ES 215 Interim Project Report

Bhavik Patel, 22110047 Hitesh Kumar, 22110098
Ruchit Jagodara, 22110102 Jinil Patel, 22110184

October 22, 2024

COA Educational GUI Tool

Project Overview

The COA Educational GUI Tool project aims to create an interactive, web-based tool to facilitate the learning of key concepts in Computer Organization and Architecture (COA). The tool will provide visualizations and simulations on topics such as CPU performance metrics, cache replacement policies, and number system conversions. The project's objective is to enhance learning by offering practical visual insights, interactive elements, and real-time analysis of COA concepts.

In response to the feedback provided on our project proposal, we have updated the project overview to incorporate the following points.

1. **Performance Metrics Analyzer:** It will evaluate and help in visualizing the CPU performance metrics like Speed Up, Throughput, Latency, etc. It will also provide a comparison of different CPUs (which is faster?) given the specifications (CPI, Instruction Count, Clock Cycle, etc.). The visualization will include how the parallelized part is divided among different processors and how all of the results are combined. The graphical visualization will also be implemented.
   In response to the feedback on our Performance Metrics Analyzer, we will also be implementing efficiency and power consumption modeling as part of our CPU performance analysis.

2. **Cache Replacement Visualizer:** It will simulate and help visualize the cache replacement policies to illustrate their impact on system performance using changeable parameters, including Cache Size, Block Size, Access Patterns, etc.

   Cache Levels:

   We will be considering two levels of caches for simplicity and will be showing how Caches works and what happens if there is a miss or hit in a particular level of cache, how they are connected with memory, etc.

   - **L1 Cache:**
     A small, very fast cache located closest to the CPU. It is typically split into instruction cache and data cache to handle specific operations more efficiently.
   - **L2 Cache:**
     Larger than L1 but slower. It may be dedicated per core or shared across multiple cores and serves as a backup when the L1 cache misses.

Replacement Policies:

- **LRU (Least Recently Used):**
  This policy replaces the cache block that has not been accessed for the longest time.
- **FIFO (First-In, First-Out):**
  Replaces the block that has been in the cache the longest, without considering access frequency.
- **LFU (Least Frequently Used):**
  Replaces the cache block that has been accessed the fewest times.
- **Random Replacement:**
  Randomly selects a cache block to replace when the cache becomes full.

Visualization Features:

- Select the cache level (L1, L2) with corresponding descriptions.
- Choose a replacement policy from a dropdown menu (LRU, FIFO, LFU, Random).
- Adjust key parameters such as cache size, block size, and access patterns using sliders or text fields.

3. **Number Conversion Simulator:** This tool is designed to facilitate inter-conversion between various number representations, including binary, decimal, octal, hexadecimal, 1's complement, 2's complement, BCD, IEEE floating point single/double precision numbers. Users will be able to input the number and see the step-by-step process for each conversion.

The tool will also include a proper user guide and documentation for the users to get information about the tool. There will be a 'Study Material' section where the user can learn about different COA concepts and get a better understanding of Computer Organization and Architecture.

Current Progress

The team has successfully made progress on various aspects of the project over the past three weeks, following the planned timeline outlined in the project proposal. Below is a summary of the activities carried out so far:

1. Front-End Design and Framework Setup:
   (a) Initial design mock-ups of the web-based GUI have been completed, incorporating essential navigation features.
   (b) The 'basic layout' and 'structure' for the three main components of the tool—Performance Metrics Analyzer, Cache Replacement Visualizer and Number Conversion Simulator have been established.
   (c) Libraries such as React.js, Plotly, and D3.js have been integrated into the development environment to allow for seamless interaction and visualizations.

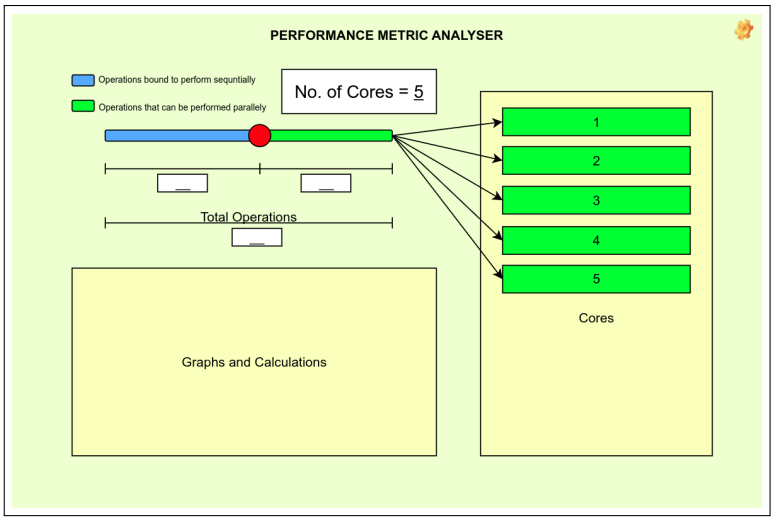

Figure 1: Basic layout of Performance Metric Analyser

2. Development of Tools
   We have developed main webpage of the web application and integrated it with the three components.

