A New AMD CPU Performance Scaling Proposal on Linux®

Huang Rui

Background

- VKD3D-Pronton Tuning on AMD APU (CPU + GPU) Platforms
  - The original problems are found by Valve Software (Link)
    - Incorrect CPU maximum frequency
      - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=3743d55b289c203d8f77b7cd47c24926b9d186ae
    - Slow slow-motion on Horizon Zero Dawn
      - Horizon Zero Dawn is Direct3D games which is using vkd3d-pronton to implement DirectX® games cross-platform on Steam Linux®
      - Issue is caused from the incorrect use with CPUFreq sysfs call on Wine
- Kernel CPUFreq on AMD Processors
  - By co-working with Valve software engineers on two issues, we found the original CPU performance scaling module was still based on the legacy kernel common ACPI cpufreq driver on AMD processors
  - This ACPI cpufreq driver is not very performance/power efficiency for modern AMD platforms right now

Legacy CPU Freq Design and Potential Conflict

- Legacy CPU Freq Design
  - Linux Kernel provides existing CPU Freq framework to use software governor (Ondemand) policy to control CPU frequency and clocks
  - Intel implemented ACPI based CPU frequency driver which is using on legacy Intel CPU (before Sandy Bridge) then switch to Intel specific Pstate driver in recent CPU series
  - Current AMD CPU platforms are still using ACPI based CPU frequency driver to manage CPU frequency and clocks with switching only in **3 P-states**
  - ACPI based CPU frequency driver develops on Intel platforms and might bring some potential issues on AMD processors
- Potential Conflict with SMU Cclk (CPU Clock) Dynamic Power Management (DPM)
  - Cclk DPM provides the firmware-based method to control the CPU frequency as well
  - The Cclk DPM and ACPI CPU Freq driver control would have the conflict to impact the target frequency

ACPI CPU Freq Driver
SMU Cclk DPM Firmware

New AMD CPU Freq Design Proposal

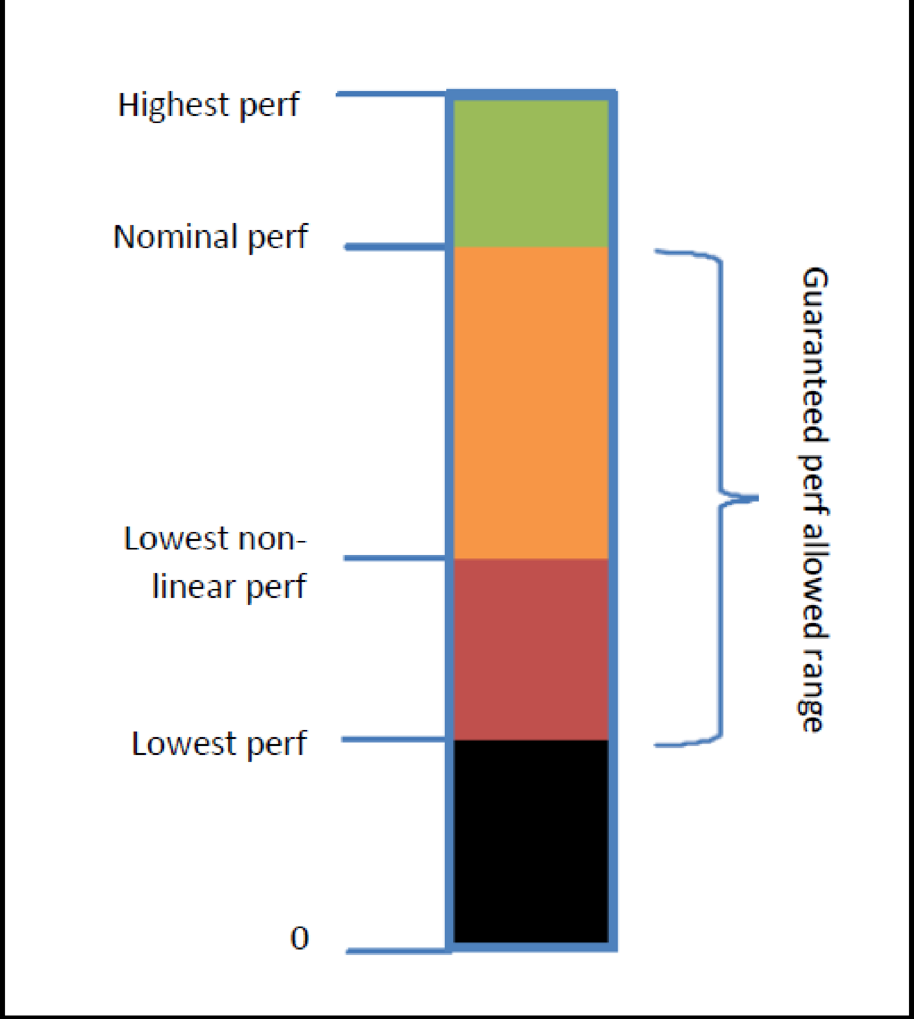
- CPU Hardware Dependencies
  - SMU Cclk DPM - Sampling C0 residency (CPU Idle)
  - SBIOS ACPI CPPC Tables
  - CPU MSR is the backend
    - The reliability and accuracy of MSR APIs should be the same with CPPC
    - MSR is the low-latency register model that is faster than ACPI AML code interpreter
- **Design a new CPPC Based CPU Freq Driver for AMD Processors**
  - Implement a new amd-pstate driver instead of acpi-cpufreq driver on AMD platform
  - Use fine grain CPPC frequency range instead of ACPI 3 P-States to control CPU frequency
  - “schedutil” governor can predict the workload to calculate more reasonable desired performance value via Linux® CPU CFS scheduler
  - Use “schedutil” governor to optimize the solution, and manage the hints to SMU Cclk DPM Arbiter and then calculate the target frequency

AMD CPU Freq Driver (CPPC Based)

