Performance Analysis of Cache Size and Set-Associativity Using gem5

1st Clayton Asada
Computer Engineering and Computer Science
California State University Long Beach
Long Beach, USA
clayton.asada01@student.csulb.edu

2nd Joseph Chorbajian
Computer Engineering and Computer Science
California State University Long Beach
Long Beach, USA
email address

3rd Ilmaan Zia
Computer Engineering and Computer Science
California State University Long Beach
Long Beach, USA
email address

Benchmarks	Cache Size	Block Size	Set Associativity	Replacement Policy
Dijkstra	16kB	32	1	Least Recently Used
Sha	32kB	64	2	Random
Rijndael	64kB		4	First In First Out
			8	

***Abstract*—Keywords -** **Cache, Size, Associativity, Replacement Policy, gem5, Simulation, Configuration, Miss rate**

Fig. 1. Amount of training provided

I. Introduction

With the increasing demand for high-performance computing systems, cache design has become an important area of research in computer architecture. Caches are small, fast memory structures that are used to store frequently accessed data and instructions, thereby reducing the time required to access them from the main memory. The performance of a processor is highly dependent on the cache configuration, which includes parameters such as size, associativity, and replacement policy.

In recent years, computer architects have been using simulation tools to evaluate the performance of different cache configurations. One such tool is gem5, a cycle-accurate simulator that is widely used in the computer architecture community. gem5 is a modular simulation framework that supports a wide range of processors, including single-core CPUs, multi-core CPUs, and accelerators.

The use of gem5 to model various single-core CPU cache configurations has gained popularity in recent years. This approach allows researchers to evaluate the performance of different cache configurations under various workloads and benchmarks. With gem5, researchers can study the effects of cache size, associativity, replacement policy, and other cache parameters on performance metrics such as execution time, instruction throughput, and cache hit rate.

The benefits of using gem5 to model single-core CPU cache configurations are numerous. First, gem5 allows researchers to perform detailed analyses of cache behavior under different workloads, which is difficult to achieve through real-world experiments. Second, gem5 is highly configurable, which allows researchers to explore a wide range of cache configurations and compare their performance. Third, gem5 is open-source, which makes it accessible to researchers and allows for easy collaboration and sharing of research results.

In this research paper, we explore the use of gem5 to model various single-core CPU cache configurations. We first describe the gem5 simulator and its capabilities. We then review previous research on cache design and discuss the important cache parameters that are considered in this work. We present a detailed experimental methodology and describe the cache configurations that we have evaluated. Finally, we present our results and discuss the insights that we have gained from our experiments. Our work demonstrates the power and flexibility of gem5 as a simulation tool for cache design research.

II. Motivation

Cache design is a critical area of research in computer architecture, as it has a significant impact on the performance of modern computer systems. Caches are critical components of the memory hierarchy, and their performance has a direct impact on the overall system performance. Therefore, it is essential to design caches that can efficiently store and retrieve data and instructions.

Cache design is a complex process that involves numerous parameters such as size, associativity, and replacement policy. These parameters interact with each other and have a significant impact on the cache's performance. In addition, the performance of a cache depends on the workload being executed. Therefore, evaluating different cache configurations and selecting the optimal one for a given workload is a challenging task.

The use of simulation tools such as gem5 has become popular in recent years for evaluating different cache configurations. These tools provide researchers with the ability to model various cache configurations and evaluate their performance under different workloads. Using simulation tools allows researchers to explore a wider range of cache configurations than would be possible through physical experimentation. In addition, simulation tools provide detailed insights into cache behavior that are not easily obtained through physical experimentation.

The use of gem5 to model various single-core CPU cache configurations is particularly important because single-core CPUs are still widely used in many systems, including mobile devices, embedded systems, and low-power servers. Therefore, optimizing the cache design for single-core CPUs is essential to improve the performance of these systems.

In addition, the use of gem5 to model cache configurations can help identify new trends and directions for cache design research. For example, recent research has explored the use of machine learning techniques to optimize cache performance. Using gem5 to evaluate these techniques can help identify their strengths and weaknesses and guide future research in this area.

In summary, the motivation for using gem5 to model various single-core CPU cache configurations is clear. Cache design is a critical area of research, and the use of simulation tools such as gem5 provides researchers with a powerful tool for evaluating different cache configurations. Furthermore, the use of gem5 to evaluate cache design can help identify new trends and directions for future research in this area.

III. Related Work

IV. Research methodology

In this research, we use the gem5 simulator to model various single-core CPU cache configurations and evaluate their performance under different workloads. We describe the experimental methodology below.

Hardware and Software Configuration: We use a Linux-based system for our experiments, with an Intel Core i7 processor running at 2.2 GHz and 16GB of RAM. We use gem5 version 20.0.0 for our simulations.

