Indian Institute of Technology Delhi

Department of Computer Science

COL - 216

Cache Simulator

Nikhil Zawar - 2022CS11106

1 Introduction

v In this section of the assignment, we evaluate the performance of our cache simulator across various configurations to gauge its effectiveness. We explore three key metrics: hit rate, average cycle count, and average cache size. Our tests involve diverse inputs, including variations in the number of sets, blocks per set, and bytes per set, as well as different write schemes, allocation strategies, and eviction methods. We conduct these evaluations using real-life trace files, namely swim.trace(303194 commands) and twolf.trace(482285 commads). Initially, we provide overarching conclusions, followed by trend graphs and detailed data logs.

2 Hit-Rate

The hit-rate trends for both swim.trace and twolf.trace indicate an increase as the cache parameters vary. For swim.trace, the hit-rate generally rises as the number of sets, blocks, and bytes increase. Similarly, for twolf.trace, higher hit-rates are observed with increasing cache parameters. These trends suggest the effectiveness of larger cache configurations in improving cache performance.

3 Average Total Cycles

The average total cycles vary across different configurations of sets, blocks, and bytes. For both number of sets and number of blocks, a decrease in the parameter leads to a reduction in total cycles, indicating a more efficient cache utilization. Similarly, as the number of bytes per block increases, the total cycles tend to decrease, suggesting improved cache efficiency. Among the configurations, smaller numbers of sets and blocks, along with larger bytes per block, result in fewer total cycles, indicating a more favorable cache configuration for minimizing cycle counts. This trend tells the importance of optimizing cache parameters to enhance performance and reduce computational overhead.

4 Average Cache Size

The average cache size varies across different configurations of sets, blocks, and bytes. Smaller values for sets and blocks result in lower average cache sizes, while larger values lead to increased cache sizes. Similarly, a higher number of bytes per block corresponds to a larger average cache size. Among the configurations, a setup with fewer sets and blocks but a higher number of bytes per block tends to yield a smaller average cache size. For instance, a configuration with 1 set, 1 block, and 64 bytes per block could potentially offer the smallest average cache size. However, the optimal configuration depends on specific performance requirements and trade-offs.

