数据缓存预取：最终报告

演讲：邓开文、赵龙征洋、杜宇航

组员：田慧楠、冯俊杰、王一鸣、张一凡、吴柏宁、冯昭栋、张乐然、金宇坤

2024/6/7

1 概述

- 基本项（必做，难度3颗星）：
  1. 理解多级缓存技术中缓存预取技术的作用与运作机理，调研先进处理器中缓存预取技术的应用。
  2. 阅读玄铁C910缓存预取单元的源码（`ct_lsu_pfu.v`），将C910预取单元的运作机制撰写为文字报告。
  3. 使用Chisel实现预取单元的硬件结构，将生成的Verilog替换玄铁源码中的`ct_lsu_pfu.v`文件，然后在开源玄铁平台上能正确跑通Coremark。
- 挑战项（选做，难度6颗星）：
  1. 调研顶会上更加先进的预取算法，并进行比较，生成洞察报告。
  2. 选择一个预取算法进行硬件实现，并和原预取单元进行PPA比较。

1 概述

- ct_lsu_pfu.v
- ct_lsu_pfu_gpfb.v
- ct_lsu_pfu_gsdb.v
- ct_lsu_pfu_pfb_entry.v
- ct_lsu_pfu_pfb_l1sm.v
- ct_lsu_pfu_pfb_l2sm.v
- ct_lsu_pfu_pfb_tsm.v
- ct_lsu_pfu_pmb_entry.v
- ct_lsu_pfu_sdb_cmp.v
- ct_lsu_pfu_sdb_entry.v

