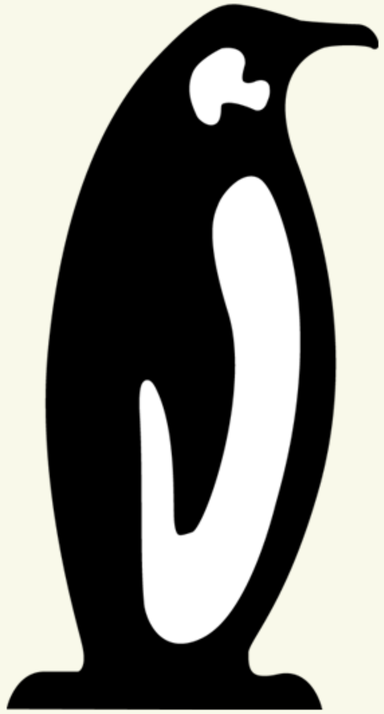


Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Performance Monitor Control Unit (PMCU)

Jie Zhan - Huawei

Contents

Motivation

- Performance Monitor Unit (PMU) are…

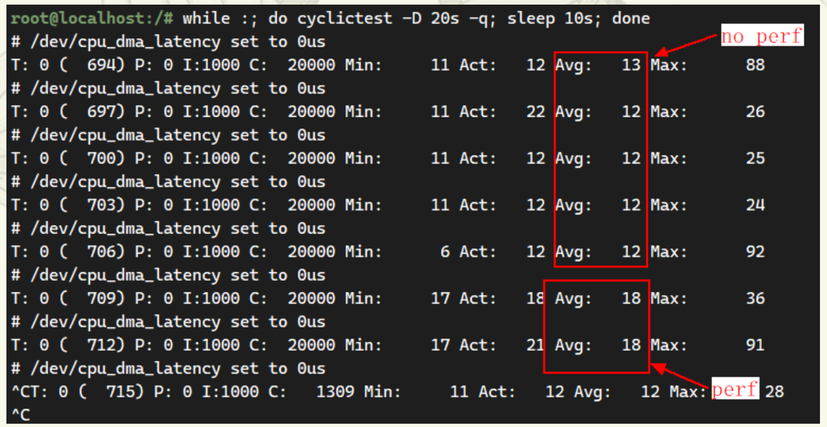
1. Counters for hardware events, e.g. cycles, cache misses, branch mispredictions
2. Great for performance analysis, resource management, debugging, task profiling, power modelling/management

BUT we found it…

1. Affecting latency/time-sensitive workloads

Example: “perf stat” monitoring VMs, 100+ events, running rt-tests

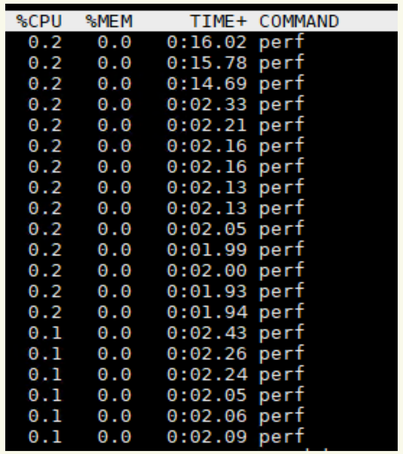
Impact: cyclictest latency increases by 50%



Latency impact on cyclictest by PMU monitoring

2. CPU utilization increases with targets

The more processes / threads we monitor, the more CPU utilization it occupies (0.1~0.2% CPU util / process)