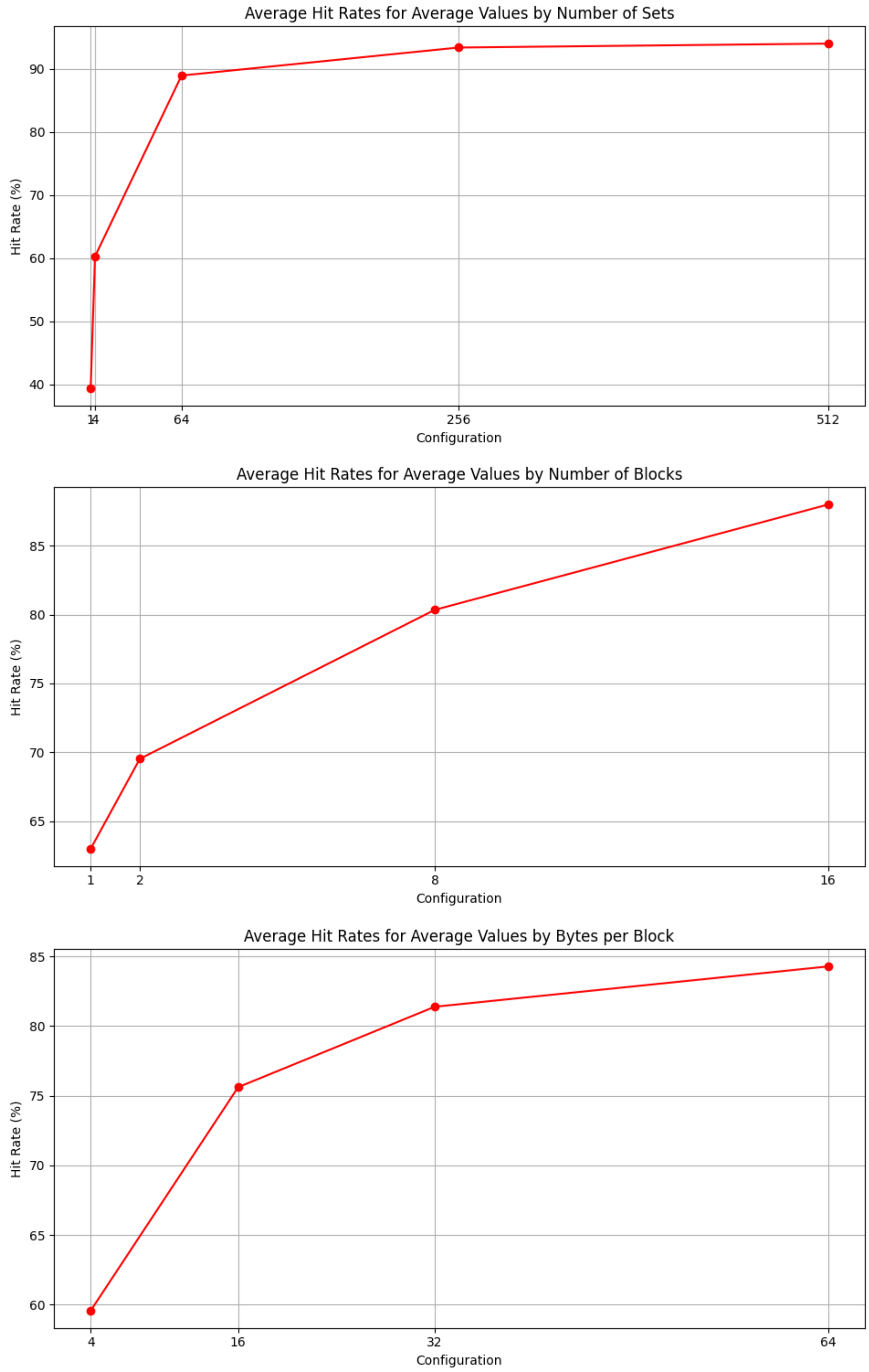

Figure 1: Hit-rate trends for swim.trace

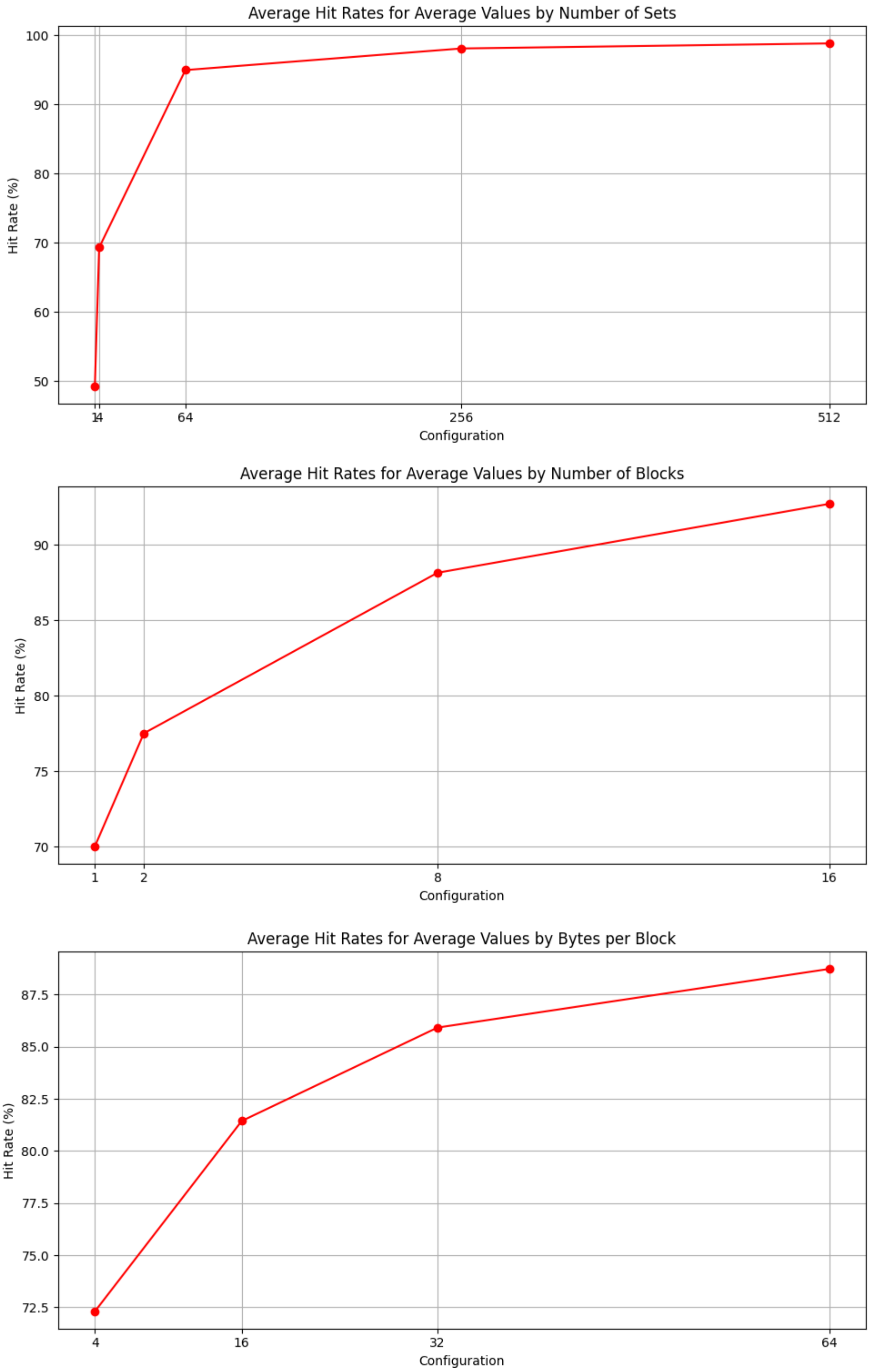

Figure 2: Hit-rate trends for twolf.trace

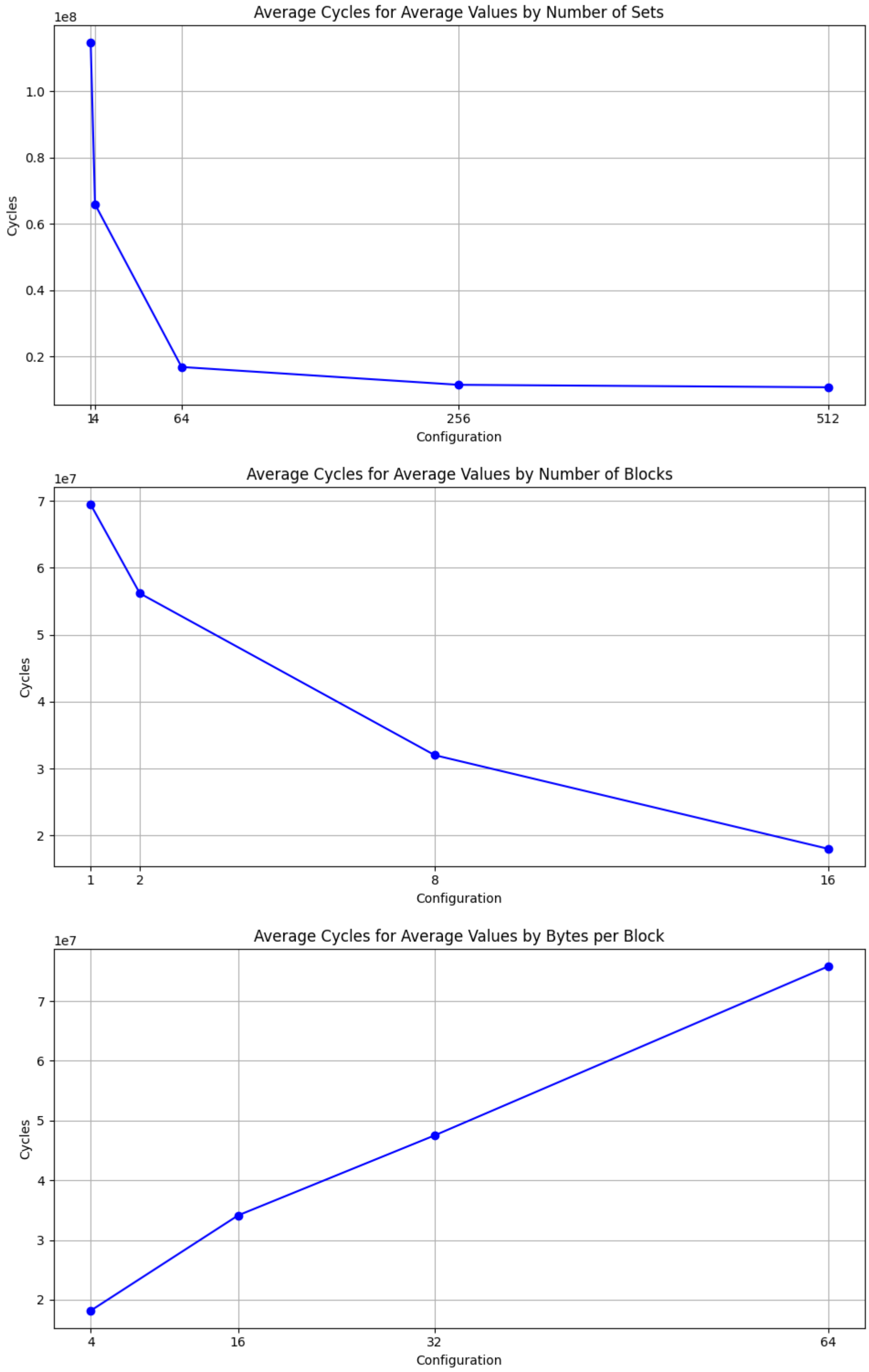

Figure 3: Average Cycles trends for swim.trace

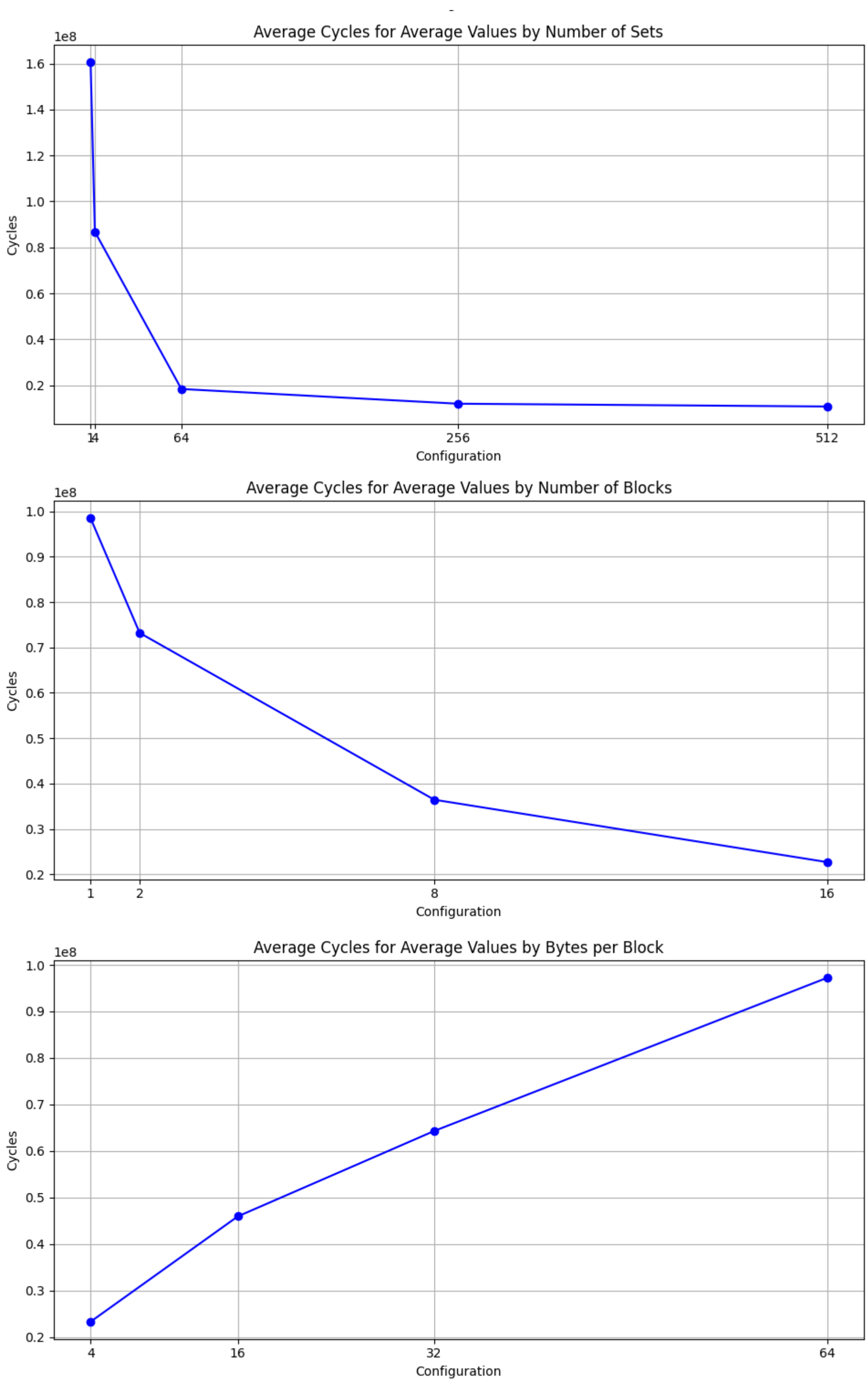

Figure 4: Average Cycles trends for twolf.trace

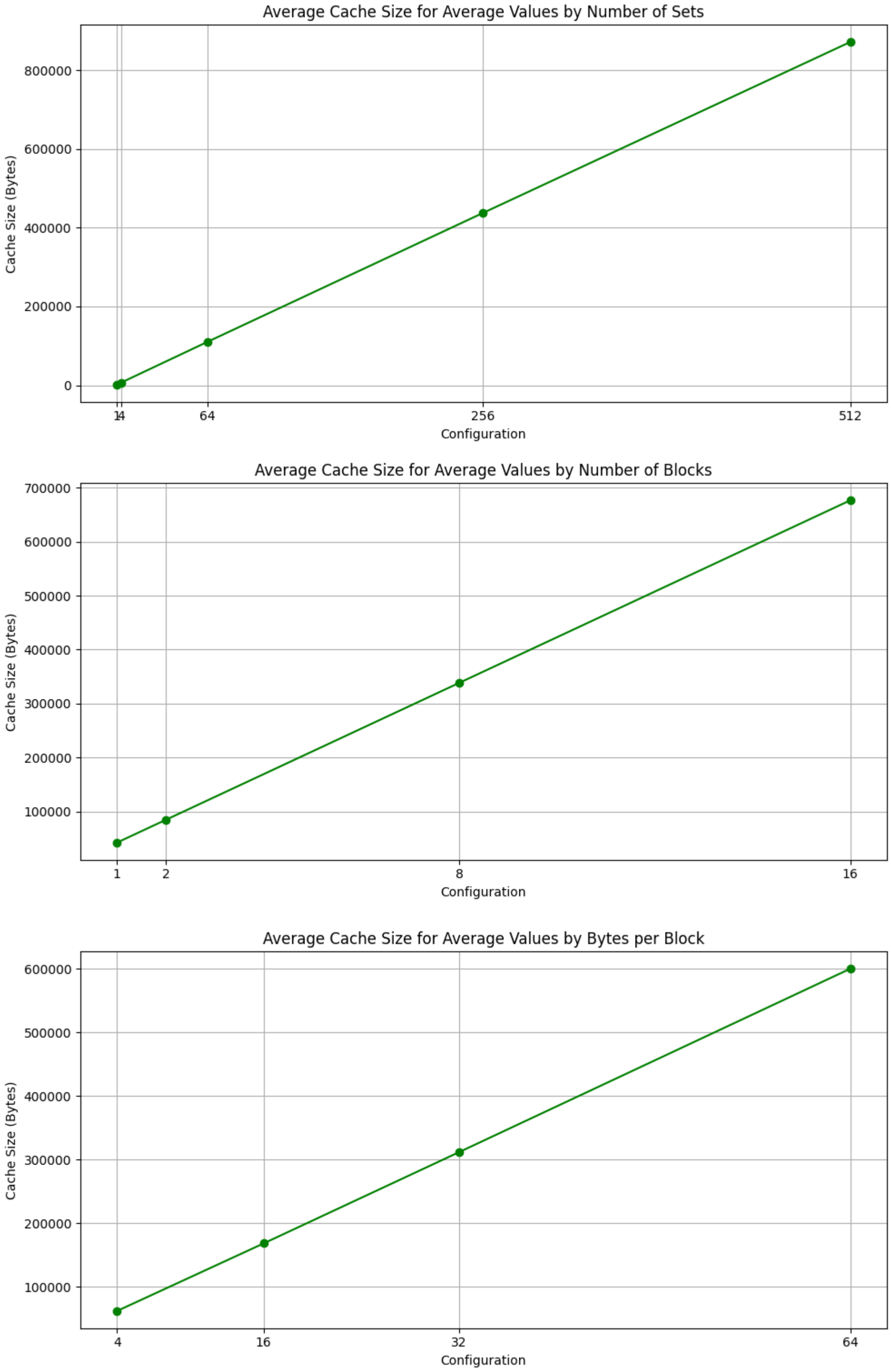

Figure 5: Average Cache Size trends for swim.trace

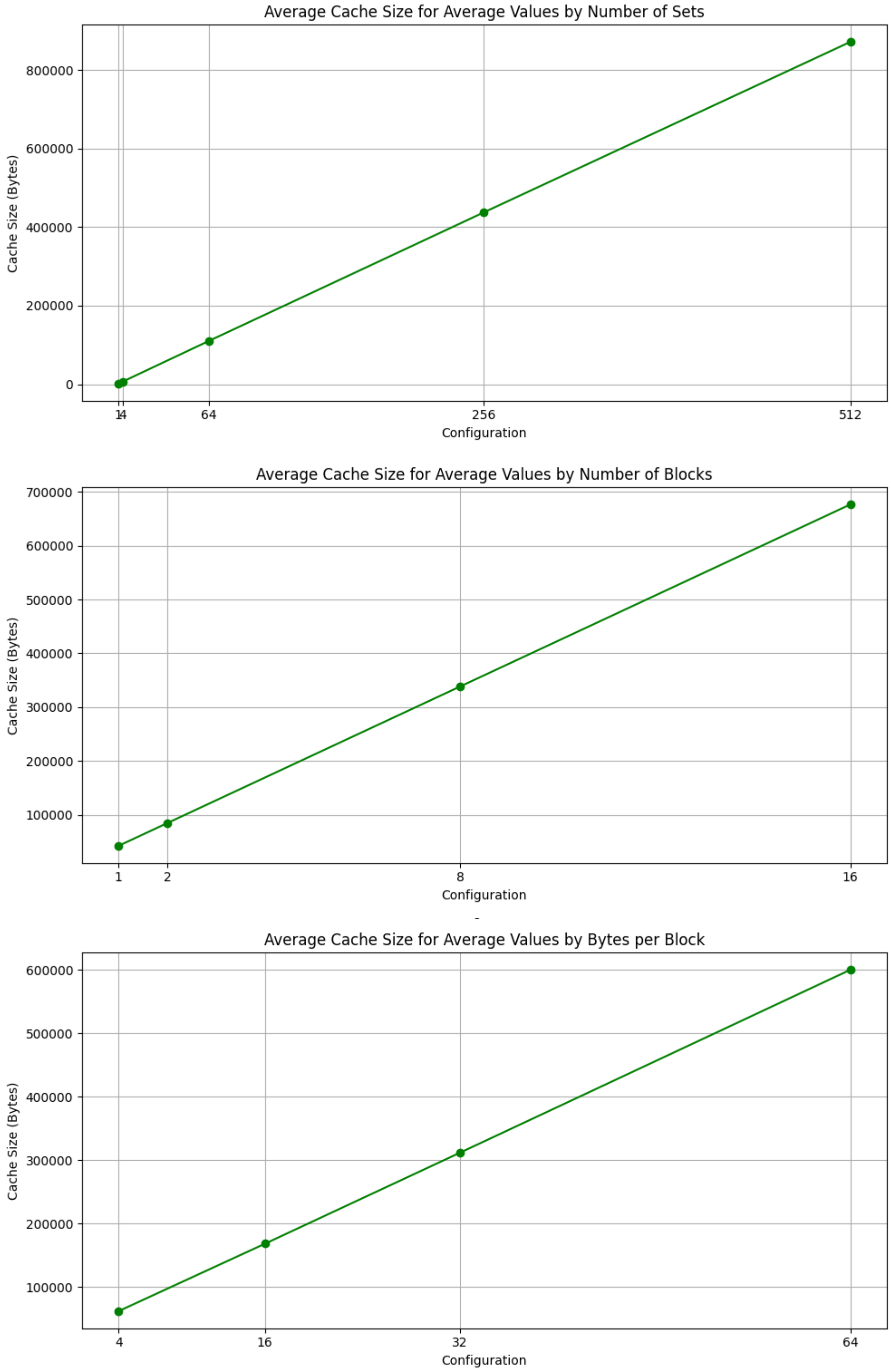

Figure 6: Average Cache Size trends for twolf.trace

Table 1: Top 10 Hit Rates for twolf.trace