Update pfu.sc
fdu-dkw on 2024/5/29

Add files via upload
wym-FDU on 2024/5/

Delete main/scala/sc
wym-FDU on 2024/5/

Merge branch 'src_n
fdu-dkw on 2024/5/29

Merge branch 'pfb
fdu-dkw on 2024/5/

Merge branch 'sr
fdu-dkw on 2024

Merge branch
fdu-dkw on 20

Update ct_lsu_
fzd on 2024/5/

Update and rena
Janleron on 2024

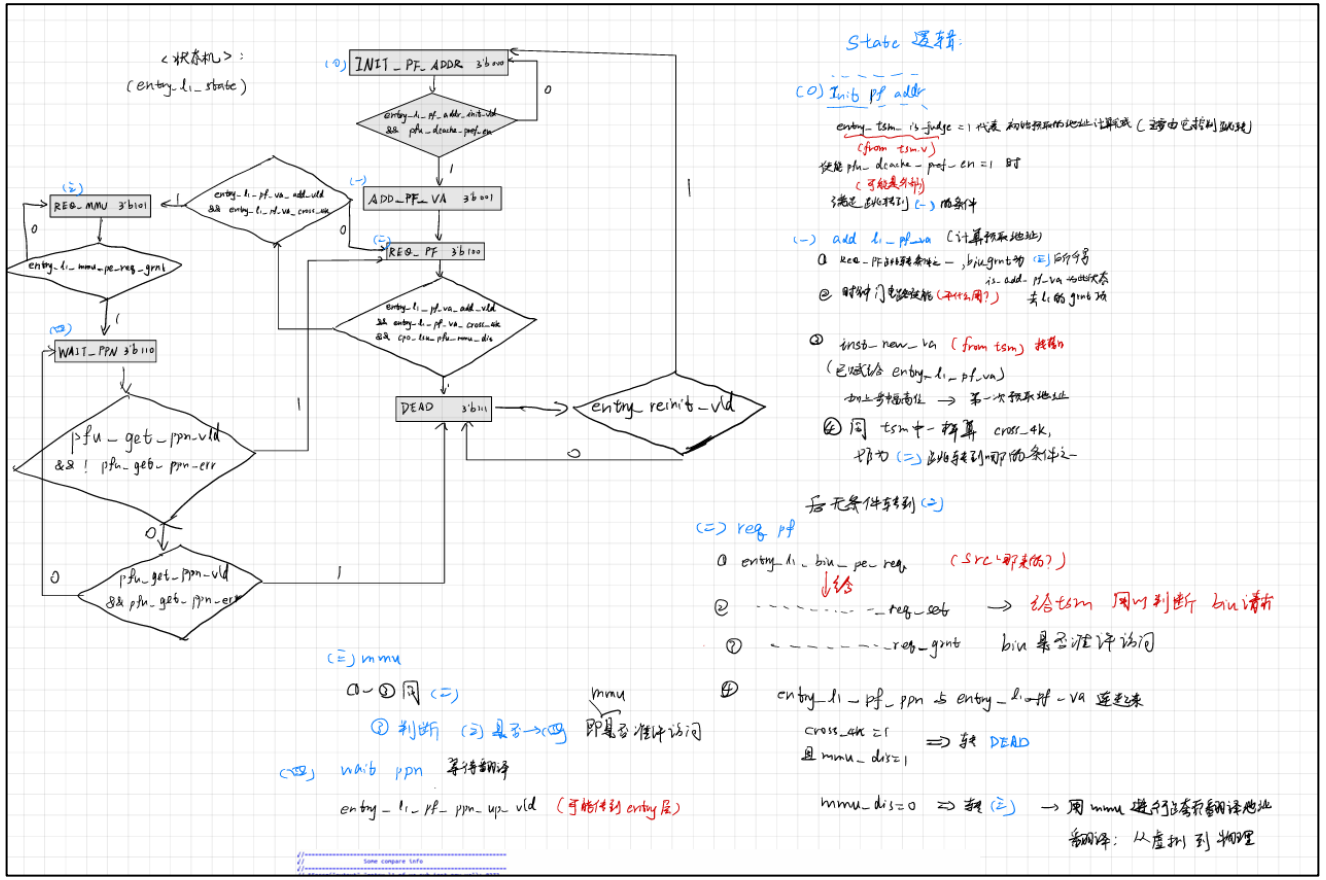
Delete main/scal
Janleron on 2024

Create pfb
Janleron on 2024

Update and rena
Janleron on 2024

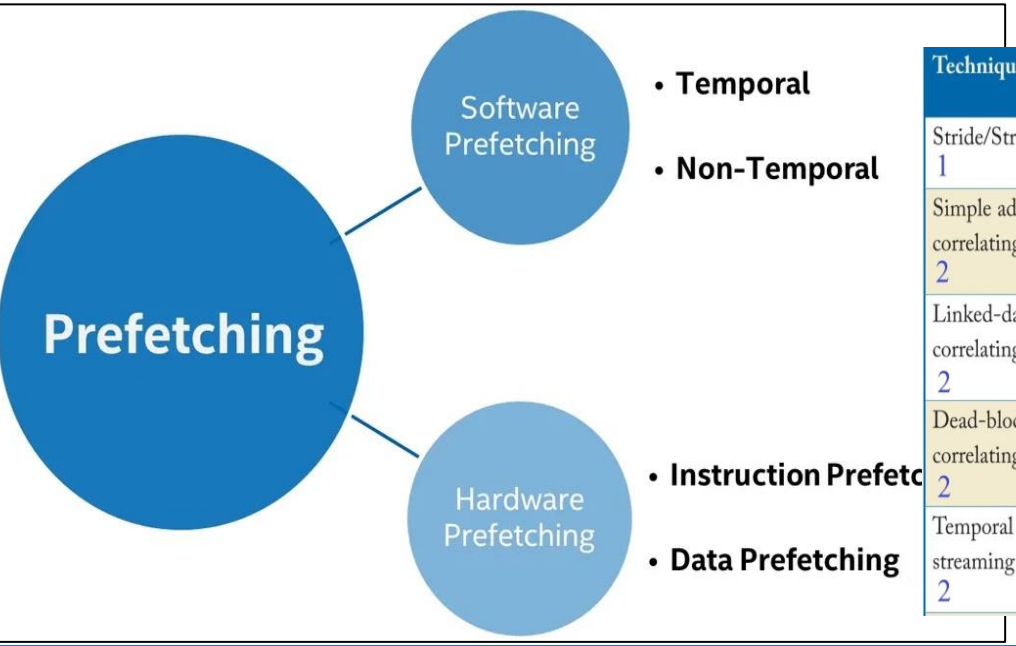
Create pfb
Janleron on 2024

Create ct_lsu_pfu
fzd on 2024/5/28



2 预取技术

- 预取方式：软件预取、硬件预取
  - 软件预取：实时、非实时
  - 硬件预取：指令预取、数据预取
- 预取技术：流式、地址相关、空间相关、操作相关



Technique	Line of Attack	Lookahead	Accuracy	Cost/ Complexity
Stride/Stream 1	Predicts cache miss strides using PCs or addresses	A few cache blocks	High for strides	< 1 KB
Simple address correlating 2	Correlates one address to one or more distinct cache miss addresses	A single cache block	> 30%	MBs
Linked-data correlating 2	Custom FSM that uses load-to-use dependencies in loops to fetch pointers	A few cache blocks	High for specific data structures	< 1 KB
Dead-block correlating 2	Correlates up to two addresses and control flow to a subsequent address	Time from death event to replacement	> 50%	MBs
Temporal streaming 2	Correlates one or more addresses to a stream of cache misses	Many cache blocks	> 50%	Off chip
Chained streaming 2	Chaining multiple address streams together using control flow or hierarchical address prediction	Many cache blocks	> 50%	Off chip
Irregular stream buffer 2	Adds a layer of address indirection to convert temporal to spatial correlation	Many cache blocks	> 50%	< 32 KB
Delta correlation 3	Records and correlates sequences of deltas between miss addresses	A few cache blocks	> 30%	< 32 KB
Spatial streaming 3	Correlates a PC with an arbitrary sequence of cache miss address deltas	A few cache blocks	> 50%	< 64 KB
Execution based 4	Helper threads or support for speculative execution	Limited by execution bandwidth	varies	--

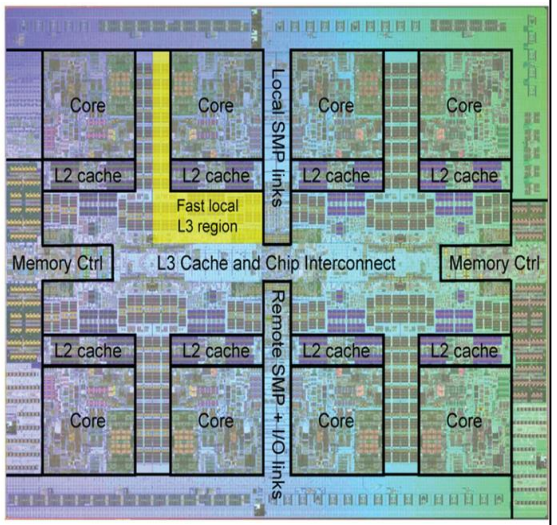
2 预取技术

- 预取方式：软件预取、硬件预取
  - 软件预取：实时、非实时
  - 硬件预取：指令预取、数据预取
- 预取技术：流式、地址相关、空间相关、操作相关
- 先进应用：可编程预取（POWER7）、自适应硬件预取（Xeon Phi）……

POWER7 prefetcher

Programmable and allow user to set different parameter including

- ***prefetch depth:*** how many lines in advance to prefetch
- ***prefetch on stores:*** whether to prefetch store operations
- ***stride-N:*** whether to prefetch streams with a stride larger than one cache block



3 Prefetching on Intel Xeon Phi[3]

3.2 Self-Adaptive hw prefetching

When complier is in -O2 or -O3 configuration

- Software prefetching on!
- Still flexible for programmer to interfere manually
- If sw prefetching works well (prefetch for the majority of the access, or reduce L2 demand misses), the hw prefetcher will not even get triggered very aggressively, i.e., hw prefetcher will throttle itself

