

## Final Project Proposal:

The project aims to extend the Instruction Pointer Classifier-based Prefetcher (IPCP) from the paper "Bouquet of Instruction Pointers: Instruction Pointer Classifier-based Hardware Prefetching" by integrating a new prefetcher to the bouquet for irregular memory access patterns. Specifically, we propose to incorporate the Irregular Stream Buffer (ISB), which was introduced in "Linearizing Irregular Memory Accesses for Improved Correlated Prefetching".

IPCP currently classifies instruction pointers into three categories: CS (Constant Stride), CPLX (Complex Patterns), and GS (Global Stride), each designed to handle regular access patterns efficiently. However, IPCP does not address the challenge of irregular memory access patterns effectively. Our objective is to introduce a new class called Irregular Access (IA) within the IPCP framework, targeting irregular accesses where existing classifications fall short.

By integrating the ISB into IPCP and defining a new IA class, we aim to improve prefetching for workloads with irregular memory access patterns. This approach is particularly relevant for applications such as graph processing, where irregular accesses dominate and are not captured well by traditional prefetching techniques. The project will evaluate the performance of the modified IPCP on the GAP and memory intensive SPEC CPU2017 benchmarks. The metrics for evaluation include speedup and MPKI compared to both no prefetching and the original IPCP, as well as the coverage and accuracy of the modified IPCP. Additionally, we will assess the contribution of each class (CS, CPLX, GS, IA) to the overall coverage, class priority, and class utility (both individual and combined).

## Related works (Any two to three papers appeared in top conferences)

1. Bouquet of Instruction Pointers: Instruction Pointer Classifier-based Hardware Prefetching, ISCA '20
2. Linearizing Irregular Memory Accesses for Improved Correlated Prefetching, MICRO '13

## What tools, simulators, benchmarks will you use?

We will use ChampSim, to implement and evaluate the extended IPCP prefetcher with the new Irregular Access (IA) class. For benchmarks, we will use the GAP Benchmark Suite to test performance on irregular memory access workloads and memory-intensive benchmarks from SPEC CPU2017 to evaluate general memory-bound scenarios.

Provide deliverables for next three checkpoints: Checkpoint-1: October 20, Checkpoint-2: November 8, Checkpoint-3: November end (as per academic calendar):

### Checkpoint-1 (October 26):

1. Complete the design and planning for the Irregular Access (IA) class extension in IPCP.
2. Implement the initial integration of the Irregular Stream Buffer (ISB) and IPCP in ChampSim.

### Checkpoint-2 (November 10):

1. Complete the integration of ISB into IPCP, ensuring proper classification of instruction pointers for irregular accesses.
2. Conduct initial performance evaluation using the GAP Benchmark Suite.

3. Analyze basic metrics like MPKI, coverage, and accuracy, comparing results against baseline IPCP and no prefetching.

Checkpoint-3 (November 26):

1. Finalize testing with both GAP Benchmark Suite and memory-intensive SPECCPU2017 benchmarks.
2. Complete analysis of speedup, MPKI, coverage, accuracy, class priority, and class utility for the extended IPCP.
3. Prepare the final report, summarizing findings over baseline IPCP.

Who will do what? Please spell out a tentative division of labor.

Sm Arif Ali:

1. Work on the design and planning for the new Irregular Access (IA) class within the IPCP framework.
2. Implement the integration of the Irregular Stream Buffer (ISB) and IPCP in ChampSim.
3. Perform initial performance evaluation and data collection using the GAP Benchmark Suite.

Soumik Dutta:

1. Focus on benchmarking, setting up the SPECCPU2017 and GAP workloads.
2. Assist in refining the classification logic for the IA class and validating the prefetching mechanism.
3. Analysis of results, including speedup, MPKI, coverage, accuracy, and class utility.

Both members will collaborate on testing, debugging, and preparing the final report and presentation.