Fine Grain Frequency Control with Kernel Governors

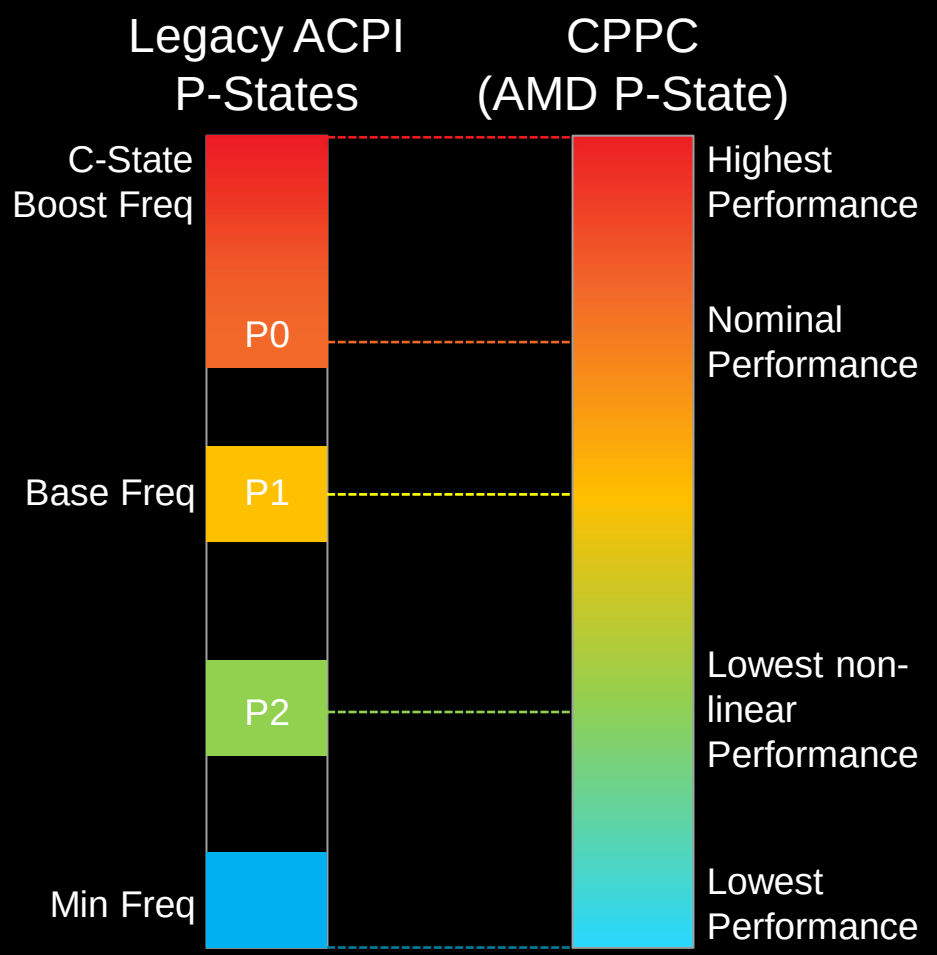
Ray Huang (Huang Rui) <ray.huang@amd.com>

AMD
together we advance_

Background

- Traditional ACPI P-States
  - C-State Boost
  - P0, P1, P2

- Collaborative Processor Power Control (CPPC) - Fine grain performance range
  - Highest Performance
  - Nominal Performance
  - Lowest non-linear performance
  - Lowest performance