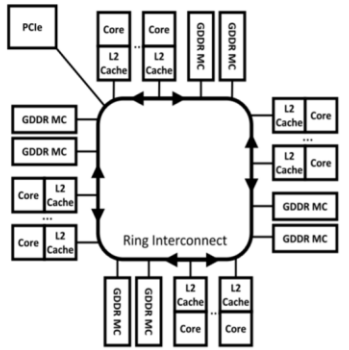
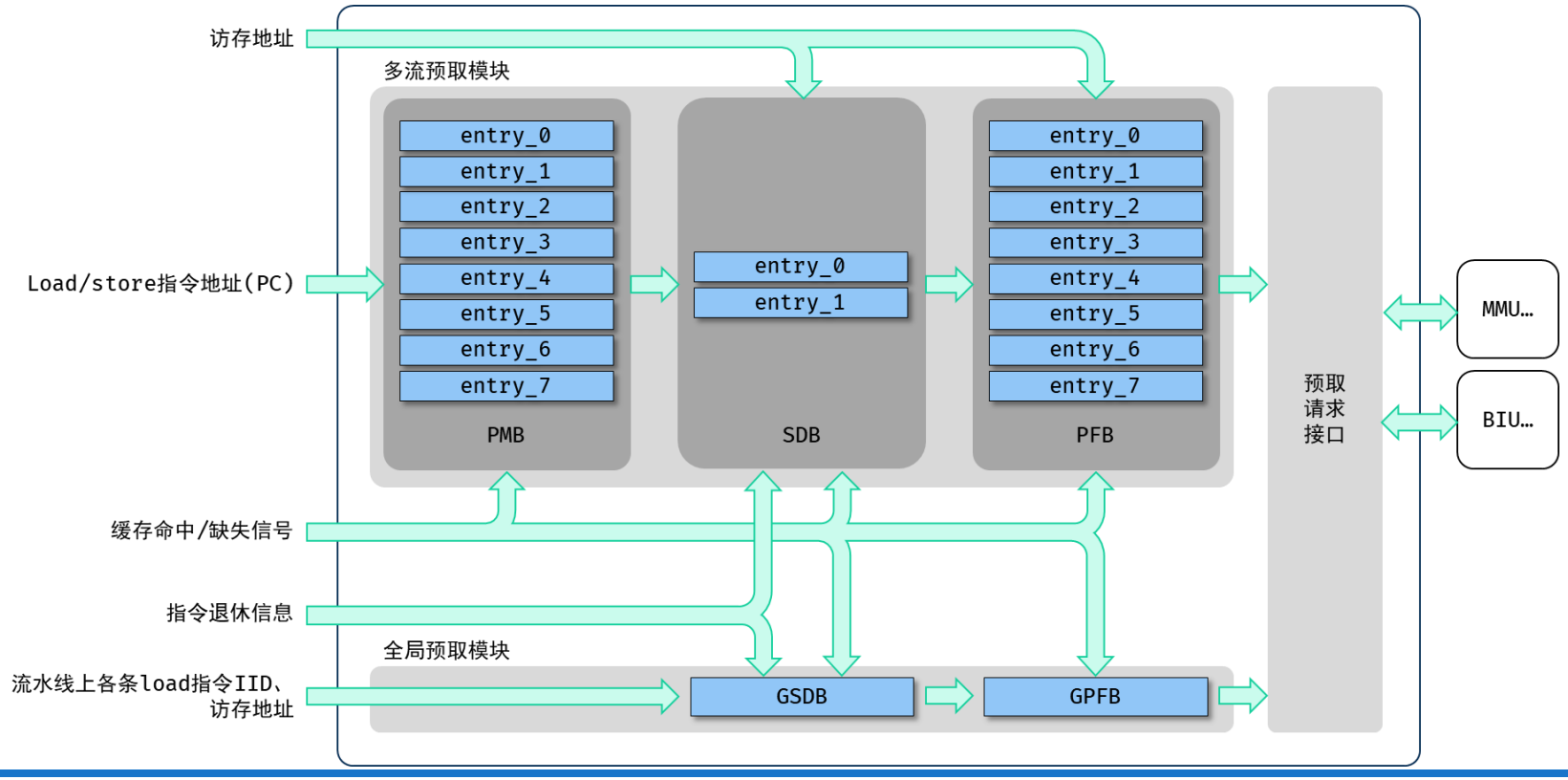


