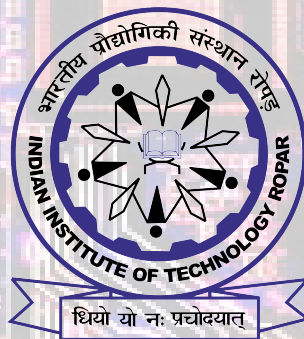


ChampSim

CS-204 Project [Computer Architecture]

Group 18



Prashant Kumar - 2022CSB1202

Kartikey Sahu - 2022CSB1087

Pranav Dipesh Bhole - 2022CSB1103

Department of Computer Science and Engineering
Indian Institute of Technology Ropar, India

Contents

1	Problem Statement	3
2	Solution	3
2.1	Remapping Method	3
2.2	Using Buffer with Remapping	3
3	Implementation	4
3.1	Important Functions	4
3.2	Remapping Method (Version 1)	6
3.3	Remapping Method with Buffer (Version 2)	6
4	Explanation Of Code and Logic	7
4.1	Remapping Method	7
4.1.1	Mapping Logic	7
4.1.2	Updation Logic	7
4.2	Remapping With Buffer	8
5	Result of Remapping With Buffer	8

1 Problem Statement

Given a fixed last-level cache configuration (2MB, 16 ways, 64-byte block size), the goal is to optimize cache performance by reducing conflict misses in response to input traces. By identifying a suitable remapping of addresses to cache lines, the aim is to minimize conflict misses, consequently improving the hit rate, average memory access time (AMAT), and overall performance.

2 Solution

2.1 Remapping Method

An issue with the fixed address-to-line mapping approach, which uses the modulo operator, is that it can lead to inefficient cache utilization. Some cache sets may be underutilized while others are heavily used. One potential solution to address this problem involves creating an intermediate data structure that facilitates the remapping of addresses to new indices, as necessary. This approach targets the reduction of conflict misses associated with frequently accessed cache lines.

2.2 Using Buffer with Remapping

Another strategy to mitigate conflict misses involves employing a small buffer. This buffer is designed to specifically address cold conflict misses, thereby further optimizing cache performance.

3 Implementation

3.1 Important Functions

Algorithm 1 void update_hot_cold()

```
function UPDATE_HOT_COLD
  if num_instr % 30000 = 0 then
    Initialize redirect array with zeros
    Empty Hot and Cold priority queues
    Initialize Hot_copy and Cold_copy priority queues
    for each element in freq array do
      Push pair of frequency and index to Hot_copy
      Push pair of frequency and index to Cold_copy
    end for
    while Hot_copy not empty and size of Hot is not 1024 do
      Get top element from Hot_copy and push it to Hot
      Pop top element from Hot_copy
    end while
    while Cold_copy not empty and size of Cold is not 1024 do
      Get top element from Cold_copy and push it to Cold
      Pop top element from Cold_copy
    end while
  end if
end function
```

Algorithm 2 bool check_hit()

```
function CHECK_HIT
  for way from 0 to NUM_WAY-1 do
    if block[final_set[set]][way] is valid and tag matches address then
      Increment num_tag_comp, set flag to true, and break
    end if
  end for
end function
```

Algorithm 3 void redirect_sets()

```
function REDIRECT_SETS
    Initialize Cold_copy priority queue with Cold
    while Cold_copy not empty do
        Get from Cold_copy
        if redirect[get]  $\leq$  256 then
            Increment redirect[get], set ind to true, increment ct
            Insert get into potential_check_sets[set], set final_set[set] to get,
and break
        end if
    end while
end function
```

Algorithm 4 void check_in_buffer()

```
function CHECK_IN_BUFFER
    if cache_type equals IS_LLC then
        if size of buffer equals MAX_SIZE_BUFFER then
            Clear buffer
        end if
    end if
    if cache_type equals IS_LLC and address is found in buffer then
        Set set_Buffer to top element of Cold
        Set way_Buffer to random number modulo NUM_WAY
        Set final_set[set] to set_Buffer
        Get previous_tag from block[set_Buffer][way_Buffer]
        Set tag of block[set_Buffer][way_Buffer] to address
        Insert set_Buffer and final_set[set] into potential_check_sets[set]
        Insert address into buffer, remove previous_tag from buffer
        return final_set[set]
    end if
end function
```

3.2 Remapping Method (Version 1)

Algorithm 5 CACHE.get_set(address)

```
function CACHE.GET_SET(address)
    set  $\leftarrow$  address bitwise AND ((1 left shift lg2(NUM_SET)) minus 1)
    update_hot_cold()
    Flag1  $\leftarrow$  check_hit()
    if cache_type equals IS_LLC and Flag1 is false and isfind_hot(set) is
true then
        redirect_sets()
    end if
    insert set into potential_check_sets[set]
    return final_set[set]
end function
```

