Intel(R) Speed Select Technology User Guide

The Intel(R) Speed Select Technology (Intel(R) SST) provides a powerful new collection of features that give more granular control over CPU performance. With Intel(R) SST, one server can be configured for power and performance for a variety of diverse workload requirements.

Refer to the links below for an overview of the technology:

- https://www.intel.com/content/www/us/en/architecture-and-technology/speed-select-technology-article.html
- https://builders.intel.com/docs/networkbuilders/intel-speed-select-technology-base-frequency-enhancing-performance.pdf

These capabilities are further enhanced in some of the newer generations of server platforms where these features can be enumerated and controlled dynamically without pre-configuring via BIOS setup options. This dynamic configuration is done via mailbox commands to the hardware. One way to enumerate and configure these features is by using the Intel Speed Select utility.

This document explains how to use the Intel Speed Select tool to enumerate and control Intel(R) SST features. This document gives example commands and explains how these commands change the power and performance profile of the system under test. Using this tool as an example, customers can replicate the messaging implemented in the tool in their production software.

intel-speed-select configuration tool

Most Linux distribution packages may include the "intel-speed-select" tool. If not, it can be built by downloading the Linux kernel tree from kernel.org. Once downloaded, the tool can be built without building the full kernel.

From the kernel tree, run the following commands:

```
# cd tools/power/x86/intel-speed-select/
# make
# make install
```

Getting Help

To get help with the tool, execute the command below:

```
# intel-speed-select --help
```

The top-level help describes arguments and features. Notice that there is a multi-level help structure in the tool. For example, to get help for the feature "perf-profile":

```
# intel-speed-select perf-profile --help
```

To get help on a command, another level of help is provided. For example for the command info "info":

```
# intel-speed-select perf-profile info --help
```

Summary of platform capability

To check the current platform and driver capabilities, execute:

```
#intel-speed-select --info
```

For example on a test system:

```
# intel-speed-select --info
Intel(R) Speed Select Technology
Executing on CPU model: X
Platform: API version : 1
Platform: Driver version : 1
Platform: mbox supported : 1
Platform: mmio supported : 1
Intel(R) SST-PP (feature perf-profile) is supported
TDP level change control is unlocked, max level: 4
Intel(R) SST-TF (feature turbo-freq) is supported
Intel(R) SST-BF (feature base-freq) is not supported
Intel(R) SST-CP (feature core-power) is supported
```

Intel(R) Speed Select Technology - Performance Profile (Intel(R) SST-PP)

This feature allows configuration of a server dynamically based on workload performance requirements. This helps users during deployment as they do not have to choose a specific server configuration statically. This Intel(R) Speed Select Technology - Performance Profile (Intel(R) SST-PP) feature introduces a mechanism that allows multiple optimized performance profiles per system. Each profile defines a set of CPUs that need to be online and rest offline to sustain a guaranteed base frequency. Once the user issues a command to use a specific performance profile and meet CPU online/offline requirement, the user can expect a change in the base frequency dynamically. This feature is called "perf-profile" when using the Intel Speed Select tool.

Number or performance levels

There can be multiple performance profiles on a system. To get the number of profiles, execute the command below:

```
# intel-speed-select perf-profile get-config-levels
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
       get-config-levels:4
package-1
 die-0
   cpu-14
       get-config-levels:4
```

On this system under test, there are 4 performance profiles in addition to the base performance profile (which is performance level 0).

Lock/Unlock status

Even if there are multiple performance profiles, it is possible that they are locked. If they are locked, users cannot issue a command to change the performance state. It is possible that there is a BIOS setup to unlock or check with your system vendor.

To check if the system is locked, execute the following command:

```
# intel-speed-select perf-profile get-lock-status
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
       get-lock-status:0
package-1
 die-0
   cpu-14
       get-lock-status:0
```

In this case, lock status is 0, which means that the system is unlocked.

Properties of a performance level

To get properties of a specific performance level (For example for the level 0, below), execute the command below:

```
# intel-speed-select perf-profile info -l 0
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     perf-profile-level-0
       cpu-count:28
       enable-cpu-mask:000003ff,f0003fff
       enable-cpu-list:0,1,2,3,4,5,6,7,8,9,10,11,12,13,28,29,30,31,32,33,34,35,36,37,38,39,40,41
       thermal-design-power-ratio:26
       base-frequency(MHz):2600
       speed-select-turbo-freq:disabled
       speed-select-base-freq:disabled
       ...
       ...
```

Here -l option is used to specify a performance level.

