CS-683 Final Presentation

Reinforcement Learned Policy(RLR)

Team: Paradox Bits

Priyanshu Gupta
210070063@iitb.ac.in

Priyansh Jain
210070064@iitb.ac.in

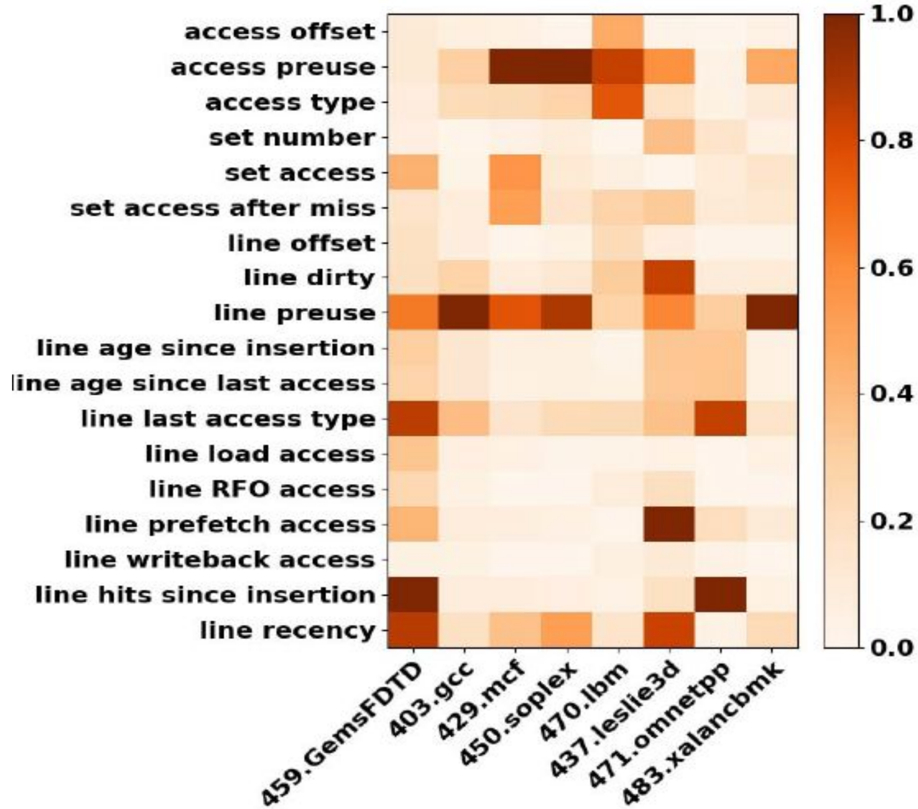
Chinmay Jadhav
210020057@iitb.ac.in

The Problem

- Motivating Task : Designing a cost-effective Cache.
  - Performance Metric : MPKI.
  - Reducing Hardware Overhead.
- Non-PC based policies : Effective only in limited cache access patterns .(eg. RRIP,LRU)
- PC- based policies :
  - Outperforms Non- PC based policies.
  - Drawbacks :
    - Requires additional storage overhead.
    - Modifications required in processor’s data path.