3.3 Remapping Method with Buffer (Version 2)

Algorithm 6 CACHE.get_set(address)

```
function CACHE.GET_SET(address)
    set  $\leftarrow$  address bitwise AND ((1 left shift lg2(NUM_SET)) minus 1)
    update_hot_cold()
    Flag2  $\leftarrow$  check_in_buffer()
    if cache_type equals IS_LLC and Flag2 is true then
        return final_set[set]
    end if
    Flag1  $\leftarrow$  check_hit()
    if cache_type equals IS_LLC and Flag1 is false and isfind_hot(set) is
true then
        redirect_sets()
    end if
    insert set into potential_check_sets[set]
    return final_set[set]
end function
```

4 Explanation Of Code and Logic

4.1 Remapping Method

4.1.1 Mapping Logic

Within the cache architecture, memory blocks are organized into sets, each set comprising multiple ways. Focusing on the last level cache in our simulation, we encounter a configuration boasting 2048 sets, with each set accommodating 16 ways.

Observing the cache's behavior, we discerned an intriguing pattern: certain sets experience saturation, while others remain underutilized, with ample vacant ways awaiting occupancy. To optimize cache efficiency, we devised a strategy of redirecting entries from heavily utilized sets to their less burdened counterparts.

We embarked on this remapping endeavor by meticulously scrutinizing the access frequency of each set during the initial 1024 calls. Subsequently, we classified the sets into distinct categories:

- The first set, enjoying the highest access frequency, earned the prestigious title of "Very Hot."
- Sets 2 through 1024 were labeled as "Hot," indicating their considerable yet not maximal utilization.
- The subsequent sets, ranging from 1025 to 2047, were dubbed "Cold," signifying their underutilized state.
- Finally, the last set, lagging behind in access frequency, was crowned "Very Cold."

Now, when we encounter an address, our first task is to determine its designated set. If the set falls within the "Hot" category, we adopt an innovative approach: amalgamating it with each of the "Cold" sets in a collaborative sharing endeavor. Essentially, each "Hot" set collaborates with all 1024 "Cold" sets, leveraging shared memory resources to enhance efficiency.

4.1.2 Updation Logic

In our pursuit of continuous optimization, we implement an update mechanism triggered after every 30000 cycles, though this frequency is adaptable.

This update involves the eviction of both the minimum and maximum heaps, recalibrating the hierarchy based on the evolving access frequencies of each set. This dynamic adjustment ensures that newcomers are seamlessly integrated into the cache hierarchy, aligning with their respective access frequencies within the refreshed heap structures.

4.2 Remapping With Buffer

In addition to the innovative remapping techniques discussed previously, we introduce an auxiliary feature: the integration of a buffer within the last level cache. This buffer serves as a transient repository for evicted memory blocks, offering a strategic alternative to direct retrieval from main memory.

Every memory block that faces eviction from the last level cache finds sanctuary within this buffer, poised for potential future access. By leveraging the buffer, we circumvent the latency associated with accessing main memory, thereby enhancing overall system responsiveness.

The buffer boasts a capacity to house 512 blocks, constituting approximately 1/32nd of the last level cache’s total capacity. Once this threshold is reached, signaling that the buffer has reached maximum occupancy, a proactive measure is initiated: the buffer undergoes a swift purging process. This ensures that the buffer remains agile and responsive, ready to accommodate forthcoming evictions and uphold seamless cache operations.

5 Result of Remapping With Buffer

- **444.namd-120B.champsimtrace.xz** [\[1\]](#)
 - IPC Value without remapping = 1.855
 - IPC Value after remapping = 1.85291
- **445.gobmk-17B.champsimtrace.xz** [\[2\]](#)
 - IPC Value without remapping = 0.786951
 - IPC Value after remapping = 0.786961
- **473.astar-153B.champsimtrace.xz** [\[3\]](#)
 - IPC Value without remapping = 0.660133

– IPC Value after remapping = 0.67963

• **605.mcf_s-1536B.champsimtrace.xz** [\[4\]](#)