Fig. 1. Intel(c) Xeon Phi(c) coprocessor block diagram

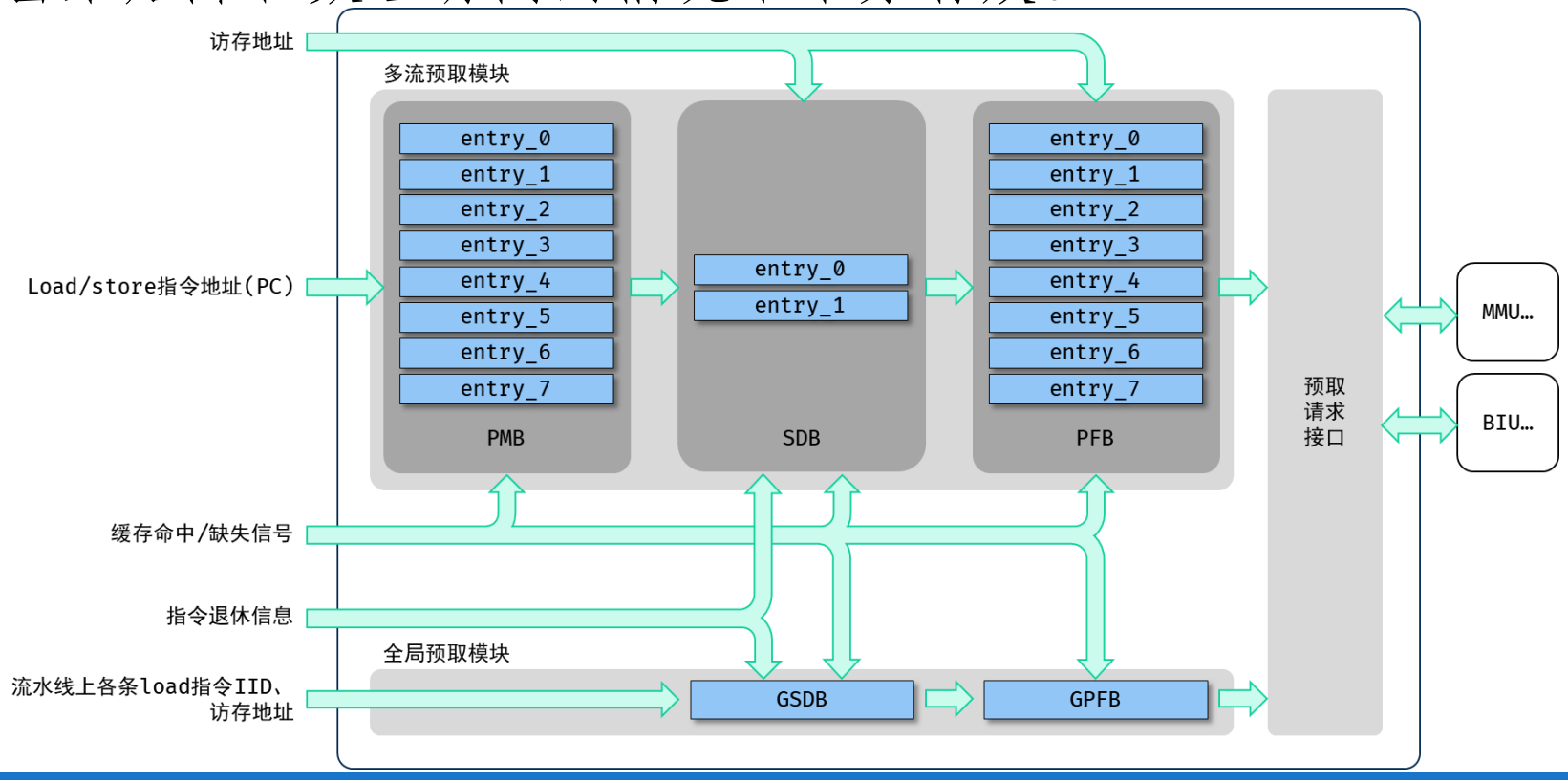
Metric	Configuration		
	HW only	SW only	HW+SW
Performance	2.70	2.63	2.77
Energy	0.37	0.45	0.42
Coverage	0.34	0.92	0.95
Efficiency	0.29	0.33	0.48

3 C910 PFU架构

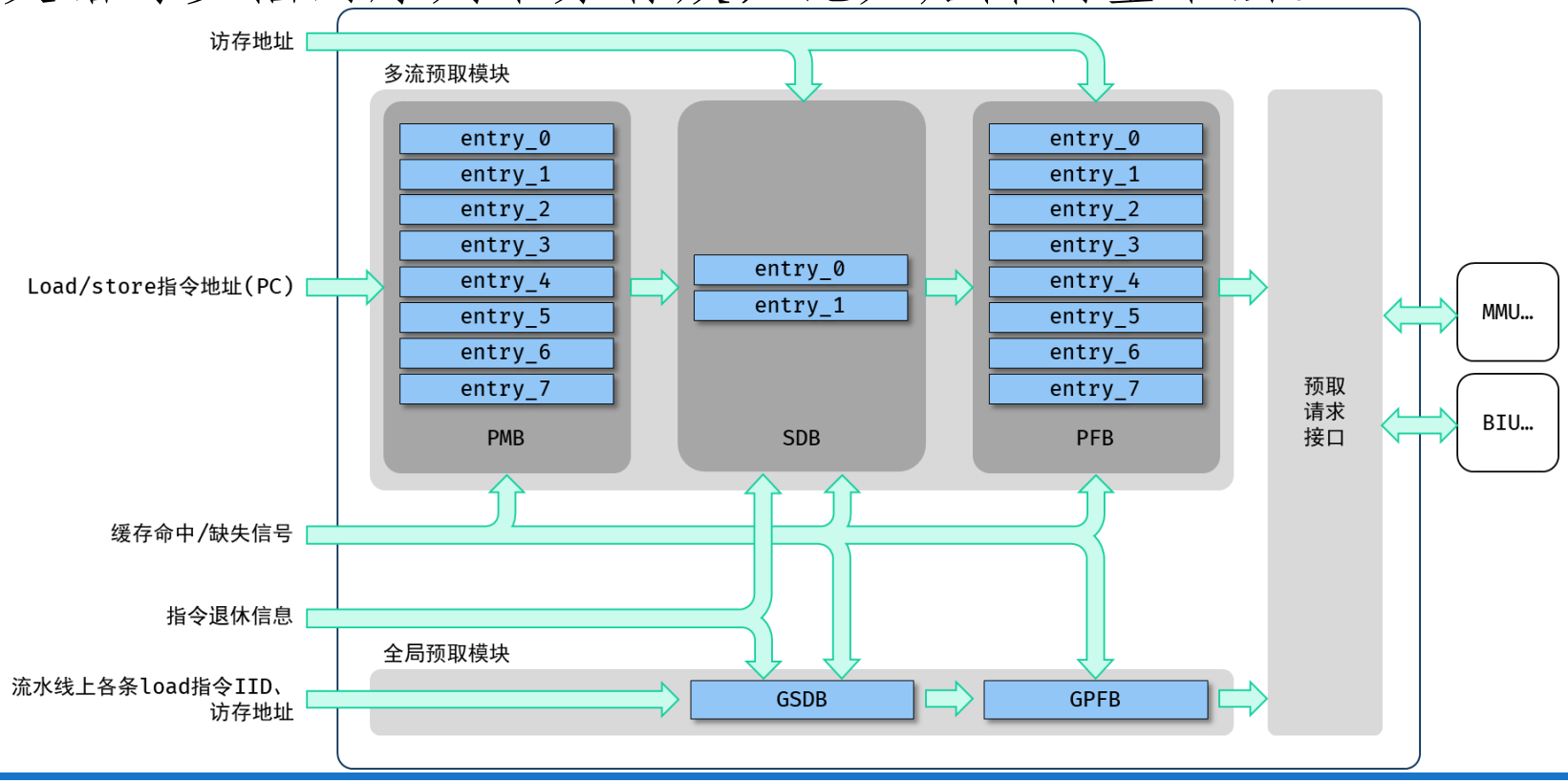
- RTL代码量：10个文件，6896行
  - 代码特点：由.vp文件生成，缩写多，持续赋值语句多，可读性较差
- 模式：全局预取模式+多流预取模式，并行+仲裁



