CS-683 Project checkpoint III
The Final Submission

Improving IPCP with Irregular Access Support using ISB

Sm Arif Ali, Soumik Dutta
Team Gandiva
23m0822@iitb.ac.in, 23m0826@iitb.ac.in

Problem statement

- Instruction Pointer Classifier-based Prefetcher (IPCP) handles regular patterns but falls short for irregular memory accesses.
- Need for efficient handling of irregular access patterns for applications like graph processing.
- An example access pattern involving *temporal locality* can be seen below:

A B C X Y X Y X Y A B C X Y ..

Where these are memory address accesses

Prior Works

- Bouquet of Instruction Pointers (ISCA ‘20): Established IPCP, categorizing regular memory access patterns.

- Linearizing Irregular Accesses (MICRO ‘13): Introduced Irregular Stream Buffer (ISB) to address irregular accesses.

First IP prefetcher: Constant stride

Second IP prefetcher: Complex stride

Third IP prefetcher: Global stream

Fourth prefetcher: Next-line

Temporal Stream:

A B C X Y X Y X Y *A* B C

GHB
A
B
C
X
Y
X
Y
X
Y
A

(a) GHB

Physical Address	Structural Address
A	19
B	20
C	21
X	22
Y	23

(b) ISB

Figure 1: Metadata for GHB (left) and ISB (right).

Image ref- fig1: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8980330;
fig2: https://dpc3.compas.cs.stonybrook.edu/slides/bouquet.pdf

Goal of the Project

- Integrate ISB into IPCP framework to create a new Irregular Access (IA) class.
- Improve prefetching for workloads with irregular memory access patterns.

