A New CPU Frequency Control Mechanism on Linux®

Ray Huang (Huang Rui) <ray.huang@amd.com>

CLK - China Linux® Kernel 2022

AMD
together we advance_

Background

- Chrome OS Requirement
  - Chrome platforms with Cezanne
  - First Full MSR support with CPPC
- Steam Deck for VKD3D-Pronton Tuning
  - 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

Legacy CPUFreq Design and Potential Conflict

- Legacy CPU Freq Design
  - Linux® Kernel provides existing CPUFreq 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 P-State 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 will bring many potential issues on AMD platforms

- Potential Conflict with SMU Cclk Dynamic Frequency Management (DPM)
  - Cclk DPM provides the firmware-based method to control the CPU frequency as well
  - The Cclk DPM and ACPI CPUFreq driver control would have the conflict to impact the target frequency

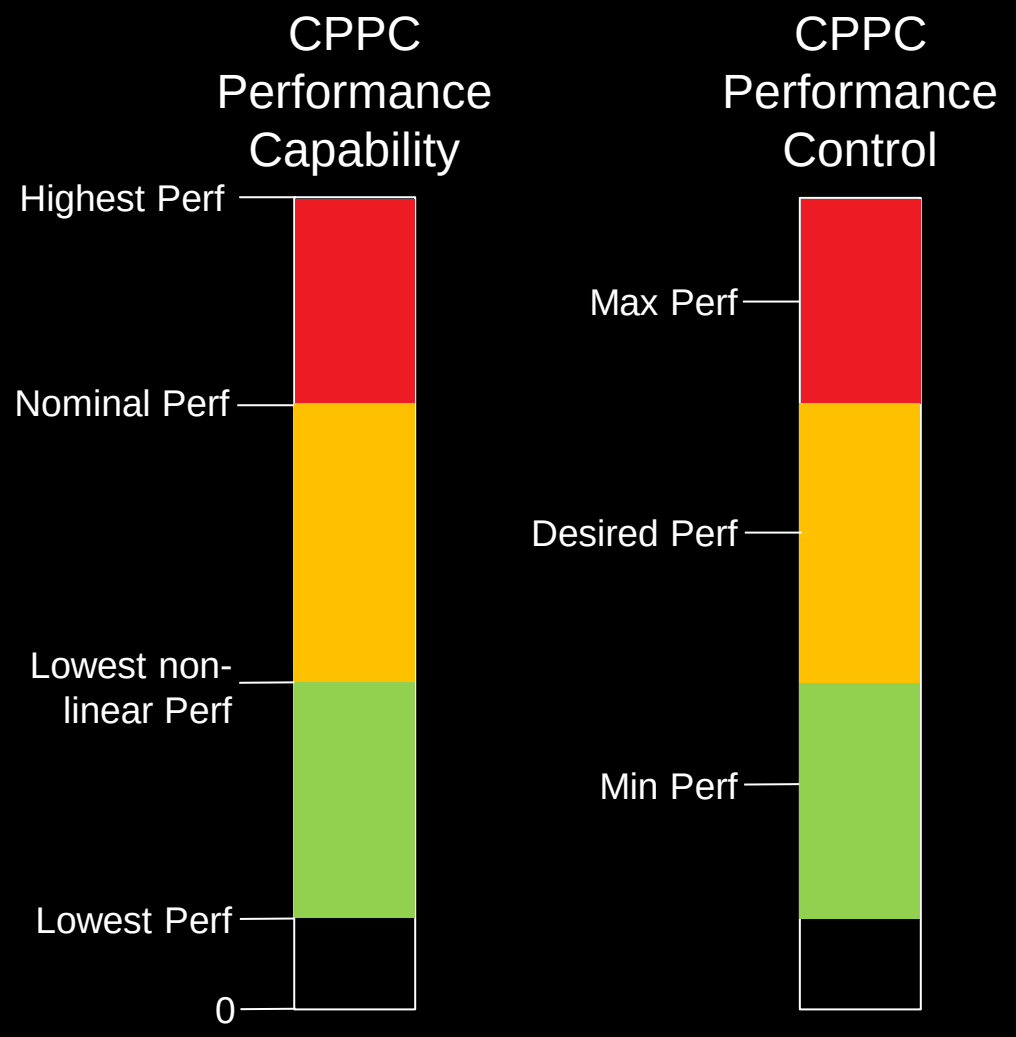
New AMD CPUFreq Design Proposal

- CPU SMU Firmware and SBIOS ACPI CPPC
  - SMU Cclk DPM - Sampling C0 residency (CPU Idle)
  - SBIOS ACPI CPPC Tables
- CPU MSR is the backend of above ACPI CPPC APIs
  - 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 CPUFreq Driver for AMD Processors
- **Implement a new AMD P-State driver instead of ACPI CPUFreq driver on AMD platform**
  - Use fine grain CPPC frequency range instead of ACPI 3 P-State 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 by default, and leverage kernel governor to manage the hints to SMU Cclk DPM Arbiter and then calculate the target frequency

CPPC Performance Capability & Control

- Highest Performance (RO)
- Nominal (Guaranteed) Performance (RO)
- Lowest non-linear Performance (RO)
- Lowest Performance (RO)

- Minimum Requested Performance (RW)
- Maximum Requested Performance (RW)
- Desired performance target (RW)
- Energy Performance Preference (EPP) (RW)