Hit Rate	Cache Size	Sets	Blocks	Bytes	Allocate	Scheme
99.80%	2179072B	256	16	64	write-allocate	write-back
99.80%	2179072B	256	16	64	write-allocate	write-back
99.80%	2174976B	256	16	64	write-allocate	write-through
99.80%	2174976B	256	16	64	write-allocate	write-through
99.80%	2174976B	512	8	64	write-allocate	write-back
99.80%	2174976B	512	8	64	write-allocate	write-back
99.80%	2170880B	512	8	64	write-allocate	write-through
99.80%	2170880B	512	8	64	write-allocate	write-through
99.80%	4349952B	512	16	64	write-allocate	write-back
99.80%	4349952B	512	16	64	write-allocate	write-back

Table 2: Top 10 Minimum Cycles for twolf.trace

Total Cycles	Cache Size	Sets	Blocks	Bytes	Allocate	Scheme
1195524	450560B	512	16	4	write-allocate	write-back
1206224	450560B	512	16	4	write-allocate	write-back
1419724	229376B	256	16	4	write-allocate	write-back
1420324	225280B	512	8	4	write-allocate	write-back
1423624	229376B	256	16	4	write-allocate	write-back
1428124	225280B	512	8	4	write-allocate	write-back
1535524	114688B	256	8	4	write-allocate	write-back
1572524	114688B	256	8	4	write-allocate	write-back
1654224	59392B	64	16	4	write-allocate	write-back