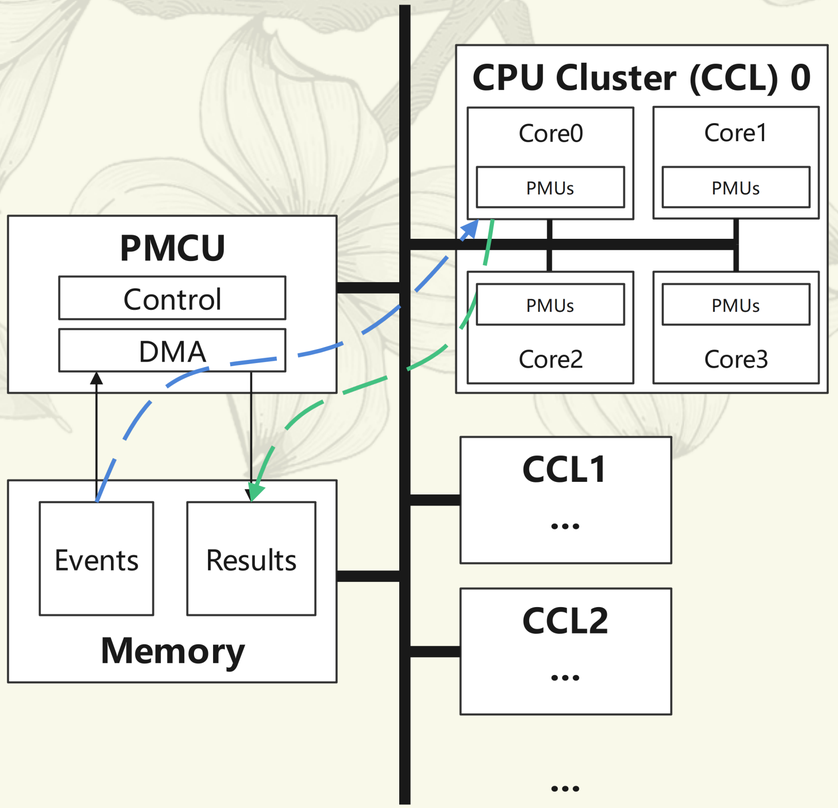
CPU utilization occupied by “perf”

What PMCU does?

A module that controls core PMUs through external memory-mapped interfaces, offloading monitor work from CPU

Function
Configure core PMUs, switch events, and save PMU readings to memory

Procedure
- Process an event list for a few rounds
- In each round:
  - Read a set of event IDs and set to PMUs for all cores
  - Start counting, wait for a period, stop counting
  - Save all PMU readings to memory
- Issue an IRQ when all rounds are done



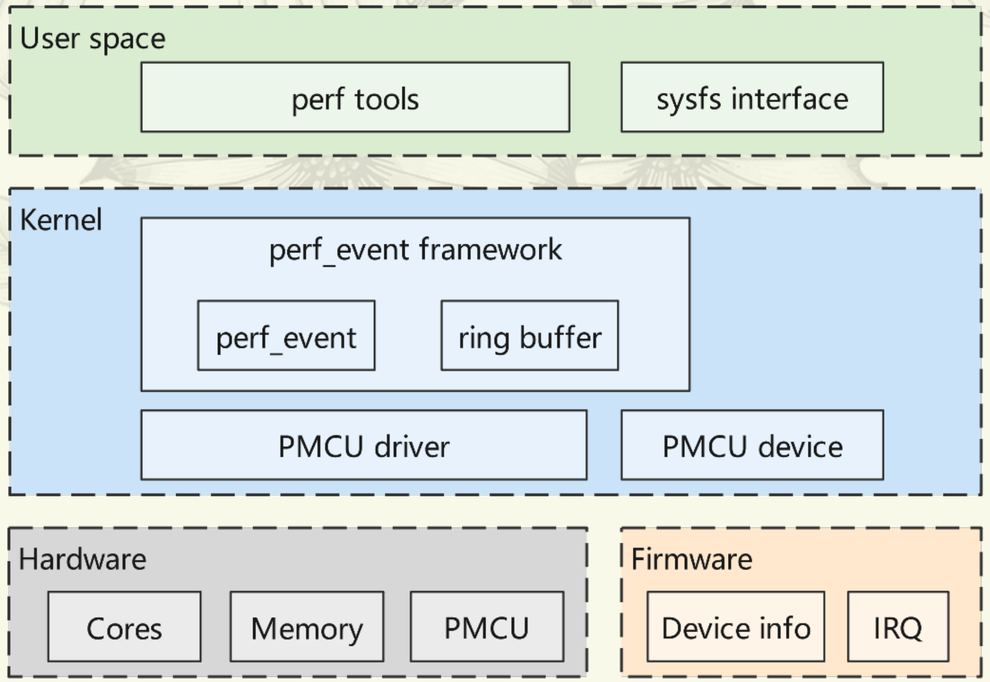
Hardware view of PMCU (simplified)

