CS810: Advanced Computer Architecture

Assignment 5

15 November 2023

Shashank P (200010048), Tabish Khalid Halim (200020049)

The objective of this assignment is to implement the Bingo prefetcher in Tejas and evaluate its efficacy in terms of both instruction throughput (IPC) and level cache hit rate and compare it with a system that uses the Power4 prefetcher in the last level and with a system that does not use any prefetcher in the last level.

1 Introduction

We know that prefetchers predict and preload data from memory to reduce memory latency, hide access delays, improve cache performance, and enhance overall system throughput and efficiency.

No Prefetcher Metric to show the system's performance without a prefetcher.

Power4 Prefetcher Default prefetcher in system architecture, created by IBM.

Simple Prefetcher Simple prefetcher a prefetching algorithm that fetches the next block instruction to achieve a higher cache hit rate and IPC.

Bingo Prefetcher Bingo Prefetcher, a prefetching algorithm developed by IPM and the Sharif University of Technology, achieves higher IPC and cache hit rate.

1.1 Bingo Prefetcher Algorithm

Bingo is a spatial data prefetcher utilizing a single history table to capture spatial patterns as the processor accesses spatial regions. It associates footprint metadata with multiple events, extracting precise spatially correlated data access patterns and reducing cache misses. The algorithm uses 'PC+Address' and 'PC+Offset' events, each looking up the history table with different events. Information is stored in the history table indexed with a hash of the shortest event but tagged with the longest event. In our case, 'PC' was unavailable to tag with the addresses; hence, only 'address' and 'offset' were used.

2 Experimental Setup

We used 4 new and 5 old benchmarks in our experiment. The benchmarks are shown below. We o**nly updated L3 cache prefetchers** while keeping other caches to have **Power4** prefetcher.

- **Number of Instructions:** 5 million

- **Auxillary Space Size**: 100 entries
- **Eviction Policy**: LRU
- **Benchmarks Used:** gcc, lbm, mcf, namd, xalancbmk, numentanab, gimp, blender, openssl

3 Plots

Let us look at the performance metrics for different benchmarks, such as IPC and L3 hit rate.

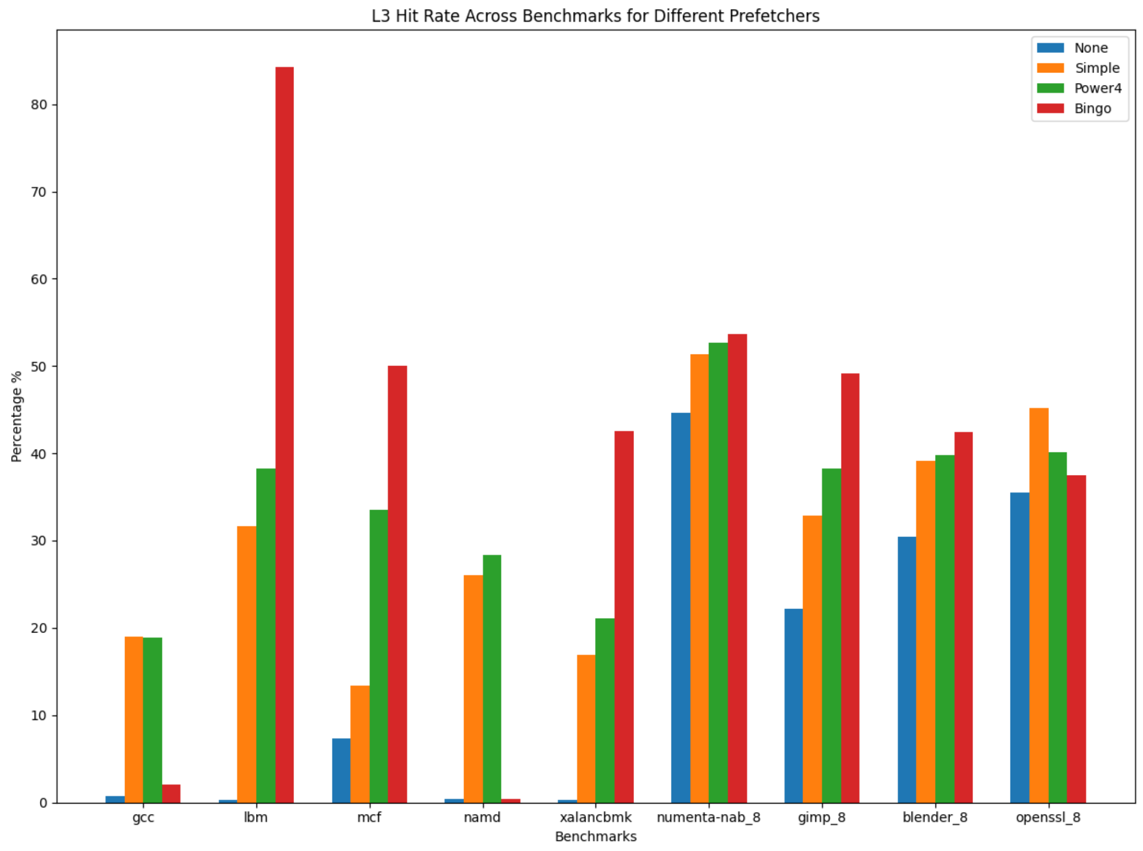

Figure 1: Histogram plot of L3 hit-rate for various Benchmarks and Prefetchers

4 Observations

- The variation in the average performance across the different benchmarks shows that overall, bingo prefetcher has performed significantly better, if not at least comparable to other prefetching algorithms.

