Technology Guide

Intel® Speed Select Technology - Core Power (Intel® SST-CP) Overview

Authors

Anatoly Burakov

Chris MacNamara

Nikhil Gupta

Srinivas Pandruvada

Vasudevan Srinivasan

Shibin Koikkara Reeny

1 Introduction

The advent of Network Function Virtualization introduced a new category of workloads running on standard servers. Such workloads adhere to strict service-level agreements and strive to achieve a high quality of experience. This means delivering deterministic, low-latency performance, where increased jitter and latency can lead to longer packet delay variation causing impacts to network subscribers.

The Intel® Speed Select Technology - Core Power (Intel® SST-CP) is available in selected models of the most recent generation of Intel processors. This guide describes how Intel SST-CP offers dynamic prioritization of CPU core power/performance by assigning priority to each CPU core, thereby satisfying the power requirements of each core in a priority order. This feature is part of the power management technology and toolbox of features that Intel provides in the CPU, allowing user control where power is applied to workloads responsible for meeting SLAs. Several use cases are presented to demonstrate the application of the technology and detailed steps are provided for reference.

This document also describes the usage of the Linux* kernel tool that was developed specifically to configure Intel SST-CP technology on a platform. The tool provides a convenient, easy-to-use interface that aids configuration and hides complexity.

This technology guide is intended for equipment manufacturers, communication service providers, and engineers who are planning and deploying workloads on the latest Intel® Xeon® Scalable processors.

This document is part of the Network & Edge Platform Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	April 2021	Initial release.
002	March 2023	Added Appendix B, Data Collection Tools.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
EPP	Energy Performance Preference
HWP	Hardware P-states
Intel® SST-BF	Intel® Speed Select Technology - Base Frequency (Intel® SST-BF)
Intel® SST-CP	Intel® Speed Select Technology - Core Power (Intel® SST-CP)
LSB	Least Significant Byte
MSR	Machine Specific Register
QoE	Quality of Experience
TDP	Thermal Design Power

1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
Intel® Speed Select Technology (Intel® SST)	https://www.intel.com/content/www/us/en/architecture-and-technology/speed-select-technology-article.html
Intel's `intel-speed-select` tool GitHub Repository	https://github.com/spandruvada/intel-speed-select-utility-src-packages

2 Overview

This guide introduces a new hardware-based CPU power prioritization mechanism that was introduced in the 3rd Generation Intel Xeon Scalable processor, named Intel Speed Select Technology - Core Power.

The mechanism works by assigning frequency bins according to requests and any surplus power available. The prioritization mechanism is illustrated in Figure 1.

2.1 Core Performance Prioritization and Existing Solutions

In telecommunications, a common platform configuration is one where specific workloads are pinned to specific CPU cores. The pinning of high-priority workloads (often data plane workloads) to specific CPU cores has several advantages. It reduces overhead and minimizes workload interruptions due to OS scheduling. The isolation of a CPU core, and other factors, leads to improved performance.

Traditionally however, there has been no notion of priority among CPU cores. All CPU cores are considered equal and have equal access to the CPU power. Currently when CPU cores are heavily loaded, the CPU equally distributes available power to the cores resulting in a uniform increase in frequency irrespective of the tasks running on either CPU core. Figure 1 shows this scenario where the Power Control Unit (PCU) distributes surplus power equally to all cores.

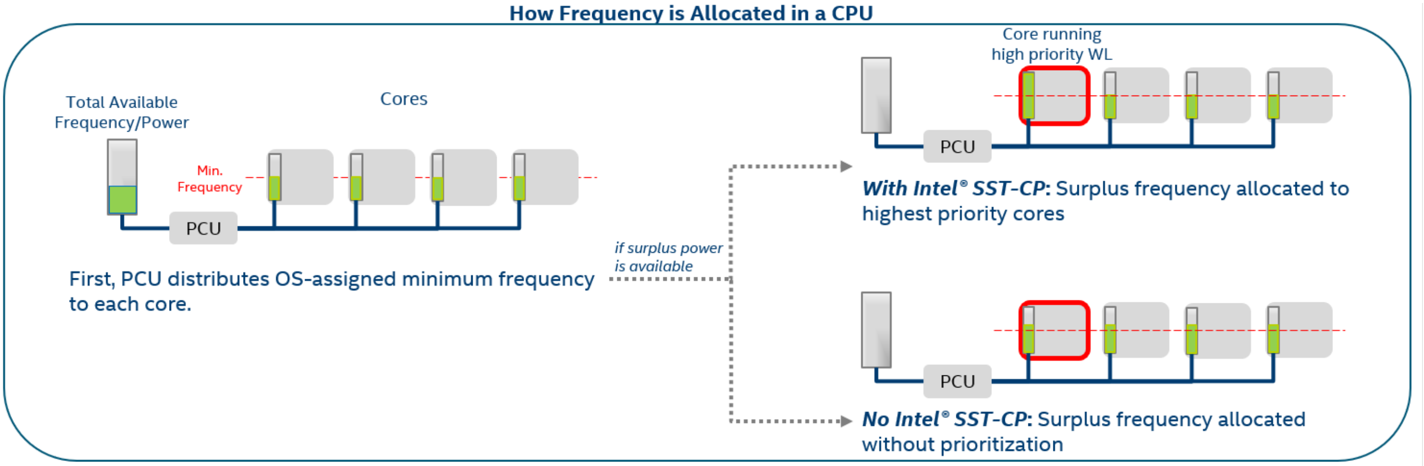
How Frequency is Allocated in a CPU
Total Available Frequency/Power
Cores
Min. Frequency
PCU
First, PCU distributes OS-assigned minimum frequency to each core.
if surplus power is available
Core running high priority WL
PCU
With Intel® SST-CP: Surplus frequency allocated to highest priority cores
PCU
No Intel® SST-CP: Surplus frequency allocated without prioritization

