Assignment 4. Monte Carlo Methods – CPU Benchmarking for High-Performance Desktop Selection

Ryan Fallon

Abstract

This study investigates CPU benchmarking to identify optimal processors for high-performance Linux desktops. Leveraging raw data from openbenchmarking.org, I used various CPU-intensive benchmarks to assess processor efficiency across scenarios. By combining pre-collected data and Monte Carlo simulations, this research evaluates metrics like (I imagine as my chosen tests abstract specifics) execution time, frames per second, and operations per second for a comprehensive view of processor capabilities. Through structured analysis using Go, I aim to present practical insights for making informed CPU choices for demanding workflows.

Introduction

In selecting hardware for high-performance computing tasks, CPU benchmarking plays a critical role, particularly for Linux desktops. The goal of this research is to examine CPU capabilities across processors to determine which models best handle a range of workloads, including rendering, data compression, browser tasks, and cryptographic functions. This benchmarking analysis not only guides individual hardware selection but also informs broader industry choices about efficient processor utilization.

Literature Review

Benchmarking studies in computer science have historically focused on identifying the best-suited hardware configurations for specific applications. Openbenchmarking.org and the Phoronix Test Suite have become valuable resources for gathering data across processors and providing a standardized approach for performance assessment. Previous research, such as studies on browser performance using Octane or Speedometer and cryptographic algorithm efficiency, has shed light on individual performance metrics but often lacks real-world synthesis. This project contributes to the field by utilizing these benchmarks to offer comprehensive recommendations based on aggregate performance indicators across different processors.

Methods

Data Collection and Processing

For this study, I utilized a pre-existing dataset from openbenchmarking.org, covering benchmarks like Speedometer, Octane, Jetstream, and Linux kernel compilation times. The dataset was exported from openbenchmarking and processing was straightforward as it just involved deleting extraneous rows/columns from the CSV export.

Monte Carlo Simulation Setup

Monte Carlo simulations were applied to randomize workload scenarios, providing insights into expected CPU performance under varying conditions. Using Go, I generated random samples from the dataset to simulate different operational environments. For example, I ran a simulation on varied browser tasks (Jetstream benchmarks in Chrome) allowing for the collection of performance distributions that reveal the Ryzen 7 8700G and Ryzen 7 7700X processors' ability to handle real-world workloads.

Benchmarking Execution

The benchmarking process, uses public results from openbenchmarking.org. Taking the outputs of those tests, I wrote code in Go (run on a MacBook Pro) to simulate 1000 runs based on returned mean and standard error. The higher-is-better performance metrics were taken and recorded.

Response Variables

The primary response variables include:

- **Score:** Selenium Jetstream 2 – Browser: Google Chrome benchmark testing metric; more is better
- **Operations per Second (OPS):** bytes/second for OpenSSL 3.3 AES-256-GCM; more is better

Each metric was logged with mean and SE to ensure organized and reproducible results and the ability to run these metrics through a Monte Carlo simulation.

Results

Through the benchmarks and Monte Carlo simulations, I evaluated how each processor handles different types of workloads. Detailed results can be found in my public repository. These results indicate that at similar price points (for my range) the Ryzen 7 7700X outclasses the Ryzen 7 8700G in both selected tests. For people with other price points or needs, by switching out the mean/SE values, one could adjust this analysis. Thus, with minor modifications one could highlight the trade-offs in processor performance, with certain CPUs excelling in browser-related tasks, while others perhaps show higher efficiency in compilation and rendering tasks. All the mean/SE data is available in the above public results link to run different Monte Carlo simulations across different processor/benchmarking combos.

Implementation Timelines

- **Time to Implement Benchmark:** The setup for processing and simulating the data was negligible having used outputs from openbenchmarking.org.
- **Data Loading Time:** The data loading time was negligible as I was able to download results to a CSV.
- **Data Extraction Time:** The data extraction time was negligible as I was able to download results to a CSV.

Discussion

Extent of Industry Utilization and Market Shares

Processors benchmarked in this study are widely used in the tech industry, particularly in data centers, high-performance computing, and workstation setups. The documentation and training resources available for each tool vary, with openbenchmarking.org providing detailed information for setup and execution of the benchmarks, while Go offers comprehensive libraries and documentation for data manipulation and analysis.

Data Science Feasibility

For a data scientist, working with openbenchmarking data and the Phoronix Test Suite is manageable due to the structured documentation and established benchmarks. The Phoronix Test Suite's accessibility enable a seamless benchmarking process. Go was a great choice for its computationally efficient and human-readable code, and it was east to run Monte Carlo simulations based on the outputs of these benchmarking tests.

Hypothetical Management Problem

This study addresses the challenge of choosing the most cost-effective and efficient CPU setup for computationally intensive applications, such as graphics rendering or real-time data compression. With the growing demands for optimal hardware in fields like AI, gaming, and digital media, selecting the right CPU can directly impact a company's operational efficiency and hardware costs.

Implementation Costs

While costs can vary based on infrastructure, using the Phoronix Test Suite and openbenchmarking data provides a low-cost solution. Openbenchmarking.org is a free resource, and Go's open-source nature further reduces costs, making this benchmarking approach financially accessible.

Conclusions

In selecting the best CPU for a high-performance desktop, this study provides insights into processor capabilities across a range of real-world tasks. The Monte Carlo simulations revealed differences in CPU performance that are crucial for decision-making in contexts requiring intense computational power. This benchmark serves as a reliable framework for companies and individuals looking to optimize their hardware configurations.

Limitations and Future Work

The benchmarking analysis relied on a pre-existing dataset, which limits control over data collection. This limitation can be overcome by cloning the Phoronix Test Suite and running tests manually. Future studies could incorporate live testing for more accurate and diverse environmental conditions. If conducted again, expanding the scope to include memory and GPU performance could enhance understanding of optimal configurations for high-performance desktops.