SMU Cclk DPM Firmware

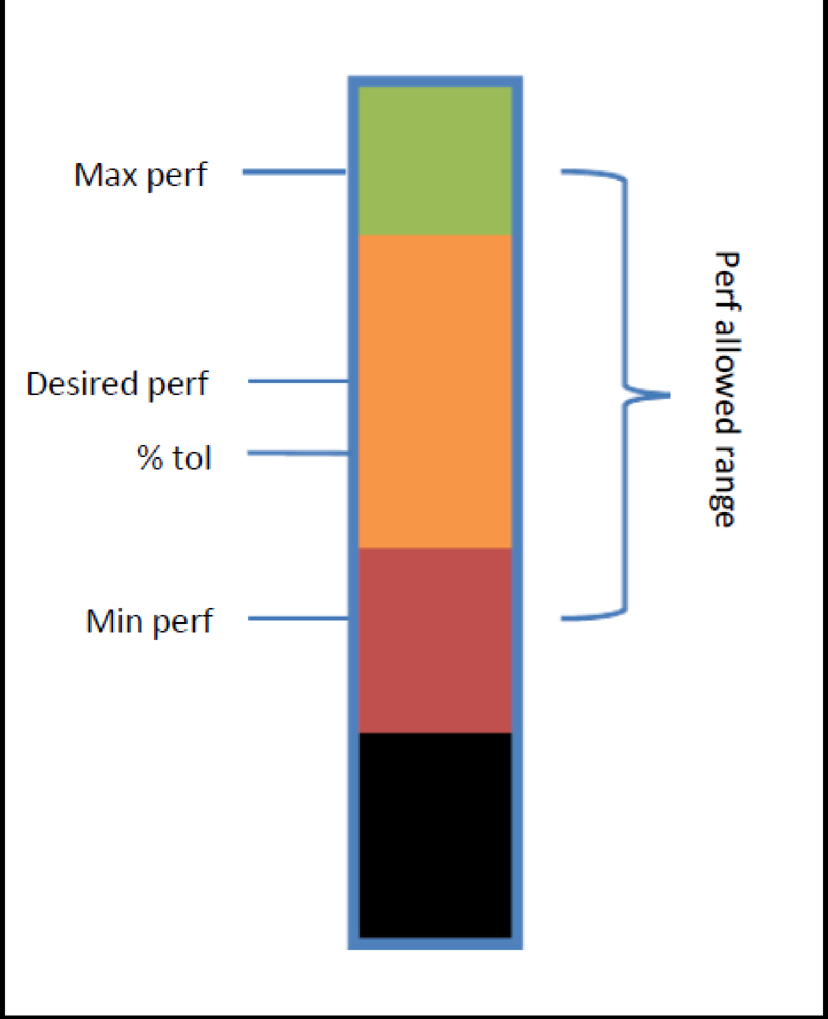
AMD P-States Performance Capability

- Highest Performance (RO)
  - It is the absolute maximum performance an individual processor may reach, assuming ideal conditions
- Nominal (Guaranteed) Performance (RO)
  - It is the maximum sustained performance level of the processor, assuming ideal operating conditions
- Lowest non-linear Performance (RO)
  - It is the lowest performance level at which nonlinear power savings are achieved, for example, due to the combined effects of voltage and frequency scaling
- Lowest Performance (RO)
  - It is the absolute lowest performance level of the processor



AMD P-States Performance Control

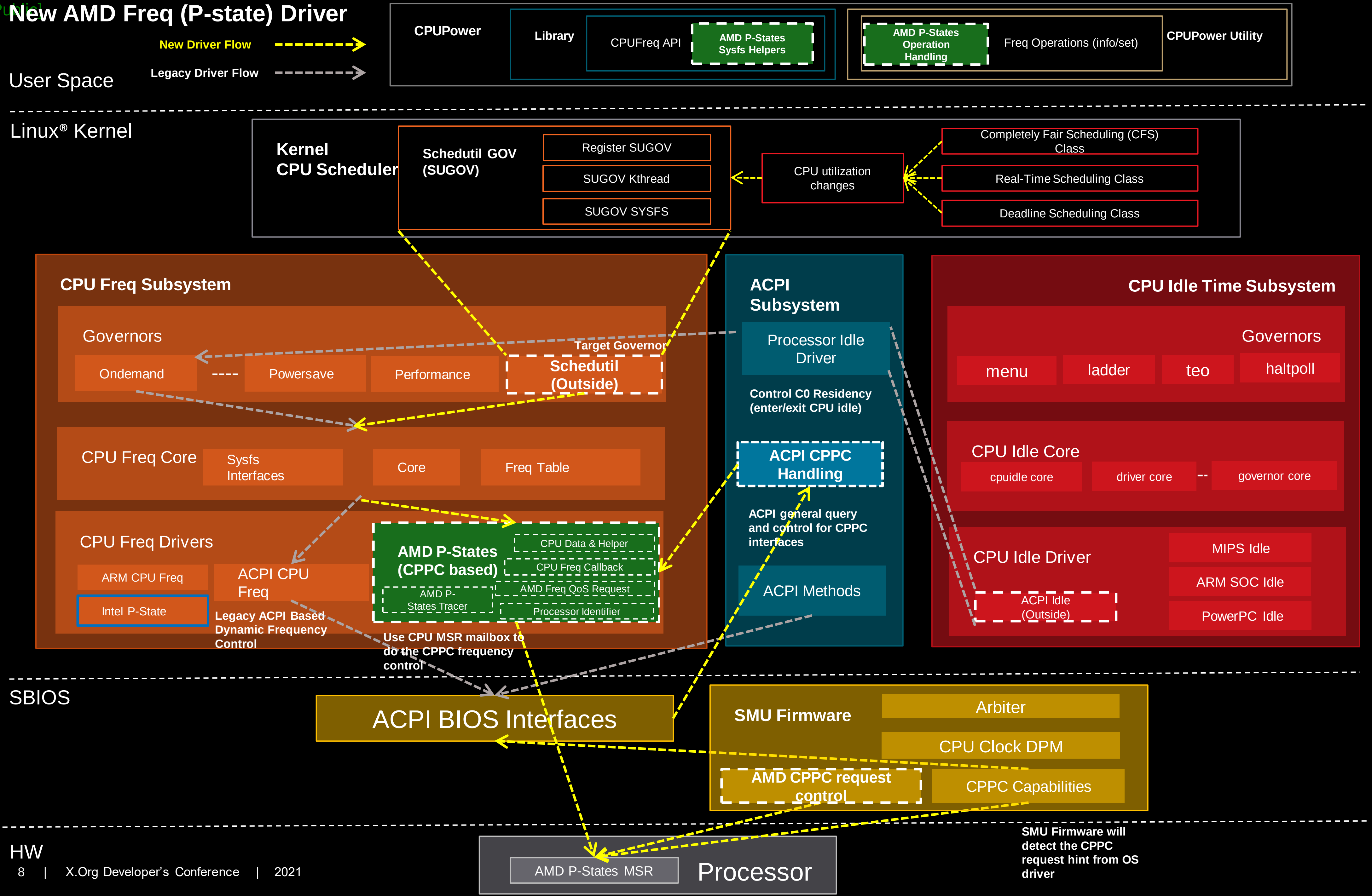
- Minimum Requested Performance (RW)
  - Software specifies the minimum allowed performance level
- Maximum Requested Performance (RW)
  - Software specifies a limit the maximum performance that is expected to be supplied by the hardware
- Desired performance target (RW)
  - Software specifies a desired QoS target in the performance scale as a relative number
- Energy Performance Preference (EPP) (RW)
  - Provides a hint to the hardware if software wants to bias toward performance (0x0) or energy efficiency (0xff)