First IP prefetcher: Constant stride

Second IP prefetcher: Complex stride

Third IP prefetcher: Global stream

ISB prefetcher: temporal stream

Fourth prefetcher: Next-line

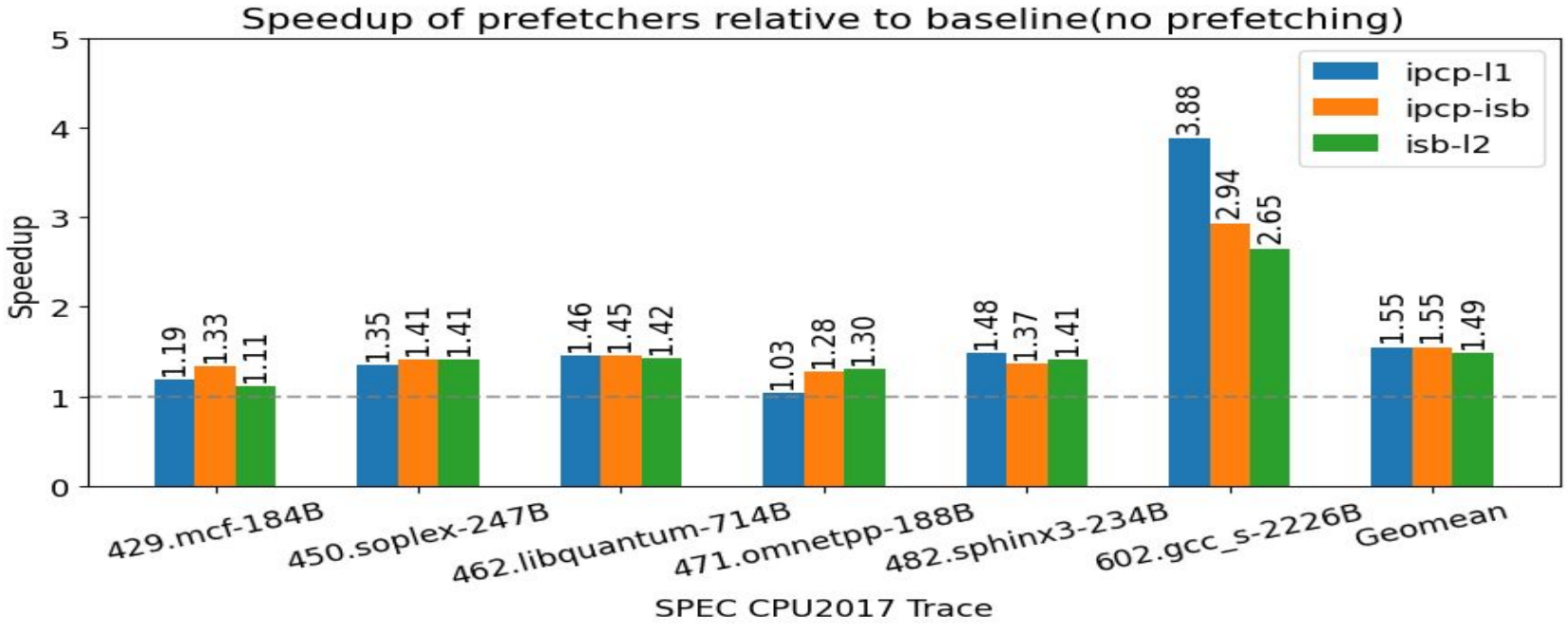
Simulator(Champsim) configuration

- Core Configuration
  - 1 core
  - 4 GHz
  - 4-wide 256-entry ROB
- TLBs
  - ITLB: 64 entries
  - DTLB: 64 entries
  - Shared L2 TLB: 1536 entries
- DRAM
  - 4GB
  - 1 channel/core
  - 6400 MT/sec
- Cache Hierarchy
  - L1 I: 32KB, 8-way, 3 cycles, PQ: 8, MSHR: 8
  - L1 D: 48KB, 12-way, 5 cycles, PQ: 8, MSHR: 16
  - L2: 512KB, 8-way, 10 cycles, PQ: 16, MSHR: 32
  - LLC: 2MB/core, 16-way, 20 cycles, PQ: 32/cores, MSHR: 64 /core
- Simulation details
  - Simulation Instructions: 50M
  - Warmup Instructions: 50M

Work done @checkpoint I

- Integrated IPCP and ISB separately in ChampSim.
- Configured and executed SPEC-CPU benchmarks.
- Initial integration of ISB within the IPCP framework completed, with early tests results.

Benchmark: SPEC-CPU17: V1 @checkpoint-I

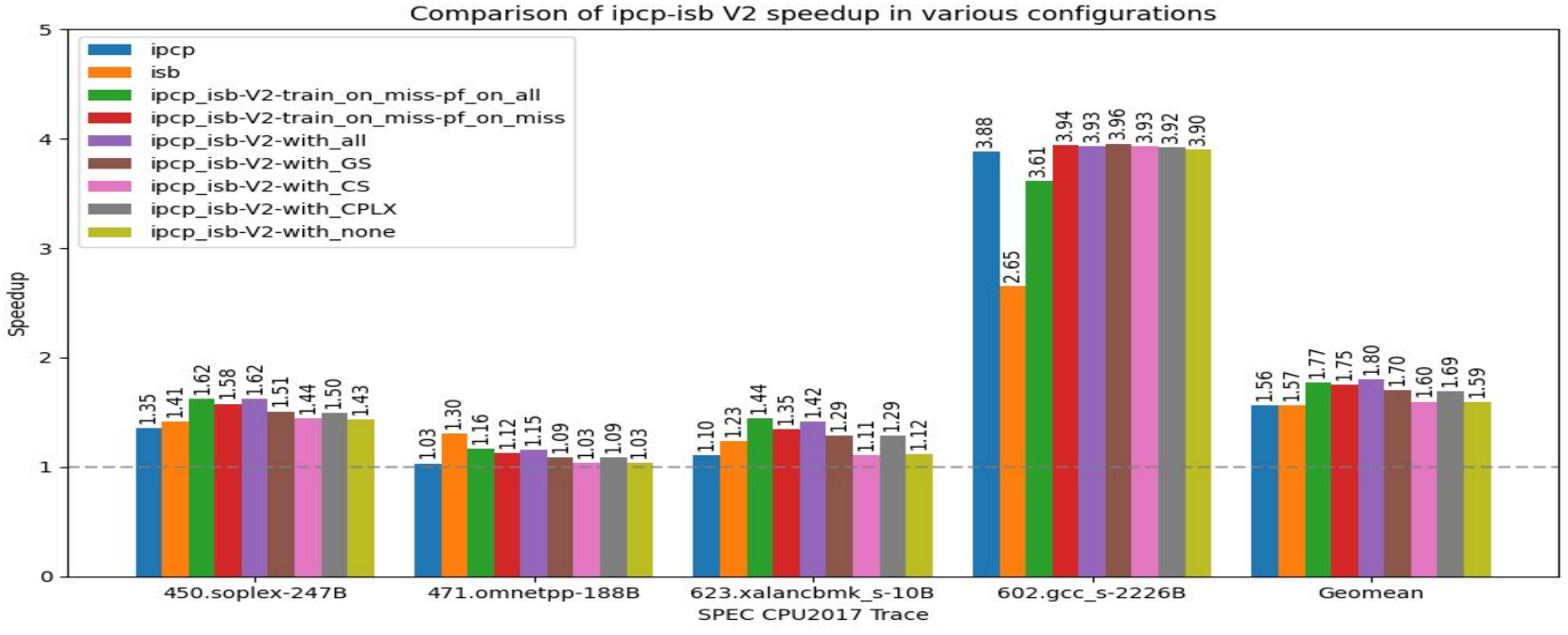


- Combined IPCP-ISB shows comparable performance to individual IPCP and ISB.
- No significant performance gain observed, but no degradation either.

Work done @checkpoint-II

- Integrated accuracy based prefetch degree adaption technique into ISB
- Implementation of Confidence counter to influence prefetch decision in ISB - found not to be beneficial.
- Tested by sharing the IPCP IP-Table with ISB to keep metadata - found not to be beneficial
- Refined classification logic for Irregular Access(IA) class

Benchmark: SPEC-CPU17: V2 @checkpoint-II

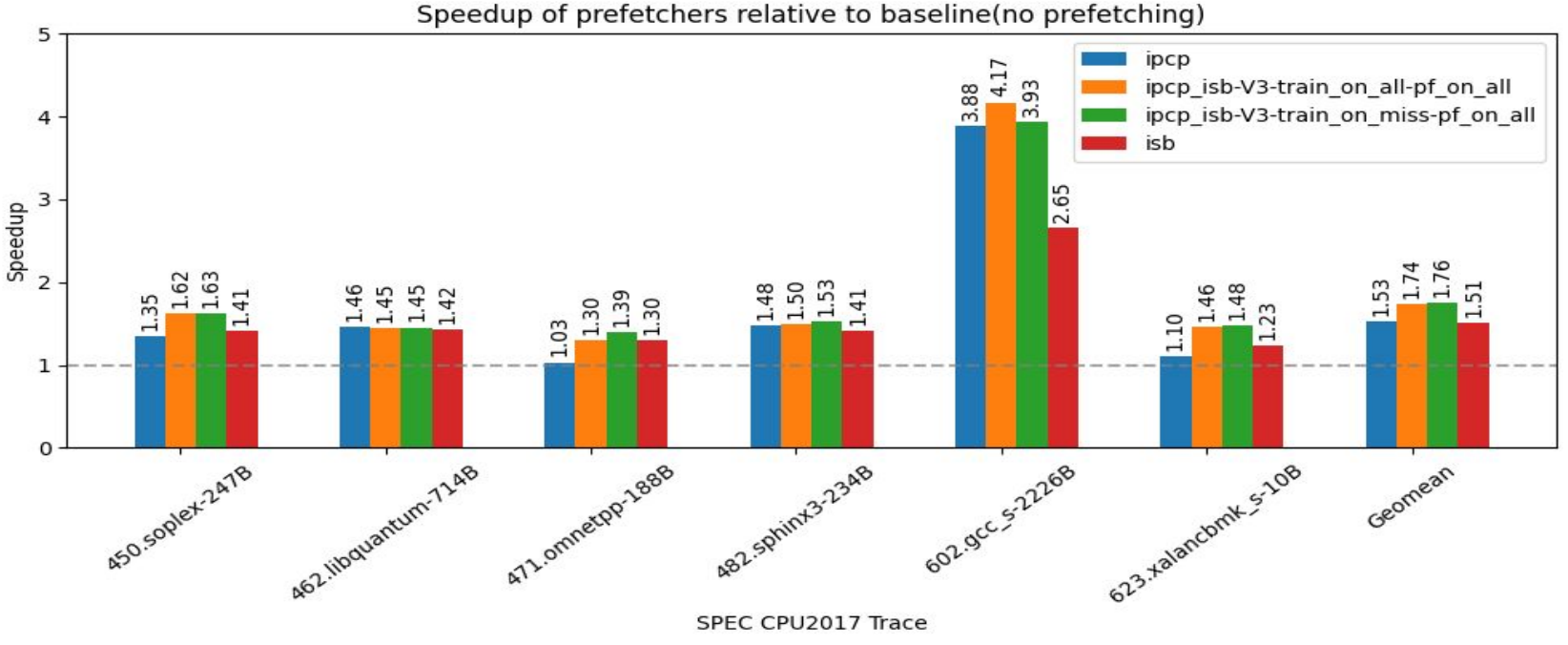


- Ipcp_isb (with_all) where isb operates simultaneously alongside all other prefetchers gives better result.
- Omnetpp is an exception.

Work done @checkpoint-III

- Refined Classification logic by evaluating more configurations.
- Found a sweet spot for handling both gcc and omnetpp gracefully.
- Adjusting ISB parameters: buffer size, and stream length - found not to be useful enough.
- Final take on optimal IPCP class design with ISB.
- Evaluated per class contribution in the combined prefetcher.
- Evaluated Accuracy, Coverage, MPKI of final version.

Benchmark: SPEC-CPU17: V3 @checkpoint-III



- Achieves better performance compared to earlier configurations. But have trade-offs as follows.

Choosing between approaches:

- ISB prefetcher statistics for Omnetpp trace:

Factors	Train on miss-pf on all	Train on all-pf on all
Triggers	16M	16M
Training Unit Size	553	1510
Addr Table Size	380K	385K
Structural address exists	9.9M(61.34%)	12.5M(77.37%)
Prefetches issued	11.7M(71.94%)	7.7M(47.77%)
No translations	3007M	3126M

Choosing between approaches:

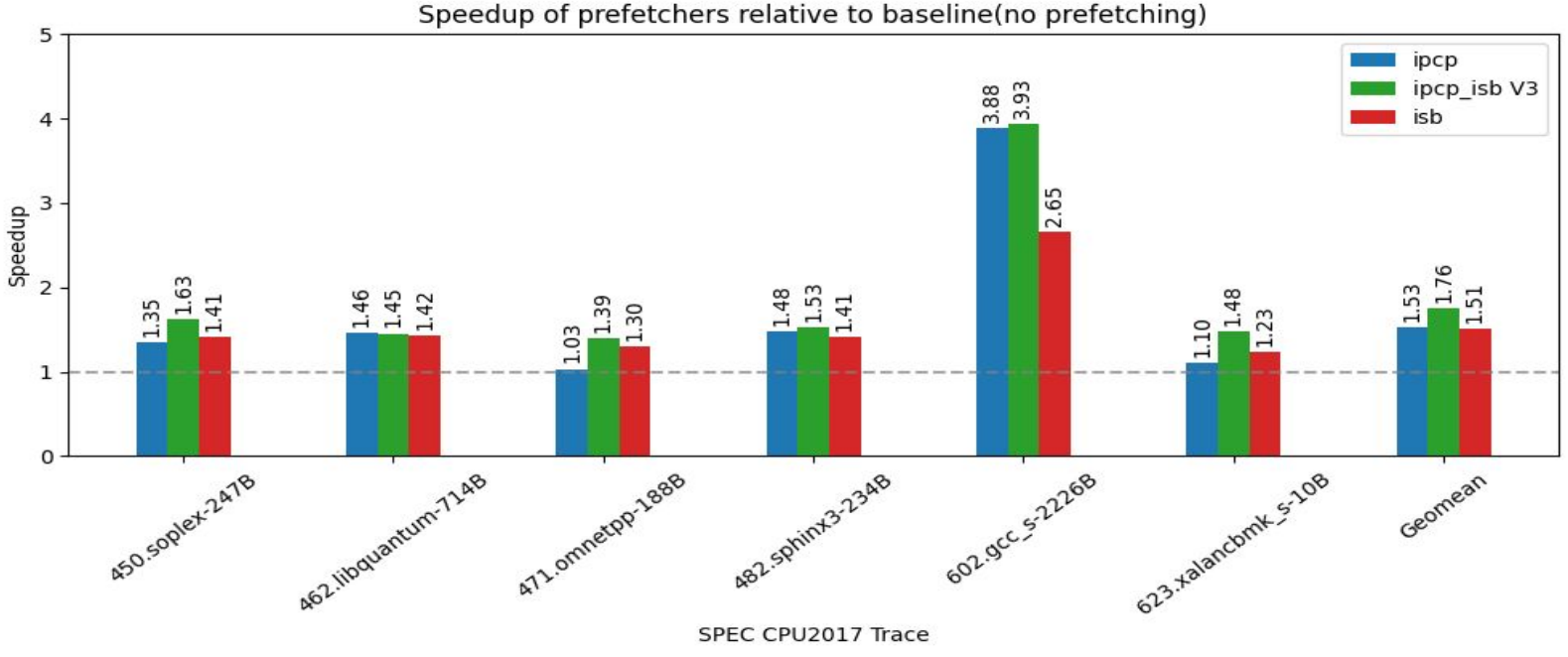
Degree	Train on miss-pf on all	Train on all-pf on all
8	5264	528
7	527	222
6	493	500
5	606	964
4	662	1636
3	806	2849
2	1120	6180
1	467048	536033
0	297003	617745

Choosing between approaches:

- ISB prefetcher statistics for gcc trace:

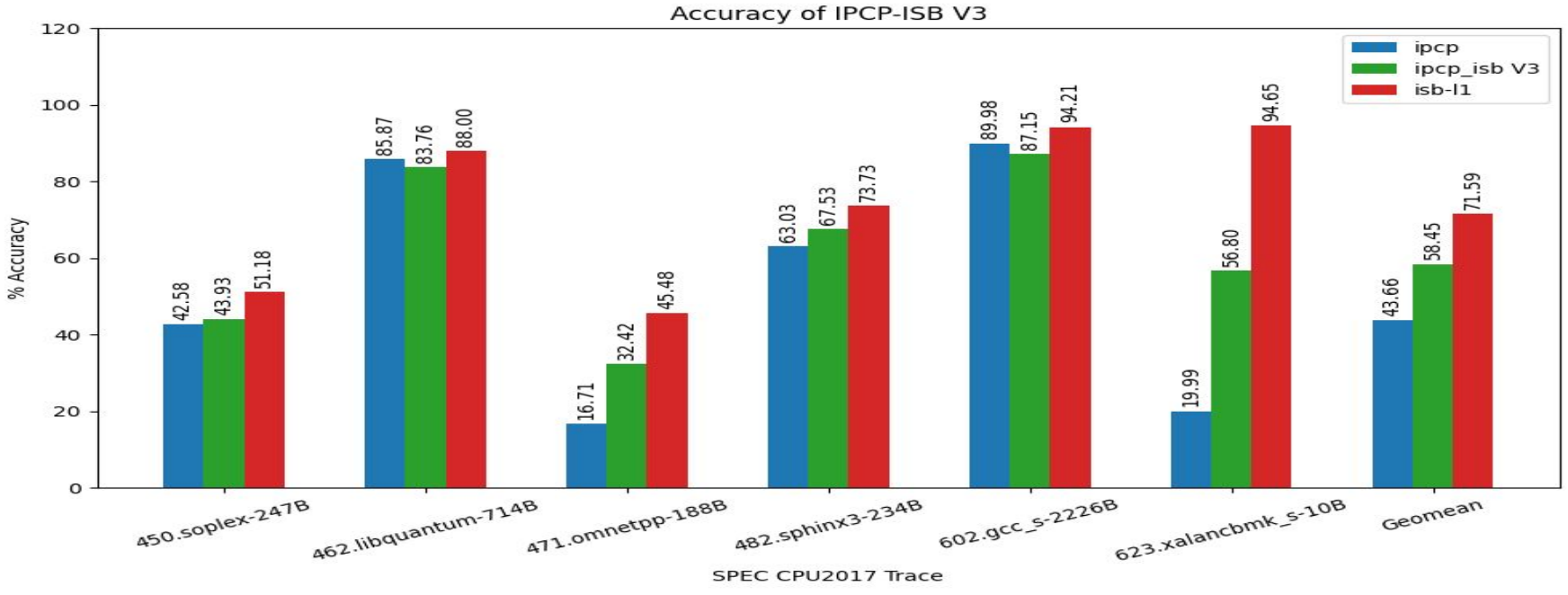
Factors	Train on miss-pf on all	Train on all-pf on all
Triggers	125M	125M
Training Unit Size	64	146
Addr Table Size	305K	2.5M
Structural address exists	873K(6.99%)	9.7M(77.74%)
Prefetches issued	787K(6.29%)	4.3M(34.5%)
No translations	583M	215M

IPCP_ISB: Final take



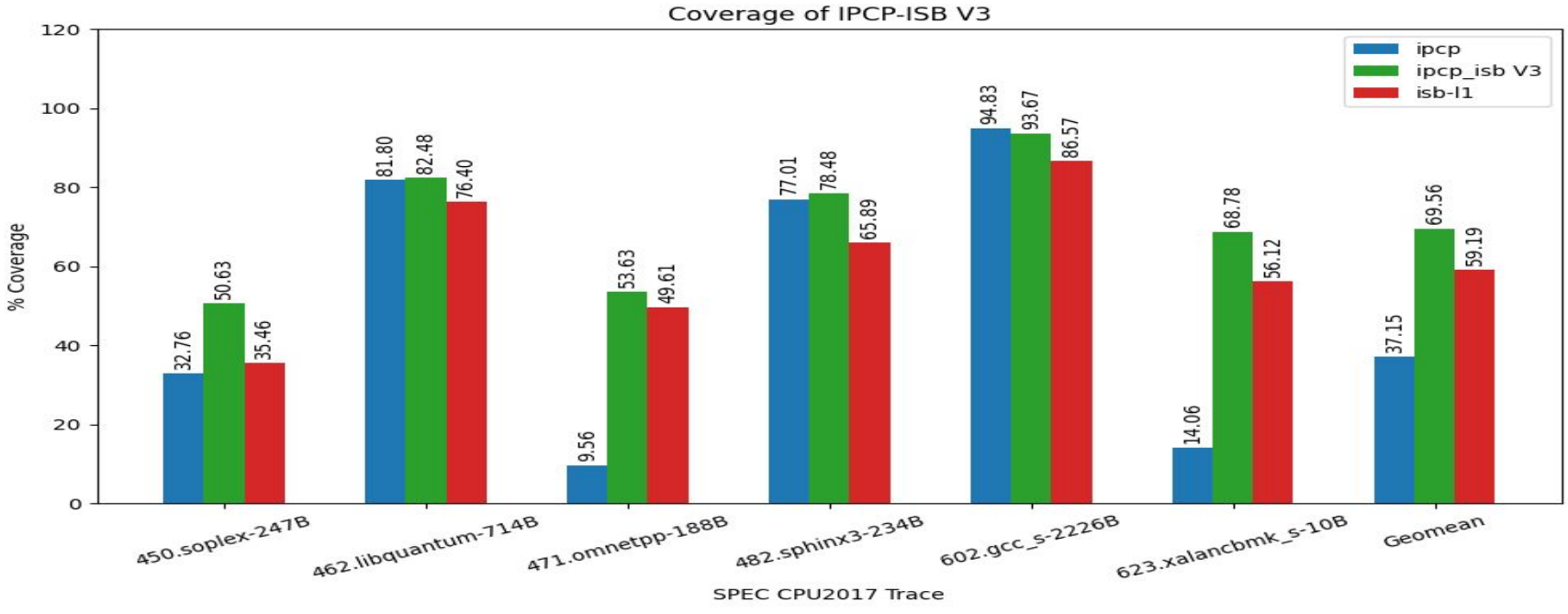
- Finally we propose this combined prefetcher with geo-mean greater than both IPCP and ISB.

IPCP_ISB: Accuracy



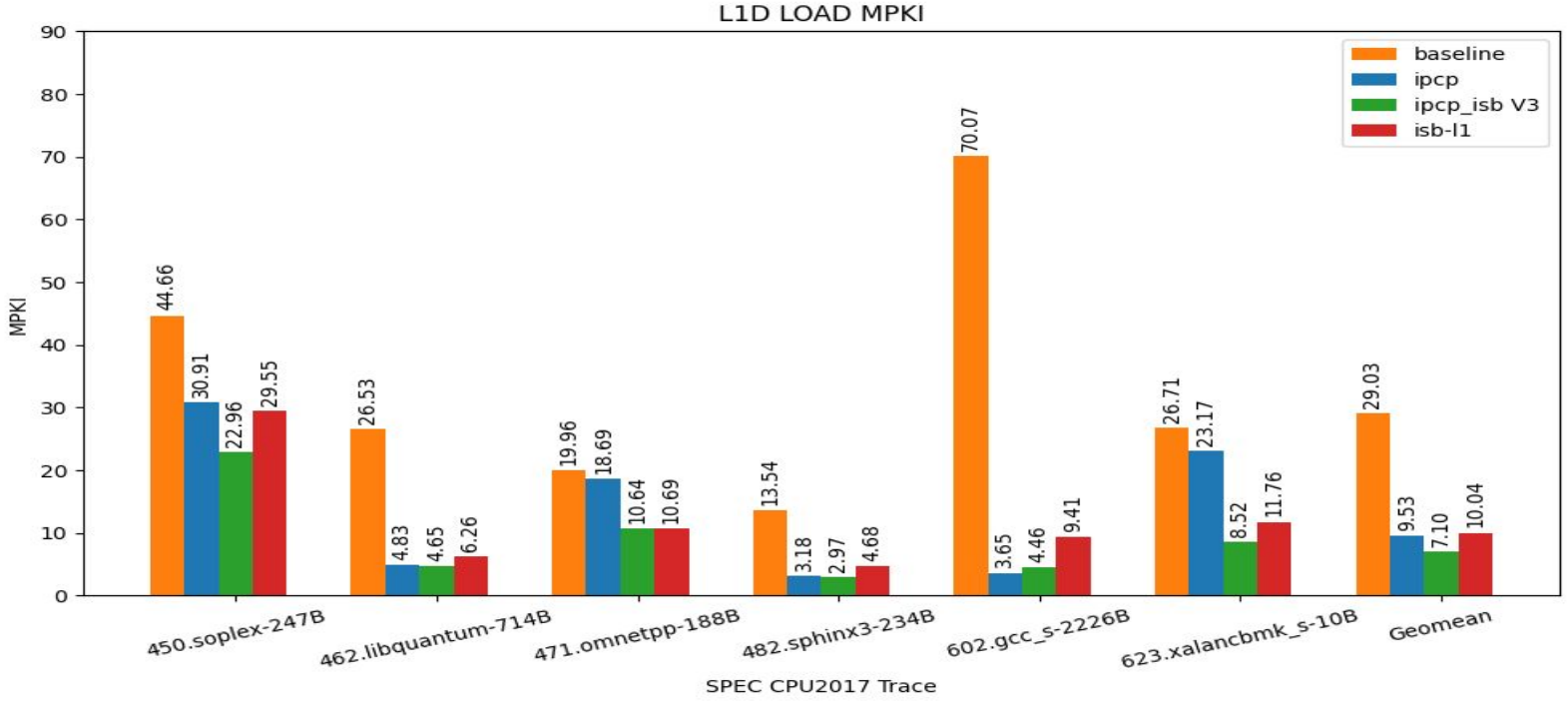
- Our proposed prefetcher Accuracy is standing in between the existing two.

IPCP_ISB: Coverage



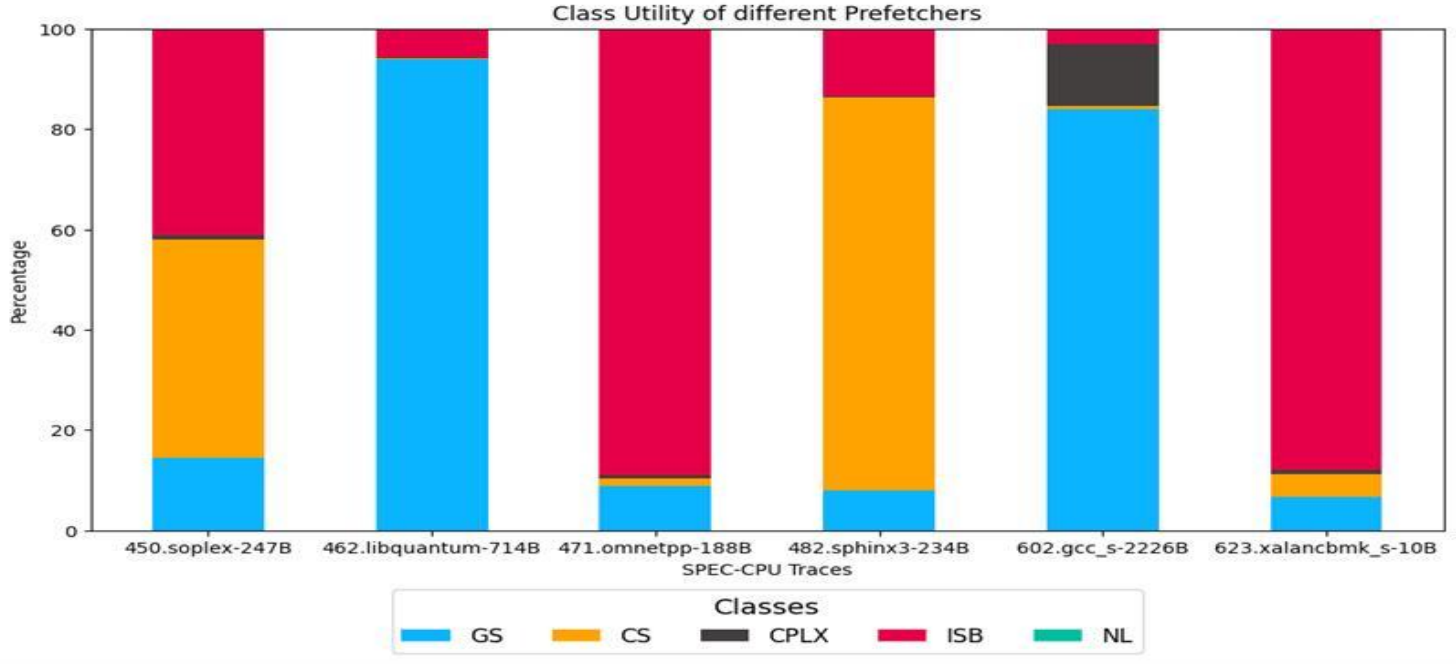
- Coverage is more in case of our proposed prefetcher.

IPCP_ISB: MPKI



- MPKI is less than others in case of our proposed prefetcher.

IPCP_ISB: class wise utilization



- Different IPCP and ISB prefetchers are utilised dynamically depending upon load.

Insights

- IPCP effectively handles regular access streams, ensuring high performance for predictable patterns.
- ISB excels in managing irregular access streams, addressing unpredictable memory accesses.
- Our proposed IPCP_ISB combined prefetcher synergizes both approaches, significantly reducing MPKI compared to standalone prefetchers.
- This hybrid solution leverages the strengths of both worlds, delivering a clear performance advantage.

Future work:

- Focus on minimizing the storage footprint of the ISB prefetcher’s address reference stream store to enhance scalability and reduce memory overhead.
- Conduct comprehensive testing using a wider variety of benchmarks to evaluate performance across diverse workloads and scenarios.

Code demo

- Walk through the key parts of the code integration.
- Codebase is available at Github repository.

Github link

- https://github.com/ArifAli-0/CS683-Project.git

Video link

- Checkpoint_1: https://youtu.be/JrZYAXMjjzY
- Checkpoint_2: https://youtu.be/AmGMXzYUihc
- Checkpoint_3: https://youtu.be/CuVzCN3LcRc

Thank you