Figure 1. Allocation of Frequency in A CPU

The Intel SST-CP feature allows the user to direct power/frequency to the highest priority cores with the goal of improving performance. In essence, Intel SST-CP allows the user to set a “Power QoS” for cores in a CPU, whereby CPU cores with highest priority get prioritized access to available power by allowing the user to set the frequency high and prioritizing power when power is limited[1]. This in turn benefits the workloads running on those cores through higher frequencies and determinism by setting up the CPU to prioritize power to those cores when there is limited power available. The focus of these examples is on a mix of data plane and control plane workloads. This guide focuses on managing frequency for this category of workloads that typically are low latency packet processing workloads. SST-CP enables flexibility for workloads that benefit from higher base frequency on a subset of the processor’s cores. While the max turbo frequency across the cores remains constant across the cores, a subset of the cores can be assigned as to run at a higher base frequency than specified, while the other cores run at lower base frequency.

[1] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

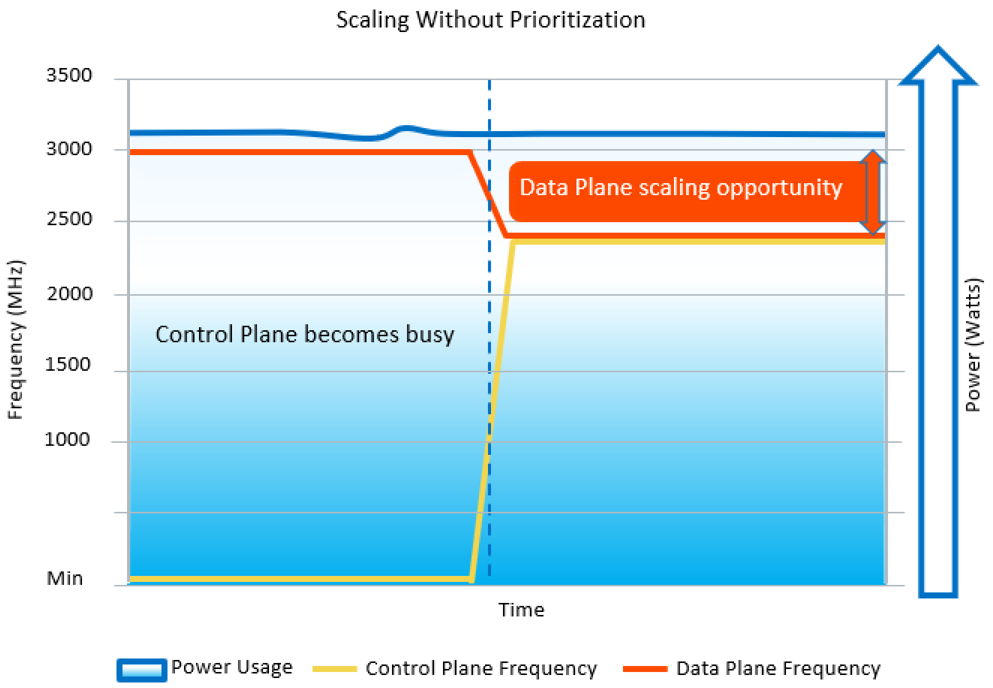

Figure 2. Data Plane Cores Scaled Down When Control Plane Becomes Busy

As an example of a scenario without prioritization, when Intel® Turbo Boost Technology is enabled, power/frequency is distributed uniformly across cores resulting in equal frequency across cores. In the data plane, control plane example we provide (Figure 2), it means when the control plane workload becomes busy, it consumes more power and the workload frequencies converge.

We can mitigate this behavior by using the per core frequency controls available in more recent Intel processors (based on the Haswell architecture and later) to limit the minimum and maximum frequencies for a given CPU core. As shown in Figure 3, we define data plane and control plane as two groups of workloads running on separate cores that have distinct priorities with data plane being the higher priority. However, such a solution requires user tuning to determine the exact frequency to be set in a way that makes best use of the available performance. In addition, dynamically adapting to workload changes (for example, the scale up of control plane cores if data plane cores are not busy) needs to be considered.

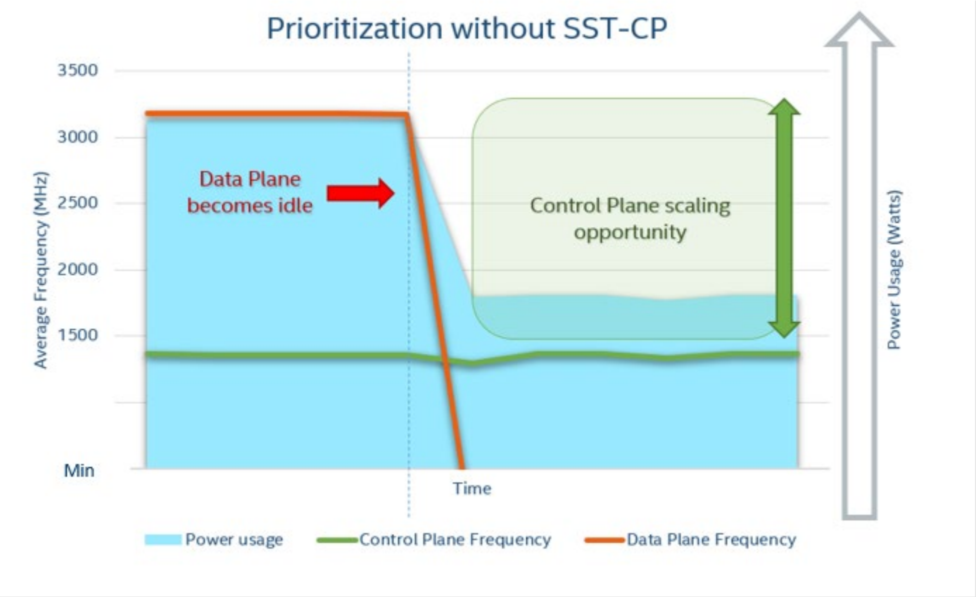

Figure 3. Control Plane Cores Not Scaling Up When Headroom Available

2.2 Core Performance Prioritization Using Intel SST-CP

Intel SST-CP is a new feature introduced in selected SKUs of the 3rd Generation Intel Xeon Scalable processor that aims to address use cases such as those described earlier in this document. The Intel SST-CP feature offers dynamic prioritization of core performance by assigning each core a priority[2].