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
  - It aims at better integration with the Linux® kernel scheduler. Load estimation is achieved through the scheduler's Per-Entity Load Tracking (PELT) mechanism, which also provides information about the recent load. This governor currently does load based DVFS only for tasks managed by CFS. RT and DL scheduler tasks are always run at the highest frequency

[Public]

New AMD P-State Driver Stack

New Driver Flow
Legacy Driver Flow

User Space

CPUPower
- Library
  - ACPI CPPC Sysfs APIs
  - CPUFreq Sysfs APIs
  - CPU Topology Sysfs APIs
- CPUPower Utility
  - Helper Functions
  - AMD P-State Specific Helpers
  - AMD P-State Operation Handling
  - Freq Operations (info/set)

X86 Power Tools
- Intel P-State Tracer
- AMD P-State Tracer

Kernel Selftest Framework
- AMD P-State Unit Test
- UT Test Script
- AMD P-State UT Test Kernel Module

Linux® Kernel

Kernel CPU Scheduler
- Schedutil GOV (SUGOV)
  - Register SUGOV
  - SUGOV Kthread
  - SUGOV SYSFS
- CPU utilization changes
  - Completely Fair Scheduling (CFS) Class
  - Real-Time Scheduling Class
  - Deadline Scheduling Class

CPU Freq Subsystem
- Governors
  - Ondemand
  - Powersave
  - Performance
  - Target Governor: Schedutil (Outside)
- CPU Freq Core
  - Sysfs Interfaces
  - Core
  - Frequency Table
- CPU Freq Drivers
  - ARM CPU Freq
  - Intel P-State
  - ACPI CPU Freq
  - AMD P-State (CPPC Based)
    - Sysfs Attributes
    - AMD CPU Data
    - AMD Freq QoS Request
    - Processor Identifier
    - AMD P-State Trace Event
    - P-State and CPPC Static Callbacks
    - AMD P-State Driver Instance

Legacy ACPI Based Dynamic Frequency Control

Use CPU MSR mailbox to do the CPPC frequency control

ACPI Subsystem
- Processor Idle Driver
- Control C0 Residency (enter/exit CPU idle)
- ACPI CPPC Handling
- ACPI general query and control for CPPC interfaces
- ACPI Methods

CPU Idle Time Subsystem
- Governors
  - menu
  - ladder
  - teo
  - haltpoll
- CPU Idle Core
  - cpuidle core
  - driver core
  - governor core
- CPU Idle Driver
  - ACPI Idle (Outside)
  - MIPS Idle
  - ARM SOC Idle
  - PowerPC Idle

SBIOS

ACPI BIOS Interfaces

SMU Firmware
- Arbiter
- CPU Clock DPM
- CPPC request control
- CPPC Capabilities

HW

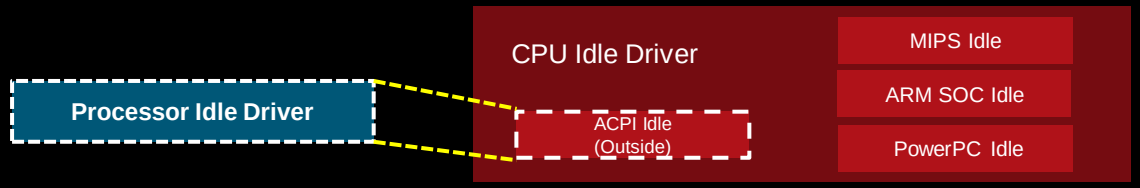
CPPC MSR

Processor

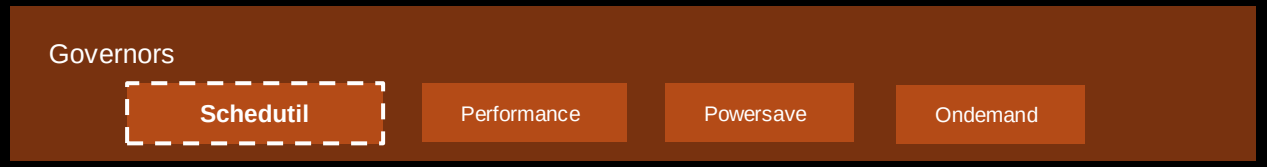
SMU Firmware will detect the CPPC request hint from OS driver.

Existing Kernel Component

- Leverage ACPI Processor Idle driver to control the C0 residency
  - AMD CPU uses an MWAIT/FFH style transition to set the CPU from C0 to C1
  - Linux® Kernel provides the CPU Idle framework to manage the timeline that when CPU will enter C1 and back to C0
  - C0 residency is the ***Activity***$_n$ as the key impact factor of final target frequency