Software Design

Designed with perf_event auxtrace framework

4 layers:
- sysfs: configs, e.g. pass event IDs
- perf tool
  perf-record controls PMCU data recording
  perf-report/script decodes PMCU data
- perf_event framework
  allocates AUX buffers, mapped to user space, passed to driver
- driver
  interacts with the framework, configures hardware to produce results into the AUX buffers

Reference: https://lwn.net/Articles/922351/



PMCU software stack

Short Demo

1. Enter event numbers in the 'user_events' file:

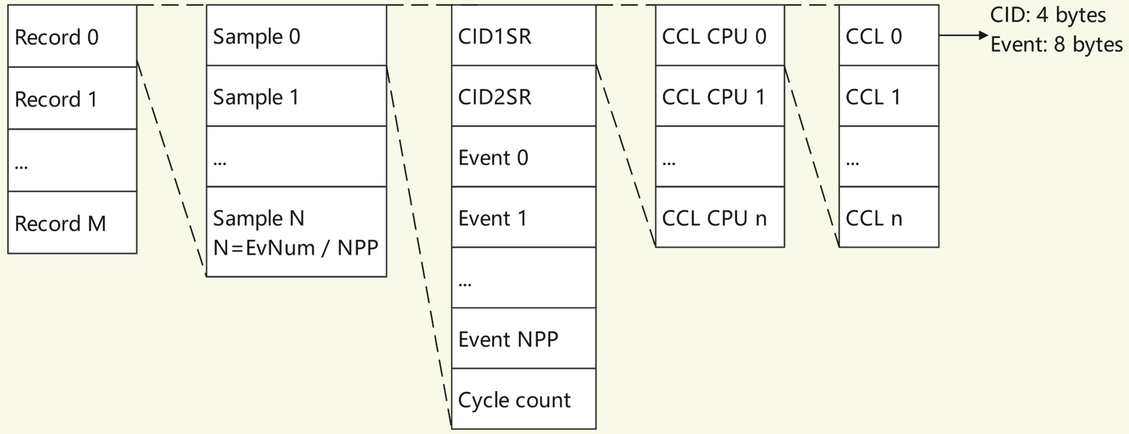
   echo “11 8 b 3c 24 23" > /sys/devices/hisi_pmcu_sccl3/user_events

2. Start the sampling with 'perf-record':

   perf record -e hisi_pmcu_sccl3/nr_sample=10,sample_period_ms=4/

3. Decode PMU data with 'perf-script’:

   perf report -D



Organization of PMCU data output

Organization of PMCU data output

Header

```
... HISI PMCU data: size 0x7840 bytes
... Header: size 0x40 bytes
 00000000:  00 00 40 00 06 00 00 00 30 00 00 00 00 00 00 00
 00000010:  00 0c 00 00 01 00 00 00 06 00 00 00 11 00 00 00
 00000020:  08 00 00 00 0b 00 00 00 3c 00 00 00 24 00 00 00
 00000030:  23 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 Auxtrace buffer max size: 0x400000
 Number of PMU counters in parallel: 6
 Number of monitored CPUs: 48
 Compatible mode: no
 Subsample size: 0xc00
 Number of subsamples per sample: 1
 Number of events: 6
 Event   0: 0x0011
 Event   1: 0x0008
 Event   2: 0x000b
 Event   3: 0x003c
 Event   4: 0x0024
 Event   5: 0x0023
... Data: size 0x7800 bytes
```

…

Data body

```
 00000300:  00000000009190c8    Event 0011 CPU 38
 00000308:  0000000000905798    Event 0011 CPU 46
 00000310:  0000000000009905    Event 0011 CPU 7
 00000318:  0000000000917760    Event 0011 CPU 15
 00000320:  000000000000a052    Event 0011 CPU 23
 00000328:  0000000000000000    Event 0011 CPU 31
 00000330:  000000000000c8b7    Event 0011 CPU 39
 00000338:  000000000000c5d14   Event 0011 CPU 47
 00000340:  000000000015ceb9    Event 0008 CPU 0
 00000348:  000000000015cd5f    Event 0008 CPU 8
 00000350:  00000000018e7a68    Event 0008 CPU 16
 00000358:  0000000000007394    Event 0008 CPU 24
 00000360:  00000000001e51ae    Event 0008 CPU 32
 00000368:  0000000000e7212c    Event 0008 CPU 40
 00000370:  000000000017468fb   Event 0008 CPU 1
 00000378:  0000000001573094    Event 0008 CPU 9
 00000380:  0000000000009f40    Event 0008 CPU 17
 00000388:  000000000180390c    Event 0008 CPU 25
```