Intel SST-CP uses a new OS mailbox interface, which is generally supported in Linux* kernel versions 5.8 or later. You can also use it on prior versions of Linux OS versions. For more information, contact your Intel representative.

The new functionality enables the OS to create up to four frequency Class of Service (CLOS) groups and assign each CPU core to one of these groups. This results in CPU cores assigned to a higher priority CLOS group having higher priority for power distribution and consequently receiving more power.

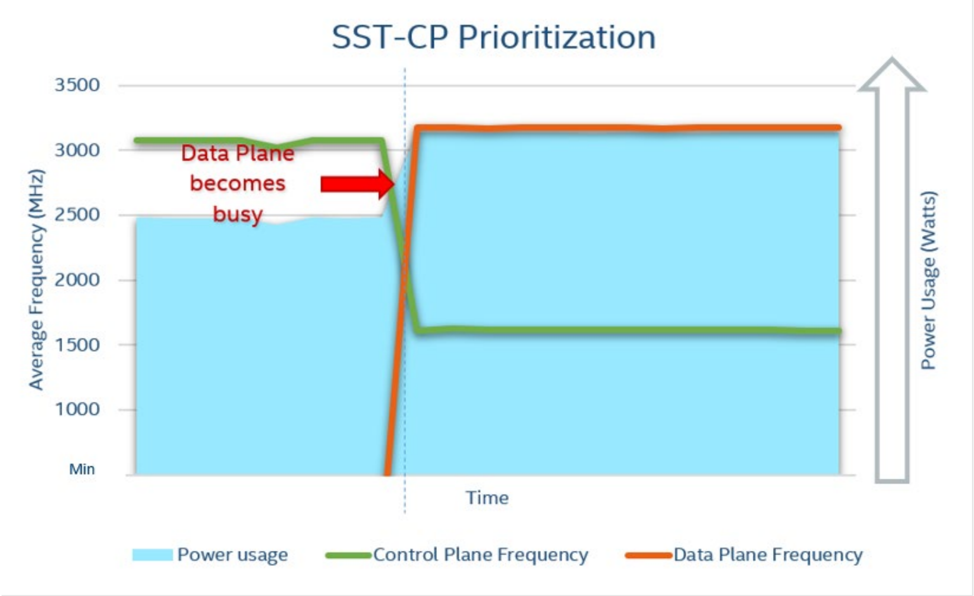

Figure 4. Data Plane Cores Stay at Maximum Even When Control Plane Busy

It is important to note that the prioritization described in this section occurs only when the CPU is power limited. Hence, when a CPU is throttling due to power limits, it applies the prioritization based on the software configuration. When software selects a subset of the cores for prioritization, the CPU throttles the remaining subset of cores before the higher priority cores. Thus, the

[2] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

prioritized cores remain at higher frequency and the workload on these cores may continue unaffected (Figure 4). Intel SST-CP works well with Intel Turbo Boost Technology as it drives to maximum frequency and maximum performance. Therefore, you may be more likely to reach a power limited scenario. With Intel SST-CP, you can protect the workload from negative impacts of throttling when power limited.

While the Intel SST-CP and the Intel® Speed Select Technology – Base Frequency (Intel® SST-BF) may seem like similar technologies, they are considered distinct due to their different usage scenario. The main difference between the two technologies is that Intel SST-CP does not provide any assurances about whether target CPU frequencies can be reached, while Intel SST-BF provides a pre-determined frequency profile. The Intel SST-CP technology is user configurable versus predetermined, allowing the user the flexibility to set the frequency as required. This can be considered an opportunistic mechanism that is similar in nature to Intel Turbo Boost Technology and hardware P-states (HWP).

The Intel hardware P-state technology allows the user to specify certain parameters of operation that are configurable per CPU core. Intel SST-CP can be used as a drop-in replacement for HWP and most of the functionality available in HWP is also available through the Intel SST-CP command-line tool. Whenever Intel SST-CP is enabled through the command-line tool, HWP configuration automatically stops taking any effect.

2.3 Power Usage Considerations

Intel SST-CP technology is aimed at high-performance use cases, and as such may result in higher power consumption if used in cases that are not performance-oriented. This is because Intel SST-CP is designed to operate at power limits and is predicated on the assumption that whatever workload is running on a high-priority core, it requires the highest frequency available at all times.

2.4 Uncore Frequency Considerations

In addition to CPU core frequencies, the Power Control Unit (PCU) also manages the uncore frequency scaling. Frequency of the uncore can go up and down depending on the overall system load, and, like CPU core frequencies, the uncore also has its own base frequency, as well as a Turbo Boost frequency range.

Under certain circumstances, the system may become power limited, resulting in uncore frequency being scaled down. This condition may adversely affect certain workloads in very rare cases. It is advised to not oversubscribe the system when the workloads are particularly sensitive to the uncore frequency.

Intel SST-CP makes no attempt to control the uncore frequency and does not change how the power is distributed between CPU cores and the uncore.

More recently Linux has been enhanced to add support for managing the uncore frequency via a new driver named `intel_uncore_frequency`.

3 Tools Overview

To configure Intel SST-CP technology effectively, a few helpful tools are referenced. This section covers usage and purpose of these tools.