Proposed Solution : RLR policy

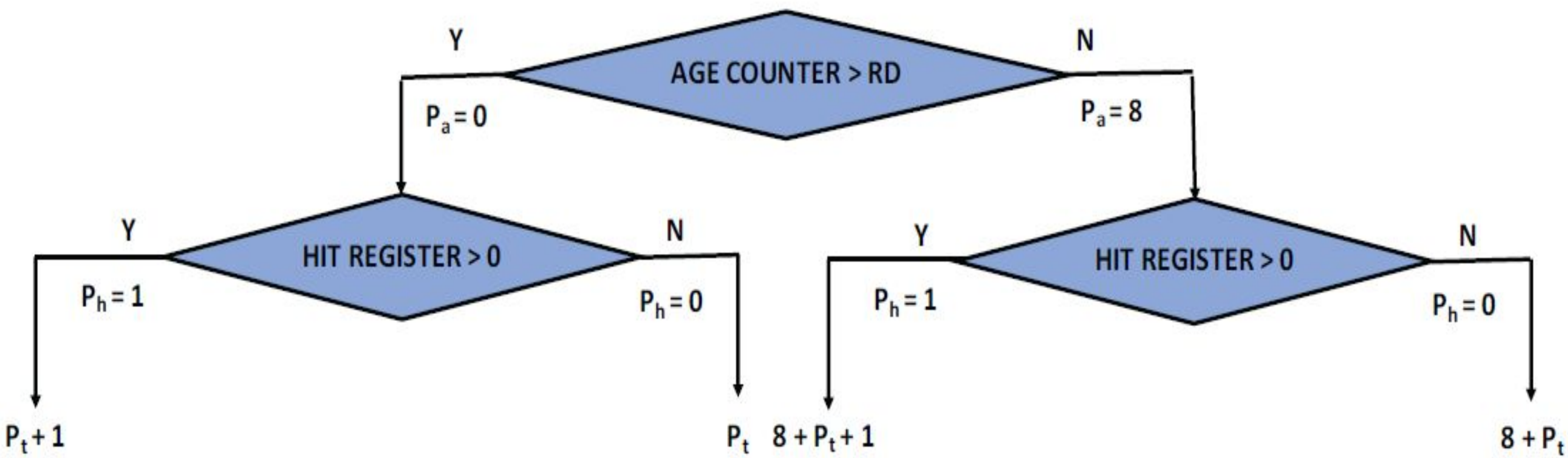
- Using RL to learn policy:
  - Uses easily obtainable features of LLC
  - Insights by observing pre trained weights to get features



About Replacement Algorithm :

- Cache lines in the set are assigned priority levels based on :
  - Age (Age priority) : $RD = 2 \times Average\ Preuse\ Distance$
  - Previous Access (Type priority)
  - Hits (Hit priority)
- On a cache miss : Eviction of line with **lowest priority.**

Priority Computation : $P_{line} = 8 \cdot P_{age} + P_{type} + P_{hit}$



Demand hits-based Priority
P_{ld} - Load Priority
P_{rfo} - RFO Priority
P_{pf} - Prefetch Priority
P_{wb} - Writeback Priority

AGE COUNTER - Set accesses since last access to the cache line
HIT REGISTER - 1 if cache line was hit at least once, 0 otherwise
P_a – Age priority
P_h – Hit priority
P_t – Type priority ($P_{ld} = P_{rfo} = P_{wb} = 1$, $P_{pf} = 0$)

