**TESTING AND ANALYZING AFFECT OF BIOS CHANGES ON PER CORE FREQUENCY**

**WHILE STRESS TESTING CPU CORES**

Among the myriad of BIOS parameters that can be varied, a system’s per core frequency can be proven to be highly influenced by the Hyperthreading and Intel Turbo Boost settings. Hyperthreading is used to enhance parallelism by primarily allocating 2 threads per core rather than 1. Since CPUs don’t always perform at maximum frequency, Intel turbo boost can be used to dynamically determine what scenarios ensue what frequencies. For example, for lighter workloads, a lesser CPU frequency may be assigned whereas for workloads demanding a higher CPU performance and hence clock speed, the CPU can run at a MAX TURBO FREQUENCY. This can definitely boost performance in scenarios where hyperthreading is enabled or disabled.

Seeing as to how the above settings can influence the system’s per core operating frequency quite significantly, it can prove to be quite insightful to experiment with how the system under test behaves (i.e. at what frequency it operates at) while being stress tested with different BIOS settings per core.

Here we have experimented with a particular scenario:

* Hyperthreading enabled
* Turbo Boost disabled

(The above system settings can be easily verified by using the Intel svr-info tool which consolidates various system characteristics into a single repository).

A computer screen capture

Description automatically generated with low confidence

The above screenshot shows the lscpu output of the system we used. We have 2 sockets with 18 cores each and Hyperthreading enabled, which gives us 72 cores totally.

The min frequency is 1000 MHz and max frequency is 2200 MHz.

**Tools Used for the experiment:**

1. **Turbostat** can be used to measure various statistics like frequency, idle power-state statistics, temperature power etc of the system.
   * Here, we are using to continuously monitor the per core frequency.
2. **S-tui** tool is used for stress testing the CPUs.
   * It can be in 2 modes: Monitor and Stress.
   * With this tool, one can visibly assess the stress on each core while applying a heavy load to the cores.

**Installation Steps:**

$ sudo apt update

$ sudo apt upgrade

$ sudo apt install pip

$ sudo pip install s-tui

$ sudo apt install stress

$ sudo apt install linux-tools-common

*Execute the following with your kernel version (can be checked by executing $ uname -a)*

$ sudo apt-get install linux-tools-**your-kernel-version**

(ex: $ sudo apt-get install linux-tools-5.4.0-122-generic)

*Now open another terminal.*

*In terminal 1, execute:*

$ sudo s-tui

(you can navigate between Monitor and Stress mode with your arrow keys and hit the space bar to select a mode)

*And in terminal 2, execute:*

$ turbostat -c 0-71 --show Package,Core,CPU,Bzy\_MHz -i 1

*In the above command, -c represents the cores to stress test. As this system has 72 cores (described above) we provide the range 0-71. We want to be able to see the package,core,cpu and bzy\_Mhz (average non idle clock rate) which use –show for, at a reporting interval of 1 second.*

*We want to be able to*

1. The left portion of the below image shows terminal 1. In the top left corner, it can be seen that s-tui is in Monitor mode and the cpu cores are not being stressed.

The right portion of the below image shows terminal 2. We can see a representation of having no load being stress tested on the cpu cores as the per cire frequency is 1000 MHz (min freq).

Graphical user interface

Description automatically generated

1. The left portion of the below image shows Terminal 1. In the top left corner, it can be seen that the s-tui tool is in stress mode (activate by hitting space bar). The frequency and cpu utilization per core are at the peak values.

The right portion shows terminal 2. We can see that the per core operating frequency is now at 2200 MHz (which is the max frequency).

Graphical user interface

Description automatically generated