Machine Learning for Memory Access Prediction and Data Prefetching

Pengmiao Zhang, Viktor K. Prasanna (Adviser)
Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California

USC University of Southern California

Introduction

Background:
- Development of processors: TPUs, accelerators, heterogenous architectures
- Data intensive workloads: graph analytics, machine learning algorithms, AI applications
- Bottleneck shifting towards memory performance

Data Prefetching:
- Predict future memory accesses
- Issue a fetch in advance of actual reference
- Hide memory latency
- Improve instructions per cycle (IPC)

Research Hypothesis:
Machine learning can be used to achieve
- High-quality memory access prediction
- High-performance data prefetching
- Overall system performance improvement

Challenges

Memory Access Prediction

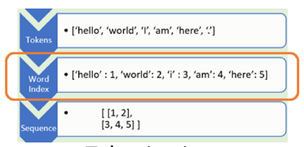
Benchmarks	# PCs	# Addresses
SPEC 06	23~893	60.0K~2.21M
SPEC 17	26~1126	62.1K~1.78M
GAP	63~118	0.56M~1.25M

Class Explosion

4049e5	d79e62a544c0
4049cc	5829a6f2f6c0
4049cc	5829a6f2f700
4049de	8ad3301ccbc0
4049de	28e837d27940

Input; Label?; Label?

Labeling

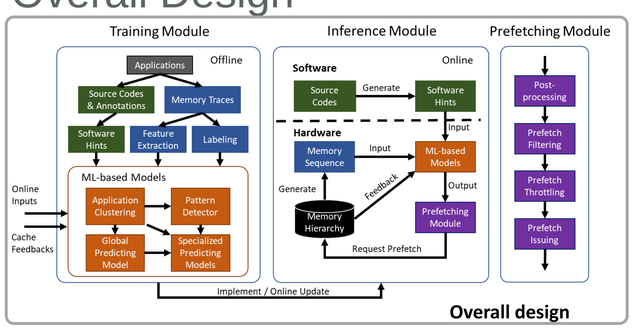


Tokenization

Latency

ML-Based Prefetching
- Integrating ML-based predictor and architecture
- Prefetching workspace: virtual/physical address
- Prefetching configuration: degree, distance
- Model online update

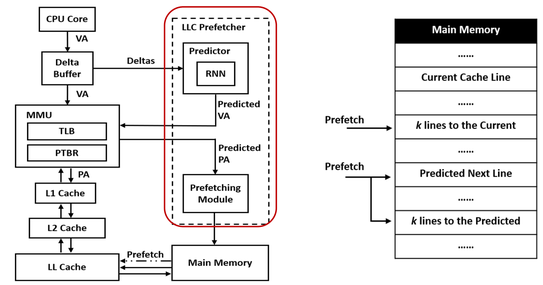
Overall Design



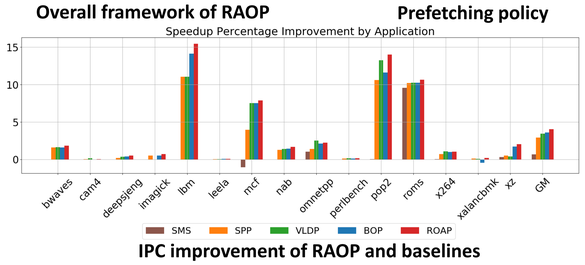
Overall design

Approach

Optimization 1: RNN Augmented Offset Prefetcher (RAOP)
- Developing a framework integrating ML-based memory access predictor, computer architecture, and an existing offset prefetcher
- Outperforms state-of-the-art prefetchers

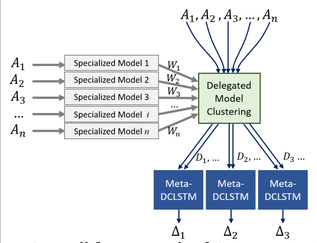


Overall framework of RAOP — Prefetching policy

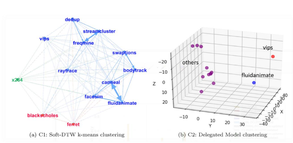


IPC improvement of RAOP and baselines

Optimization 2: Clustering-Driven Meta-LSTM for Memory Access Prediction (C-MemMAP)
- Can m models predict A applications ($m \ll A$)?
- Trace clustering: delegated model (DM) clustering
- Multi-task: meta-learning for LSTM
- Shows higher adaptability and generalizability