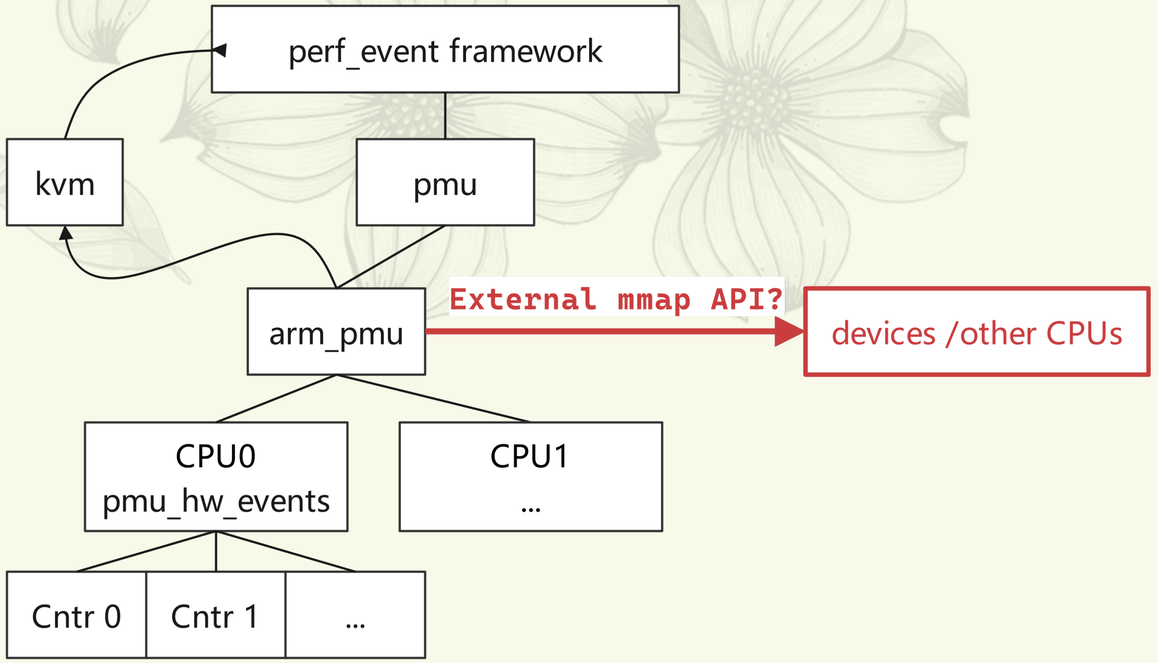
Main Challenge

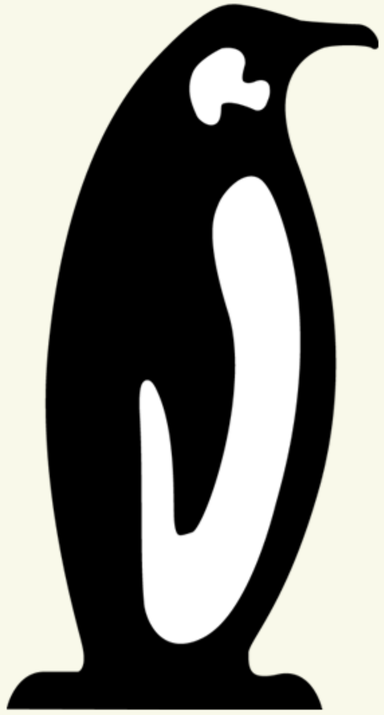
1. Synchronization between external and internal PMU accesses

ARM PMUs can be accessed from both internal sysreg (from local CPU) and external memory-mapped interfaces (from other CPUs or devices, e.g. PMCU) simultaneously.
However, **a mechanism for synchronizing counter accesses or reserving counters is missing**.

2. Kernel support for ARM PMU external memory-mapped accesses?

ARM PMUs are currently controlled from internal sysreg in Linux. External memory-mapped interfaces are not yet enabled.
Shall we enable support for it? Prioritized cores could avoid handling perf events and focus on the main workload.





Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023