Frequency Control Governors in Linux Kernel

- Traditional Max/Min Governor: Performance/Powersave
  - Performance
    - Set the minimum performance as the nominal performance value
    - Set the maximum performance as the highest performance value
  - Powersave
    - Set the minimum performance as the lowest nonlinear performance value
    - Set the maximum performance as the lowest nonlinear performance value
- Legacy Dynamic Control Governor: Ondemand
  - This governor sets the CPU frequency depending on the current system load
- CPU CFS Scheduler Governor: Schedutil
  - This "schedutil" governor uses PELT-based next frequency formula
  - Per-Entity Load Tracking (PELT) mechanism in the Linux® kernel scheduler

New AMD Freq (P-state) Driver



Control C0 Residency (enter/exit CPU idle)

ACPI general query and control for CPPC interfaces

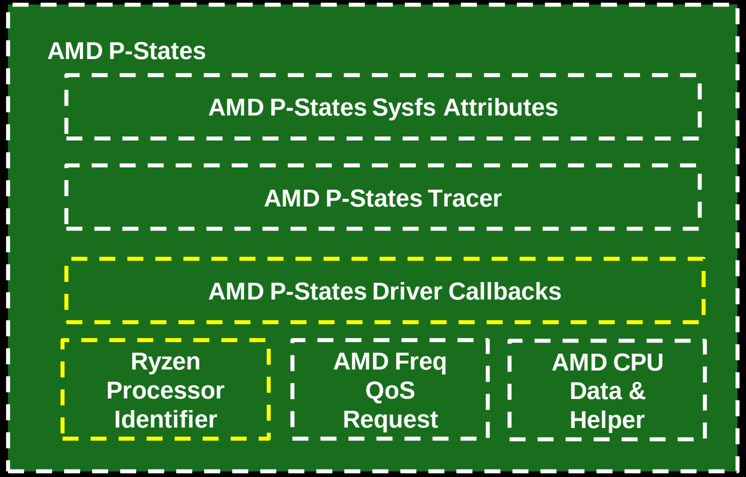
Legacy ACPI Based Dynamic Frequency Control

Use CPU MSR mailbox to do the CPPC frequency control

SMU Firmware will detect the CPPC request hint from OS driver

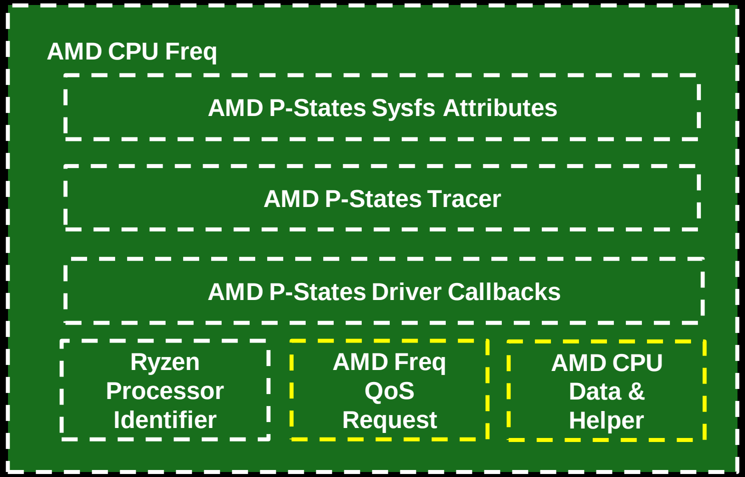
New AMD-PSTATE Core Design and Implementation (1)

- **Introduce a new CPU Freq Kernel Module: "AMD_PSTATE"**
  - Manage the CPU frequency and performance via AMD P-States API with multiple Linux® kernel governors.
  - Ryzen Processor Identifier
    - Identify the AMD P-States with _CPC check.
    - Identify the AMD P-States for full MSR support with bit 27 of *CPUID Fn80000008_EBX*
  - AMD CPU Freq Driver Callbacks
    - *amd_pstate_driver* instance
    - Callback functions of *struct cpufreq_driver*
    - *target* and *adjust_perf* method to support Ondemand and Schedutil governors
  - AMD CPU P-State reStructuredText (kernel documentation)
    - Implement amd-pstate CPU performance scaling driver RST documentation


AMD P-States
AMD P-States Sysfs Attributes
AMD P-States Tracer
AMD P-States Driver Callbacks
Ryzen Processor Identifier
AMD Freq QoS Request
AMD CPU Data & Helper

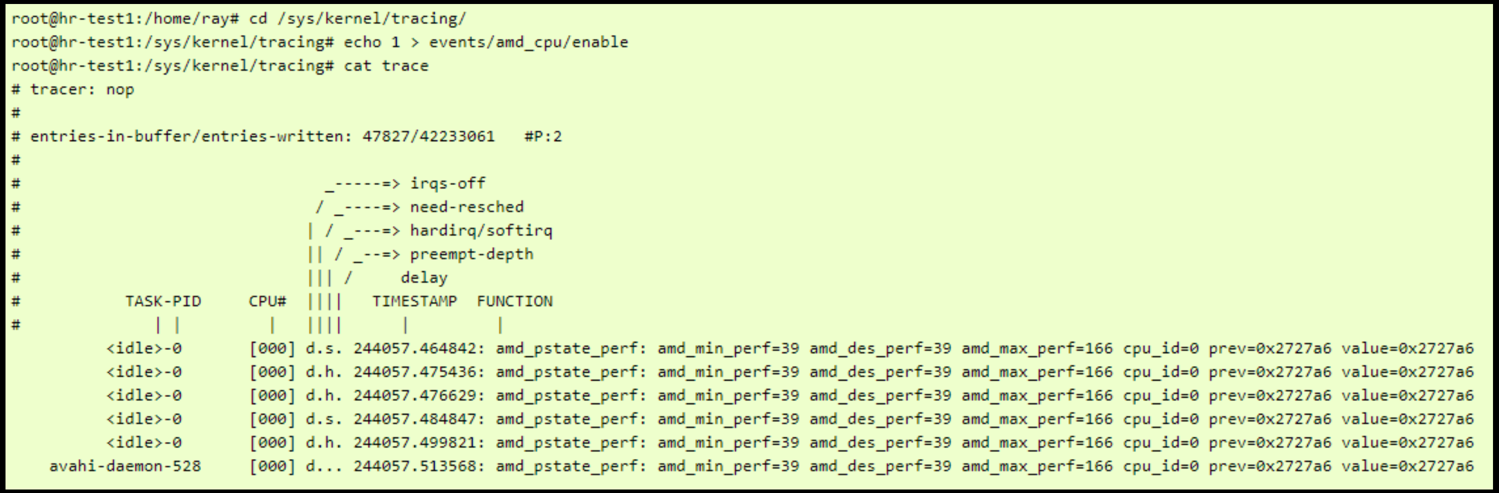
New AMD-PSTATE Core Design and Implementation (2)