If the option -l is omitted, then this command will print information about all the performance levels. The above command is printing properties of the performance level 0.

For this performance profile, the list of CPUs displayed by the "enable-cpu-mask/enable-cpu-list" at the max can be "online." When that condition is met, then base frequency of 2600 MHz can be maintained. To understand more, execute "intel-speed-select perf-profile info" for performance level 4:

```
# intel-speed-select perf-profile info -l 4
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     perf-profile-level-4
       cpu-count:28
       enable-cpu-mask:000000fa,f0000faf
       enable-cpu-list:0,1,2,3,5,7,8,9,10,11,28,29,30,31,33,35,36,37,38,39
       thermal-design-power-ratio:28
       base-frequency(MHz):2800
```

```
        speed-select-turbo-freq:disabled
        speed-select-base-freq:unsupported
        ...
        ...
```

There are fewer CPUs in the "enable-cpu-mask/enable-cpu-list". Consequently, if the user only keeps these CPUs online and the rest "offline," then the base frequency is increased to 2.8 GHz compared to 2.6 GHz at performance level 0.

Get current performance level

To get the current performance level, execute:

```
# intel-speed-select perf-profile get-config-current-level
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
       get-config-current_level:0
```

First verify that the base_frequency displayed by the cpufreq sysfs is correct:

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/base_frequency
2600000
```

This matches the base-frequency (MHz) field value displayed from the "perf-profile info" command for performance level 0(cpufreq frequency is in KHz).

To check if the average frequency is equal to the base frequency for a 100% busy workload, disable turbo:

```
# echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo
```

Then runs a busy workload on all CPUs, for example:

```
#stress -c 64
```

To verify the base frequency, run turbostat:

```
#turbostat -c 0-13 --show Package,Core,CPU,Bzy_MHz -i 1

 Package       Core    CPU     Bzy_MHz
               -       -       2600
 0             0       0       2600
 0             1       1       2600
 0             2       2       2600
 0             3       3       2600
 0             4       4       2600
 .             .       .       .
```

Changing performance level

To the change the performance level to 4, execute:

```
# intel-speed-select -d perf-profile set-config-level -l 4 -o
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     perf-profile
       set_tdp_level:success
```

In the command above, "-o" is optional. If it is specified, then it will also offline CPUs which are not present in the enable_cpu_mask for this performance level.

Now if the base_frequency is checked:

```
#cat /sys/devices/system/cpu/cpu0/cpufreq/base_frequency
2800000
```

Which shows that the base frequency now increased from 2600 MHz at performance level 0 to 2800 MHz at performance level 4. As a result, any workload, which can use fewer CPUs, can see a boost of 200 MHz compared to performance level 0.

Check presence of other Intel(R) SST features

Each of the performance profiles also specifies weather there is support of other two Intel(R) SST features (Intel(R) Speed Select Technology - Base Frequency (Intel(R) SST-BF) and Intel(R) Speed Select Technology - Turbo Frequency (Intel SST-TF)).

For example, from the output of "perf-profile info" above, for level 0 and level 4:

For level 0::

speed-select-turbo-freq:disabled speed-select-base-freq:disabled

For level 4::

speed-select-turbo-freq:disabled speed-select-base-freq:unsupported

Given these results, the "speed-select-base-freq" (Intel(R) SST-BF) in level 4 changed from "disabled" to "unsupported" compared to performance level 0.

This means that at performance level 4, the "speed-select-base-freq" feature is not supported. However, at performance level 0, this feature is "supported", but currently "disabled", meaning the user has not activated this feature. Whereas "speed-select-turbo-freq" (Intel(R) SST-TF) is supported at both performance levels, but currently not activated by the user.

The Intel(R) SST-BF and the Intel(R) SST-TF features are built on a foundation technology called Intel(R) Speed Select Technology - Core Power (Intel(R) SST-CP). The platform firmware enables this feature when Intel(R) SST-BF or Intel(R) SST-TF is supported on a platform.

Intel(R) Speed Select Technology Core Power (Intel(R) SST-CP)

Intel(R) Speed Select Technology Core Power (Intel(R) SST-CP) is an interface that allows users to define per core priority. This defines a mechanism to distribute power among cores when there is a power constrained scenario. This defines a class of service (CLOS) configuration.

The user can configure up to 4 class of service configurations. Each CLOS group configuration allows definitions of parameters, which affects how the frequency can be limited and power is distributed. Each CPU core can be tied to a class of service and hence an associated priority. The granularity is at core level not at per CPU level.

Enable CLOS based prioritization

To use CLOS based prioritization feature, firmware must be informed to enable and use a priority type. There is a default per platform priority type, which can be changed with optional command line parameter.

To enable and check the options, execute:

```
# intel-speed-select core-power enable --help
Intel(R) Speed Select Technology
Executing on CPU model: X
Enable core-power for a package/die
       Clos Enable: Specify priority type with [--priority|-p]
                0: Proportional, 1: Ordered
```

There are two types of priority types:

- Ordered

Priority for ordered throttling is defined based on the index of the assigned CLOS group. Where CLOS0 gets highest priority (throttled last).

Priority order is: CLOS0 > CLOS1 > CLOS2 > CLOS3.

- Proportional

When proportional priority is used, there is an additional parameter called frequency_weight, which can be specified per CLOS group. The goal of proportional priority is to provide each core with the requested min., then distribute all remaining (excess/deficit) budgets in proportion to a defined weight. This proportional priority can be configured using "core-power config" command.

To enable with the platform default priority type, execute:

```
# intel-speed-select core-power enable
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     core-power
       enable:success
package-1
 die-0
   cpu-6
     core-power
       enable:success
```

The scope of this enable is per package or die scoped when a package contains multiple dies. To check if CLOS is enabled and get priority type, "core-power info" command can be used. For example to check the status of core-power feature on CPU 0, execute:

```
# intel-speed-select -c 0 core-power info
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
```

```
      core-power
        support-status:supported
        enable-status:enabled
        clos-enable-status:enabled
        priority-type:proportional
package-1
 die-0
   cpu-24
     core-power
       support-status:supported
       enable-status:enabled
       clos-enable-status:enabled
       priority-type:proportional
```

Configuring CLOS groups

Each CLOS group has its own attributes including min, max, freq_weight and desired. These parameters can be configured with "core-power config" command. Defaults will be used if user skips setting a parameter except clos id, which is mandatory. To check core-power config options, execute:

```
# intel-speed-select core-power config --help
Intel(R) Speed Select Technology
Executing on CPU model: X
Set core-power configuration for one of the four clos ids
       Specify targeted clos id with [--clos|-c]
       Specify clos Proportional Priority [--weight|-w]
       Specify clos min in MHz with [--min|-n]
       Specify clos max in MHz with [--max|-m]
```

For example:

```
# intel-speed-select core-power config -c 0
Intel(R) Speed Select Technology
Executing on CPU model: X
clos epp is not specified, default: 0
clos frequency weight is not specified, default: 0
clos min is not specified, default: 0 MHz
clos max is not specified, default: 25500 MHz
clos desired is not specified, default: 0
package-0
 die-0
   cpu-0
     core-power
       config:success
package-1
 die-0
   cpu-6
     core-power
       config:success
```

The user has the option to change defaults. For example, the user can change the "min" and set the base frequency to always get guaranteed base frequency.

Get the current CLOS configuration

To check the current configuration, "core-power get-config" can be used. For example, to get the configuration of CLOS 0:

```
# intel-speed-select core-power get-config -c 0
Intel(R) Speed Select Technology
Executing on CPU model: X
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
package-1
 die-0
   cpu-24
     core-power
       clos:0
       epp:0
       clos-proportional-priority:0
       clos-min:0 MHz
       clos-max:Max Turbo frequency
       clos-desired:0 MHz
```

Associating a CPU with a CLOS group

To associate a CPU to a CLOS group "core-power assoc" command can be used:

```
# intel-speed-select core-power assoc --help
Intel(R) Speed Select Technology
Executing on CPU model: X
Associate a clos id to a CPU
       Specify targeted clos id with [--clos|-c]
```

For example to associate CPU 10 to CLOS group 3, execute:

```
# intel-speed-select -c 10 core-power assoc -c 3
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-10
     core-power
       assoc:success
```

Once a CPU is associated, its sibling CPUs are also associated to a CLOS group. Once associated, avoid changing Linux "cpufreq" subsystem scaling frequency limits.

To check the existing association for a CPU, "core-power get-assoc" command can be used. For example, to get association of CPU 10, execute:

```
# intel-speed-select -c 10 core-power get-assoc
Intel(R) Speed Select Technology
Executing on CPU model: X
package-1
 die-0
   cpu-10
     get-assoc
       clos:3
```

This shows that CPU 10 is part of a CLOS group 3.

Disable CLOS based prioritization

To disable, execute:

```
# intel-speed-select core-power disable
```

Some features like Intel(R) SST-TF can only be enabled when CLOS based prioritization is enabled. For this reason, disabling while Intel(R) SST-TF is enabled can cause Intel(R) SST-TF to fail. This will cause the "disable" command to display an error if Intel(R) SST-TF is already enabled. In turn, to disable, the Intel(R) SST-TF feature must be disabled first.

Intel(R) Speed Select Technology - Base Frequency (Intel(R) SST-BF)

The Intel(R) Speed Select Technology - Base Frequency (Intel(R) SST-BF) feature lets the user control base frequency. If some critical workload threads demand constant high guaranteed performance, then this feature can be used to execute the thread at higher base frequency on specific sets of CPUs (high priority CPUs) at the cost of lower base frequency (low priority CPUs) on other CPUs. This feature does not require offline of the low priority CPUs.

The support of Intel(R) SST-BF depends on the Intel(R) Speed Select Technology - Performance Profile (Intel(R) SST-PP) performance level configuration. It is possible that only certain performance levels support Intel(R) SST-BF. It is also possible that only base performance level (level = 0) has support of Intel SST-BF. Consequently, first select the desired performance level to enable this feature.

In the system under test here, Intel(R) SST-BF is supported at the base performance level 0, but currently disabled. For example for the level 0:

```
# intel-speed-select -c 0 perf-profile info -l 0
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     perf-profile-level-0
       ...

       speed-select-base-freq:disabled
       ...
```

Before enabling Intel(R) SST-BF and measuring its impact on a workload performance, execute some workload and measure performance and get a baseline performance to compare against.

Here the user wants more guaranteed performance. For this reason, it is likely that turbo is disabled. To disable turbo, execute:

```
#echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo
```

Based on the output of the "intel-speed-select perf-profile info -l 0" base frequency of guaranteed frequency 2600 MHz.

Measure baseline performance for comparison

To compare, pick a multi-threaded workload where each thread can be scheduled on separate CPUs. "Hackbench pipe" test is a good example on how to improve performance using Intel(R) SST-BF.

Below, the workload is measuring average scheduler wakeup latency, so a lower number means better performance:

```
# taskset -c 3,4 perf bench -r 100 sched pipe
# Running 'sched/pipe' benchmark:
# Executed 1000000 pipe operations between two processes
    Total time: 6.102 [sec]
      6.102445 usecs/op
        163868 ops/sec
```

While running the above test, if we take turbostat output, it will show us that 2 of the CPUs are busy and reaching max. frequency (which would be the base frequency as the turbo is disabled). The turbostat output:

```
#turbostat -c 0-13 --show Package,Core,CPU,Bzy_MHz -i 1
Package        Core    CPU     Bzy_MHz
0              0       0       1000
0              1       1       1005
0              2       2       1000
0              3       3       2600
0              4       4       2600
0              5       5       1000
0              6       6       1000
0              7       7       1005
0              8       8       1005
0              9       9       1000
0              10      10      1000
0              11      11      995
0              12      12      1000
0              13      13      1000
```

From the above turbostat output, both CPU 3 and 4 are very busy and reaching full guaranteed frequency of 2600 MHz.

Intel(R) SST-BF Capabilities

To get capabilities of Intel(R) SST-BF for the current performance level 0, execute:

```
# intel-speed-select base-freq info -l 0
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     speed-select-base-freq
       high-priority-base-frequency(MHz):3000
       high-priority-cpu-mask:00000216,00002160
       high-priority-cpu-list:5,6,8,13,33,34,36,41
       low-priority-base-frequency(MHz):2400
       tjunction-temperature(C):125
       thermal-design-power(W):205
```

The above capabilities show that there are some CPUs on this system that can offer base frequency of 3000 MHz compared to the standard base frequency at this performance levels. Nevertheless, these CPUs are fixed, and they are presented via high-priority-cpu-list/high-priority-cpu-mask. But if this Intel(R) SST-BF feature is selected, the low priorities CPUs (which are not in high-priority-cpu-list) can only offer up to 2400 MHz. As a result, if this clipping of low priority CPUs is acceptable, then the user can enable Intel SST-BF feature particularly for the above "sched pipe" workload since only two CPUs are used, they can be scheduled on high priority CPUs and can get boost of 400 MHz.

Enable Intel(R) SST-BF

To enable Intel(R) SST-BF feature, execute:

```
# intel-speed-select base-freq enable -a
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     base-freq
       enable:success
package-1
 die-0
```

```
    cpu-14
      base-freq
        enable:success
```

In this case, -a option is optional. This not only enables Intel(R) SST-BF, but it also adjusts the priority of cores using Intel(R) Speed Select Technology Core Power (Intel(R) SST-CP) features. This option sets the minimum performance of each Intel(R) Speed Select Technology - Performance Profile (Intel(R) SST-PP) class to maximum performance so that the hardware will give maximum performance possible for each CPU.

If -a option is not used, then the following steps are required before enabling Intel(R) SST-BF:

- Discover Intel(R) SST-BF and note low and high priority base frequency
- Note the high priority CPU list
- Enable CLOS using core-power feature set
- Configure CLOS parameters. Use CLOS.min to set to minimum performance
- Subscribe desired CPUs to CLOS groups

With this configuration, if the same workload is executed by pinning the workload to high priority CPUs (CPU 5 and 6 in this case):

```
#taskset -c 5,6 perf bench -r 100 sched pipe
# Running 'sched/pipe' benchmark:
# Executed 1000000 pipe operations between two processes
    Total time: 5.627 [sec]
      5.627922 usecs/op
        177685 ops/sec
```

This way, by enabling Intel(R) SST-BF, the performance of this benchmark is improved (latency reduced) by 7.79%. From the turbostat output, it can be observed that the high priority CPUs reached 3000 MHz compared to 2600 MHz. The turbostat output:

```
#turbostat -c 0-13 --show Package,Core,CPU,Bzy_MHz -i 1
Package        Core    CPU     Bzy_MHz
0              0       0       2151
0              1       1       2166
0              2       2       2175
0              3       3       2175
0              4       4       2175
0              5       5       3000
0              6       6       3000
0              7       7       2180
0              8       8       2662
0              9       9       2176
0              10      10      2175
0              11      11      2176
0              12      12      2176
0              13      13      2661
```

Disable Intel(R) SST-BF

To disable the Intel(R) SST-BF feature, execute:

```
# intel-speed-select base-freq disable -a
```

Intel(R) Speed Select Technology - Turbo Frequency (Intel(R) SST-TF)

This feature enables the ability to set different "All core turbo ratio limits" to cores based on the priority. By using this feature, some cores can be configured to get higher turbo frequency by designating them as high priority at the cost of lower or no turbo frequency on the low priority cores.

For this reason, this feature is only useful when system is busy utilizing all CPUs, but the user wants some configurable option to get high performance on some CPUs.

The support of Intel(R) Speed Select Technology - Turbo Frequency (Intel(R) SST-TF) depends on the Intel(R) Speed Select Technology - Performance Profile (Intel SST-PP) performance level configuration. It is possible that only a certain performance level supports Intel(R) SST-TF. It is also possible that only the base performance level (level = 0) has the support of Intel(R) SST-TF. Hence, first select the desired performance level to enable this feature.

In the system under test here, Intel(R) SST-TF is supported at the base performance level 0, but currently disabled:

```
# intel-speed-select -c 0 perf-profile info -l 0
Intel(R) Speed Select Technology
package-0
 die-0
   cpu-0
     perf-profile-level-0
       ...
       ...
       speed-select-turbo-freq:disabled
       ...
       ...
```

To check if performance can be improved using Intel(R) SST-TF feature, get the turbo frequency properties with Intel(R) SST-TF enabled and compare to the base turbo capability of this system.

Get Base turbo capability

To get the base turbo capability of performance level 0, execute:

```
# intel-speed-select perf-profile info -l 0
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-0
     perf-profile-level-0
       ...
       ...
       turbo-ratio-limits-sse
         bucket-0
           core-count:2
           max-turbo-frequency(MHz):3200
         bucket-1
           core-count:4
           max-turbo-frequency(MHz):3100
         bucket-2
           core-count:6
           max-turbo-frequency(MHz):3100
         bucket-3
           core-count:8
           max-turbo-frequency(MHz):3100
         bucket-4
           core-count:10
           max-turbo-frequency(MHz):3100
         bucket-5
           core-count:12
           max-turbo-frequency(MHz):3100
         bucket-6
           core-count:14
           max-turbo-frequency(MHz):3100
         bucket-7
           core-count:16
           max-turbo-frequency(MHz):3100
```

Based on the data above, when all the CPUS are busy, the max. frequency of 3100 MHz can be achieved. If there is some busy workload on cpu 0 - 11 (e.g. stress) and on CPU 12 and 13, execute "hackbench pipe" workload:

```
# taskset -c 12,13 perf bench -r 100 sched pipe
# Running 'sched/pipe' benchmark:
# Executed 1000000 pipe operations between two processes
    Total time: 5.705 [sec]
      5.705488 usecs/op
        175269 ops/sec
```

The turbostat output:

```
#turbostat -c 0-13 --show Package,Core,CPU,Bzy_MHz -i 1
Package        Core    CPU     Bzy_MHz
0              0       0       3000
0              1       1       3000
0              2       2       3000
0              3       3       3000
0              4       4       3000
0              5       5       3100
0              6       6       3100
0              7       7       3000
0              8       8       3100
0              9       9       3000
0              10      10      3000
0              11      11      3000
0              12      12      3100
0              13      13      3100
```

Based on turbostat output, the performance is limited by frequency cap of 3100 MHz. To check if the hackbench performance can be improved for CPU 12 and CPU 13, first check the capability of the Intel(R) SST-TF feature for this performance level.

Get Intel(R) SST-TF Capability

To get the capability, the "turbo-freq info" command can be used:

```
# intel-speed-select turbo-freq info -l 0
Intel(R) Speed Select Technology
```

```
Executing on CPU model: X
package-0
 die-0
   cpu-0
     speed-select-turbo-freq
         bucket-0
           high-priority-cores-count:2
           high-priority-max-frequency(MHz):3200
           high-priority-max-avx2-frequency(MHz):3200
           high-priority-max-avx512-frequency(MHz):3100
         bucket-1
           high-priority-cores-count:4
           high-priority-max-frequency(MHz):3100
           high-priority-max-avx2-frequency(MHz):3000
           high-priority-max-avx512-frequency(MHz):2900
         bucket-2
           high-priority-cores-count:6
           high-priority-max-frequency(MHz):3100
           high-priority-max-avx2-frequency(MHz):3000
           high-priority-max-avx512-frequency(MHz):2900
         speed-select-turbo-freq-clip-frequencies
           low-priority-max-frequency(MHz):2600
           low-priority-max-avx2-frequency(MHz):2400
           low-priority-max-avx512-frequency(MHz):2100
```

Based on the output above, there is an Intel(R) SST-TF bucket for which there are two high priority cores. If only two high priority cores are set, then max. turbo frequency on those cores can be increased to 3200 MHz. This is 100 MHz more than the base turbo capability for all cores.

In turn, for the hackbench workload, two CPUs can be set as high priority and rest as low priority. One side effect is that once enabled, the low priority cores will be clipped to a lower frequency of 2600 MHz.

Enable Intel(R) SST-TF

To enable Intel(R) SST-TF, execute:

```
# intel-speed-select -c 12,13 turbo-freq enable -a
Intel(R) Speed Select Technology
Executing on CPU model: X
package-0
 die-0
   cpu-12
     turbo-freq
       enable:success
package-0
 die-0
   cpu-13
     turbo-freq
       enable:success
package--1
 die-0
   cpu-63
     turbo-freq --auto
       enable:success
```

In this case, the option "-a" is optional. If set, it enables Intel(R) SST-TF feature and also sets the CPUs to high and low priority using Intel Speed Select Technology Core Power (Intel(R) SST-CP) features. The CPU numbers passed with "-c" arguments are marked as high priority, including its siblings.

If -a option is not used, then the following steps are required before enabling Intel(R) SST-TF:

- Discover Intel(R) SST-TF and note buckets of high priority cores and maximum frequency
- Enable CLOS using core-power feature set - Configure CLOS parameters
- Subscribe desired CPUs to CLOS groups making sure that high priority cores are set to the maximum frequency

If the same hackbench workload is executed, schedule hackbench threads on high priority CPUs:

```
#taskset -c 12,13 perf bench -r 100 sched pipe
# Running 'sched/pipe' benchmark:
# Executed 1000000 pipe operations between two processes
    Total time: 5.510 [sec]
      5.510165 usecs/op
        180826 ops/sec
```

This improved performance by around 3.3% improvement on a busy system. Here the turbostat output will show that the CPU 12 and CPU 13 are getting 100 MHz boost. The turbostat output:

```
#turbostat -c 0-13 --show Package,Core,CPU,Bzy_MHz -i 1
Package        Core    CPU     Bzy_MHz
...
```

```
0               12      12      3200
0               13      13      3200
```