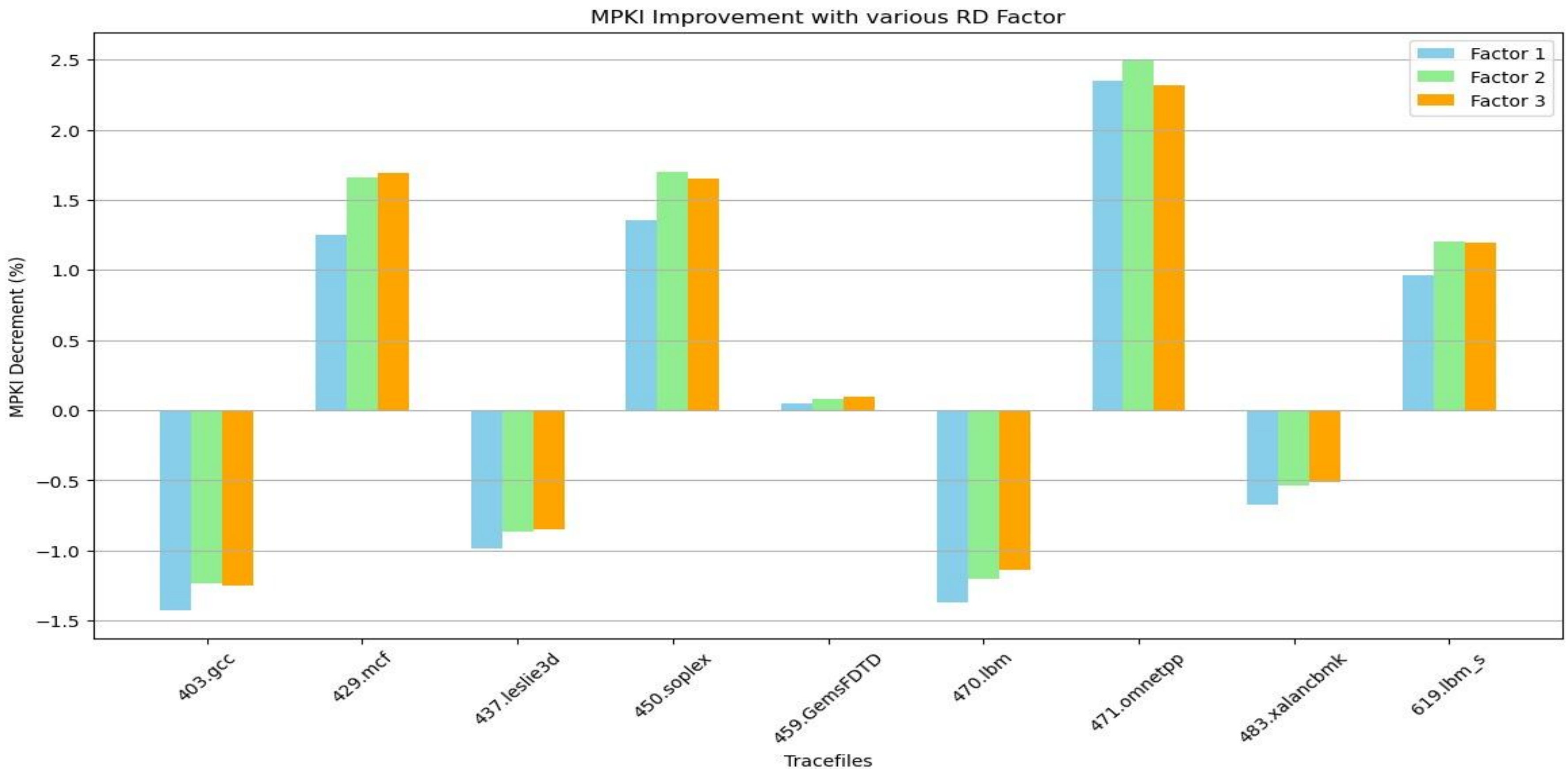
Priority:
0 Evict
10 Retain

Implementation in ChampSim:

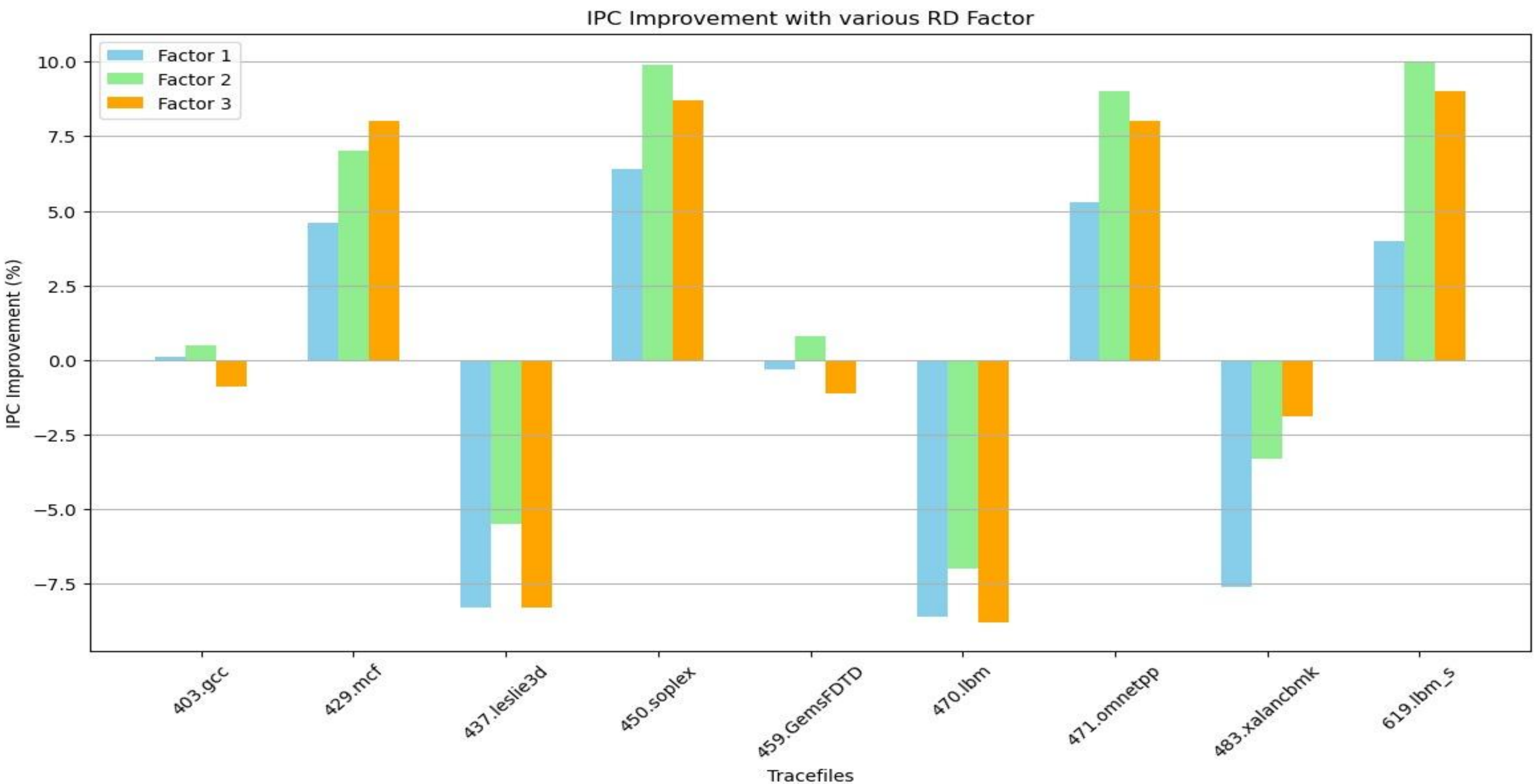
- Benchmarks : 459.GemsFDTD, 403.gcc, 429.mcf, 450.soplex, 470.lbm, 437.leslie3d, 471.omnetpp, 483.xalancbmk
- Cache Configuration :

Cache	Size	Sets	Ways	Latency(cycles)
L1I	64KB	64	8	4
L1D	64KB	64	8	5
L2C	512KB	1024	8	10
LLC	2MB	2048	16	20

Results : RD Factor (MPKI)



Results : RD Factor (IPC)

