Assignment:1

CSL3020: Computer Architecture
AY 2024-25, Semester – V
Due on: 17-08-2024

Total:50 Marks

General Instructions:

1. Clearly mention the assumptions you have made, if any.
2. Clearly report any resources you have used while attempting the assignment.
3. Any submission received in another format or after the deadline will not be evaluated.
4. Make sure to add references to the resources that you have used while attempting the assignment.
5. Plagiarism of any kind will not be tolerated and will result in zero marks.
6. Select the correct and working program for the comparison.

Submission Guidelines:

1.Submit a single report depicting comparisons, methods, results, and observations. Preparing a report is mandatory; failing it will lead to non-evaluation of the assignment.

2. Name your report as YourRollNo.pdf. And your program codes as yourRollNo.py/ yourRollNo.c/yourRollNo.cpp,

3. There is no need to make a zip file. Just upload the report and program directly on the google-classroom, that is, submission will contain{YourRollNo.pdf,YourRollNo.py/c/c++}. Do not upload files in any other format.

4. Do not copy-paste code or screenshots, etc. in the report. The report should look like a technical document, containing plots, tables, etc. whenever necessary.

5. Adhere to the instructions given, failing them may result in a penalty.

Performance Analysis Using `perf`

Objective

In this assignment, you will use the `perf` tool to analyze the performance of a program/commands. You will capture various performance metrics, interpret the data, and understand how these metrics reflect the program's behavior on the hardware.

Perf Wiki (Official Documentation)

Requirements

- **Linux Machine**: Ensure you have a Linux system with `perf` installed.'
- **Python/c/c++ program**
- **Compiler/Interpreter of your selected programming language**

Notes:

You need to adjust the security and access control settings for performance monitoring on a Linux system. Clearly explain the variables involved, the modifications made to the kernel, and the reasons for these changes in the context of performance evaluation.

Create or find a Python, C, or C++ script that performs a CPU-intensive task. For example, the script could handle matrix multiplication, sort a large list, or compute the Fibonacci sequence recursively.

You'll earn bonus marks if your program is computationally demanding, including floating point operations, recursions and distinct from all of your classmates' submissions. [10] (Marks will be awarded in this section only if you complete the task outlined below.)

<u>You need to compare the performance of the program/commands in two or more different linux machines.</u>
Key metrics to analyze:

- **Task Clock:**
- **Context Switches:**
- **CPU Migrations:**
- **Page Faults:**
- **Cycles:**
- **Stalled Cycles:**
- **Instructions:**
- **Branches:**
- **Branch Misses:**

Your Task:

Use perf to capture performance metrics by executing your script and saving the performance statistics to a text file. Make sure your script runs for more than 1 second. Then, run the same program on different Linux machines and compare the performance results using the metrics mentioned above. **[15]**

No need to submit the .txt file of perf results.

Identify potential bottlenecks and areas for optimization, providing a clear explanation of how and where optimization is needed. Specify which computational aspect in function is consuming the most resources and specify the reasons behind it. **[15]**

Analyze the **resource usage(**using above metrics**)** of 'linux commands' (ls, pwd, cd, etc.) you can choose any 2 commands for performance evaluation using perf.   **[10]**

1. **Create and Submit Your Report along with the program's code files.**
   ○ **Summarize your findings based on the performance metrics.**
   ○ **Compare and analyze the performance metrics of same programs run on two or more different linux machines. Mention the configurations of the selected machines.**
   ○ **Discuss how these metrics reflect the performance characteristics of your programs.**
   ○ **Include observations on CPU utilization, instruction execution efficiency, and potential bottlenecks.**
   ○ **Screenshots or text excerpts from the `perf` report.**
   ○ **Conclusions and any recommendations for performance improvements.**
2. **Optional: Visualize the Data**

Evaluation Criteria

- **Correctness: Accurate execution and analysis of the `perf` tool.**
- **Comprehensiveness: Thorough analysis of performance metrics.**
- **Clarity: Clear and well-organized report presentation.**
- **Insightfulness: Depth of understanding and interpretation of performance data.**