3.1 Intel Speed Select Technology kernel Tool

Starting with Linux kernel version 5.3, an Intel SST technology tool (`intel-speed-select`) is included with kernel source releases. We recommend using the version from Linux kernel 5.8. This tool is aimed at configuring the entire family of Intel SST technologies, including Intel SST-CP technology through the OS mailbox interface. By using this tool, the user can enable Intel SST-CP itself, as well as configure CLOS priority groups that are used by Intel SST-CP technology to make decisions about the distribution of power among CPU cores[3].

Note: The version coming with your kernel may not be the latest version of the `intel-speed-select` tool. Refer to Appendix A: Installing and Configuring Intel SST-CP for more information on how to get the latest version.

3.1.1 Displaying Help Text

To display the `intel-speed-select` help text, run the tool without any options or include the `-h` option. The help text displays information about the tool usage and the available options as follows:

```
# intel-speed-select -h
Intel(R) Speed Select Technology
```

[3] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

```
Executing on CPU model:106[0x6a]

Usage:
intel-speed-select [OPTIONS] FEATURE COMMAND COMMAND_ARGUMENTS

Use this tool to enumerate and control the Intel Speed Select Technology features:

FEATURE : [perf-profile|base-freq|turbo-freq|core-power]

For help on each feature, use -h|--help
        For example:  intel-speed-select perf-profile -h

For additional help on each command for a feature, use --h|--help
        For example:  intel-speed-select perf-profile get-lock-status -h
                 This will print help for the command "get-lock-status" for the feature "perf-
profile"

OPTIONS
        [-c|--cpu] : logical cpu number
                Default: Die scoped for all dies in the system with multiple dies/package
                         Or Package scoped for all Packages when each package contains one die
        [-d|--debug] : Debug mode
        [-f|--format] : output format [json|text]. Default: text
        [-h|--help] : Print help
        [-i|--info] : Print platform information
        [-o|--out] : Output file
                        Default : stderr
        [-v|--version] : Print version

Result format
        Result display uses a common format for each command:
        Results are formatted in text/JSON with
                Package, Die, CPU, and command specific results.

Examples
        To get platform information:
                intel-speed-select –info
        To get full perf-profile information dump:
                intel-speed-select perf-profile info
        To get full base-freq information dump:
                intel-speed-select base-freq info -l 0
        To get full turbo-freq information dump:
                intel-speed-select turbo-freq info -l 0
```

However, this help text is generic and is not specific to any Intel Speed Select Technology feature. To display help message that is specific to Intel SST-CP technology, the `core-power` option must be specified, as well as (optionally) a command to display help message for. For example, the following command lists all available Intel SST-CP-related commands:

```
# intel-speed-select core-power -h
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
core-power:     Interface that allows user to define per core/tile
                priority.

Commands : For feature=core-power
        info
        enable
        disable
        config
        get-config
        assoc
        get-assoc
```

To list help for a specific command out of the above list, the command must be added as a command line parameter, similarly followed by a `-h` option:

```
# intel-speed-select core-power enable -h
Intel(R) Speed Select Technology
```

```
Executing on CPU model:106[0x6a]
Enable core-power for a package/die
        Clos Enable: Specify priority type with [--priority|-p]
                 0: Proportional, 1: Ordered
```

The help text for each command explains the various options available. We explore many of these options in the later sections of this document.

3.1.2 Displaying System Status and Configuration

When you attempt to configure or receive status reports from the `intel-speed-select` tool, it is important to always specify the CPU core mask that the command affects. This is because most of the configuration is package- and core-specific, so it is imperative to indicate which specific cores are affected. For package-wide settings (such as CLOS group configuration), it is enough to specify one CPU core number belonging to the package. Since the changes are package-wide, they are applied regardless of which specific CPU core number was specified as a configuration endpoint.

To display the general package status and whether the CPU supports Intel SST-CP technology, the `info` command should be specified, along with a CPU core list. The following is an example output:

```
# intel-speed-select -c 0 core-power info
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      core-power
        support-status:supported
        enable-status:enabled
        clos-enable-status:enabled
        priority-type:ordered
```

The above output indicates that Intel SST-CP technology is supported and enabled on this CPU.

To display more specific package-wide information (such as CLOS group configuration), a similar command can be executed, specifying one of the cores belonging to the target package, as well as any command-specific arguments (such as CLOS group number):

```
# intel-speed-select -c 0 core-power get-config -c 0
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      core-power
        clos:0
        epp:0
        clos-proportional-priority:0
        clos-min:0 MHz
        clos-max:Max Turbo frequency
        clos-desired:0 MHz
```

Displaying per-CPU core information (such as which CPU core is assigned to which CLOS group) works in a similar way:

```
# intel-speed-select -c 0-3 core-power get-assoc
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      get-assoc
        clos:0
 package-0
  die-0
    cpu-1
      get-assoc
        clos:0
 package-0
  die-0
    cpu-2
```

```
      get-assoc
        clos:0
 package-0
  die-0
    cpu-3
      get-assoc
        clos:0
```

In the above example, all CPU cores belong to the same CLOS group.

3.1.3 Setting per-CPU and per-Core Values

In a similar way to how reading the configuration works, the configuration can be changed using the command-line tool. For per-package configuration, specifying one CPU core belonging to the target package is enough, while for per-CPU core configuration, each affected CPU core number must be specified.