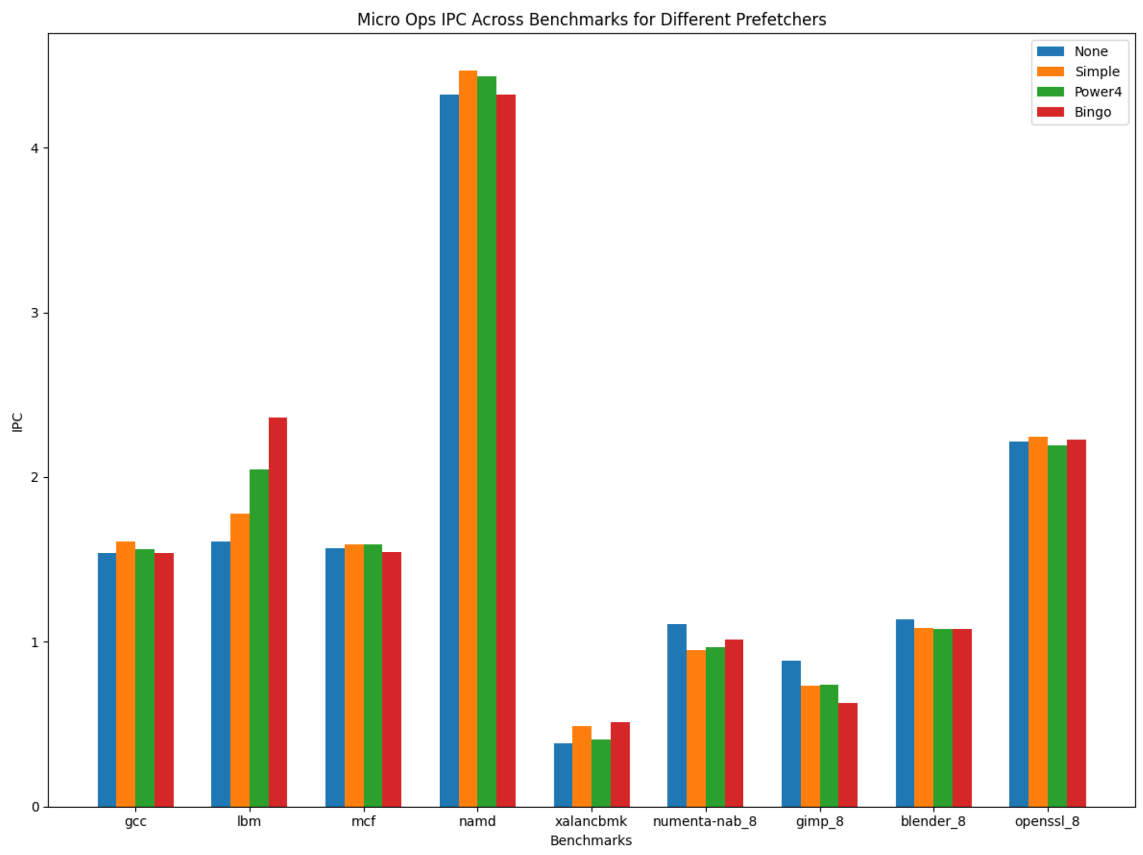

Figure 2: Histogram plot of Micro-op IPC for various Benchmarks and Prefetchers

- The average Micro-ops IPC metric, as well as the L3 hit rate for the Bingo Prefetcher, is found to be the highest among the prefetchers, standing out at IPC = 1.691 & L3 hit-rate = 0.402
- For spatially located memory access-based benchmarks (like **lbm** and **mcf**) the number of prefetches for **Bingo** are **10k** more than **Power4** (per $5mil$ instructions). If there is no spatial locality, then there is very little perfecting.
- From the graph, we can see **lbm** benefited the most from using Bingo. This could be because of extremely high spatial locality in memory access. It also had a boost in IPC.
- The accuracy of the prediction can be increased using the 'PC' parameter which was not available directly in **Tejas** events

Prefetchers	**None**	**Simple**	**Power**	**Bingo**
Micro-ops IPC	1.6405	1.667	1.669	1.691
L3 hit-rate	0.157	0.306	0.345	0.402

Table 1: Average IPC and Average L3 hit rate for various prefetches in the L3 cache.

5 Conclusion

Our observations show that the bingo prefetcher achieves higher IPC and L3 cache hit rates, especially in the case of benchmarks with spatially located accesses. It performs worse if the accesses do not follow a fixed long-term pattern like **gcc** and **namd** where the pattern is short-term next-in-line prefetcher works well. This might be because the data in the cache displays spacial locality, exploiting which the bingo prefetcher performs better for **lbm** benchmark.