441.0,INST_SPEC,26361919003.0,0.02117757212350388,VFP_instruction_rate_per_instructions=VFP_SPEC/INST_SPEC
26 L2_cache_eviction_rate,L2D_CACHE_WB_VICTIM,980463.0,L2D_CACHE,90815459.0,0.018892162422901159,L2_cache_eviction_rate=L2D_CACHE_WB_VICTIM/L2D_CACHE
27 L2_TLB_misses_per_context_swap,L2D_TLB_REFILL,1085.0,TTBR_WRITE_RETIRED,8.0,135.625,L2_TLB_misses_per_context_swap=L2D_TLB_REFILL/TTBR_WRITE_RETIRED
28 DSB_rate_over_time,DSB_SPEC,3176.0,CPU_CYCLES,12024140829.0,2.6413529624836697e-07,DSB_rate_over_time=DSB_SPEC/CPU_CYCLES
29 CRYPTO_instruction_rate_over_time,CRYPTO_SPEC,0.0,CPU_CYCLES,12024124568.0,0.0,CRYPTO_instruction_rate_over_time=CRYPTO_SPEC/CPU_CYCLES
30 L1_D-cache_eviction_rate,L1D_CACHE_WB_VICTIM,13575270.0,L1D_CACHE,2209845378.0,0.006143085907796034,L1_D-cache_eviction_rate=L1D_CACHE_WB_VICTIM/L1D_CACHE
31 L1_data_TLB_write_miss_rate,L1D_TLB_REFILL_WR,498236.0,L1D_TLB_WR,957842331.0,0.0005201649414260435,L1_data_TLB_write_miss_rate=L1D_TLB_REFILL_WR/L1D_TLB_WR
32 BR_RETURN_instruction_rate_per_instructions,BR_RETURN_SPEC,299813278.0,INST_SPEC,26371446087.0,0.011368859978740231,BR_RETURN_instruction_rate_per_instructions=BR
_RETURN_SPEC/INST_SPEC
33 L1_instruction_TLB_misses_per_context_swap,L1I_TLB_REFILL,2490.0,TTBR_WRITE_RETIRED,8.0,311.25,L1_instruction_TLB_misses_per_context_swap=L1I_TLB_REFILL/TTBR_WRI
TE_RETIRED
34 BR_RETURN_instruction_rate_over_time,BR_RETURN_SPEC,299133665.0,CPU_CYCLES,12024139926.0,0.02487759809928547,BR_RETURN_instruction_rate_over_time=BR_RETURN_SPEC/
CPU_CYCLES
35 BR_IMMED_instruction_rate_per_instructions,BR_IMMED_SPEC,4397074925.0,INST_SPEC,26360233187.0,0.16680713307075343,BR_IMMED_instruction_rate_per_instructions=BR_IM
MED_SPEC/INST_SPEC
```