For example, CLOS group configuration is package-wide, so changing it only requires one CPU core as a parameter:

```
# intel-speed-select -c 0 core-power config --clos 0 --min 1200
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
clos epp is not specified or invalid, default: 0
clos frequency weight is not specified or invalid, default: 0
clos max is not specified, default: Max frequency (ratio 0xff)
clos desired is not supported on this platform
 package-0
  die-0
    cpu-0
      core-power
        config:success
```

As another example, CLOS group association is per-CPU core, so a list of affected CPU cores should be specified when setting them up:

```
# intel-speed-select -c 0-1 core-power assoc --clos 1
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      core-power
        assoc:success
 package-0
  die-0
    cpu-1
      core-power
        assoc:success
```

Using the command given in the previous section, we can now check if the associations have been set up correctly:

```
# intel-speed-select -c 0-3 core-power get-assoc
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      get-assoc
        clos:1
 package-0
  die-0
    cpu-1
      get-assoc
        clos:1
 package-0
  die-0
    cpu-2
      get-assoc
        clos:0
 package-0
  die-0
    cpu-3
      get-assoc
        clos:0
```

As in the above example, CPU cores 0 and 1 are now associated with CLOS group 1, while CPU cores 2 and 3 are still associated with CLOS group 0, as they were before. By default, all cores are associated with CLOS group 0.

4 Configuring Prioritization

The `intel-speed-select` tool has many configuration options. This document concentrates on the most important and relevant of them. However, to make full use of these options, refer to Appendix A on how to ensure that the minimum recommended version of the `intel-speed-select` tool is installed on the system[4].

4.1 Priority Assignment

Intel SST-CP currently supports two modes of operation. These modes are referred to as “priority types” in the `intel-speed-select` tool. The two priority types available are “proportional” and “ordered” priority. This guide uses the “ordered” priority type as it is the recommended way to set up Intel SST-CP.

The “ordered” feature allows the user to explicitly specify the cores to prioritize, hence, power is explicitly applied to this core or group of cores before any remainder is applied to other priorities. In contrast, the proportional configuration allows for the power to be shared across all priorities with a higher proportion to the higher priority core group.

The priority is assigned to CLOS groups (to which CPU cores can be assigned), so CLOS groups are the granularity at which priority is assigned and redistribution of power is happening. In Ordered Priority mode, the CLOS group number determines priority, starting from CLOS group 0, which has the highest priority.

Therefore, systems wishing to take advantage of Intel SST-CP should associate CPU cores used for high priority tasks with CLOS group 0 (in Ordered Priority mode) and assign all other CPU cores to other CLOS groups depending on the priority of the task.

4.2 Frequency Requirements

Every CLOS group can also have optional minimum and maximum frequency requirements. This allows for various use cases associated with setting specific frequency points for all cores in a specific CLOS group, such as:

- Limiting maximum frequency
- Ensuring best attempts are made in reaching a specified minimum frequency
- Fixing the frequency for all cores by setting minimum and maximum to the same value

As mentioned in Section 2.2, Intel SST-CP prioritization only takes place if system power use reaches its TDP. For the purposes of prioritization, only the minimum frequency requirements are considered. For power budgeting, all parameters are considered.

The notion of “priority” only applies when the Power Management Unit (PMU) attempts to satisfy minimum frequency requirements for all CLOS groups. As noted previously, it does so in order, so whenever frequency requirements cannot be satisfied for all CLOS groups, the higher priority CLOS groups (starting with CLOS group 0) satisfy their requirements at the expense of lower priority CLOS groups (starting with CLOS group 3).

5 Example Use Cases

The following sections describe several example use cases, together with suggested configurations, and the `intel-speed-select` tool commands necessary to achieve the required configurations. Before attempting to configure a system, refer to the instructions in Appendix A to set up the system correctly. To observe the runtime power and core frequencies shown in the graphs in the following sections, use the instructions in Appendix B.

5.1 Use Case 1: 20-core Data Plane and 4-core Control Plane

In this use case, we use Intel SST-CP technology to prioritize 20 data plane cores over 4 control plane cores. To always prioritize the data plane cores over the control plane cores, we must set the minimum frequency of the data plane cores to be higher than that of the control plane cores and assign the high priority cores to CLOS group 0. This use case assumes that cores 0 through 3 are the control plane cores, and cores 4 through 23 are the data plane cores.

[4] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

5.1.1 Resetting Configuration

Before configuring anything, it is important to bring the system to a known state. To do that, we can reset all CLOS groups to their default values, assign all cores to CLOS group 0, and enable Ordered Priority mode using the following commands:

```
# intel-speed-select -c 0-23 core-power config -c 0  # reset CLOS group 0
# intel-speed-select -c 0-23 core-power config -c 1  # reset CLOS group 1
# intel-speed-select -c 0-23 core-power config -c 2  # reset CLOS group 2
# intel-speed-select -c 0-23 core-power config -c 3  # reset CLOS group 3
# intel-speed-select -c 0-23 core-power assoc -c 0  # assign all cores to CLOS group 0
# intel-speed-select -c 0-23 core-power disable  # disable any currently active priority mode
# intel-speed-select -c 0-23 core-power enable --priority 1  # enable Ordered Priority mode
```

To ensure that correct priority mode is enabled, issue an `info` command; its output should indicate that the Intel SST-CP is enabled in Ordered Priority mode:

```
# intel-speed-select -c 0 core-power info
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      core-power
        support-status:supported
        enable-status:enabled
        clos-enable-status:enabled
        priority-type:ordered
```

Thereafter, we can use the `intel-speed-select` tool to set up the desired configuration.

5.1.2 Configuring Intel SST-CP

While there are multiple ways to configure the system to achieve high performance for priority cores, to get the highest performance, it is recommended to pick the highest frequency available and setting it as the minimum frequency. When Intel Turbo Boost Technology is enabled, the highest frequency a CPU core can reach when all cores are active is called the all core turbo frequency. The exact frequency varies based on which processor the system is running. For the purposes of this guide, the all-core turbo frequency of 3.2 GHz is assumed.