- 全局预取模式
  - 对流水线上的访存指令做地址预测。
  - GSDB分析步长并维护置信度，GPFB计算预取地址。
  - 对于密集矩阵和数组访问的情况下十分有效。



- 多流预取模式
  - 对最近出现缓存缺失的若干条访存指令做地址预测。
  - PMB记录并识别待测指令，SDB分析步长并维护置信度，PFB计算预取地址。
  - 对于交错跨步幅的序列十分有效，比如矩阵向量乘法。



22307130211@m.fudan.edu.cn

4 Chisel化报告

赵龙征洋 2024/06/07

4 Chisel化报告


```
VCUNT_SIM: CoreMark 1.0 : 6.367926 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_in iis not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*************************************************
*    simulation finished successfully           *
*************************************************
$finish called from file "../logical/tb/tb.v", line 273.
$finish at simulation time          35040550000
           V C S   S i m u l a t i o n   R e p o r t
Time: 3504055000000 fs
CPU Time:    218.550 seconds;       Data structure size:  73.9Mb
```

服务器上源码跑分结果

4 Chisel化报告


```
VCUNT_SIM: CoreMark has been run 2 times, one times cost 156706 cycles !

VCUNT_SIM: CoreMark 1.0 : 6.381377 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_int is not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*********************************************
*    simulation finished successfully       *
*********************************************
```

pfu顶层

```
Notice: timing checks disabled with +notimingcheck at compile-time
Chronologic VCS simulator copyright 1991-2018
Contains Synopsys proprietary information.
Compiler version O-2018.09-SP2_Full64; Runtime version O-2018.09-SP2_Full64;  May 31 01:04 2024
        ********* Init Program *********
        ********* Wipe memory to 0 *********
        ********* Read program *********
        ********* Load program to memory *********
ERROR! Please define ee_ptr_int to a type that holds a pointer!

VCUNT_SIM: CoreMark has been run 2 times, one times cost 156706 cycles !

VCUNT_SIM: CoreMark 1.0 : 6.381377(iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_int is not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*********************************************
*    simulation finished successfully       *
*********************************************
```

gpfb

```
VCUNT_SIM: CoreMark 1.0 : 6.383087 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_int is not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*************************************************
*    simulation finished successfully           *
*************************************************
```

sdb_entry

- 跑分均为6.38，两个6.381377，一个6.383087。
- 目前还没有分析跑分比原版高的原因，应该要分析生成的Verilog和原来的Verilog，猜测应该是Chisel对原来的逻辑进行了优化，裁剪了冗余。

4 Chisel化报告


```
VCUNT_SIM: CoreMark 1.0 : 6.367926 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_in iis not a datatype that holds an int po
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*********************************************
*    simulation finished successfully       *
*********************************************
```

pfb

```
VCUNT_SIM: CoreMark has been run 2 times, one times cost 157037 cycles !

VCUNT_SIM: CoreMark 1.0 : 6.367926 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_in iis not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*********************************************
*    simulation finished successfully       *
*********************************************
```

pmb

```
VCUNT_SIM: CoreMark 1.0 : 6.367926 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_in iis not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*********************************************
*    simulation finished successfully       *
*********************************************
$finish called from file "../logical/tb/tb.v", line 273.
```

sdb_cmp

```
ERROR! Pease  define ee_ptr_int to a type that holds a pointer!

VCUNT_SIM: CoreMark has been run 2 times, one times cost 157037 cycles !

VCUNT_SIM: CoreMark 1.0 : 6.367926 (iterations/sec)/MHz
2K performance run parameters for coremark.
ERROR: ee_ptr_in iss not a datatype that holds an int pointer!
ERROR: Please modify the datatypes in core_portme.h!
CoreMark Size    : 666
Total ticks      : -1
Total time (secs): 42
Iterations/Sec   : 0
Iterations       : 2
Compiler version : GCC10.2.0
Compiler flags   : -O
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Errors detected
*********************************************
*    simulation finished successfully       *
*********************************************
```

gsdb

Coremark跑分均为6.367926，与源代码跑分一致

4 Chisel化报告

一些建议：

- 虚拟机的作用在于一些软件必须跑在Linux，或者在Linux比Windows系统更好。
- 对于IDEA也许可以出一个Windows部署Chisel项目的教程，我们小组最终所有人都是在Windows上面重新下载了一个IDEA并克隆了一个Chisel模板来做，给的虚拟机并没有发挥大的作用。
- 建议虚拟机可以打包好一个包含有VCS和Verdi的破解版的，大家也不用频繁麻烦助教上传文件了，可以很方便在自己本地debug，根据一位同学在自己虚拟机上面跑Coremark来看，个人电脑跑仿真10分钟左右一次Coremark。

