MTCS-102(P)
Advanced Architecture

Practical Assignment

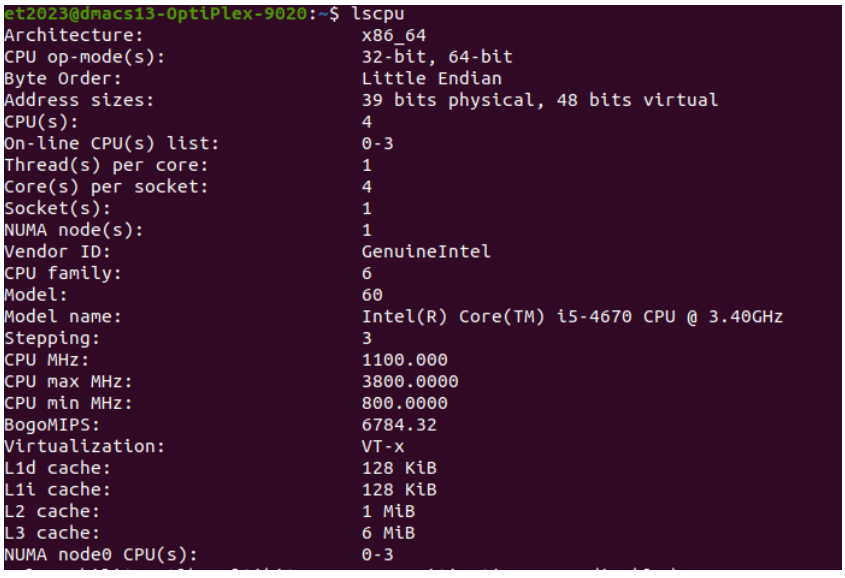
Reg. No.: 23361

Introduction:

Here in this experiment , we have seen the behavior of CPUs in terms of process scheduling and dynamic frequency scaling. We explored how processes are distributed across multiple CPUs, how CPU affinity can be controlled, and how dynamic frequency scaling affects CPU performance and power consumption.

Determining Number of CPUs:

We determined the number of CPUs on the system using the ***lscpu*** command. This revealed that there were four CPUs numbered from 0 to 3.



Observing Process Affinity:

To know the behavior of the cpu threads and the execution program, a code was run that contained a large loop which was run throughout this experiment.

We opened two terminals and ran a program named ***a.out*** on one terminal while monitoring process activity using the ***top*** command on the other terminal. The 'P' column in the ***top*** display indicated the process ID that was most recently run by each process.

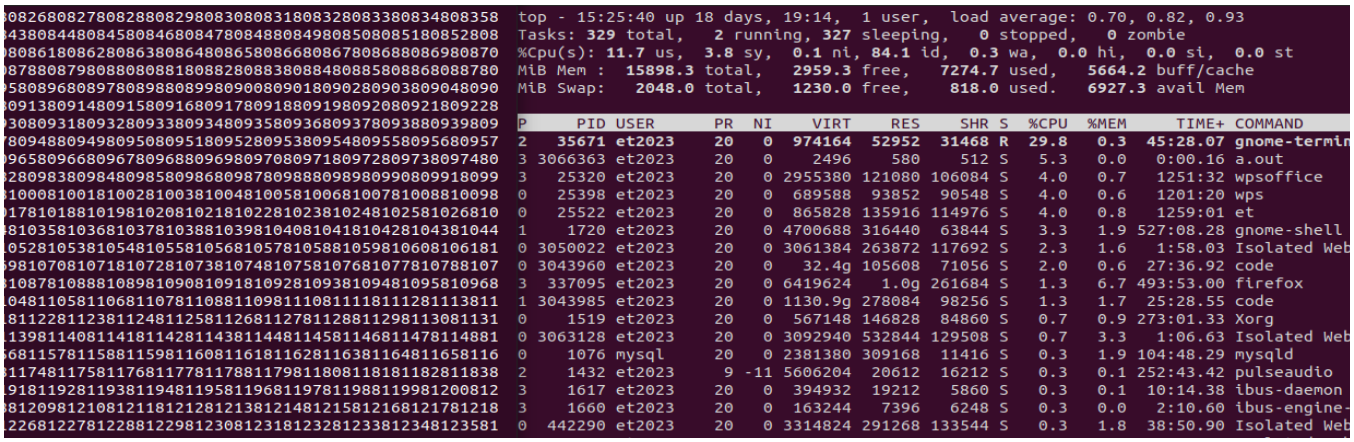


Image1

Process Affinity:

We observed that the program ***a.out*** was being executed on different CPUs at different points in time. The 'P' column in the ***top*** output reflected the process ID that was most recently run by each process. For instance, we noticed that the process corresponding to ***a.out*** switched between different CPUs when ran at two different instance of the execution of the same code(Image1 and Image2).