To set up the high-priority CPU cores to have their minimum frequency at all core turbo, we first have to ensure that our high priority CPU cores are assigned to the CLOS group 0, while the rest of them are assigned to CLOS group 1:

```
# intel-speed-select -c 0-19 core-power assoc -c 0  # associate cores 0-19 with CLOS 0
# intel-speed-select -c 20-23 core-power assoc -c 1  # associate cores 20-23 with CLOS 1
```

After that is done, we can configure the minimum frequency for CLOS group 0:

```
# intel-speed-select -c 0 core-power config -c 0 --min 3200
```

To ensure that the correct configuration is entered, the `get config` command can be issued:

```
# intel-speed-select -c 0 core-power get-config -c 0
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      core-power
        clos:0
        epp:0
        clos-proportional-priority:0
        clos-min:3200 MHz
        clos-max:Max Turbo frequency
        clos-desired:0 MHz
```

The workload configuration is now complete. We can then run the workload and ensure that the configuration satisfies the performance constraints[5]. See Figure 5.

[5] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

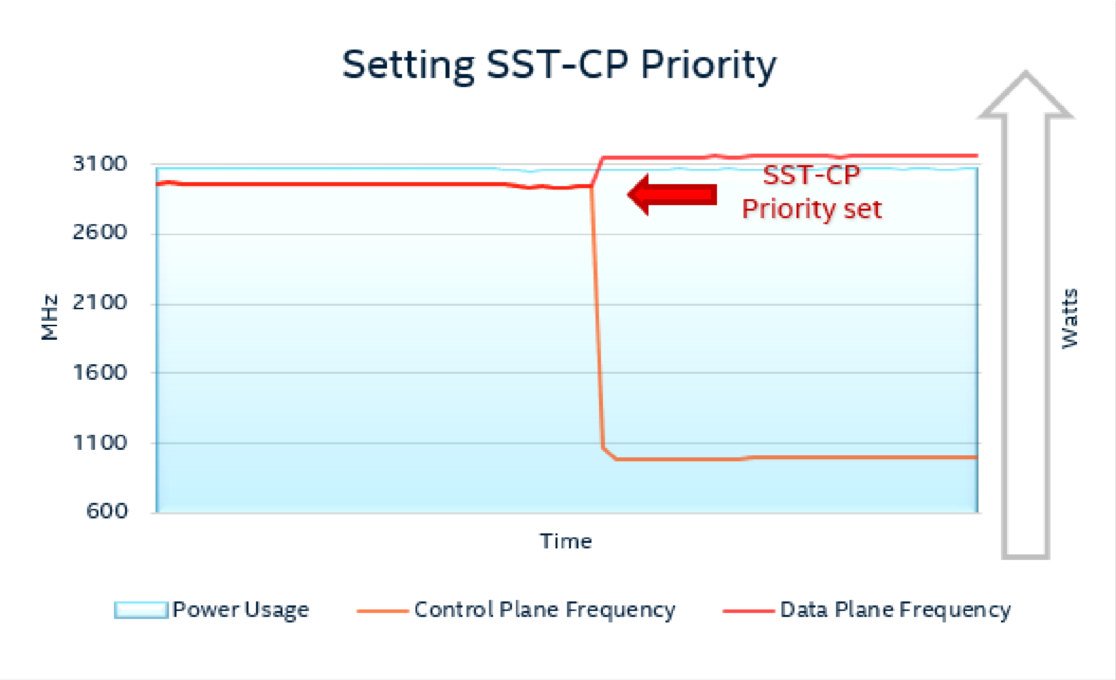

Figure 5. Verifying Performance of 20-core Data Plane and 4-core Control Plane Configuration

Be aware that using Intel SST-CP technology to set the minimum frequency does not ensure that the data plane cores always operate at requested frequency. Intel SST-CP technology provides no such assurance. Instead, the intention is that data plane cores have the highest priority, they always get prioritized access to power, and they maintain the minimum frequency unless it is not possible to do so. By default, the maximum frequency value for the data plane cores is set to maximum Turbo-Boost frequency value, which means that the high-priority CPU core may go higher than minimum frequency if enough CPU cores are in the Idle state.

Since the maximum frequency value for control plane cores is also set at the maximum supported frequency, control plane cores can also operate at full speed, but only if data plane cores are idle enough to allow for that. The Intel SST-CP technology ensures that control plane tasks do not interfere with high-priority tasks, as the low-priority tasks automatically get lower frequencies if there is a demand to process high-priority tasks[6]. This is completely automatic.

5.2 Use Case 2: 16-core Data Plane and 8-core Control Plane with Jitter Sensitivity

For this use case, we assume a configuration with a 16-core data plane and an 8-core control plane. In addition, we assume that the workload running on the data plane cores is sensitive to frequency transitions, and that it would be preferable to always have it running at constant frequency rather than allowing frequency transitions to higher Turbo Boost range frequencies whenever some cores enter Idle state.

Note: Before applying any configuration, it is highly recommended to bring the system to a known state by following a configuration reset procedure that is described in Section 5.1.1.

5.2.1 Limiting Data Plane Frequency Transitions

In this example, we again assume that our data plane workload benefits from running at highest frequency possible at all times. On Intel processors, such frequency is referred to as all core turbo frequency, as this is the highest frequency any given CPU core can run at when all other cores are also active and is a typical configuration.

When not all cores are active, the remaining CPU cores' frequencies may go higher than all core turbo frequency, up to maximum Turbo Boost frequency. Running at higher frequencies is generally preferable as it allows for faster processing and thus more throughput, but if the Control Plane workload is such that CPU cores are constantly oscillating between idle and active states, this may negatively affect certain latency-sensitive workloads as such a situation may result in frequency jitter for the Data Plane CPU cores.