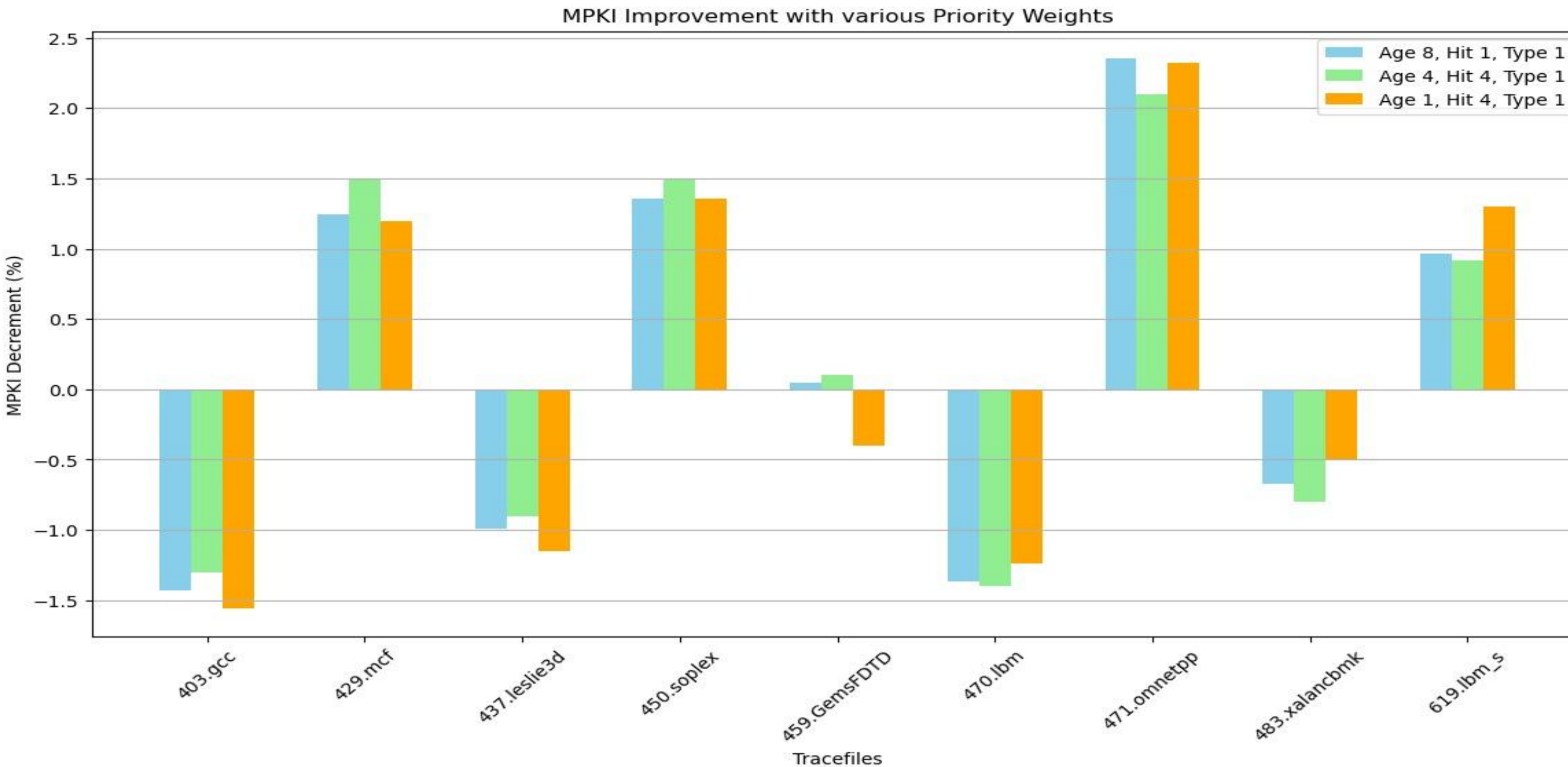


Insights from varying RD Factor :

