



Pythia

A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning

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



1

Mainly use one program context info. for prediction

2

Lack system awareness

3

Lack in-silicon customizability

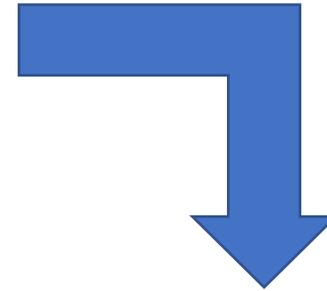
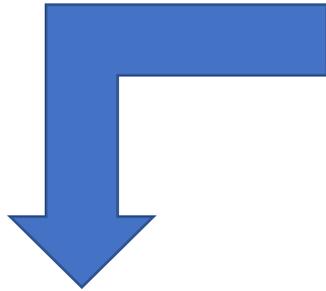


Why prefetchers do not perform well?





Pythia



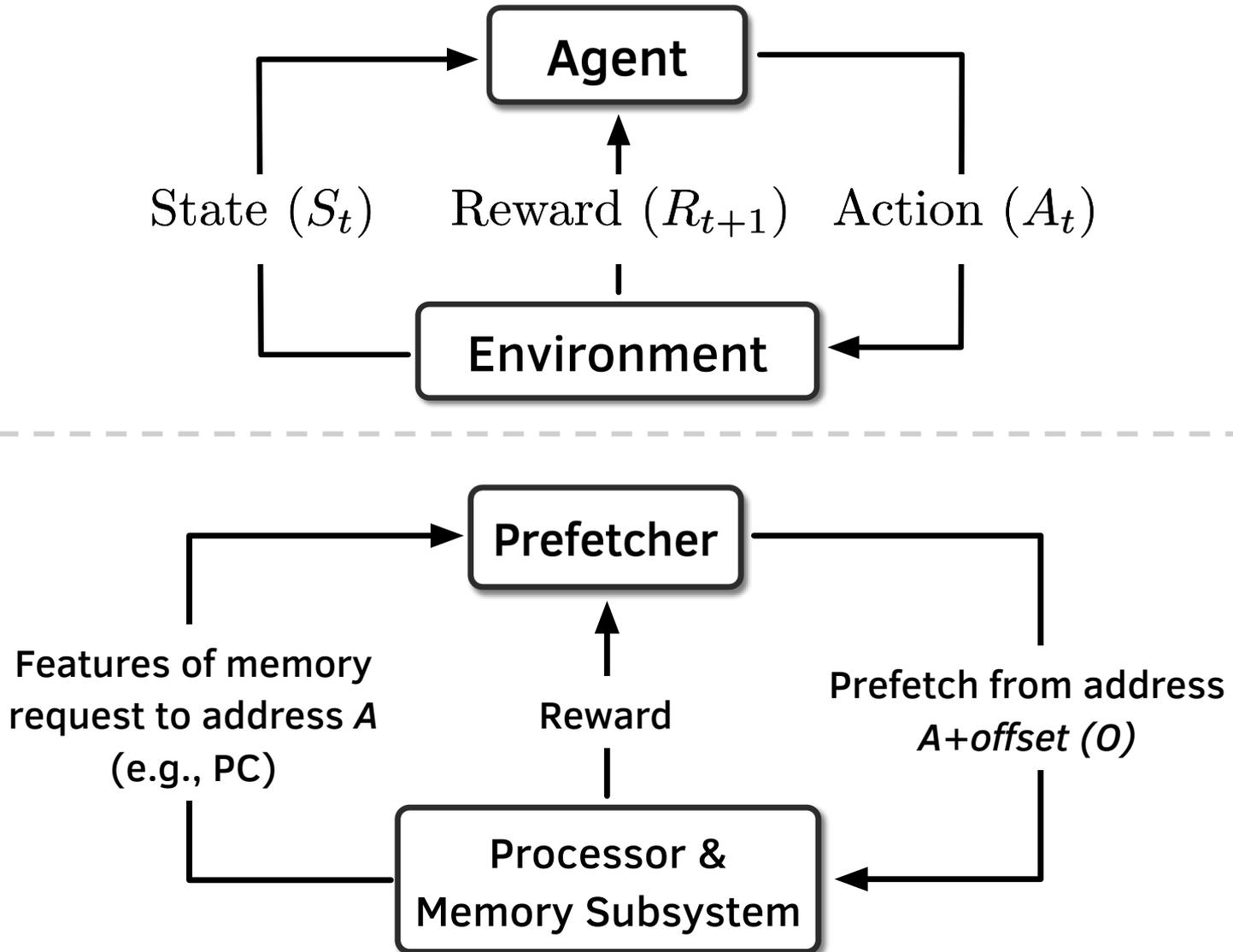
Autonomously learns to prefetch using multiple program context information and system-level feedback

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



1

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

2

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

3

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



The screenshot shows the GitHub repository for CMU-SAFARI/Pythia. The repository is public and has 1 branch and 5 tags. The commit history shows a recent bump to v1.3 by rahulbera 5 days ago. The repository contains several folders and files, including branch, config, experiments, inc, prefetcher, replacement, scripts, src, tracer, .gitignore, LICENSE, LICENSE.champsim, Makefile, README.md, build_champsim.sh, build_champsim_highcore.sh, logo.png, and setvars.sh. The repository also has a README, a View license button, and a Releases section with 4 releases, including the latest v1.3. The Packages section shows no packages published. The Languages section shows the following distribution: C++ 56.3%, XS 7.8%, Raku 3.3%, Perl 28.0%, C 2.5%, and Other 1.0%.

The screenshot shows the README.md file for the Pythia project. The README describes Pythia as a Customizable Hardware Prefetching Framework Using Online Reinforcement Learning. It includes a Table of Contents with 11 sections: 1. What is Pythia?, 2. About the Framework, 3. Prerequisites, 4. Installation, 5. Preparing Traces (with sub-sections for More Traces), 6. Experimental Workflow (with sub-sections for Launching Experiments and Rolling up Statistics), 7. HDL Implementation, 8. Citation, 9. License, 10. Contact, and 11. Acknowledgements. The README also includes a section titled "What is Pythia?" which states that Pythia is a hardware-realizable, light-weight data prefetcher that uses reinforcement learning to generate accurate, timely, and system-aware prefetch requests. It further explains that Pythia formulates hardware prefetching as a reinforcement learning task, observing multiple different types of program context information to take a prefetch decision. For every prefetch decision, Pythia receives a numerical reward that evaluates prefetch quality under the current memory bandwidth utilization. Pythia uses this reward to reinforce the correlation between program context information and prefetch decision to generate highly accurate, timely, and system-aware prefetch requests in the future.



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

SAFARI
SAFARI Research Group
safari.ethz.ch

ETH zürich

intel

TU Delft