[6] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

With Intel SST-CP, it is possible to both prioritize Data Plane workloads over Control Plane workloads, as well as limiting the maximum frequency the Data Plane workload may go up to. As in previous scenarios, the first configuration step would be to set up core associations:

```
# intel-speed-select -c 0-15 core-power assoc -c 0
# intel-speed-select -c 16-23 core-power assoc -c 1
```

The next step would be to configure the CLOS group 0 to be limited to all core turbo frequency:

```
# intel-speed-select -c 0 core-power config -c 0 --min 3200 --max 3200
```

We then check if the configuration has applied to CLOS group 0:

```
# intel-speed-select -c 0 core-power get-config -c 0
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
 package-0
  die-0
    cpu-0
      core-power
        clos:0
        epp:0
        clos-proportional-priority:0
        clos-min:3200 MHz
        clos-max:3200 MHz
        clos-desired:0 MHz
```

We can now verify that this configuration is working by running the workload and observing frequency for Data Plane and Control Plane CPU cores.

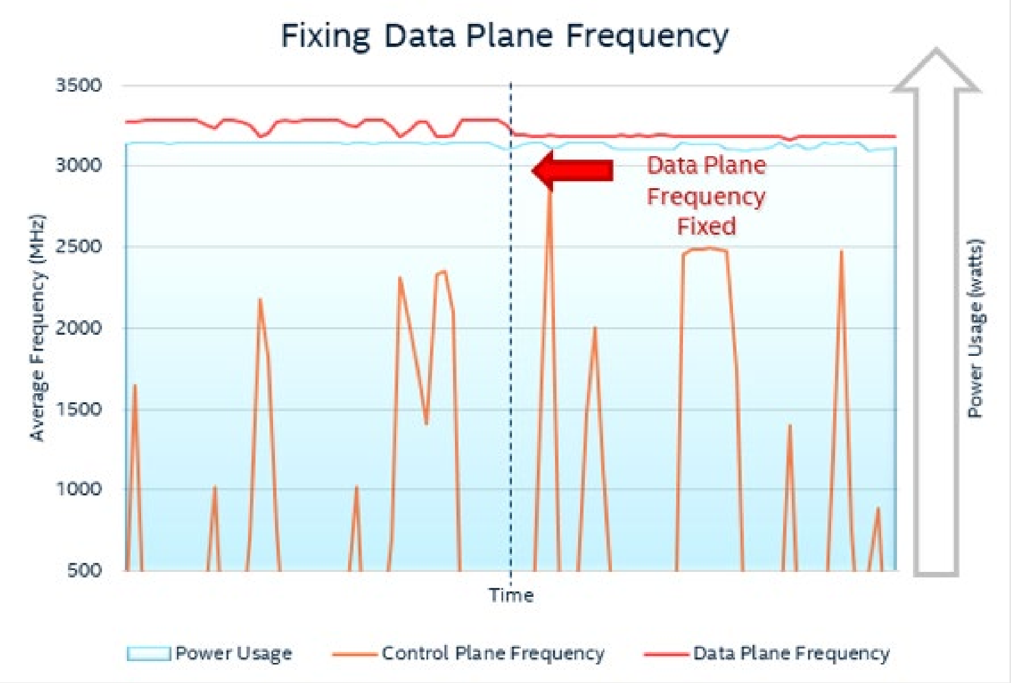

Figure 6. Verifying Performance of 16-core Data Plane and 8-core Control Plane with Fixed Data Plane Frequency

As can be seen in Figure 6, without fixing the data plane frequency, it can oscillate between different Turbo Boost frequencies because of the transition of control plane cores from idle state to busy state. Normally this is not a problem and is preferred due to higher average operating frequency (and thus higher throughput), but in cases where latency matters, these frequency transitions can be undesirable. With Intel SST-CP, it is possible to avoid these frequency switches in most cases[7].

[7] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

5.3 Use Case 3: 3-Tier Configuration

Until now, the examples were dealing with a 2-tier setup, with a critical workload in one tier, and a non-critical one in the other. Real deployments may have different things running on different cores, and each of those would have different priority and different requirements when it comes to performance. In this example use case, we assume a 3-tier configuration: a 12-core data plane with highest performance requirements, an 8-core workload with lower performance requirements, and a 4-core control plane.

5.3.1 Configuring 3 CLOS Groups

In Ordered Priority mode, the prioritization of CPU cores in a particular CLOS group is implicit and is decided based on CLOS group number. Therefore, to prioritize certain CPU cores, they need to be assigned to correct CLOS groups. In this scenario, we make the following assumptions:

- 10-core data plane with highest performance requirements
- 8-core workload that doesn't necessarily benefit from highest performance, but does need to be at a certain level at all times
- 6-core control plane workload that does not have performance requirements

The above CPU cores are assigned to CLOS groups 0, 1 and 2 respectively:

```
# intel-speed-select -c 0-9 core-power assoc --clos 0
# intel-speed-select -c 10-17 core-power assoc --clos 1
# intel-speed-select -c 18-23 core-power assoc --clos 2
```

We also have to specify minimum frequencies these CLOS groups will be operating in. The assumptions made in this scenario are as follows:

- CLOS group 0 is the highest priority group, so it is configured to have its minimum frequency set to all-core turbo frequency of 3.2 GHz
- CLOS group 1 has its performance requirements, but a minimum of 2.8 GHz and a maximum of 3.0 GHz is sufficient to satisfy them
- CLOS group 2 has no performance requirements, so its minimum is set to the lowest possible value (in our case, it is assumed to be 1 GHz)

The following command line sets up this scenario:

```
# intel-speed-select -c 0 core-power config --clos 0 --min 3200
# intel-speed-select -c 0 core-power config --clos 1 --min 2800 --max 3000
# intel-speed-select -c 0 core-power config --clos 2 --min 1000
```

After applying this configuration, the control plane workload can still scale up to maximum frequency possible, but never at the expense of other workloads. Whenever a CPU core belonging to a higher priority CLOS group becomes busy, the control plane gets out of the way.