- **Introduce a new CPU Freq Kernel Module: "AMD_PSTATE"**
  - AMD CPU Data & Helper
    - Implement a *struct amd_cpudata* to store private AMD specific information and function callbacks
    - Implement MSR register write/read helpers
      - AMD P-States Capability
      - AMD P-States Request
  - AMD Freq QoS (Quality of Service) Request
    - *struct freq_qos_request*
  - Future Design Work (under planning)
    - Kernel parameters to switch different frequency management policies
    - Customization for the different management policy for specific product
    - EPP (Energy Performance Preference)
    - Preferred Core

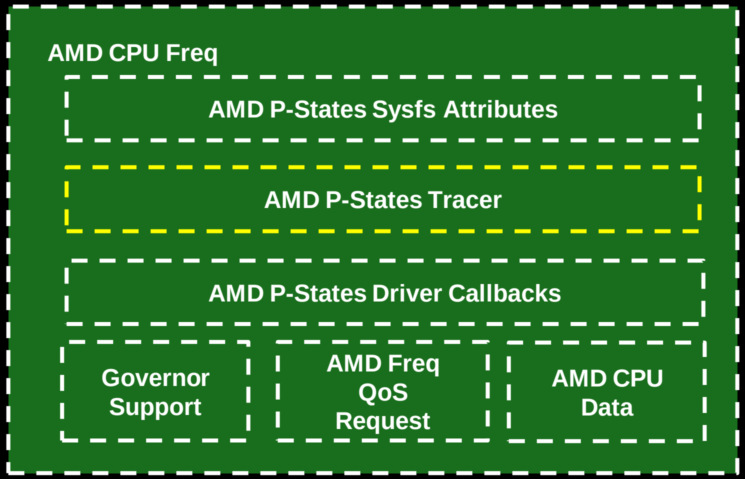

AMD CPU Freq
AMD P-States Sysfs Attributes
AMD P-States Tracer
AMD P-States Driver Callbacks
Ryzen Processor Identifier
AMD Freq QoS Request
AMD CPU Data & Helper

AMD-PSTATE Tracer Design and Implementation

- **Two Trace Events for Diagnostics and Tunning**
  - *cpu_frequency* trace event from CPUFreq core
  - *amd_pstate_perf* trace event specific to amd-pstate



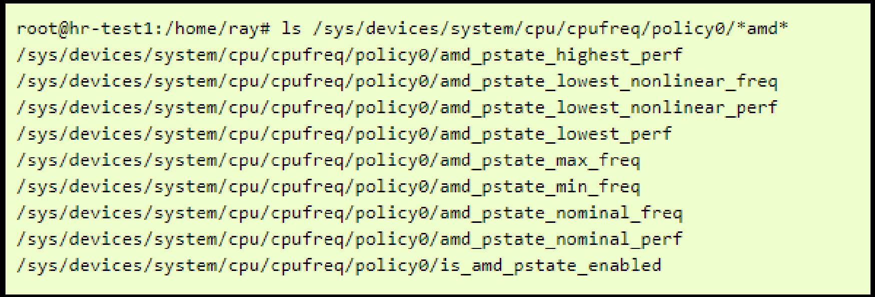
```
root@hr-test1:/home/ray# cd /sys/kernel/tracing/
root@hr-test1:/sys/kernel/tracing# echo 1 > events/amd_cpu/enable
root@hr-test1:/sys/kernel/tracing# cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 47827/42233061   #P:2
#
#                                _-----=> irqs-off
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| /     delay
#           TASK-PID     CPU#  ||||   TIMESTAMP  FUNCTION
#              | |         |   ||||      |         |
          <idle>-0       [000] d.s. 244057.464842: amd_pstate_perf: amd_min_perf=39 amd_des_perf=39 amd_max_perf=166 cpu_id=0 prev=0x2727a6 value=0x2727a6
          <idle>-0       [000] d.h. 244057.475436: amd_pstate_perf: amd_min_perf=39 amd_des_perf=39 amd_max_perf=166 cpu_id=0 prev=0x2727a6 value=0x2727a6
          <idle>-0       [000] d.h. 244057.476629: amd_pstate_perf: amd_min_perf=39 amd_des_perf=39 amd_max_perf=166 cpu_id=0 prev=0x2727a6 value=0x2727a6
          <idle>-0       [000] d.s. 244057.484847: amd_pstate_perf: amd_min_perf=39 amd_des_perf=39 amd_max_perf=166 cpu_id=0 prev=0x2727a6 value=0x2727a6
          <idle>-0       [000] d.h. 244057.499821: amd_pstate_perf: amd_min_perf=39 amd_des_perf=39 amd_max_perf=166 cpu_id=0 prev=0x2727a6 value=0x2727a6
    avahi-daemon-528     [000] d... 244057.513568: amd_pstate_perf: amd_min_perf=39 amd_des_perf=39 amd_max_perf=166 cpu_id=0 prev=0x2727a6 value=0x2727a6
```



AMD-PSTATE Sysfs Attributes Design and Implementation

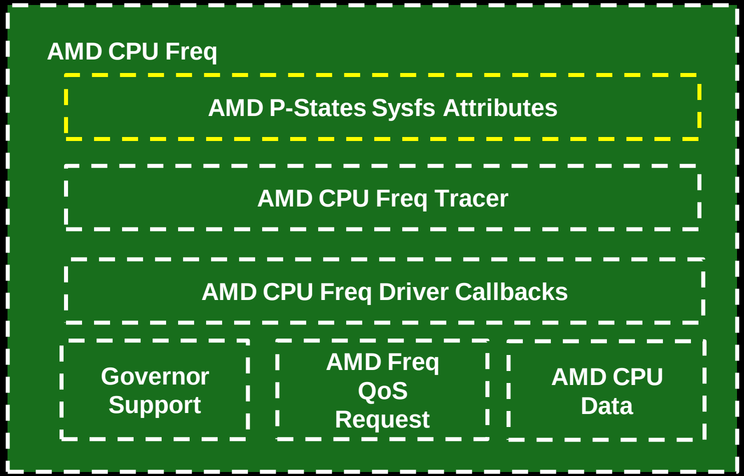
Global Kernel Sysfs Attributes For User Mode Library Query

- amd-pstate exposes several global attributes (files) in sysfs to control its functionality at the system level. They located in the /sys/devices/system/cpu/cpufreq/policyX/ directory and affect all CPUs