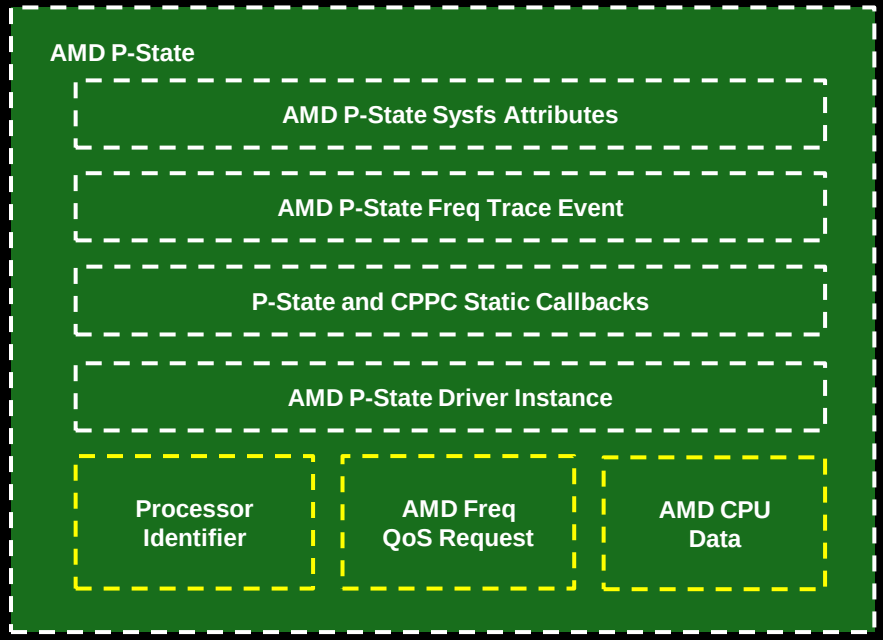


- Leverage and modify the governor “Schedutil” to set it as default policy for AMD P-State driver
  - SMU frequency control needs to sample the C0 or Idle residency to inference the target frequency
  - “Schedutil” can predict the workload directly and manage the CPPC performance with SMU firmware by new AMD P-State driver. This way is more efficiency



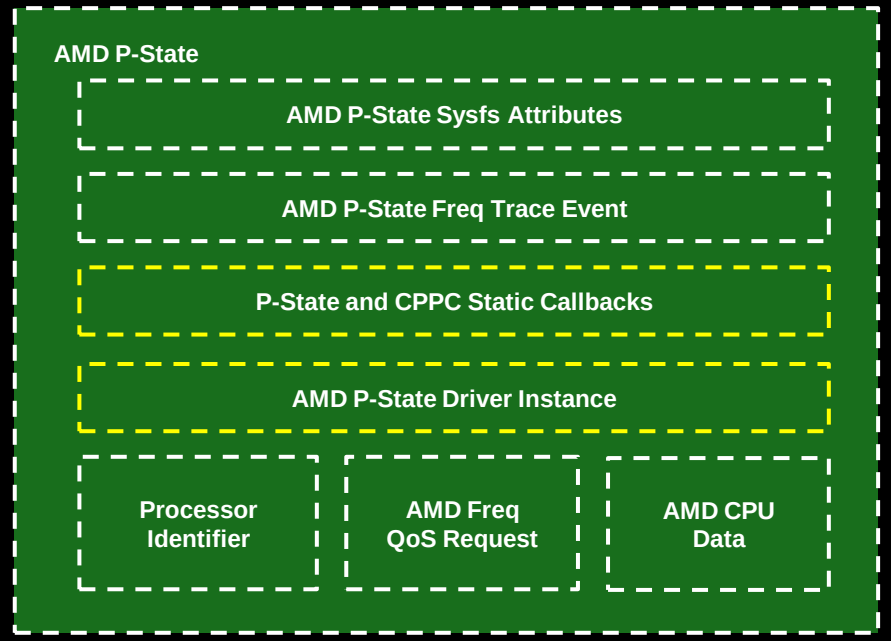
New AMD P-State Core Design and Implementation (1)

- **Introduce a new CPUFreq Kernel Module: “AMD-PSTATE”**
  - Manage the CPU frequency and performance via AMD CPPC MSR API with multiple Linux® kernel governors. Following is Component Design
  - Processor Identifier
    - Check ACPI _CPC Object present in SBIOS
    - Identify CPPC MSR feature with bit 27 of *CPUID Fn80000008_EBX*
  - AMD CPU Data & Module Param
    - Implement a *struct amd_cpudata* to store AMD specific information as the private data for each cpu core
  - AMD Freq QoS (Quality of Service) Request
    - Implement *struct freq_qos_request* instances for *freq_constraints* set the limit range between maximum and minimum frequency to register into the Linux® PM QoS framework
  - AMD CPU Freq reStructuredText (kernel documentation)
    - Implement amd-pstate CPU performance scaling driver RST documentation
    - https://www.kernel.org/doc/html/latest/admin-guide/pm/amd-pstate.html



New AMD P-State Core Design and Implementation (2)