– IPC Value without remapping = 0.150799

– IPC Value after remapping = 0.151351

Average IPC overall without remapping = 0.86322075

Average IPC overall after remapping = 0.867713

Appendices

```

CPU 0 cumulative IPC: 1.85291 instructions: 100000002 cycles: 53969091
L1D TOTAL      ACCESS:   19539449 HIT:   19327008 MISS:    212441
L1D LOAD       ACCESS:   16044222 HIT:   15842148 MISS:    202074
L1D RFO        ACCESS:    3495227 HIT:    3484860 MISS:     10367
L1D PREFETCH   ACCESS:         0 HIT:         0 MISS:         0
L1D WRITEBACK  ACCESS:         0 HIT:         0 MISS:         0
L1D PREFETCH REQUESTED:         0 ISSUED:         0 USEFUL:         0 USELESS:         0
L1D AVERAGE MISS LATENCY: 20.0381 cycles
L1I TOTAL      ACCESS:   17066485 HIT:   17066465 MISS:         20
L1I LOAD       ACCESS:   17066485 HIT:   17066465 MISS:         20
L1I RFO        ACCESS:         0 HIT:         0 MISS:         0
L1I PREFETCH   ACCESS:         0 HIT:         0 MISS:         0
L1I WRITEBACK  ACCESS:         0 HIT:         0 MISS:         0
L1I PREFETCH REQUESTED:         0 ISSUED:         0 USEFUL:         0 USELESS:         0
L1I AVERAGE MISS LATENCY: 160.35 cycles
L2C TOTAL      ACCESS:    250192 HIT:    236880 MISS:    13312
L2C LOAD       ACCESS:    202076 HIT:    188794 MISS:    13282
L2C RFO        ACCESS:    10364 HIT:    10337 MISS:         27
L2C PREFETCH   ACCESS:         0 HIT:         0 MISS:         0
L2C WRITEBACK  ACCESS:    37752 HIT:    37749 MISS:         3
L2C PREFETCH REQUESTED:         0 ISSUED:         0 USEFUL:         0 USELESS:         0
L2C AVERAGE MISS LATENCY: 80.1629 cycles
LLC TOTAL      ACCESS:    15933 HIT:     9222 MISS:    6711
LLC LOAD       ACCESS:    13282 HIT:     6632 MISS:    6650
LLC RFO        ACCESS:         27 HIT:         21 MISS:         6
LLC PREFETCH   ACCESS:         0 HIT:         0 MISS:         0
LLC WRITEBACK  ACCESS:    2624 HIT:    2569 MISS:         55
LLC PREFETCH REQUESTED:         0 ISSUED:         0 USEFUL:         0 USELESS:         0
LLC AVERAGE MISS LATENCY: 99.3578 cycles

```

Figure 1: 444.namd-120B.champsimtrace.xz Result

```

CPU 0 cumulative IPC: 0.786961 instructions: 100000000 cycles: 127071030
L1D TOTAL ACCESS: 34319548 HIT: 34242055 MISS: 77493
L1D LOAD ACCESS: 17968896 HIT: 17939180 MISS: 29716
L1D RFO ACCESS: 16350652 HIT: 16302875 MISS: 47777
L1D PREFETCH ACCESS: 0 HIT: 0 MISS: 0
L1D WRITEBACK ACCESS: 0 HIT: 0 MISS: 0
L1D PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
L1D AVERAGE MISS LATENCY: 17.5901 cycles
L1I TOTAL ACCESS: 19409816 HIT: 19171731 MISS: 238085
L1I LOAD ACCESS: 19409816 HIT: 19171731 MISS: 238085
L1I RFO ACCESS: 0 HIT: 0 MISS: 0
L1I PREFETCH ACCESS: 0 HIT: 0 MISS: 0
L1I WRITEBACK ACCESS: 0 HIT: 0 MISS: 0
L1I PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
L1I AVERAGE MISS LATENCY: 14.06 cycles
L2C TOTAL ACCESS: 375236 HIT: 374168 MISS: 1068
L2C LOAD ACCESS: 267800 HIT: 266808 MISS: 992
L2C RFO ACCESS: 47777 HIT: 47701 MISS: 76
L2C PREFETCH ACCESS: 0 HIT: 0 MISS: 0
L2C WRITEBACK ACCESS: 59659 HIT: 59659 MISS: 0
L2C PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
L2C AVERAGE MISS LATENCY: 187.392 cycles
LLC TOTAL ACCESS: 1549 HIT: 485 MISS: 1064
LLC LOAD ACCESS: 992 HIT: 41 MISS: 951
LLC RFO ACCESS: 76 HIT: 0 MISS: 76
LLC PREFETCH ACCESS: 0 HIT: 0 MISS: 0
LLC WRITEBACK ACCESS: 481 HIT: 444 MISS: 37
LLC PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
LLC AVERAGE MISS LATENCY: 157.984 cycles