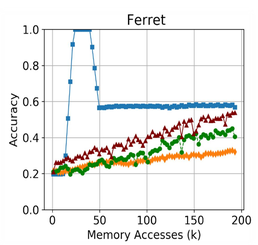


Overall framework of C-MemMAP

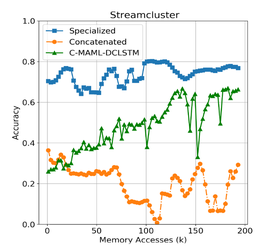


(a) C1: Soft-DTW k-means clustering (b) C2: Delegated Model clustering

Trace clustering

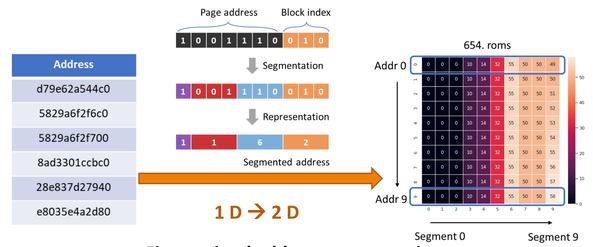


Adaptability



Generalizability

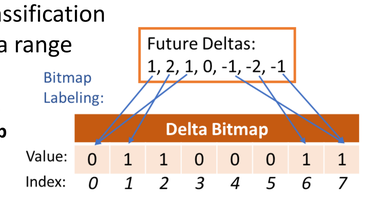
Optimization 3: Address Segmentation for Attention-Based Prefetching (TransFetch)
- Goal: address the class explosion, labeling, tokenization, and latency challenges
- Input: fine-grained address segmentation



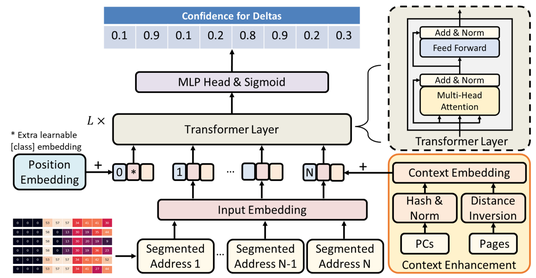
Fine-grained address segmentation

- Labeling/output: delta bitmap
  - Multi-label classification
  - Deltas within a range

Mapping address deltas to a bitmap

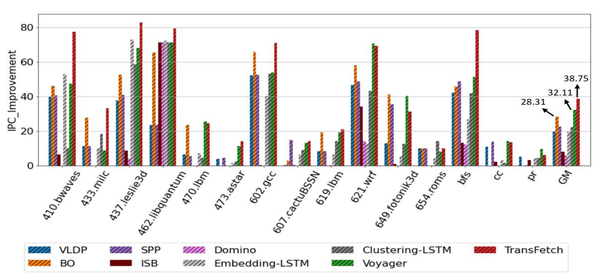


- Model: attention-based predictor
  - Multi-head self-attention
  - Powerful and parallelizable



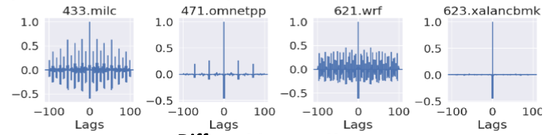
Attention-based memory access predictor

- Results: TransFetch achieves 38.75% IPC improvement, outperforming rule-based prefetcher BOP by 10.44%, outperforming ML-based prefetcher Voyager by 6.64%

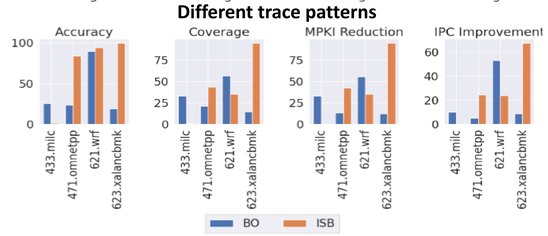


TransFetch performance

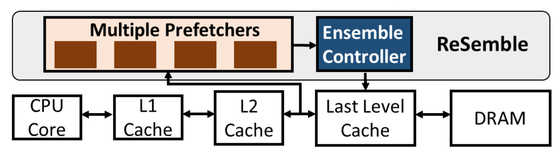
Optimization 4: Reinforced Ensemble Framework for Prefetching (ReSemble)
- Different trace patterns benefit from different prefetchers
- We propose a reinforcement learning-based ensemble framework that enables multiple prefetchers to complement each other



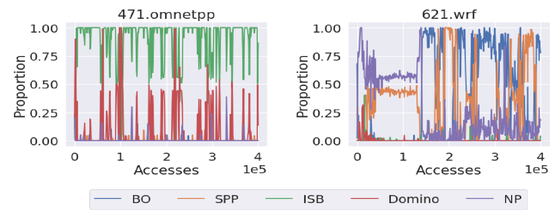
Different trace patterns



Different trace benefits from different prefetchers



Overall structure of ReSemble



Learning process of ReSemble

Conclusion

- We developed RAOP for hardware prefetching framework, C-MemMAP for smaller model size, TransFetch for higher prediction performance and parallelizability, and ReSemble for online adaptation to various trace patterns
- In the future we will incorporate more software and context information for higher prediction and prefetching performance

Acknowledgements
This work is supported by U.S. National Science Foundation under grant numbers CCF-1912680 and PPoSS-2119816.

Website