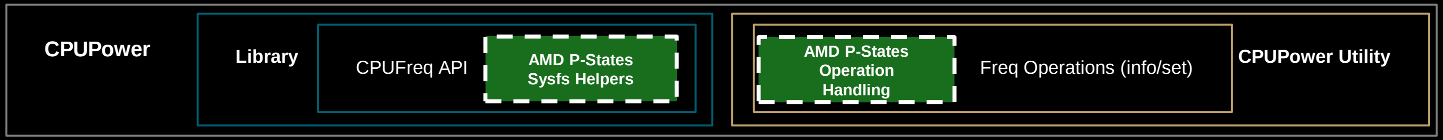
```
root@hr-test1:/home/ray# ls /sys/devices/system/cpu/cpufreq/policy0/*amd*
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_highest_perf
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_lowest_nonlinear_freq
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_lowest_nonlinear_perf
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_lowest_perf
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_max_freq
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_min_freq
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_nominal_freq
/sys/devices/system/cpu/cpufreq/policy0/amd_pstate_nominal_perf
/sys/devices/system/cpu/cpufreq/policy0/is_amd_pstate_enabled
```

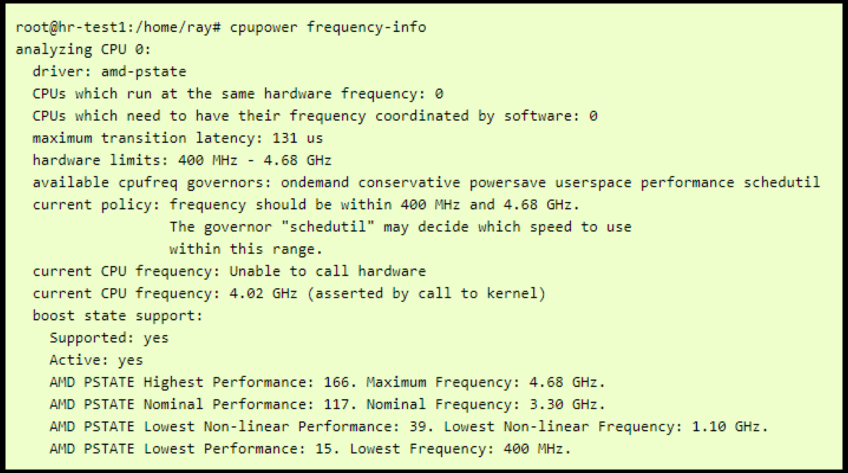
- *is_amd_pstate_enabled*
- *amd_pstate_highest_perf / amd_pstate_max_freq*
- *amd_pstate_nominal_perf / amd_pstate_nominal_freq*
- *amd_pstate_lowest_nonlinear_perf / amd_pstate_lowest_nonlinear_freq*
- *amd_pstate_lowest_perf / amd_pstate_min_freq*



CPU Power Library and Tool Support

- CPUPower
  - cpupower is a set of user space utilities designed to assist with CPU frequency scaling
    - Implement the support for new amd-pstate module which expose the sysfs API from kernel to use space
      - Expose AMD P-States performance capability into the library of cpupower
      - Implement the frequency operation control into cpupower utilities for AMD P-States

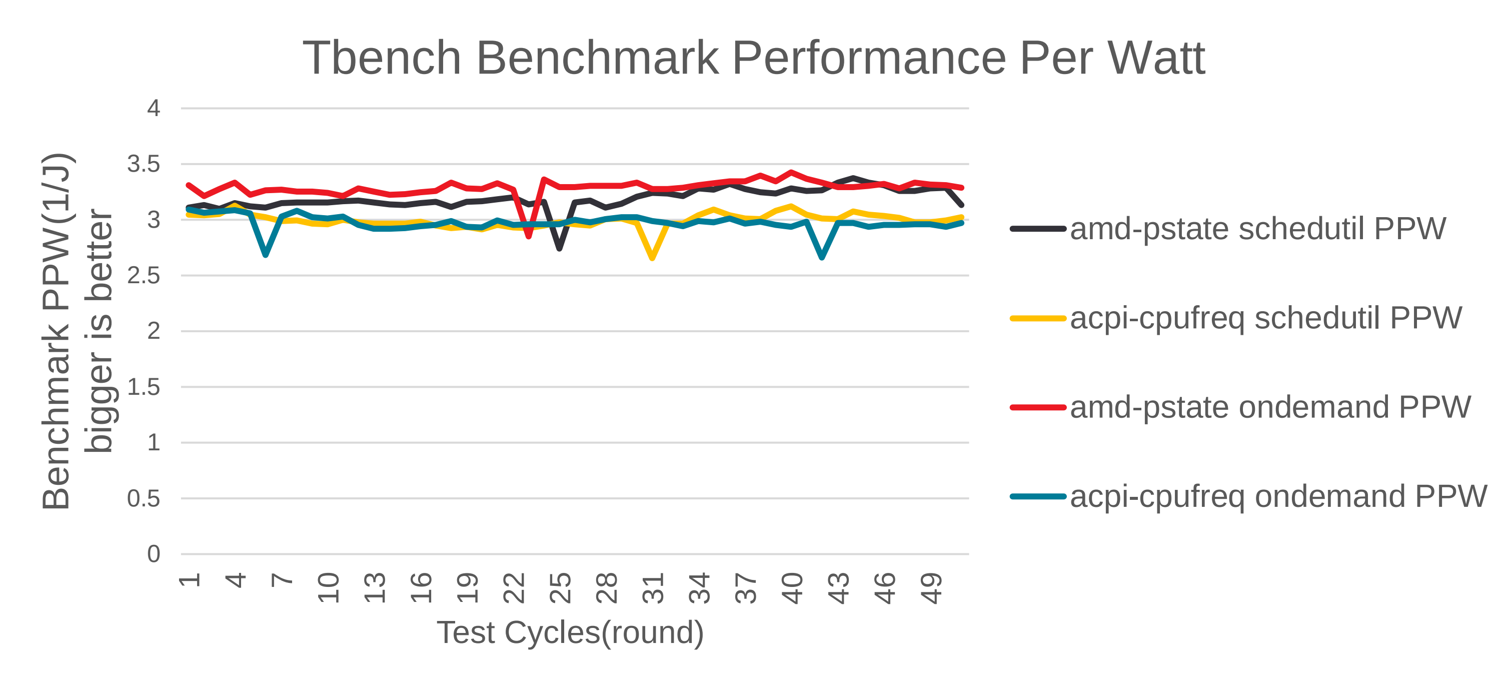

CPUPower
Library
CPUFreq API
AMD P-States Sysfs Helpers
AMD P-States Operation Handling
Freq Operations (info/set)
CPUPower Utility



```
root@hr-test1:/home/ray# cpupower frequency-info
analyzing CPU 0:
  driver: amd-pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 131 us
  hardware limits: 400 MHz - 4.68 GHz
  available cpufreq governors: ondemand conservative powersave userspace performance schedutil
  current policy: frequency should be within 400 MHz and 4.68 GHz.
                  The governor "schedutil" may decide which speed to use
                  within this range.
  current CPU frequency: Unable to call hardware
  current CPU frequency: 4.02 GHz (asserted by call to kernel)
  boost state support:
    Supported: yes
    Active: yes
    AMD PSTATE Highest Performance: 166. Maximum Frequency: 4.68 GHz.
    AMD PSTATE Nominal Performance: 117. Nominal Frequency: 3.30 GHz.
    AMD PSTATE Lowest Non-linear Performance: 39. Lowest Non-linear Frequency: 1.10 GHz.
    AMD PSTATE Lowest Performance: 15. Lowest Frequency: 400 MHz.
```

