Intel Uncore Frequency Scaling

Copyright: © 2022 Intel Corporation
Author: Srinivas Pandruvada <srinivas.pandruvada@linux.intel.com>

Introduction

The uncore can consume significant amount of power in Intel's Xeon servers based on the workload characteristics. To optimize the total power and improve overall performance, SoCs have internal algorithms for scaling uncore frequency. These algorithms monitor workload usage of uncore and set a desirable frequency.

It is possible that users have different expectations of uncore performance and want to have control over it. The objective is similar to allowing users to set the scaling min/max frequencies via cpufreq sysfs to improve CPU performance. Users may have some latency sensitive workloads where they do not want any change to uncore frequency. Also, users may have workloads which require different core and uncore performance at distinct phases and they may want to use both cpufreq and the uncore scaling interface to distribute power and improve overall performance.

Sysfs Interface

To control uncore frequency, a sysfs interface is provided in the directory: */sys/devices/system/cpu/intel_uncore_frequency/*.

There is one directory for each package and die combination as the scope of uncore scaling control is per die in multiple die/package SoCs or per package for single die per package SoCs. The name represents the scope of control. For example: 'package_00_die_00' is for package id 0 and die 0.

Each package_*_die_* contains the following attributes:

`initial_max_freq_khz`
> Out of reset, this attribute represent the maximum possible frequency. This is a read-only attribute. If users adjust max_freq_khz, they can always go back to maximum using the value from this attribute.

`initial_min_freq_khz`
> Out of reset, this attribute represent the minimum possible frequency. This is a read-only attribute. If users adjust min_freq_khz, they can always go back to minimum using the value from this attribute.

`max_freq_khz`
> This attribute is used to set the maximum uncore frequency.

`min_freq_khz`
> This attribute is used to set the minimum uncore frequency.

`current_freq_khz`
> This attribute is used to get the current uncore frequency.