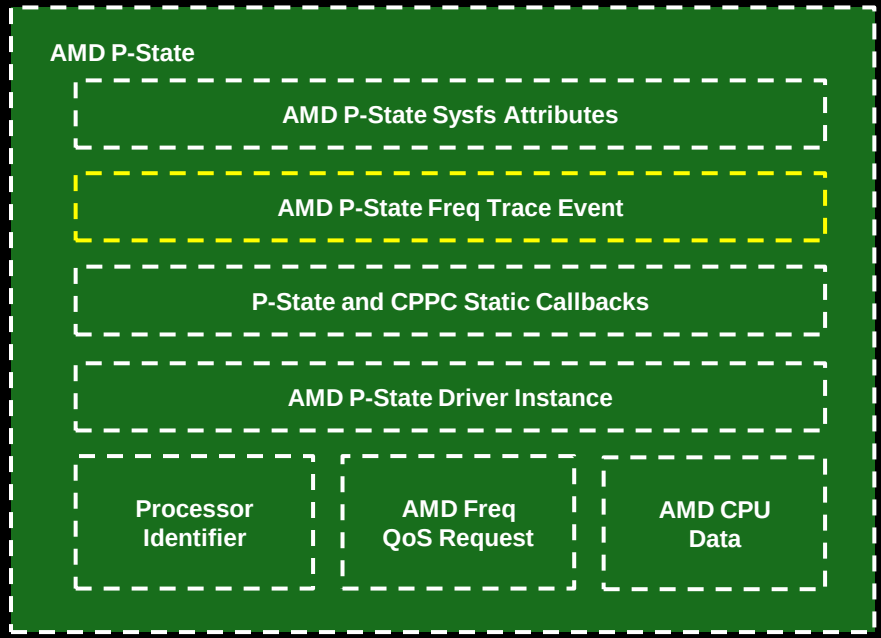
- **Introduce a new CPU Freq Kernel Module: "AMD-PSTATE"**
  - AMD P-State Static Callbacks (MSR and Shared Memory)
    - Use static call APIs to Implement CPPC register helpers
      - MSR Register Operations
      - Shared Memory Operations with ACPI CPPC library
    - Implement *target* and *adjust_perf* method to support Ondemand and Schedutil governors
  - AMD P-State Driver Instance
    - Implement and initialize the *amd_pstate_driver* instance
    - Implement all callback functions of *struct cpufreq_driver* structure
    - Implement *target* and *adjust_perf* method to support Ondemand and Schedutil governors
  - Future Work (under planning)
    - Resolve the performance and power issues on multiple benchmarks
    - Customize the different management policy for specific product such as chrome book
    - Implement the support of EPP (Energy Performance Preference)
    - Implement the support of Preferred Core
    - Implement the support of Fast CPPC



AMD P-State Trace Event Design and Implementation

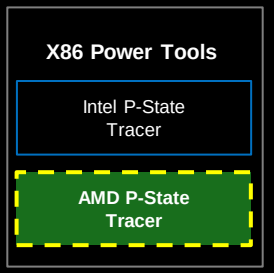
- Two Trace Events for Diagnostics and Tunning
  - *cpu_frequency* trace event from CPUFreq core
  - *amd_pstate_perf* trace event specific to AMD P-State driver

```
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       [015] dN...  4995.979886: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=15 changed=false fast_switch=true
         <idle>-0       [007] d.h..  4995.979893: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=7 changed=false fast_switch=true
            cat-2161    [000] d....  4995.980841: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=0 changed=false fast_switch=true
          sshd-2125     [004] d.s..  4995.980968: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=4 changed=false fast_switch=true
         <idle>-0       [007] d.s..  4995.980968: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=7 changed=false fast_switch=true
         <idle>-0       [003] d.s..  4995.980971: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=3 changed=false fast_switch=true
         <idle>-0       [011] d.s..  4995.980996: amd_pstate_perf: amd_min_perf=85 amd_des_perf=85 amd_max_perf=166 cpu_id=11 changed=false fast_switch=true
```

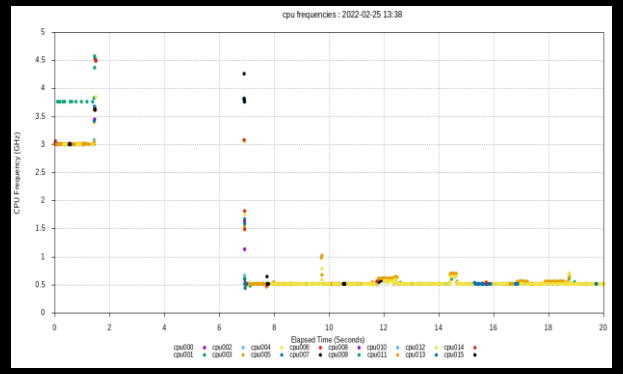


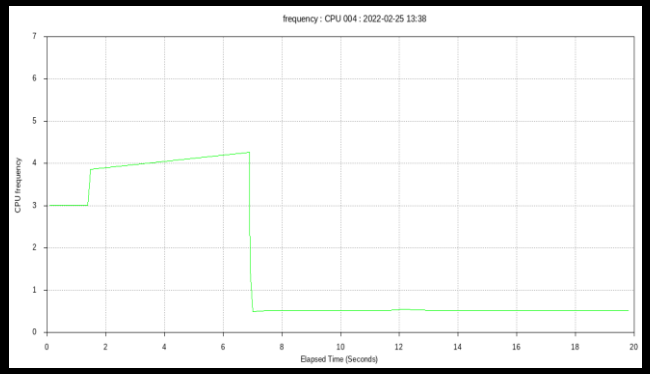
AMD P-State Tracer Tool Support

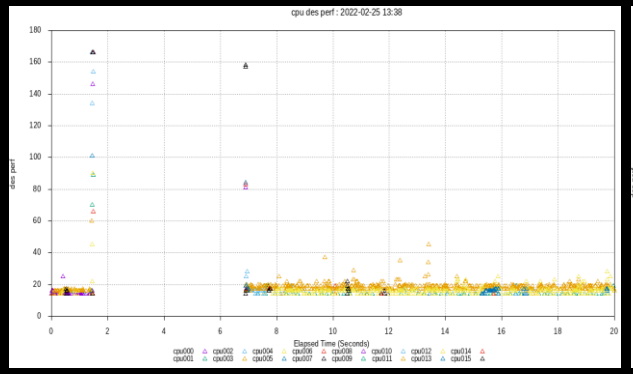
- AMD P-State Tracer
  - *amd_pstate_tracer.py* is a python script which can record and parse the trace log
    - According to the AMD P-State trace event, this tool generates performance plots
    - It is used for debugging and tuning the performance of AMD P-State driver
    - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/tools/power/x86/amd_pstate_tracer/amd_pstate_trace.py