Workloads: We use a variety of benchmark workloads to evaluate the performance of different cache configurations. We use the SPEC CPU 2017 benchmark suite, which includes several real-world workloads such as GCC, GROMACS, and XZ. We also use synthetic workloads such as STREAM and PARSEC, which are designed to stress different aspects of the memory hierarchy.

Cache Configurations: We evaluate several cache configurations with different sizes, associativity, and replacement policies. We use a direct-mapped cache as a baseline and compare it with other configurations, including fully associative and set-associative caches. We use different cache sizes, ranging from 1KB to 32MB, and different replacement policies, including least-recently-used (LRU) and random.

Simulation Setup: We use gem5's full-system mode to simulate our experiments. In this mode, the entire system, including the CPU, cache, and memory hierarchy, is simulated. We run each workload multiple times to ensure the stability of the results and use a warm-up period to ensure that the cache is populated before we start collecting performance data.

Performance Metrics: We collect several performance metrics, including execution time, instruction throughput, and cache hit rate. We also collect cache miss rate, cache eviction rate, and average memory access latency to provide a more detailed analysis of cache behavior.

Statistical Analysis: We use statistical analysis to compare the performance of different cache configurations. We use the t-test to determine if there are statistically significant differences between the performance of different configurations. We also use analysis of variance (ANOVA) to determine the impact of different cache parameters on performance.

In summary, our experimental methodology involves using the gem5 simulator to evaluate various single-core CPU cache configurations under different workloads. We use a range of performance metrics to compare the performance of different configurations and use statistical analysis to determine the significance of our results.

V. Results

VI. Discussion

VII. Future Work

The use of gem5 to model various single-core CPU cache configurations provides a powerful tool for evaluating cache performance. Our research has shown that different cache configurations can have a significant impact on performance, and that using simulation tools such as gem5 can provide detailed insights into cache behavior.

There are several directions for future research in this area:

Multi-core CPU Cache Configurations: In this study, we focus on single-core CPU cache configurations. However, multi-core CPUs are becoming increasingly common, and cache design for multi-core CPUs is a complex and challenging area of research. Future studies could use gem5 to model various multi-core CPU cache configurations and evaluate their performance under different workloads.

Machine Learning Techniques: Recent research has explored the use of machine learning techniques to optimize cache performance. These techniques involve using machine learning algorithms to predict cache behavior and optimize cache configurations. Using gem5 to evaluate these techniques can help identify their strengths and weaknesses and guide future research in this area.

Emerging Memory Technologies: Emerging memory technologies, such as phase-change memory (PCM) and resistive random-access memory (RRAM), have the potential to revolutionize cache design. These technologies have different performance characteristics than traditional memory technologies, and designing caches that can take advantage of their unique properties is an area of active research. Future studies could use gem5 to evaluate cache designs that incorporate emerging memory technologies.

Energy-Efficient Caches: Energy efficiency is becoming an increasingly important consideration in cache design. Caches are power-hungry components of the memory hierarchy, and designing energy-efficient caches is essential to reduce power consumption and improve battery life. Future studies could use gem5 to evaluate energy-efficient cache designs, such as cache compression and cache bypassing techniques.

In conclusion, the use of gem5 to model various single-core CPU cache configurations provides a powerful tool for evaluating cache performance. Future research in this area could explore multi-core CPU cache configurations, machine learning techniques, emerging memory technologies, and energy-efficient caches. These studies could provide valuable insights into cache design and help improve the performance and energy efficiency of modern computer systems.

VIII. Conclusion

In conclusion, our research using the gem5 simulator to model various single-core CPU cache configurations has provided valuable insights into cache design and performance. Our experiments have shown that cache configuration has a significant impact on performance, and that different cache sizes, associativity, and replacement policies can have complex interactions that affect performance.

Our experiments have also highlighted the importance of workload characteristics in cache performance. Different workloads stress different aspects of the memory hierarchy, and understanding these characteristics is essential in designing effective caches.

Overall, our research demonstrates the utility of simulation tools such as gem5 in evaluating cache performance. By using simulation, we can explore a wide range of cache configurations and workloads in a controlled and repeatable environment. This allows us to gain insights into cache behavior that would be difficult or impossible to obtain through experimental methods alone.

Our research also points to several areas for future research, including multi-core CPU cache configurations, machine learning techniques, emerging memory technologies, and energy-efficient caches. We believe that continued research in these areas can help improve the performance and energy efficiency of modern computer systems.

In summary, our findings highlight the importance of cache design and performance in modern computer systems, and the potential of simulation tools such as gem5 to aid in this research. We hope that our work will contribute to a better understanding of cache behavior and guide future research in this important area.