**Advanced Computer Architecture**

LAB2 – Get familiar with System-level Power Estimation Tool

**Deadline: 2024/12/20**

Student : M11215075, 胡劭

**1 Introduction**

You need to have access to a Unix/Linux computer to be able to perform the laboratory exercises. All installation instructions refer to Linux distributions, but they may work on Sun/Solaris or PC/Solaris installations as well.

**※ Note. Server IP: 140.118.115.163**

**The account and password to access the server is your student ID number. You may change the password using the following command: passwd <yourID i.e Old password><New password>**

**2 Objective of the laboratory exercise**

After this laboratory exercise, you should:

․Understand how program behavior relates to power consumption and performance.

․Be able to make tradeoffs related to save more power usage.

․Understand the bottleneck of a program and how to improve the performance.

**3 Powertop assignments**

**3‐1 Program behavior**

Run the program pi, test1 and make an observation with top.

Table 1: process state in top.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| process | VIRT | RES | SHR | S | %CPU | Time+ |
| Pi | 2200 | 548 | 504 | R | 11.7 | 0:02.45 |
| Test1 | 2068 | 544 | 500 | R | 11.4 | 0:01.88 |

**Q3‐1.1** What is the difference in process state S(D、R、S、T、Z)?

Ans: D (TASK\_UNINTERRUPTIBLE):

* A sleep state where the process cannot be interrupted
* Typically occurs during I/O operations
* The process cannot respond to any signals or be killed
* Must wait for specific conditions to be met before waking up

R (TASK\_RUNNING):

* The process is either currently executing or waiting in the run queue
* The only state where a process can be executed on the CPU
* Includes both actively running processes and those ready to run

S (TASK\_INTERRUPTIBLE):

* A sleep state where the process can be interrupted
* Waiting for certain conditions (e.g., keyboard input, network data) to be met
* Can be awakened by signals

T (TASK\_STOPPED or TASK\_TRACED):

* The process execution is suspended
* Usually caused by receiving a SIGSTOP signal
* May indicate the process is being debugged

Z (TASK\_DEAD - EXIT\_ZOMBIE):

* The process has terminated but remains in the process table
* Waiting for its parent process to read its exit status
* Retains minimal information for the parent process to retrieve

**Q3‐1.2** Try to use powertop calibrate function and generate a csv report. Using this report to fill the Table 2 with total power consumption(wakeups/second) and top four usage process.

