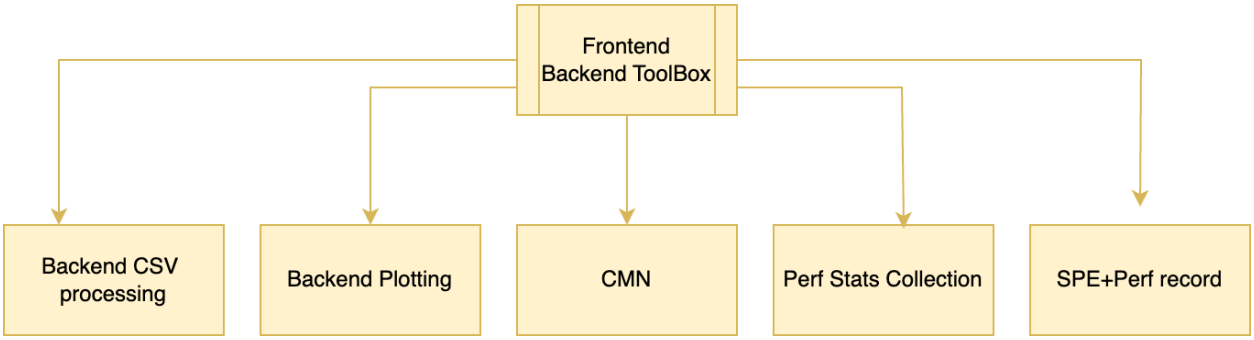
Frontend and Backend Tool – Perf stats

Table of Contents

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

Request access if not given already to 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 parallely 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

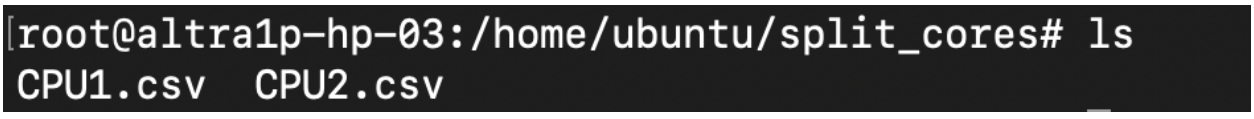
csv_split.py

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.




```
vim CPU1.csv
```

```
Metrics,Name_1,Event_1,Name_2,Event_2,Event_1/Event_2,Graph_Xlabel
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
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
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
D-side_page_table_walk_rate,DTLB_WALK,1207.0,L1D_TLB,2239299992.0,5.3900772755417 4e-07,D-side_page_table_walk_rate=DTLB_WALK/L1D_TLB
CRYPTO_instruction_rate_per_instructions,CRYPTO_SPEC,0.0,INST_SPEC,26351513572.0,0.0,CRYPTO_instruction_rate_per_instructions=CRYPTO_SPEC/INST_SPEC
ST_instruction_rate_per_instructions,ST_SPEC,973375796.0,INST_SPEC,26341816834.0,0.036951733516863615,ST_instruction_rate_per_instructions=ST_SPEC/INST_SPEC
Front_end_stall_rate,STALL_FRONTEND,306877491.0,CPU_CYCLES,12024133878.0,0.0255217959242353,Front_end_stall_rate=STALL_FRONTEND/CPU_CYCLES
Speculatively_executed_IPC,INST_SPEC,26353329486.0,CPU_CYCLES,12024122764.0,2.191704958710 2837,Speculatively_executed_IPC=INST_SPEC/CPU_CYCLES
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
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_WR
Branch_MPKI,BR_MIS_PRED_RETIRED,49533800.0,INST_RETIRED,24509143583.0,0.0020210334903075 75,Branch_MPKI=BR_MIS_PRED_RETIRED/INST_RETIRED
Back_end_stall_rate,STALL_BACKEND,2513413986.0,CPU_CYCLES,12024126585.0,0.2090308986879316,Back_end_stall_rate=STALL_BACKEND/CPU_CYCLES
L2_cache_read_rate,L2D_CACHE_RD,45030404.0,L2D_CACHE,89995480.0,0.5003629515615673,L2_cache_read_rate=L2D_CACHE_RD/L2D_CACHE
ISB_rate_over_time,ISB_SPEC,10466.0,CPU_CYCLES,12024126972.0,8.704166235412904e-07,ISB_rate_over_time=ISB_SPEC/CPU_CYCLES
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
DMB_rate_over_time,DMB_SPEC,17751.0,CPU_CYCLES,12024142037.0,1.4762799662027978e-06,DMB_rate_over_time=DMB_SPEC/CPU_CYCLES
DP_instruction_rate_over_time,DP_SPEC,18197316148.0,CPU_CYCLES,12024129771.0,1.5133998463563323,DP_instruction_rate_over_time=DP_SPEC/CPU_CYCLES
Exception_rate_per_instructions,EXC_TAKEN,1017.0,INST_RETIRED,24517803401.0,4.148061788835 45e-08,Exception_rate_per_instructions=EXC_TAKEN/INST_RETIRED
BR_INDIRECT_instruction_rate_over_time,BR_INDIRECT_SPEC,307511635.0,CPU_CYCLES,12024137056.0,0.025574528431256762,BR_INDIRECT_instruction_rate_over_time=BR_INDIRECT_SPEC/CPU_CYCLES
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
Branch_misprediction_rate,BR_MIS_PRED_RETIRED,49607962.0,BR_RETIRED,4524656622.0,0.0109639175178054,Branch_misprediction_rate=BR_MIS_PRED_RETIRED/BR_RETIRED
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
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
VFP_instruction_rate_per_instructions,VFP_SPEC,558281441.0,INST_SPEC,26361919003.0,0.021177572123503 88,VFP_instruction_rate_per_instructions=VFP_SPEC/INST_SPEC
L2_cache_eviction_rate,L2D_CACHE_WB_VICTIM,980463.0,L2D_CACHE,90015459.0,0.010892162422901159,L2_cache_eviction_rate=L2D_CACHE_WB_VICTIM/L2D_CACHE
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
DSB_rate_over_time,DSB_SPEC,3176.0,CPU_CYCLES,12024140829.0,2.6413529624836697e-07,DSB_rate_over_time=DSB_SPEC/CPU_CYCLES
CRYPTO_instruction_rate_over_time,CRYPTO_SPEC,0.0,CPU_CYCLES,12024124568.0,0.0,CRYPTO_instruction_rate_over_time=CRYPTO_SPEC/CPU_CYCLES
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
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
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
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_WRITE_RETIRED
BR_RETURN_instruction_rate_over_time,BR_RETURN_SPEC,299133665.0,CPU_CYCLES,12024139926.0,0.024877759809928547,BR_RETURN_instruction_rate_over_time=BR_RETURN_SPEC/CPU_CYCLES
BR_IMMED_instruction_rate_per_instructions,BR_IMMED_SPEC,4397074925.0,INST_SPEC,26360233187.0,0.16680713307075343,BR_IMMED_instruction_rate_per_instructions=BR_IMMED_SPEC/INST_SPEC
```