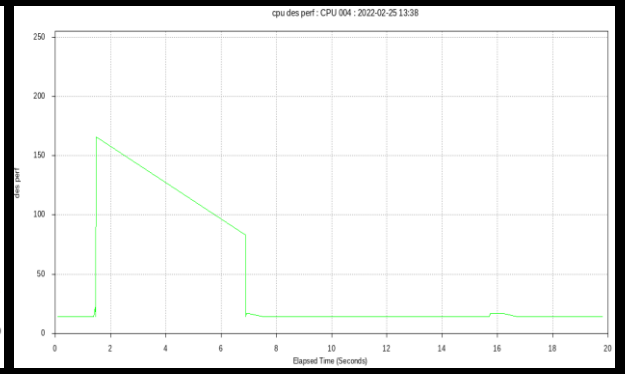


common_cpu	common_secs	common_usecs	min_perf	des_perf	max_perf	freq	mperf	apef	tsc	load	duration_ms	sample_num	elapsed_time	common_comm
CPU_005	712	116384	39	49	166	0.7565	9645075	2214891	38431470	25.1	11.646	469	2.496	kworker/5:0-40
CPU_006	712	116408	39	49	166	0.6769	8950227	1839034	37192089	24.06	11.272	470	2.496	kworker/6:0-1264









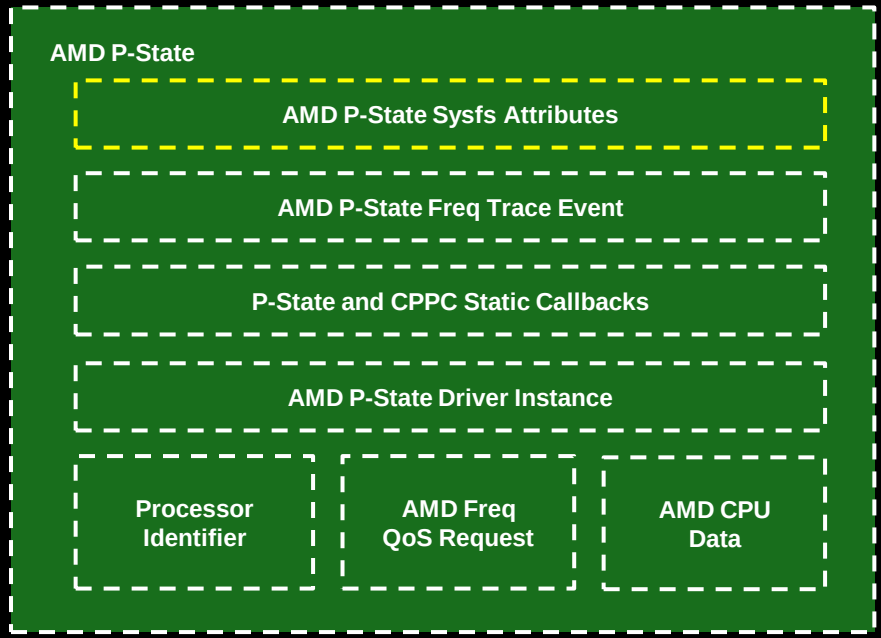
AMD P-State Sysfs Attributes Design and Implementation

- 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_max_freq
```

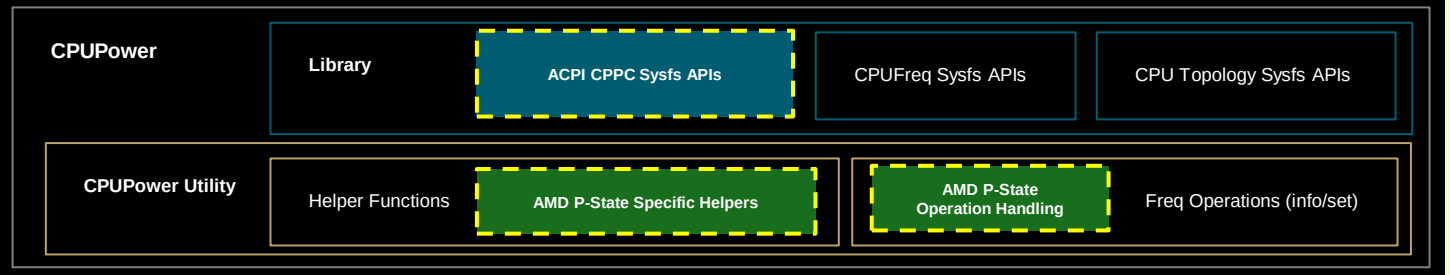
- *amd_pstate_highest_perf*
- *amd_pstate_lowest_nonlinear_freq*
- *amd_pstate_max_freq*
- *lowest_freq (cppc)*
- *lowest_perf (cppc)*
- *…*