TBench Benchmark for AMD-PSTATE vs ACPI-CPUFREQ

- AMD-PSTATE vs ACPI-CPUFREQ on AMD Cezanne APU
  - TBench Benchmark
    - CPU: "Cezanne" Ryzen 7 5750G, 8C16T, disabled iGPU
    - MEM: DDR4 2666/8G*2/two channel
    - GPU: dGPU Radeon VII
    - Storage: NVME SSD, PCI-e x4, Samsung 980 512GB
    - OS: ubuntu 20.04.2 LTS
    - Kernel: test kernel, 5.12.0-rc5
    - Filesystem: ext4
    - BIOS: WA21407N 04/06/2021
    - Motherboard: Artic, known as B550
    - TBench Client number: 128
    - TBench Link: https://dbench.samba.org/web/download.html

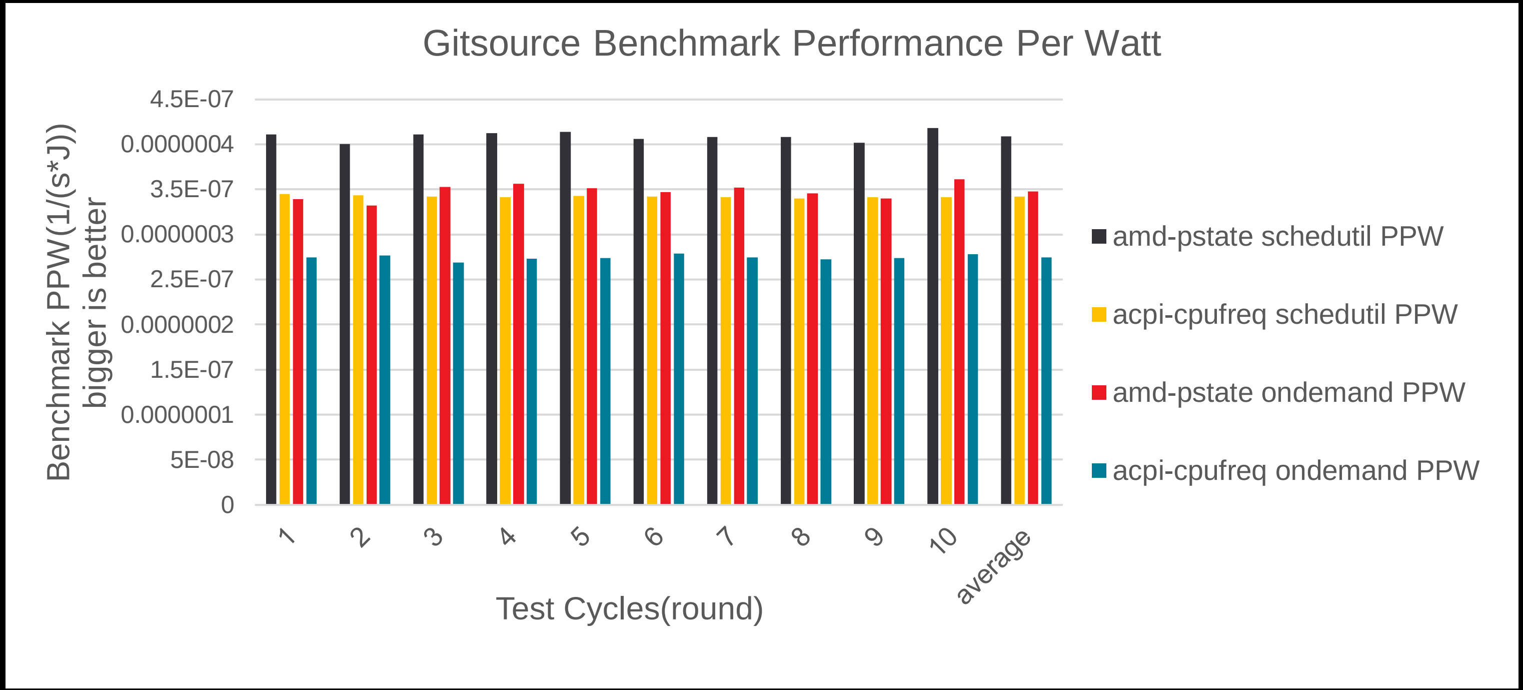


TBench

Formula	Kernel Module	Performance Per Watt (Schedutil) Unit: MB/(s*Joules)	Performance Per Watt (Ondemand) Unit: MB/(s*Joules)
Performance Per Watts (PPW) = F/E (F as Throughput MB/s, E as Energy Joules, the result unit is MB/(s*Joules))	acpi-cpufreq (legacy)	3.022	2.969
	amd-pstate	3.131 (+ 3.6%)	3.284 (+ 10.6%)

Gitsource Benchmark for AMD-PSTATE vs ACPI-CPUFREQ

- AMD-PSTATE vs ACPI-CPUFREQ on AMD Cezanne APU
  - Gitsource Benchmark
    - CPU: "Cezanne" Ryzen 7 5750G, 8C16T, disabled iGPU
    - MEM: DDR4 2666/8G*2/two channel
    - GPU: dGPU Radeon VII
    - Storage: NVME SSD, PCI-e x4, Samsung 980 512GB
    - OS: ubuntu 20.04.2 LTS
    - Kernel: test kernel, 5.12.0-rc5
    - Filesystem: ext4
    - BIOS: WA21407N 04/06/2021
    - Motherboard: Artic, known as B550
    - git version: 2.15.1
    - Benchmark Link: https://github.com/git/git