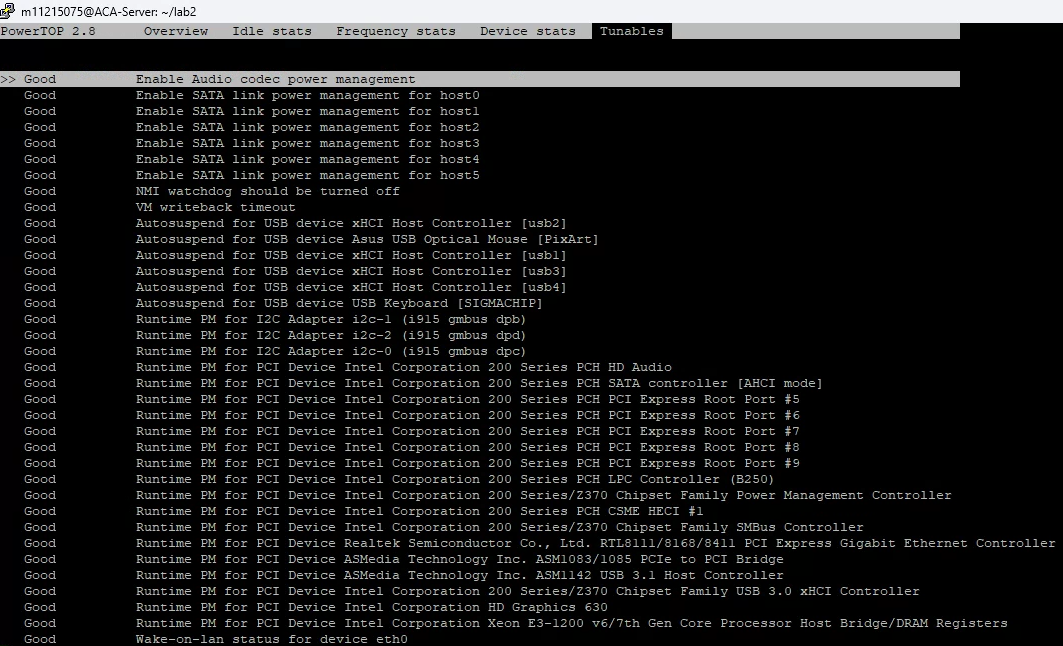
Table 2: powertop csv report

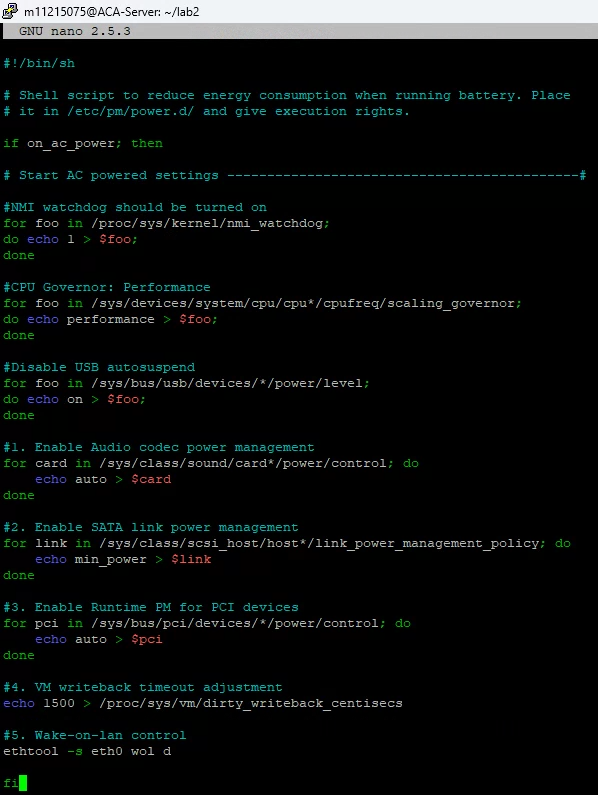
|  |  |  |  |
| --- | --- | --- | --- |
| total power consumption | | ????(wakeups/second) | |
| Usage | Category | Description | Events/s |
| 402% | Process | ./pi | 0.00 |
| 263% | Process | ./test1 | 0.00 |
| 24.6% | Process | /usr/lib/linux-tools/4.4.0-210-generic/perf top -p 25823 | 0.00 |
| 16.7% | Process | /home/m11302129/lab1/simplescalar/simplesim-3.0/sim-outorder basicmath\_small | 0.00 |
| 9.3% | Process | ./pi –user=e11302007 | 0.00 |
| 7.0% | Process | .p1 ^ | 0.00 |
| 5.0% | Process | htop | 0.00 |
| 0.1% | Process | /usr/lib/vino/vino-server --sm-disable | 0.00 |
| 2.3% | Process | powertop | 0.00 |
| 0.5% | Process | top | 0.00 |

**4 Powertop power-saving policy**

Because powertop tunable page can't be store permanently. Try to add five more power saving policy in file "power" which is located in your folder. You have to make a brief explain to each policy. (Can just Print Screen and Paste below)

Ans:

**Power-saving policy**  




1. **Enable Audio codec power management**

* Purpose: Reduces power consumption of audio hardware components
* When enabled, audio devices will automatically enter low-power mode when not in use
* Particularly effective for laptops and systems with integrated audio

1. **Enable SATA link power management**

* Purpose: Optimizes power usage of SATA connections to storage devices
* Allows SATA links to enter lower power states during periods of inactivity
* Helps reduce overall system power consumption without impacting performance

1. **Enable Runtime PM for PCI devices**

* Purpose: Implements dynamic power management for PCI devices
* Automatically puts inactive PCI devices into low-power states
* Particularly beneficial for systems with multiple PCI devices like graphics cards and network adapters

1. **VM writeback timeout adjustment**

* Purpose: Optimizes the frequency of writing cached data to disk
* Setting it to 1500 centiseconds increases the interval between write operations
* Reduces disk activity and power consumption while maintaining system stability

1. **Wake-on-lan control**

* Purpose: Disables Wake-on-LAN functionality
* Prevents unnecessary power consumption by the network interface
* Especially useful when the system doesn't require remote wake-up capabilities

These policies work together to create a comprehensive power management strategy that reduces system power consumption while maintaining functionality. Each policy targets a specific hardware component or system function, allowing for granular control over power usage.

**5 Perf**

We will use the phonebook case to understand what perf tool can help us to improve program performance. There are two method can let us find the right name in phonebook. The only different place is how they claim their entry structure. But in this observation, you can find the original one cost more time and also get lots of cache misses. Using this perf tool to fill the Table 3.

Table 3: Cache miss and performance with phonebook case.

|  |  |  |
| --- | --- | --- |
|  | % of FindName | % of all cache ref |
| Main\_origin | 20.97% | 91.738% |
| Main\_optimal | 11.43% | 58.485% |

**Q5.1** What is the key point let these two method can get different performance.

Ans: The size of entry in main\_origin is 128 bytes, and the size of entry in main\_optimal is 24 bytes. This significant size difference leads to clear performance differences as shown by the perf statistics:

1. FindName function overhead:

* main\_origin spends 20.97% of execution time in FindName
* main\_optimal only spends 11.43% of execution time in FindName

1. Cache efficiency:

* main\_origin has a high cache reference rate of 91.738%
* main\_optimal achieves better cache efficiency with only 58.485% cache references

These numbers clearly demonstrate that the smaller entry structure in main\_optimal results in better memory access patterns and CPU cache utilization.

**6 Conclusions**

At the end of this laboratory exercise, you should be able to answer following questions:

**Q6.1** What benefits can we obtain when using powertop?

Ans: Powertop provides several key benefits:

1. It can estimate the total power consumption resulting from system usage and monitor individual processes separately.
2. It helps us identify power-draining process.es and components.
3. It can help us save power by suggesting and implementing power saving policies.
4. It provides real-time monitoring of system power usage through its interactive interface.
5. It can generate detailed reports for power consumption analysis.

**Q6.2** What benefits can we obtain when using perf?

Ans: Perf offers multiple advantages for performance analysis:

1. It can analyze the performance of different processes in detail.
2. It supports various performance monitoring features:

* Hardware performance counters
* Tracepoints
* Software performance counters (e.g. hrtimer)
* Dynamic tracing probes (kprobes/uprobes)

1. It helps identify performance bottlenecks through cache misses and CPU cycle analysis.
2. It provides detailed statistical data for program optimization.

If you have any problems, please contact with the TA.

E-mail: D10602805@mail.ntust.edu.tw (Walle Haileeyesus Engdaw)