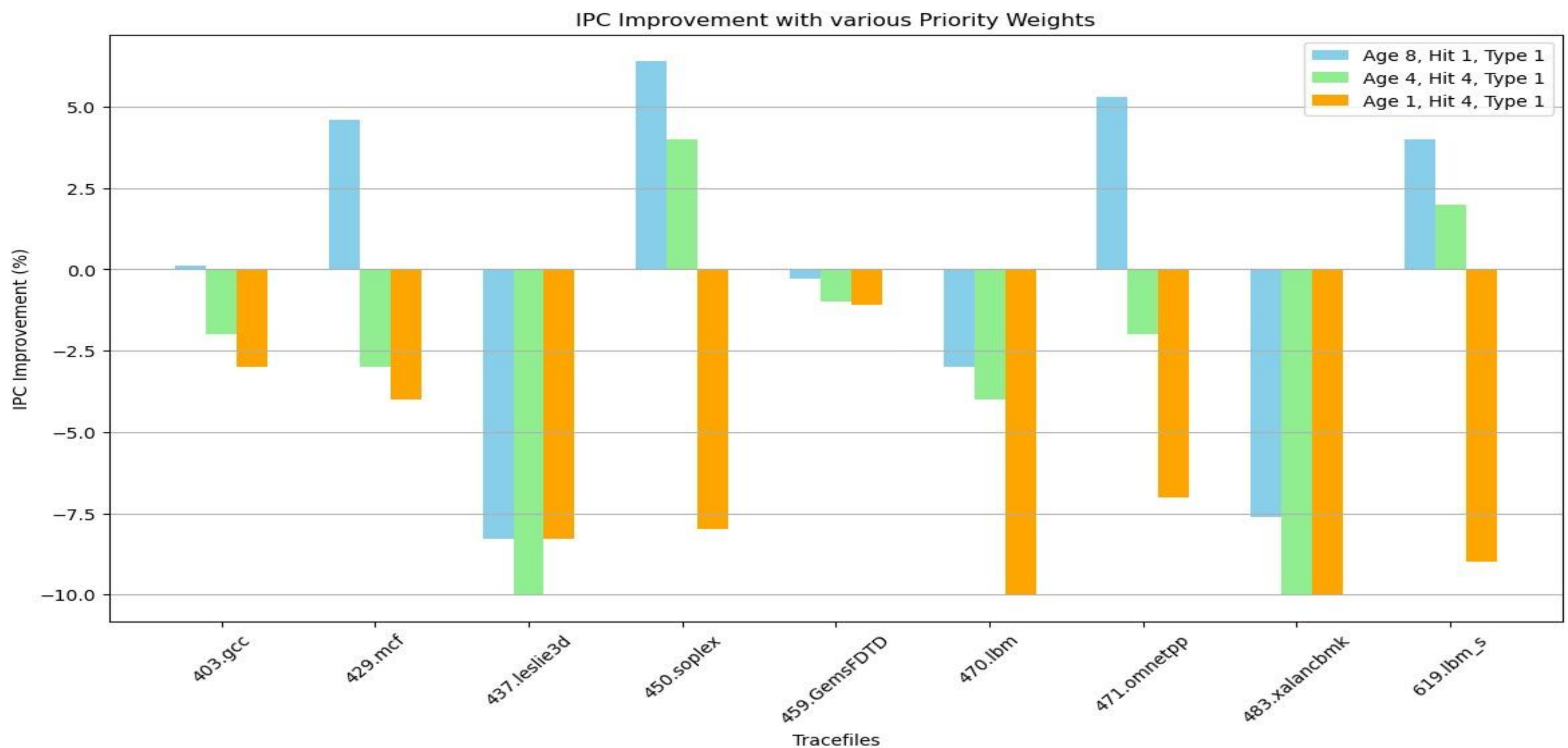
RD Factor	Overall MPKI Improvement	Overall IPC Speedup
1	0.15%	-0.22%
2	0.22%	2.15%
3	0.19%	1.85%

Optimum RD Factor: 2

Results : Weight Priority (MPKI)



Results : Weight Priority (IPC)



Insights from varying Weight Priorities

Weight Priority (Age, Hit, Type)	Overall MPKI Improvement	Overall IPC Speedup
(8,1,1)	0.22%	2.15%
(4,4,1)	0.18%	1.04%
(1,4,1)	0.10%	0.57%

Optimum Weights: (8,1,1)

Observations

- Decreasing Age Priority affects the speedup by a lot
- Incresing Hit Rate Priority did not have significant effects
- RLR shows better overall performance than LRU

Thank You