```

Figure 2: 445.gobmk-17B.champsimtrace.xz Result

```

CPU 0 cumulative IPC: 0.67963 instructions: 10000001 cycles: 147138877
L1D TOTAL ACCESS: 35776973 HIT: 35451898 MISS: 325075
L1D LOAD ACCESS: 22788171 HIT: 22525842 MISS: 262329
L1D RFO ACCESS: 12988802 HIT: 12926056 MISS: 62746
L1D PREFETCH ACCESS: 0 HIT: 0 MISS: 0
L1D WRITEBACK ACCESS: 0 HIT: 0 MISS: 0
L1D PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
L1D AVERAGE MISS LATENCY: 47.1643 cycles
L1I TOTAL ACCESS: 15556200 HIT: 15556200 MISS: 0
L1I LOAD ACCESS: 15556200 HIT: 15556200 MISS: 0
L1I RFO ACCESS: 0 HIT: 0 MISS: 0
L1I PREFETCH ACCESS: 0 HIT: 0 MISS: 0
L1I WRITEBACK ACCESS: 0 HIT: 0 MISS: 0
L1I PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
L1I AVERAGE MISS LATENCY: -nan cycles
L2C TOTAL ACCESS: 503435 HIT: 351447 MISS: 151988
L2C LOAD ACCESS: 262326 HIT: 168685 MISS: 93641
L2C RFO ACCESS: 62746 HIT: 4403 MISS: 58343
L2C PREFETCH ACCESS: 0 HIT: 0 MISS: 0
L2C WRITEBACK ACCESS: 178363 HIT: 178359 MISS: 4
L2C PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
L2C AVERAGE MISS LATENCY: 67.9378 cycles
LLC TOTAL ACCESS: 266203 HIT: 220467 MISS: 45736
LLC LOAD ACCESS: 93641 HIT: 71611 MISS: 22030
LLC RFO ACCESS: 58343 HIT: 41807 MISS: 16536
LLC PREFETCH ACCESS: 0 HIT: 0 MISS: 0
LLC WRITEBACK ACCESS: 114219 HIT: 107049 MISS: 7170
LLC PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS: 0
LLC AVERAGE MISS LATENCY: 125.741 cycles

```

Figure 3: 473.astar-153B.champsimtrace.xz Result

```

CPU 0 cumulative IPC: 0.151351 instructions: 100000001 cycles: 660717181
L1D TOTAL      ACCESS: 18562051 HIT: 13017689 MISS: 5544362
L1D LOAD       ACCESS: 14127866 HIT: 8860666 MISS: 5267200
L1D RFO        ACCESS: 4434185 HIT: 4157023 MISS: 277162
L1D PREFETCH   ACCESS: 0 HIT: 0 MISS: 0
L1D WRITEBACK  ACCESS: 0 HIT: 0 MISS: 0
L1D PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS:
L1D AVERAGE MISS LATENCY: 122.266 cycles
L1I TOTAL      ACCESS: 19738247 HIT: 19738247 MISS: 0
L1I LOAD       ACCESS: 19738247 HIT: 19738247 MISS: 0
L1I RFO        ACCESS: 0 HIT: 0 MISS: 0
L1I PREFETCH   ACCESS: 0 HIT: 0 MISS: 0
L1I WRITEBACK  ACCESS: 0 HIT: 0 MISS: 0
L1I PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS:
L1I AVERAGE MISS LATENCY: -nan cycles
L2C TOTAL      ACCESS: 7761451 HIT: 5150778 MISS: 2610673
L2C LOAD       ACCESS: 5267200 HIT: 2675894 MISS: 2591306
L2C RFO        ACCESS: 277162 HIT: 257795 MISS: 19367
L2C PREFETCH   ACCESS: 0 HIT: 0 MISS: 0
L2C WRITEBACK  ACCESS: 2217089 HIT: 2217089 MISS: 0
L2C PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS:
L2C AVERAGE MISS LATENCY: 196.709 cycles
LLC TOTAL      ACCESS: 4827097 HIT: 2214937 MISS: 2612160
LLC LOAD       ACCESS: 2591307 HIT: 92209 MISS: 2499098
LLC RFO        ACCESS: 19367 HIT: 18914 MISS: 453
LLC PREFETCH   ACCESS: 0 HIT: 0 MISS: 0
LLC WRITEBACK  ACCESS: 2216423 HIT: 2103814 MISS: 112609
LLC PREFETCH REQUESTED: 0 ISSUED: 0 USEFUL: 0 USELESS:
LLC AVERAGE MISS LATENCY: 165.697 cycles

```

Figure 4: 605.mcf_s-1536B.champsimtrace.xz Result