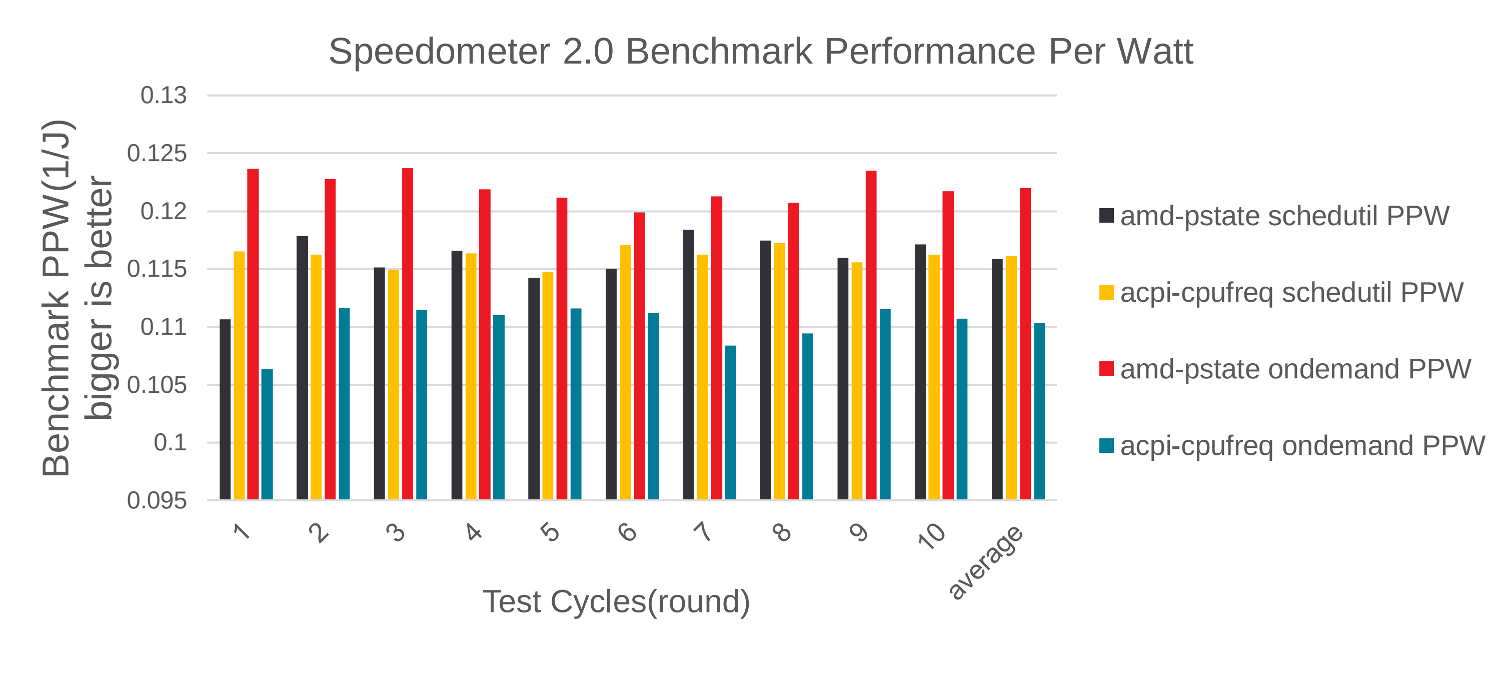


Gitsource

Formula	Kernel Module	Performance Per Watt (Schedutil) Unit: 1/(s*Joules)	Performance Per Watt (Ondemand) Unit: 1/(s*Joules)
Performance Per Watts (PPW) = F/E (F as the reciprocal of time 1/s, E as Energy Joules, the result unit is 1/(s*Joules))	acpi-cpufreq (legacy)	3.42E-07	2.74508E-07
	amd-pstate	4.09E-07 (+ 19.5%)	3.4761E-07 (+ 26.6%)

Speedometer 2.0 Benchmark for AMD-PSTATE vs ACPI-CPUFREQ

- AMD-PSTATE vs ACPI-CPUFREQ on AMD Cezanne APU
  - Speedometer 2.0 (For Chrome Book)
    - CPU: "Cezanne" Ryzen 7 5750G, 8C16T, disabled iGPU
    - MEM: DDR4 2666/8G*2/two channel
    - GPU: dGPU Radeon VII
    - Storage: NVME SSD, PCI-e x4, Samsung 980 512GB
    - OS: ubuntu 20.04.2 LTS
    - Kernel: test kernel, 5.12.0-rc5
    - Filesystem: ext4
    - BIOS: WA21407N 04/06/2021
    - Motherboard: Artic, known as B550
    - Speedometer 2.0 link: https://browserbench.org/Speedometer2.0/