1664824	1220608B	512	16	16	write-allocate	write-back

Table 3: Top 10 Hit Rates for swim.trace

Hit Rate	Cache Size	Sets	Blocks	Bytes	Allocate	Scheme
98.86%	2179072B	256	16	64	write-allocate	write-back
98.86%	2174976B	256	16	64	write-allocate	write-through
98.86%	2174976B	512	8	64	write-allocate	write-back
98.86%	2170880B	512	8	64	write-allocate	write-through
98.86%	4349952B	512	16	64	write-allocate	write-back
98.86%	4349952B	512	16	64	write-allocate	write-back
98.86%	4341760B	512	16	64	write-allocate	write-through
98.86%	4341760B	512	16	64	write-allocate	write-through
98.86%	2179072B	256	16	64	write-allocate	write-back
98.86%	2174976B	256	16	64	write-allocate	write-through

Table 4: Top 10 Minimum Cycles for swim.trace

Total Cycles	Cache Size	Sets	Blocks	Bytes	Allocate	Scheme
4207793	450560B	512	16	4	write-allocate	write-back
4295793	450560B	512	16	4	write-allocate	write-back
4477993	229376B	256	16	4	write-allocate	write-back
4480493	225280B	512	8	4	write-allocate	write-back
4689393	114688B	256	8	4	write-allocate	write-back
4693293	229376B	256	16	4	write-allocate	write-back
4695593	225280B	512	8	4	write-allocate	write-back
5168493	114688B	256	8	4	write-allocate	write-back
5214693	59392B	64	16	4	write-allocate	write-back
5375793	56320B	512	2	4	write-allocate	write-back

5 Conclusion

Configurations with a cache size of 256 sets, 16 blocks per set, and 64 bytes per block consistently achieve a high hit rate of 99.80%. Moreover, setups with a cache size of 512 sets, 16 blocks per set, and 4 bytes per block demonstrate the lowest total cycles, indicating efficient performance. The combination of write-allocate, write-back, and LRU eviction policy tends to yield optimal results across both hit rate and cycle count metrics.