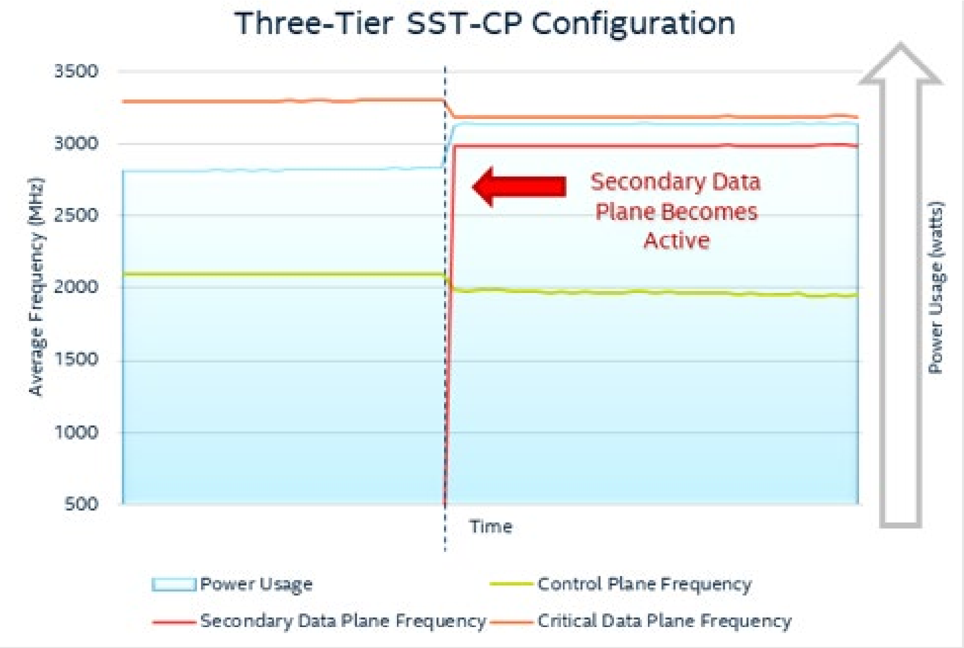

Figure 7. Verifying Performance of Three Tier SST-CP Configuration

As can be seen on Figure 7, when non-critical data plane cores become active, the control plane frequency is automatically lowered. The critical data plane frequency gets lowered also, but that has to do with fewer cores being idle, so the critical data plane always operates at whatever maximum frequency it can get for a given number of active cores, and that frequency is also maintained by Intel SST-CP[8].

6 Summary

The example scenarios discussed in this guide have shown that Intel SST-CP can be applied to implement a "Power QoS" to direct power to the most important cores matching power/frequency for the most important workloads running on CPU. By prioritizing power to cores, it allows users who deploy workloads to meet SLAs and deliver QoE for subscribers.

[8] See backup for workloads and configurations or visit www.Intel.com/PerformanceIndex. Results may vary.

Appendix A Installing and Configuring Intel SST-CP

A.1 OS Configuration

To use Intel SST-CP, the Linux OS must use a supported kernel version. The recommended kernel version is 5.8.

To use uncore frequency scaling, if the msr kernel driver is available, use the driver. Otherwise, install msr-tools and use MSR writing to control uncore frequency.

To use core frequencies and package power, the turbostat tool needs to be installed, which is available in kernel releases. In Red Hat-based distributions, it is available in Kernel tools packages and in Debian-based distributions it is available in the linux-tools-common package.

A.2 Installing and Using the intel-speed-select Tool

This guide references a command line intel-speed-select tool. It requires a Linux kernel version 5.8 or higher and is often installed by default. For this guide, version 1.4 was used, and this is the minimum recommended version to use. To find the version of the tool, simply add -v as a command line parameter:

```
# intel-speed-select -v
Intel(R) Speed Select Technology
Executing on CPU model:106[0x6a]
Version v1.4
```

If the version that comes with the distribution is older than 1.4, it is advised to either upgrade the kernel or compile the tool from source. The source code can be downloaded either from the Linux kernel source tree or from GitHub.

Run the intel-speed-select tool from the root account.

A.3 Enabling the MSR Driver

The MSR Tools package referenced in this guide depends on having the msr Linux kernel driver loaded. On some distributions, this module may be built into the kernel and does not need to be loaded. On others, it is necessary to load this driver before using the MSR Tools package.

To load the msr driver, issue the following command:

```
# modprobe msr
```

If the msr Linux kernel driver is not loaded, programs from the MSR Tools package display an error.

Appendix B Data Collection Tools

B.1 turbostat

turbostat is a Linux tool used to report the processor topology, frequency, idle power-state statistics, temperature, and power usage on Intel processors. Use turbostat tool as shown in the following example:

```
# turbostat --show Bzy_MHz --show PkgWatt --show Core -i 2 --cpu 0-23
Options:
--show column: show only the specified built-in columns.
Core: Display the core number.
Bzy_MHz: average clock rate while the CPU was not idle.
PkgWatt: Watts consumed by the whole package.
-i sec: overrides the default 5.0 second measurement interval.
--cpu cpu-set: limit output to system summary plus the specified cpu-set.
```

B.2 rdmsr

rdmsr is available with the MSR tools, as mentioned in Appendix A.3. It is used for reading a CPU's machine-specific registers (MSR).

To read the core frequency using the legacy MSR 0x198, use the following example command:

```
# rdmsr -a 0x198
```

This command gives a six byte hex value and the second LSB byte represents core frequency.

To read the current uncore frequency using the legacy MSR 0x621, use the following example command:

```
# rdmsr -a 0x621
```

This command gives a six byte hex value and the first LSB byte represents the uncore frequency.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

0323/DN/WIT/PDF 638103-002US