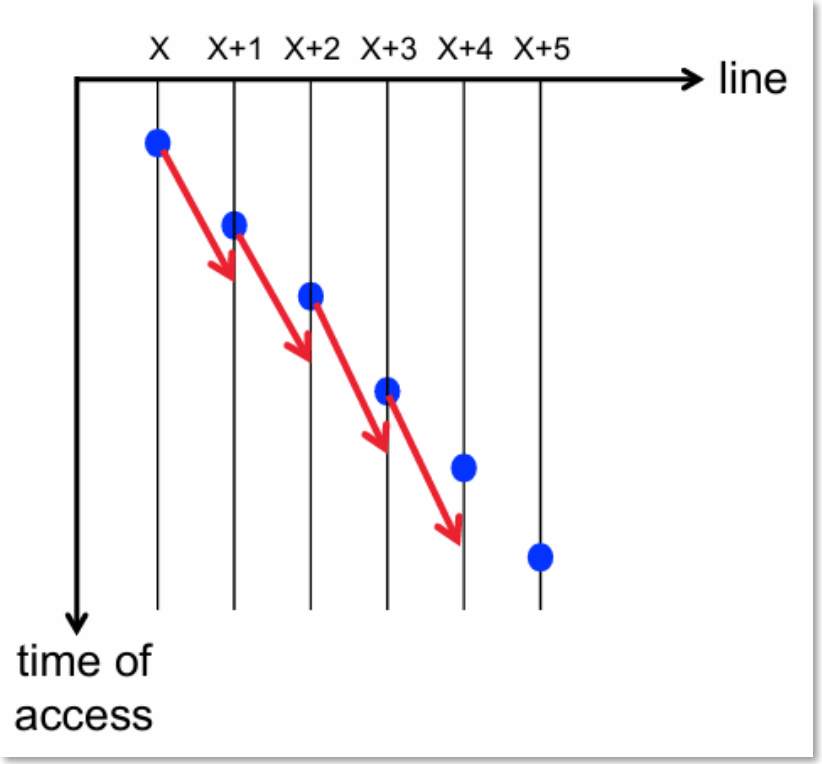
5 顶会论文阅读洞察

杜宇航 2024/06/07

Best-Offset Hardware Prefetching [1]

Offset-prefetching

When access x, prefetch x+D



Late prefetches time of access hurt performance

Best-Offset Hardware Prefetching [1]

Best-Offset-prefetching

- Try to identify the single best offset.
- New method for evaluating offsets.

  Recent Requests(RR) table

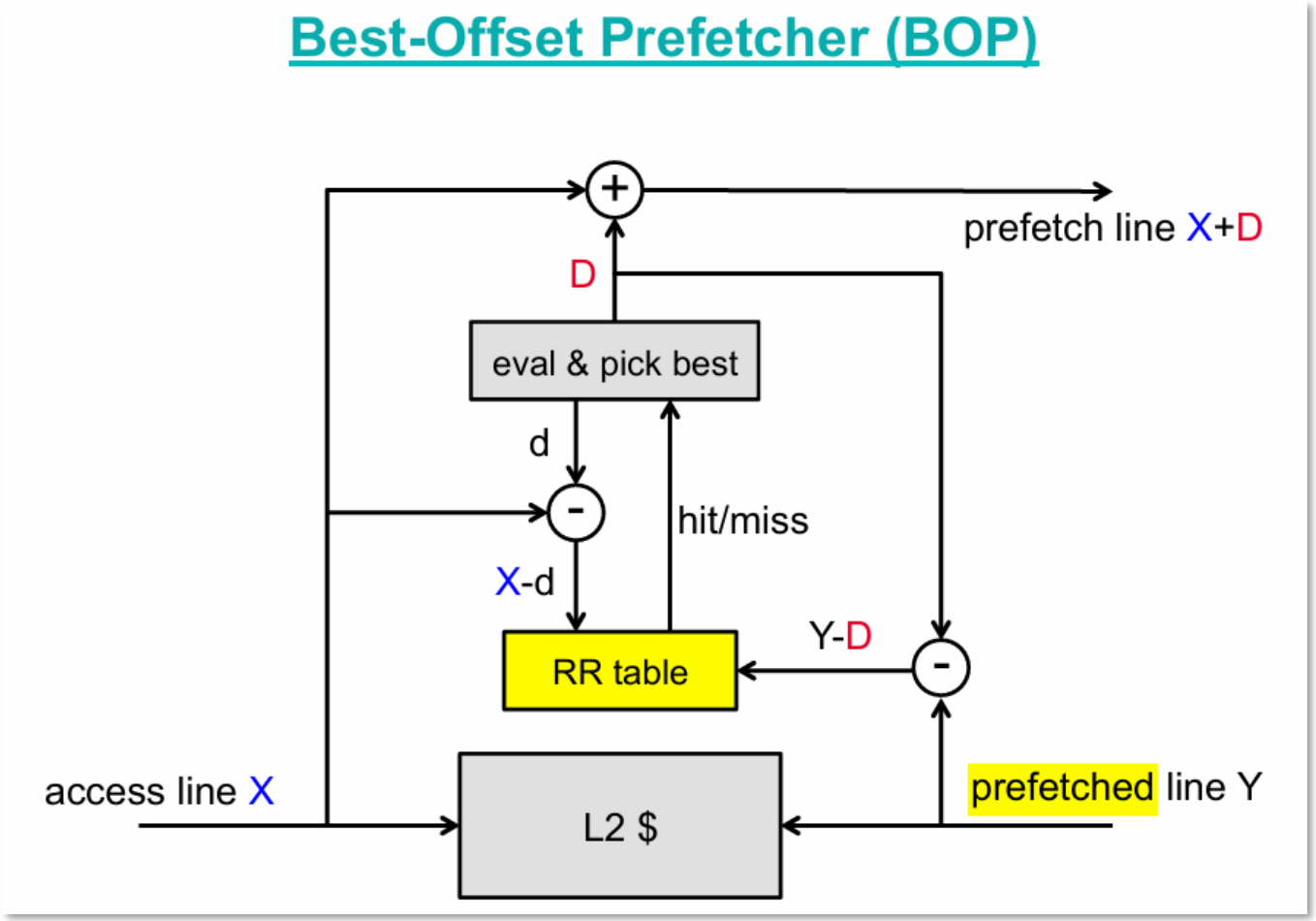
  evaluate offset d: if hit, increase score and pick the **highest one**.
- Take into account both coverage and timeliness.

Best-Offset Hardware Prefetching [1]

Best-Offset-prefetching

- Prefetch and evaluate.
- Shut the prefetch sometimes.
- When prefetch is off,

  offset evaluation continues.



Best-Offset Hardware Prefetching [1]

Hardware evaluation

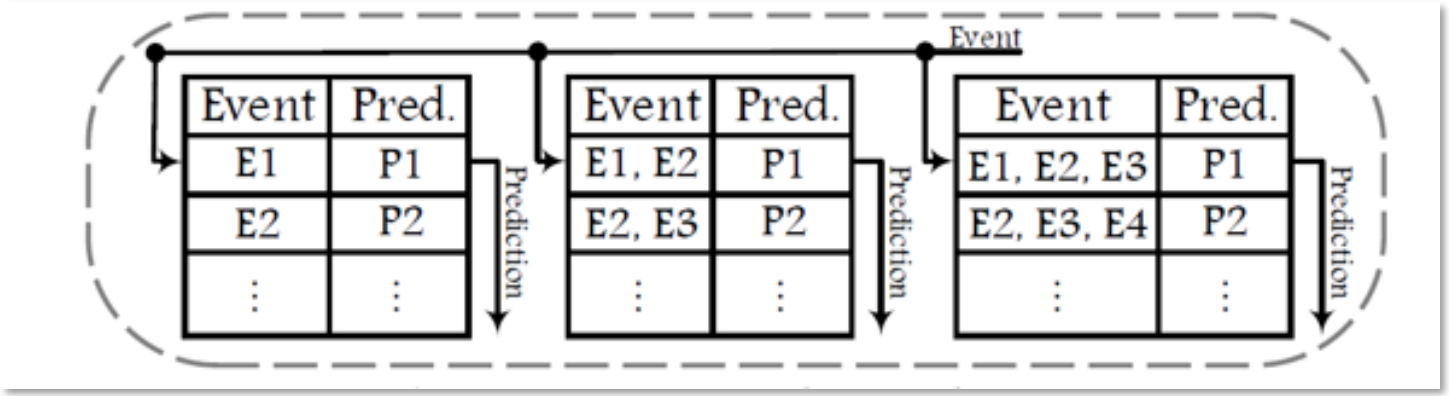
- One score per offset in the list: 52offsets 5-bit scores →260bits
- RR table: 256 entries, 12-bit tags → 3072 bits
- 3 adders: 64B line, 2MB page → 15-bit adders
- Misc. logic

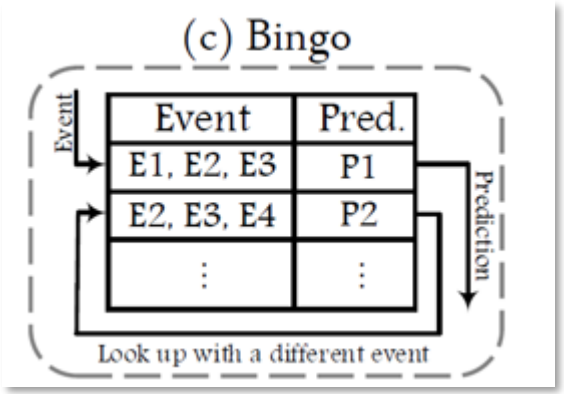
Main Weakness

- Tradeoff between prefetch coverage and timeliness

Bingo Spatial Data Prefetcher [2]

From TAGE to BINGO



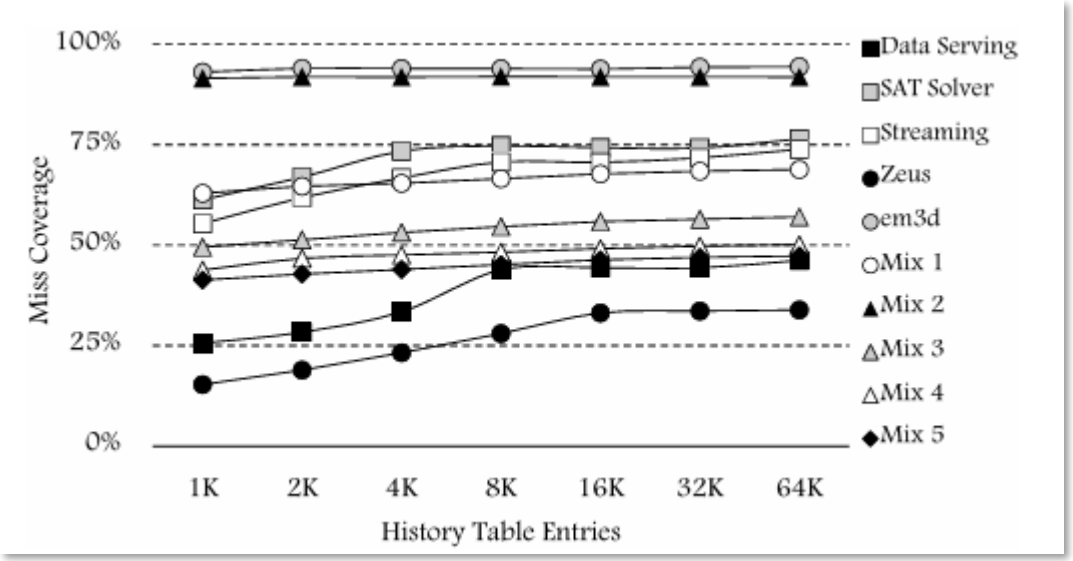


TAGE-like Prefetcher: use different table to predict

BINGO: single history table using different index

Bingo Spatial Data Prefetcher [2]