ACPI CPUFreq vs AMD P-State

- ACPI CPUFreq
  - Using in traditional AMD CPUs only switching in 3 P-States

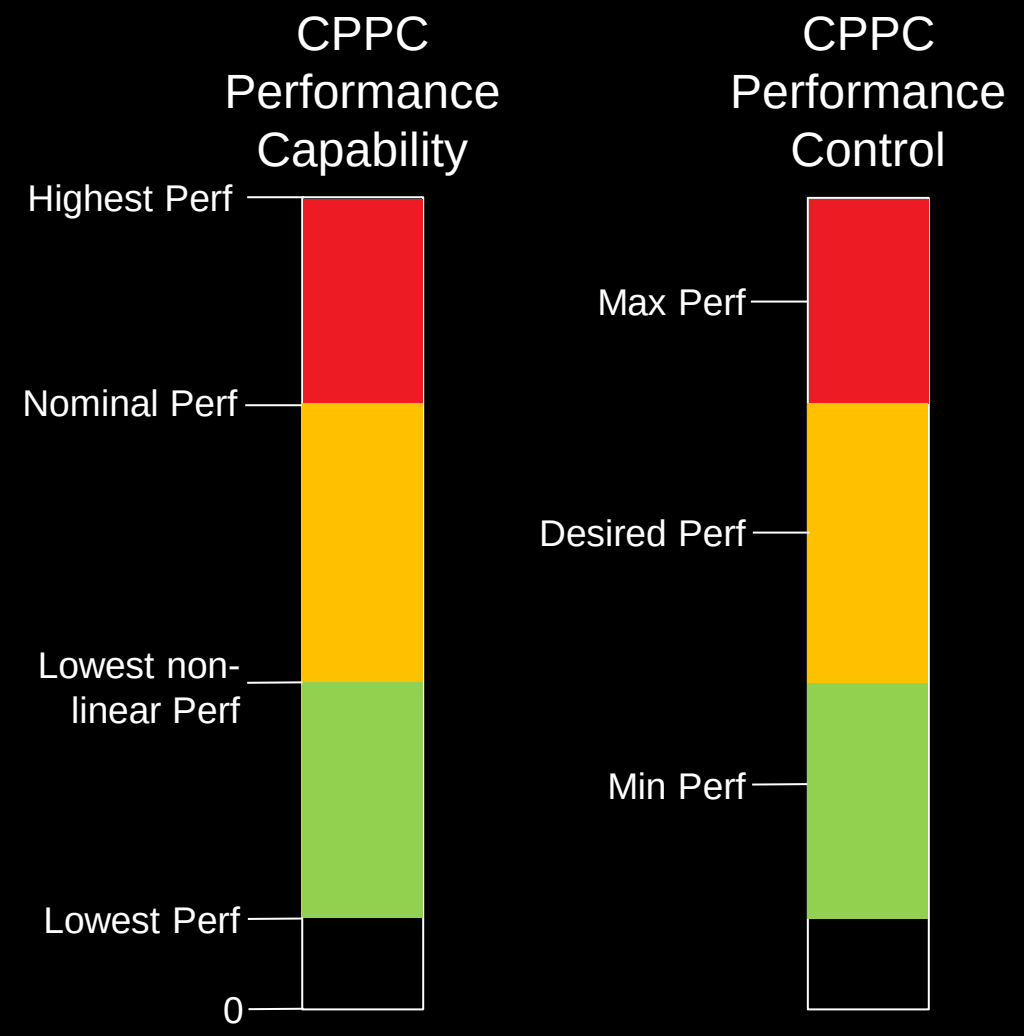
```
Name (_PSS, Package (0x03)  // _PSS: Performance Supported States
    {
      Package (0x06)
      {
        0x000009C4,
        0x00000ABE,
        0x00000000,
        0x00000000,
        0x00000000,
        0x00000000
      },
      Package (0x06)
      {
        0x0000085C,
        0x00000834,
        0x00000000,
        0x00000000,
        0x00000001,
        0x00000001
      },
      Package (0x06)
      {
        0x00000708,
        0x00000654,
        0x00000000,
        0x00000000,
        0x00000002,
        0x00000002
      }
    })
```

- AMD P-State
  - Supported on partial of Zen2, Zen3, and future CPUs
  - Full MSR Solution
    - New version of CPPC on recent Zen processors
      - MSR_AMD_CPPC_CAP1
      - MSR_AMD_CPPC_ENABLE
      - MSR_AMD_CPPC_CAP2
      - MSR_AMD_CPPC_REQ
      - MSR_AMD_CPPC_STATUS
  - Shared Memory Solution
    - First version of CPPC on old Zen processors

```
Name (_CPC, Package (0x17)  // _CPC: Continuous Performance Control
    {
      0x17,
      0x03,
      ResourceTemplate ()
      {
        Register (PlatformCommChannel,
          0x20,             // Bit Width
          0x00,             // Bit Offset
          0x0000000000000000, // Address
          ,)
      },
    ......
```

Fine Grain Performance Control

- AMD P-State is fine grain performance control with CPPC + kernel governors
  - CPPC Performance Capability
    - Highest / Nominal / Lowest non-linear / Lowest Perf
  - CPPC Performance Control
    - Max / Desired / Min Perf
  - Support governors
    - **Schedutil / Ondemand** / Conservative / Performance / Powersave
- **Performance Issue on Shared Memory CPUs**
  - https://bugzilla.kernel.org/show_bug.cgi?id=215135
  - ACPI P-State vs AMD P-State (discussion?)
    - “shared memory” processors uses a system memory mailbox mechanism to implement the fine grain performance control is not as good as “actual MSR”
    - However, in this kind of processors, the legacy ACPI P-State control in _PSS object is “actual MSR” which is faster than “share memory” with CPPC
    - **How to enhance or optimize the kernel to improve “share memory” support? – Discussion**



Energy Performance Preference

- What is Energy Performance Preference (EPP)
  - Provide a hint to hardware if driver wants to bias toward performance (0x0) or energy efficiency (0xff)
  - If EEP is enabled, the desired perf will be inactive
    - Set desired perf as 0 to enable EPP

- Current Solution:
  - Provide 4 OS profiles with different EPP hints which can be controlled by user space and do hardware-based dynamic frequency management
    - Performance (0x0)
    - Balance performance (0x80)
    - Balance powersave (0xBF)
    - Powersave (0xFF)

- How to manage max perf / min perf / epp hint with kernel governor? – Discussion
  - Linux® kernel doesn’t have the management for max/min perf.