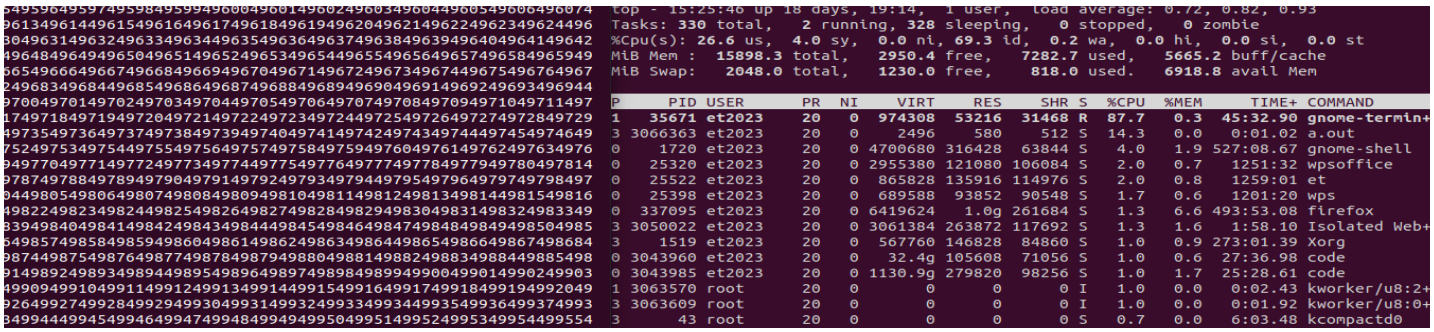
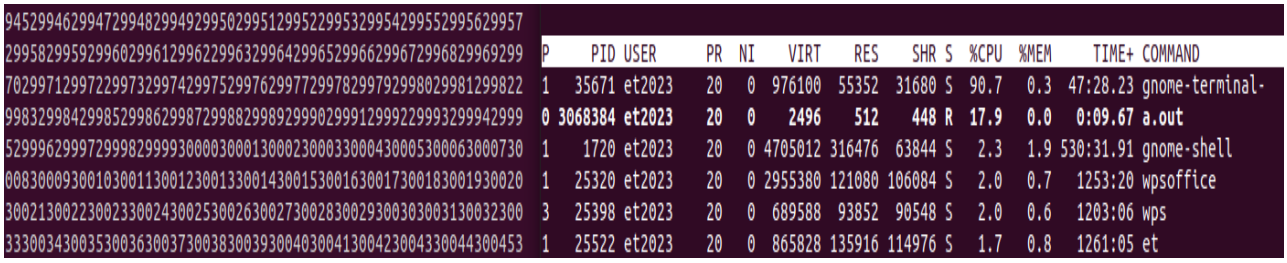


Image2

Restricting CPU Affinity:

To restrict the affinity of a particular CPU, we used the `taskset` command. For example, running the command ***`taskset -c 0 ./a.out`*** limited the execution of a `.out` to CPU 0. This resulted in the ***'P'*** column consistently displaying 0 for the ***`a.out`*** process, indicating that it was consistently running on CPU 0(Image3).



```
    P    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1  35671 et2023    20   0  976100  55352  31680 S  90.7   0.3  47:28.23 gnome-terminal-
    0 3068384 et2023    20   0    2496    512    448 R  17.9   0.0   0:09.67 a.out
    1   1720 et2023    20   0 4705012 316476  63844 S   2.3   1.9 530:31.91 gnome-shell
    1  25320 et2023    20   0 2955380 121080 106084 S   2.0   0.7   1253:20 wpsoffice
    3  25398 et2023    20   0  689588  93852  90548 S   2.0   0.6   1203:06 wps
    1  25522 et2023    20   0  865828 135916 114976 S   1.7   0.8   1261:05 et
```

Image3

Dynamic Frequency Scaling:

Modern CPUs utilize dynamic frequency scaling to manage power consumption and heat generation. We investigated the dynamic nature of CPU frequency using the ***`/sys/devices/system/cpu/cpu0/scaling_cur_freq`*** file(Image4). This provided the current frequency of CPU 0 at various points in time. We found that the frequency wasn't fixed and changed based on the workload.

```
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1284906
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1374017
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1334410
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1252026
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1360201
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1273106
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1433344
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1354249
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1402504
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1322221
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_cur_freq
1377878
```

Image4

The range of possible frequencies was determined using the ***scaling_min_freq*** and ***scaling_max_freq*** commands.

```
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_min_freq
800000
```

```
et2023@dmacs13-OptiPlex-9020:/sys/devices/system/cpu/cpu0/cpufreq$ cat scaling_max_freq
3800000
```

Conclusion:

This experiment provided insights into the behavior of CPUs, process scheduling, and dynamic frequency scaling. We observed how processes are distributed across CPUs, how CPU affinity can be controlled, and how dynamic frequency scaling influences CPU performance and energy consumption. These observations shed light on the efficient management of CPU resources and power consumption in modern computing systems.