```
$ ls -lR  /sys/devices/system/cpu/cpu0/acpi_cppc/
/sys/devices/system/cpu/cpu0/acpi_cppc/:
total 0
-r--r--r-- 1 root root 65536 Mar  5 19:38 feedback_ctrs
-r--r--r-- 1 root root 65536 Mar  5 19:38 highest_perf
-r--r--r-- 1 root root 65536 Mar  5 19:38 lowest_freq
-r--r--r-- 1 root root 65536 Mar  5 19:38 lowest_nonlinear_perf
-r--r--r-- 1 root root 65536 Mar  5 19:38 lowest_perf
-r--r--r-- 1 root root 65536 Mar  5 19:38 nominal_freq
-r--r--r-- 1 root root 65536 Mar  5 19:38 nominal_perf
-r--r--r-- 1 root root 65536 Mar  5 19:38 reference_perf
-r--r--r-- 1 root root 65536 Mar  5 19:38 wraparound_time
```



CPUPower Library and TurboStat 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-State performance capability into the library of cpupower
      - Add the ACPI CPPC library support in CPUPower
      - Implement the frequency operation control into cpupower utilities for AMD P-State




```
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.
```

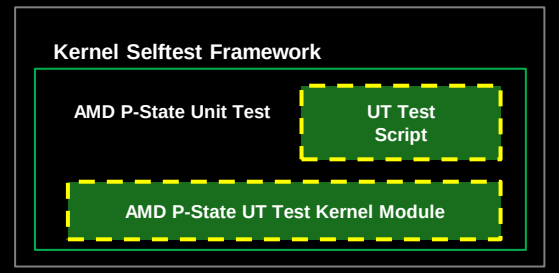
- TurboStat
  - turbostat can display the frequency, power consumption, idle status and other statistics of the modern Intel and AMD CPUs
    - Implement the support of new amd-pstate module which expose the sysfs API on boost state control
      - Leverage the CPPC API to implement the maximum performance working behavior

AMD P-State Unit Test Support

- AMD P-State UT (Unit Test)
  - amd-pstate-ut is a test kernel module to test AMD P-State functionalities
  - It's for verifying the support of SBIOS/Firmware on existing AMD CPUs
  - Add more functional and performance tests for performance and power optimization

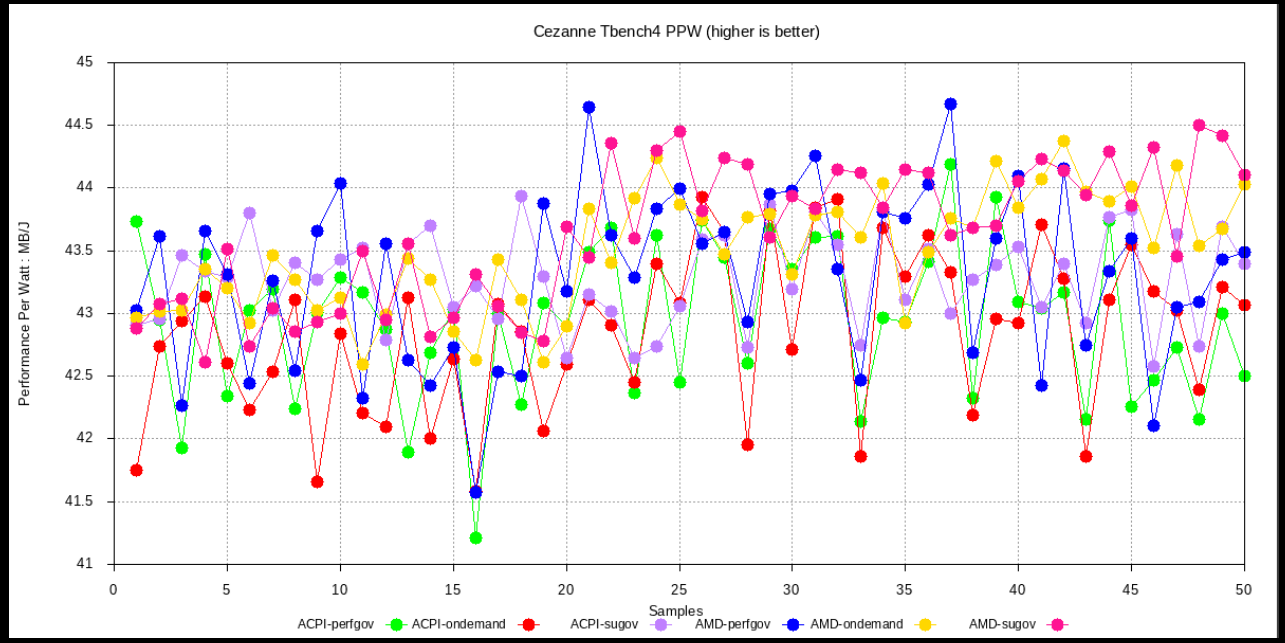
```
jasmine@jasmine-MayanDAP-RMB:~/amd-brahma/linux$ make -C tools/testing/selftests install INSTALL_PATH=~/kselftest
jasmine@jasmine-MayanDAP-RMB:~$ sudo ./kselftest/run_kselftest.sh -c amd-pstate
TAP version 13
1..1
# selftests: amd-pstate: amd-pstate-ut.sh
# amd-pstate-ut: ok
ok 1 selftests: amd-pstate: amd-pstate-ut.sh
```

```
jasmine@jasmine-MayanDAP-RMB:~$ dmesg | grep "amd-pstate-ut" | tee log.txt
[76697.480217] amd-pstate-ut: loaded.
[76697.480222] amd-pstate-ut: ****** Begin 1            acpi_cpc_valid          ******
[76697.480227] amd-pstate-ut: ****** End   1            acpi_cpc_valid          ******
[76697.480228] amd-pstate-ut: ****** Begin 2            check_enabled           ******
[76697.480253] amd-pstate-ut: ****** End   2            check_enabled           ******
[76697.480255] amd-pstate-ut: ****** Begin 3            check_perf              ******
[76697.480554] amd-pstate-ut: ****** End   3            check_perf              ******
[76697.480556] amd-pstate-ut: ****** Begin 4            check_freq              ******
[76697.480558] amd-pstate-ut: ****** End   4            check_freq              ******
[76697.480559] amd-pstate-ut: all 4 tests passed
[76697.482507] amd-pstate-ut: unloaded.
```



TBench CPU Benchmark for AMD P-State vs ACPI CPUFreq

- AMD P-State 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

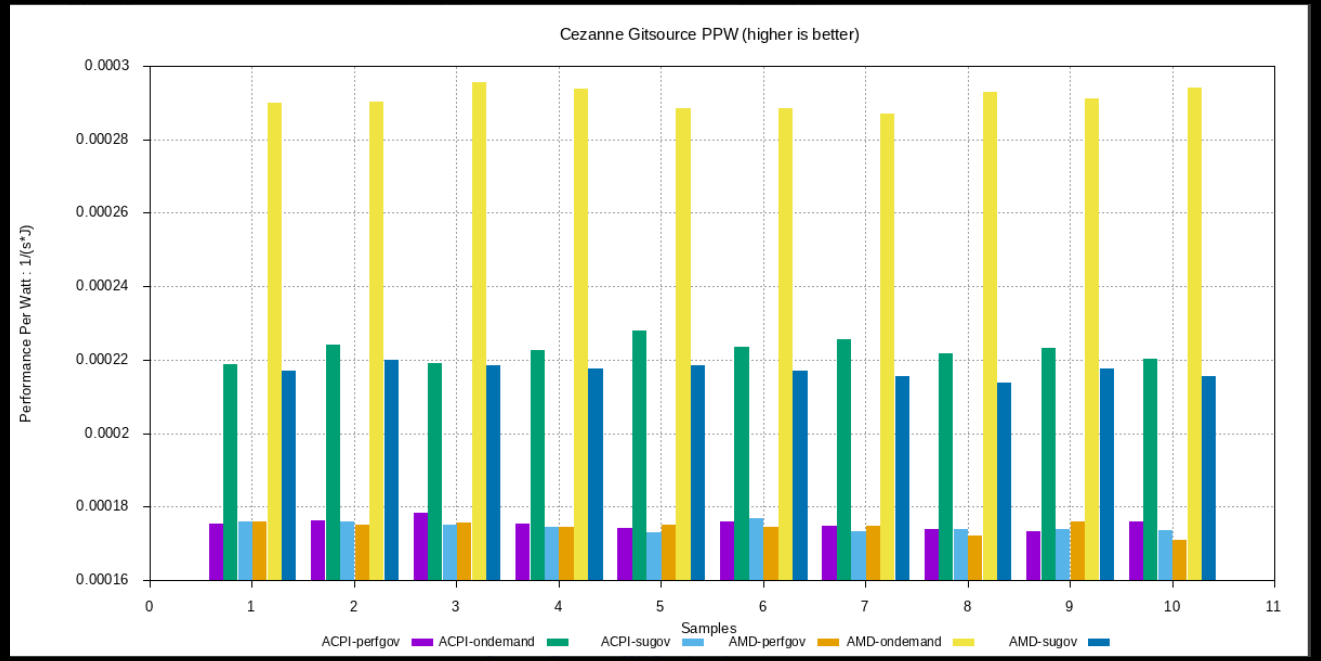


TBench4

Kernel Module	Schedutil			Ondemand			Performance		
	Performance Unit: MB/S	Energy Unit: J	Performance Per Watt Unit: MB/J	Performance Unit: MB/s	Energy Unit: J	Performance Per Watt Unit: MB/J	Performance Unit: MB/S	Energy Unit: J	Performance Per Watt Unit: MB/J
acpi-cpufreq	2214	9074	48.5662	2221	9039	48.89	2214	9218	47.8084
amd-pstate	2207	9080	48.383 (- 0.4%)	2182	9167	47.3753 (- 3.1%)	2228	9093	48.7735 (+ 2.0%)

Gitsource CPU Benchmark for AMD P-State vs ACPI CPUFreq

- AMD P-State 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

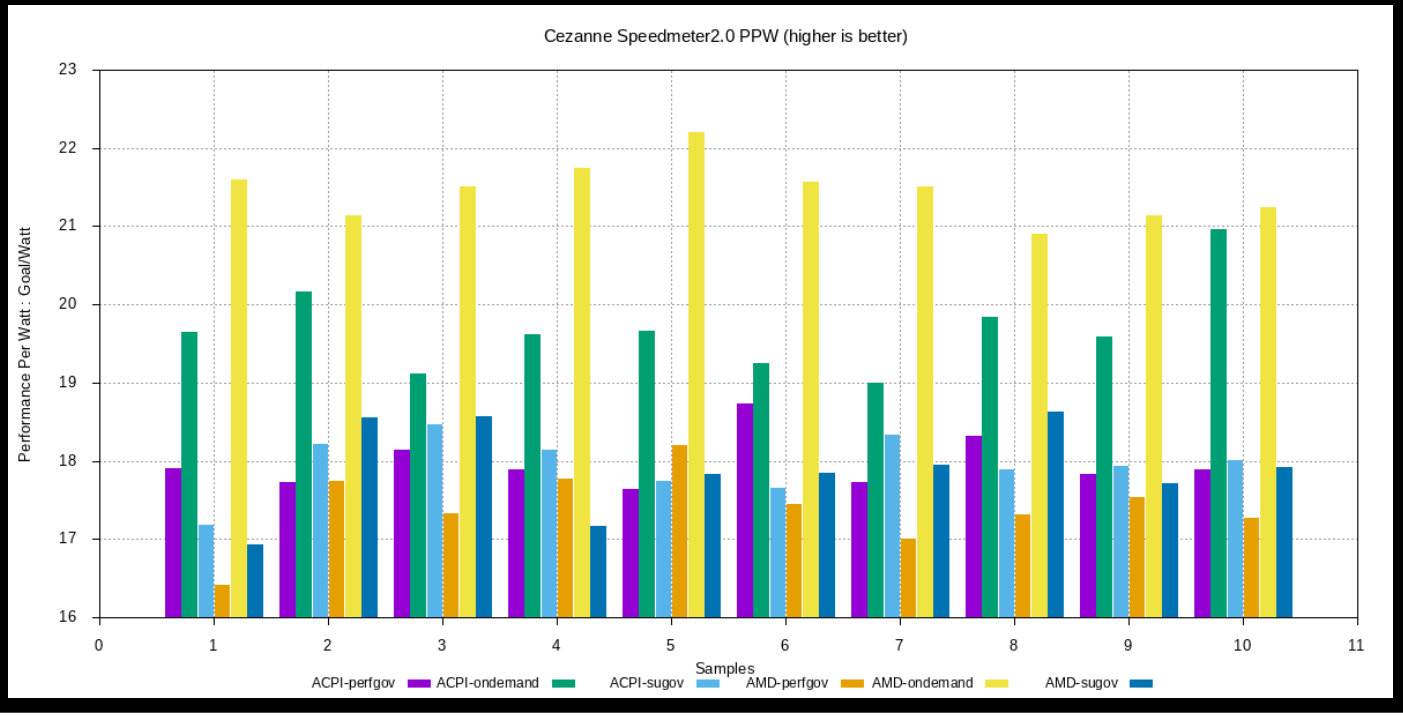


Gitsource

Kernel Module	Schedutil			Ondemand			Performance		
	Performance (Time, low is better)	Energy Unit: J	Performance Per Watt Unit:1/(s*watt)	Performance (Time, low is better)	Energy Unit: J	Performance Per Watt Unit:1/(s*watt)	Performance (Time, low is better)	Energy Unit: J	Performance Per Watt Unit:1/(s*watt)
acpi-cpufreq	360.2185517	5728.061311	0.000174579	501.1471876	4491.407676	0.000222647	357.112983	5732.135085	0.000174455
amd-pstate	401.0577597	4606.293384	0.000217094 (+ 24.35%)	578.8245807	3435.134909	0.000291109 (+ 30.75%)	353.7065307	5701.750577	0.000175385 (+ 0.53%)

Speedometer CPU Benchmark for AMD P-State vs ACPI CPUFreq

- AMD P-State 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

Kernel Module	Schedutil				Ondemand				Performance			
	Performance (Goal, higher is better)	Time Unit: S	Energy Unit: J	Performance Per Watt Unit:Goal/Watt	Performance (Goal, higher is better)	Time Unit: S	Energy Unit: J	Performance Per Watt Unit:Goal/Watt	Performance (Goal, higher is better)	Time Unit: S	Energy Unit: J	Performance Per Watt Unit:Goal/Watt
acpi-cpufreq	194.5942439	69	747.8670161	17.95373047	189.3893419	75	721.8567372	19.67731256	195.1934774	68	738.30419	17.97789671
amd-pstate	188.2814733	68	715.0619	17.9049398 (- 0.27%)	180.6754877	81	682.2303	21.45128302 (+ 9.02%)	194.500182	68	760.2146	17.39773457 (- 3.23%)

The Best is Yet to Come

- Improvement from legacy acpi-cpufreq to amd-pstate solution
- Linux® Kernel Upstream
  - AMD P-State Driver – Kernel 5.17
  - CPUPower Support – Kernel 5.18
  - AMD P-State Tracer – Kernel 5.18
  - AMD P-State Unit Tests – Kernel 6.1
  - EPP / Preferred Core support – In progress
    - Below is our working git repo for upstream
    - https://git.kernel.org/pub/scm/linux/kernel/git/rui/linux.git/
- Challenges:
  - Resolve the performance and power issues on multiple benchmarks
    - Resolve the Shared Memory performance per watt drop which compared with ACPI CPUFreq driver
  - Cover all existing recent CPUs with different types including mobile, desktop, etc
  - Optimize the power on steam games with Proton (Vulkan to D3D)

Reference

- AMD P-State Kernel Documentation
  - https://www.kernel.org/doc/html/latest/admin-guide/pm/amd-pstate.html
- Open Source Summit Europe 2022
  - https://sched.co/15yzz
- Linux® Plumbers Conference 2022
  - https://lpc.events/event/16/contributions/1258/
- X.Org Developer Conference 2021 (initial proposal)
  - https://indico.freedesktop.org/event/1/contributions/5/
- CPUFreq and The Scheduler
  - http://events17.linuxfoundation.org/sites/events/files/slides/cpufreq_and_scheduler_0.pdf
- 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

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