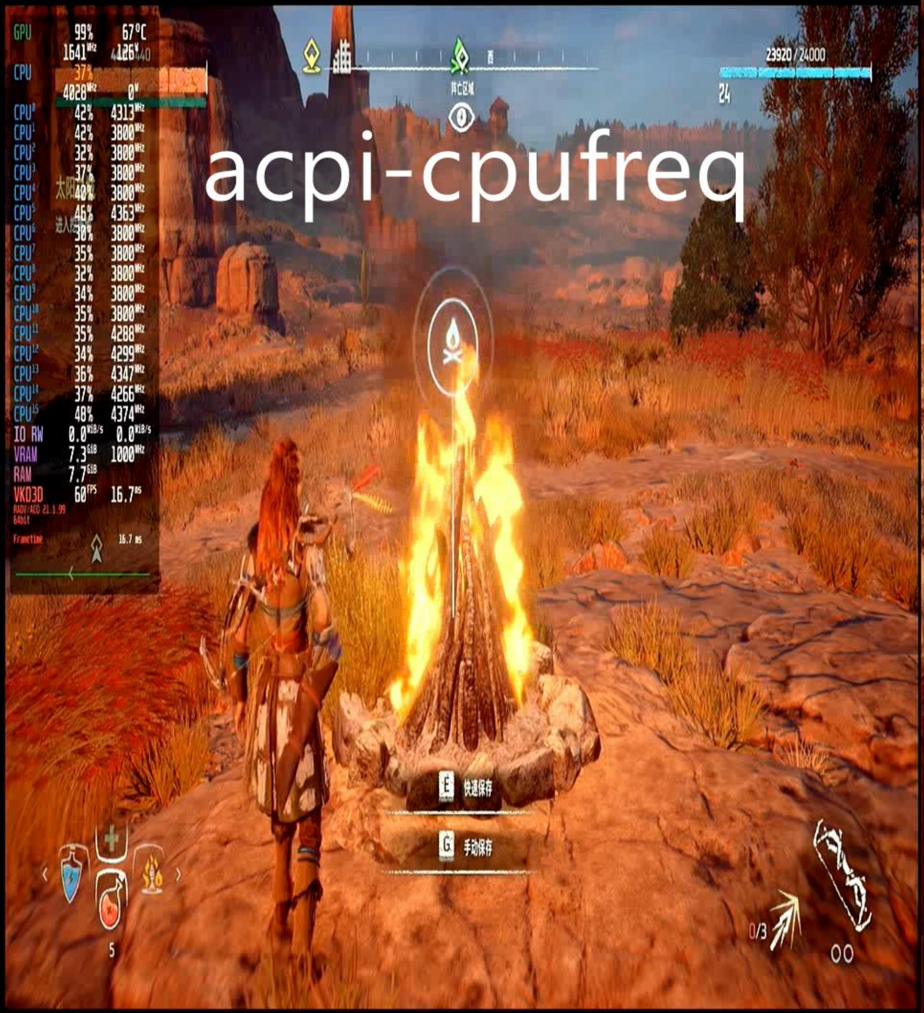


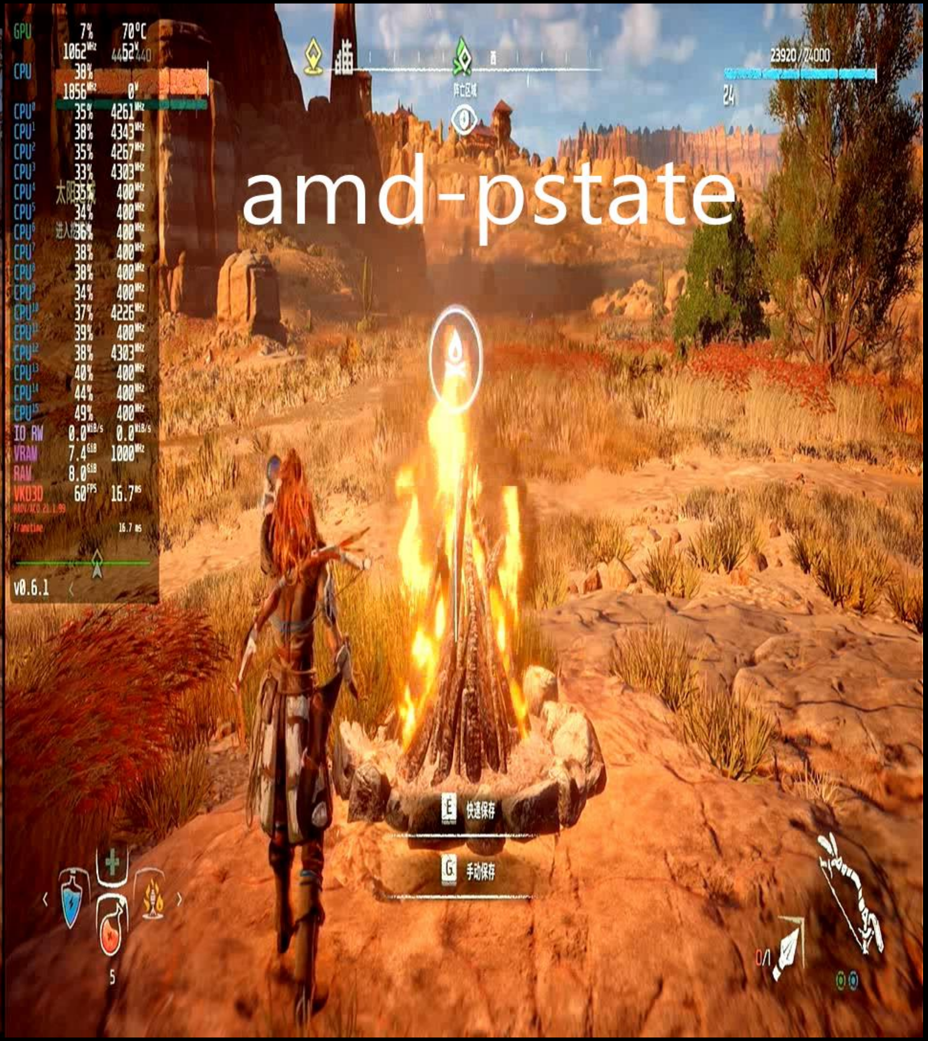
Speedometer 2.0

Formula	Kernel Module	Performance Per Watt (Schedutil) Unit: 1/Joules	Performance Per Watt (Ondemand) Unit: 1/Joules
Performance Per Watts (PPW) = F/E (F as benchmark goal, E as Energy Joules, the result unit is 1/Joules)	acpi-cpufreq (legacy)	0.116111767	0.110321664
	amd-pstate	0.115825281 (-0.25%)	0.122024299 (+ 10.6%)

Horizon Zero Dawn for AMD-PSTATE vs ACPI-CPUFREQ

- AMD-PSTATE vs ACPI-CPUFREQ on AMD Cezanne APU
  - Horizon Zero Dawn (Provided by Valve)
    - APU/CPU: Cezanne, Ryzen 7 pro 5750g, disabled iGPU (make sure no GPU bottleneck)
    - dGPU: VEGA20, Radeon VII
    - Memory: DDR4 8+8GB, double channel
    - Game config, HZD on steam with VKD3D Proton, @1080P, 60Hz, same graphics configuration
  - It is easy to observe the right platform with amd-pstate module faster and lower average power consumption





Challenges and Future

- Improvement from legacy acpi-cpufreq to amd-pstate solution
  - We have tested many CPU benchmarks and get positive improvement (please refer TBench/Gitsource/Speedometer/Horizon Zero Dawn comparison result)
    - These Benchmarks covered generic Linux and Chrome platforms
- Linux® Kernel Upstream
  - We are in progress to upstream the whole solution into official Linux® kernel (https://www.kernel.org/)
    - Below is our working git repo for upstream
    - https://git.kernel.org/pub/scm/linux/kernel/git/rui/linux.git/
- Challenges
  - AMD P-States MSR based APIs are the first time (new) for Linux® solution, it may have potential stability issues that need more testing and verification
  - Hope this solution to be accepted by Linux® community

Reference

- Linux® amd-pstate driver source
  - https://lore.kernel.org/linux-pm/20210908150001.3702552-1-ray.huang@amd.com/
- CPUFreq and The Scheduler
  - http://events17.linuxfoundation.org/sites/events/files/slides/cpufreq_and_scheduler_0.pdf
- Linux® cpufreq governor
  - https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt
- Prior Work
  - https://lkml.org/lkml/2019/7/10/682
- Windows® Server CPPC
  - https://docs.microsoft.com/en-us/windows-server/administration/performance-tuning/hardware/power/power-performance-tuning

Thank You and Q&A

Contributors:

- Paul Blinzer <paul.blinzer@amd.com>
- Alex Deucher <alexander.deucher@amd.com>
- Deepak Sharma <deepak.sharma@amd.com>
- Jinzhou Su <Jinzhou.Su@amd.com>
- Xiaojian Du <Xiaojian.Du@amd.com>
- Mario Limonciello <mario.limonciello@amd.com>
- Nathan Fontenot <nathan.fontenot@amd.com>

AMD

Disclaimer:

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD