Two-Level Cache Performance Analysis Report

Project 2: Two-Level Performance Cache Simulator

Course:	Computer Organization and assembly
Project:	Two-Level Performance Cache Simulator
Date:	July 2024
Team Members:	[Salma Elmarakby, Marina Nazeh, Mahinour]

Department of Computer Science
[University Name]

1 Introduction and Methodology

This report presents a comprehensive analysis of cache performance using a two-level cache hierarchy simulator. The objective was to evaluate the impact of L1 cache line size on overall system performance across different memory access patterns.

1.1 Simulation Configuration

The simulator implements a realistic two-level cache hierarchy with the following specifications:

- **L1 Cache:** 16KB, 4-way set associative, variable line size (16B, 32B, 64B, 128B), 1-cycle hit time
- **L2 Cache:** 128KB, 8-way set associative, 64B line size, 10-cycle hit time
- **Main Memory:** 64GB DRAM, 50-cycle access penalty
- **Write Policy:** Write-back with random replacement
- **Workload:** 35% memory instructions (50% reads, 50% writes), 1M instruction simulation

1.2 Memory Access Patterns

Five distinct memory generators were implemented to represent different application behaviors:

1. **memGen1:** Sequential access pattern (addr++)
2. **memGen2:** Random access within 24KB locality
3. **memGen3:** Random access across entire 64MB space
4. **memGen4:** Sequential access within 4KB working set
5. **memGen5:** Strided access pattern (stride = 32B) within 1MB

2 Experimental Results

2.1 CPI Performance Analysis

Table 1 presents the Cycles Per Instruction (CPI) results across all memory generators and L1 line sizes:

Table 1: CPI Results

Generator	16B Line	32B Line	64B Line	128B Line
memGen1	1.925	1.686	1.554	1.277
memGen2	3.065	3.064	3.067	3.052
memGen3	32.451	32.445	32.479	32.494
memGen4	1.006	1.004	1.004	1.002
memGen5	21.702	21.702	19.241	11.113

2.2 Cache Hit Rate Analysis

Table 2: L1 Cache Hit Rates (%)

Generator	16B Line	32B Line	64B Line	128B Line
memGen1	93.7	96.9	98.4	99.2
memGen2	66.4	66.5	66.5	66.8
memGen3	0.0	0.0	0.0	0.0
memGen4	99.9	100.0	100.0	100.0
memGen5	0.0	0.0	50.0	75.0

Table 3: L2 Cache Hit Rates (%)

Generator	16B Line	32B Line	64B Line	128B Line
memGen1	84.7	71.4	46.1	46.2
memGen2	99.8	99.8	99.8	99.8
memGen3	21.5	26.9	30.5	15.8
memGen4	75.0	50.0	0.0	0.0
memGen5	64.5	64.5	40.3	45.5

3 Data Analysis and Discussion

3.1 Impact of Spatial Locality

Sequential Access Patterns (memGen1, memGen4): These generators demonstrate excellent spatial locality benefits. MemGen1 shows a 33% CPI improvement (1.925 $\rightarrow$ 1.277) when increasing line size from 16B to 128B, with L1 hit rates improving from 93.7% to 99.2%. MemGen4, with its small 4KB working set, achieves near-optimal performance ($\sim$1.00 CPI) across all line sizes due to perfect cache containment.

Strided Access Pattern (memGen5): This generator provides the most dramatic demonstration of spatial locality effects. When the stride (32B) exceeds the cache line size (16B, 32B), spatial locality cannot be exploited, resulting in poor performance (CPI > 21). However, when line sizes (64B, 128B) exceed the stride, significant performance improvements occur, with CPI dropping to 11.113 for 128B lines.

3.2 Cache Size vs. Working Set Relationships

MemGen2 accesses a 24KB working set, which exceeds the 16KB L1 cache capacity. This results in consistent ∼66% L1 hit rates regardless of line size, as the fundamental issue is capacity rather than spatial locality. However, the excellent L2 hit rates (99.8%) demonstrate effective cache hierarchy operation.

MemGen3 represents the worst-case scenario with random access across 64MB, far exceeding both cache capacities. The near-zero L1 hit rates and poor L2 hit rates (15-30%) result in CPI values exceeding 32, highlighting the critical importance of locality in memory access patterns.

3.3 Cache Hierarchy Effectiveness

The L2 cache effectively captures misses from L1, particularly evident in memGen2 where L2 hit rates remain consistently high (99.8%) despite moderate L1 performance. This demonstrates the importance of multi-level cache hierarchies in handling diverse access patterns.

4 Conclusions

1. **Spatial Locality is Critical:** Applications with sequential or predictable stride patterns benefit significantly from larger cache lines, with performance improvements of up to 95% observed.

2. **Working Set Size Dominates:** When the application working set exceeds cache capacity, line size optimizations provide minimal benefit. Capacity-based solutions become more important.

3. **Access Pattern Characterization:** Random access patterns with large working sets severely degrade cache performance, emphasizing the need for algorithm design that considers memory hierarchy characteristics.

4. **Optimal Line Size Selection:** The optimal L1 line size depends heavily on application access patterns. Sequential applications favor larger lines (128B), while random access patterns show minimal sensitivity to line size.

5. **Cache Hierarchy Value:** The two-level hierarchy effectively captures different types of locality, with L2 providing excellent coverage for medium-sized working sets that exceed L1 capacity.

4.1 Design Recommendations

- **For sequential workloads:** Implement larger cache lines (128B) to maximize spatial locality benefits
- **For random workloads:** Focus on cache capacity rather than line size optimization
- **For mixed workloads:** Consider adaptive cache line sizing or application-specific cache configurations

These results provide valuable insights for both cache design optimization and application development strategies in modern computing systems.