Pythia
A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera, Konstantinos Kanellopoulos, Anant V. Nori, Taha Shahroodi, Sreenivas Subramoney, Onur Mutlu
Mainly use **one** program context info. for prediction

Lack system awareness

Lack in-silicon customizability

Why prefetchers do not perform well?
Autonomously learns to prefetch using multiple program context information and system-level feedback

In-silicon customizable to change program context information or prefetching objective on the fly
Brief Overview of Pythia

Pythia formulates prefetching as a **reinforcement learning** problem.

**Agent**
- State ($S_t$)
- Reward ($R_{t+1}$)
- Action ($A_t$)

**Environment**

**Processor & Memory Subsystem**
- Features of memory request to address $A$ (e.g., PC)
- Reward

**Prefetcher**
- Prefetch from address $A+offset$ ($O$)
We evaluate Pythia using a wide-range of workloads

Pythia improves performance by
3.4% and 3.8% in single-core
7.7% and 9.6% in twelve-core
16.9% and 20.2% in memory bandwidth-constrained core
over state-of-the-art MLOP and Bingo prefetchers

We gain 7.8% more performance on top of basic Pythia configuration by simply customizing reward values for graph workloads

Realistic, practical implementation
No complex structures, only simple tables.
Only 1.03% area and 0.4% power of a desktop-class processor
Pythia is Open-sourced

https://github.com/CMU-SAFARI/Pythia

• MICRO’21 artifact evaluated
• Champsim source code + Chisel modeling code
• All traces used for evaluation
Pythia
A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

Rahul Bera, Konstantinos Kanellopoulos, Anant V. Nori, Taha Shahroodi, Sreenivas Subramoney, Onur Mutlu

Session 9B, October 21, 8:00 PM CEST