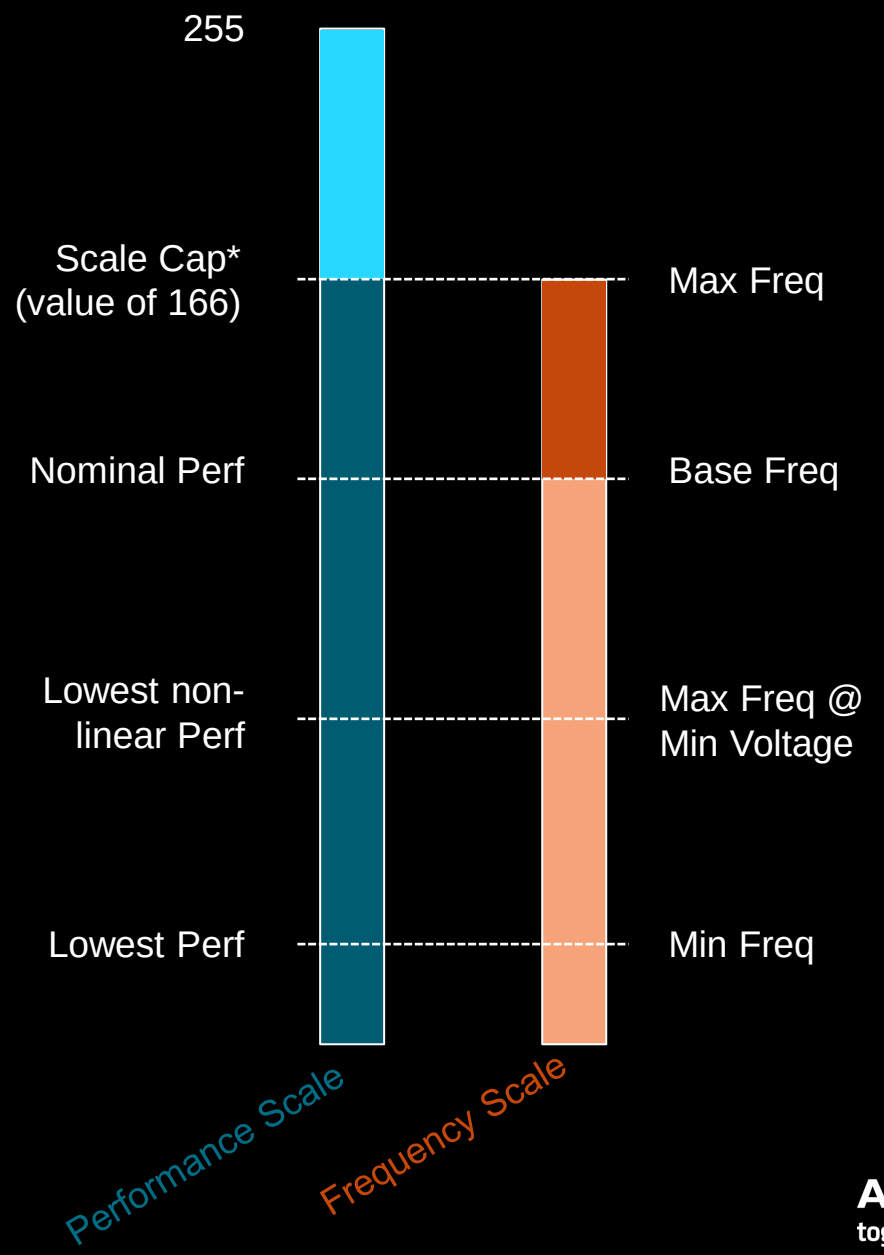
Preferred Core

- What is Preferred Core
  - Growing number of cores + Chiplet -> A wider range of frequency (Scale Cap to 255)
  - Needs an algorithm that characterizes the capabilities of the cores under various system parameters and generates a list of cores in an order of preference

Core 6 is preferred

0	1	2	3	4	5	6	7	8	9	10	11
122	231	236	221	201	191	241	186	181	176	171	166

  - Region between Scale Cap and 255 is used for communicating core ordering with CPPC highest performance
- How to design the support for Preferred Core in Linux® kernel? – Discussion
  - How about leveraging cpu capacity approach?
    - arch_scale_cpu_capacity



More Introduction

- The following detail introduction on LinuxCon @ Open Source Submit 2022 - Europe
  - https://sched.co/15yzz
- Kernel documentation
  - https://www.kernel.org/doc/html/latest/admin-guide/pm/amd-pstate.html
- Initial proposal presentation last year in XDC2021
  - https://indico.freedesktop.org/event/1/contributions/5/

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