   (a) **Performance Metrics Analyzer:** Initial algorithms have been developed to calculate and visualize CPU metrics such as speedup, throughput, and latency. User inputs for parameters such as CPI (Cycles Per Instruction), clock cycle, and instruction count have been implemented to dynamically calculate the results. Working on how to display the results for better understanding of the concepts.

   (b) **Cache Replacement Visualizer:** Work has begun on simulating cache replacement policies. The tool currently allows users to adjust parameters such as cache size, block size, and access patterns. The visualization part is in development.

   (c) **Number System Conversion Simulator:** The number system conversion module is partially functional (need to fix some bugs). It currently supports conversions between binary, decimal, octal and hexadecimal systems with plans to expand it to include, BCD, IEEE floating-point single/double precision and other important representations. We are exploring different approaches to simulate the conversion process within a web environment.

Objectives Achieved

1. A basic front-end layout for the web-based GUI has been created and is fully functional, with navigational features to access different modules of the tool.

2. Performance Metrics Analyzer is partially implemented, and users can now input CPU specifications to visualize key performance metrics.

3. The Cache Replacement Visualizer allows for parameter adjustments, with ongoing work on graphical representations and visualization of the replacement policies.

4. The Number Conversion Simulator supports conversions between binary, decimal, and hexadecimal, with more conversions in progress.

Individual Contributions

1. **Bhavik Patel :** Coordinated the integration of the various components, ensuring that the modules being developed fit cohesively into the web application. Contributing to the design and layout of the main page of the web application.
2. **Hitesh Kumar :** Led the implementation of the Performance Metrics Analyzer component and is working on setting up the interactive visualizations for CPU metrics.
3. **Ruchit Jagodara :** Focused on the development of the Number Conversion Simulator, ensuring smooth integration with the front-end. Contributing to the back-end logic for handling different number systems.
4. **Jinil Patel :** Helped develop the main page and is working on developing the core logic for the Cache Replacement Policy.

Challenges Faced

1. Learning the libraries like react, node.js, etc. for the first time is a bit of a hectic and time consuming task.
2. Thinking about how to present the data to the user so that the user can get the most out of the data in less time.

Plan of Work for Next Stage Development

In the upcoming weeks, the team will focus on the following tasks:

- Finalize the Performance Metrics Analyzer by implementing parallelized computation visualization and efficiency modeling.
- Complete the Number Conversion Simulator by adding for IEEE floating-point numbers, BCD, and other representations.
- Conduct integration testing to ensure that all modules interact seamlessly within the web application.

Tentative Timeline for the Project

This is the tentative timeline for the project's workflow. The heavy part of the project will remain in the starting phase. Near the end of the project, only the testing and deployment part will be left to manage the workload from other courses, as the workload increases with the progress of the course.

- **Week 1,2: Front-end Design and Framework Setup** (Completed With No-Backlog)
  - Prepare initial design mock-ups for the web-based GUI.
  - Implement the basic layout and navigation of the web application.
  - Set up of development environment and integration of necessary libraries.

- **Week 3,4: Development of Tools** (Ongoing)
  - Development of Python algorithms for required tools.
  - Implementation interactive visualizations using Plotly and D3.js
  - Integration of Cache Simulation features into the front-end.

——— Mid-Sem Break ———

- **Week 5,6: Integration and Initial Testing**
  - Integration of all the modules into the cohesive web application.
  - Adding User Guide, Documentation, and Study Material.
  - Conducting initial integrated tool testing to identify and fix any major issues.
- **Week 7,8: Final Testing and Documentation**
  - Refining features and bug fixing.
  - Deployment of the web application.

Conclusion

The project is progressing as planned, with significant portions of the tool's core functionality developed. The next phase will involve completing the remaining modules, integrating them fully, and conducting initial testing. Overall, the team is on track to meet the project objectives and deliver an interactive, educational tool for COA concepts.