Hardware Evaluation



Using 16 K entries

Total storage requirement is 119KB

Accounting for 6% of the LLC

22307130436@m.fudan.edu.cn

Make prefetch a reinforcement learning problem

Problem: Lack inherent systems awareness;

Lack in-silicon customizability

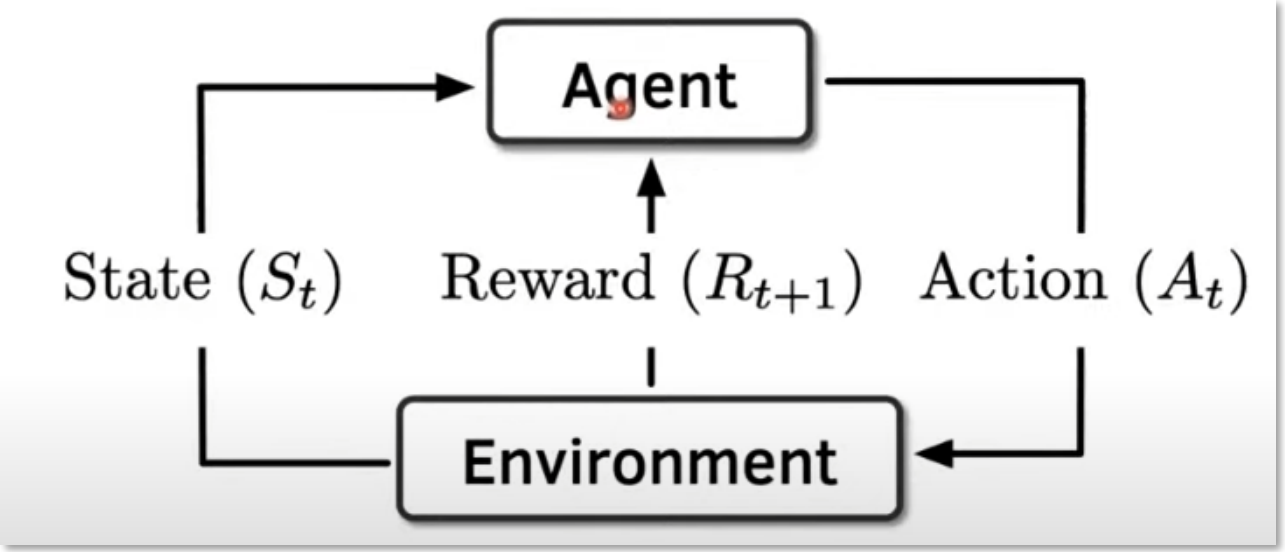
Pythia: takes adaptive prefetch decisions using multiple features

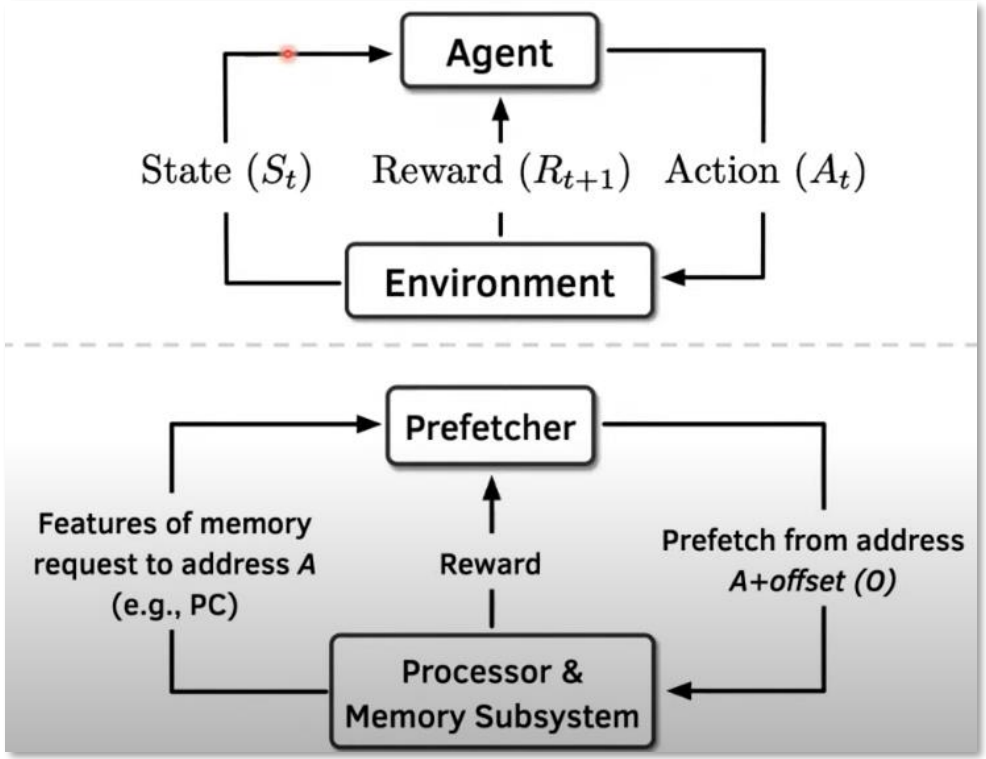
can be customized in silicon for target workloads

propose a practical implementation of RL in hardware

22307130436@m.fudan.edu.cn

Basics of RL and formulating prefetching as RL

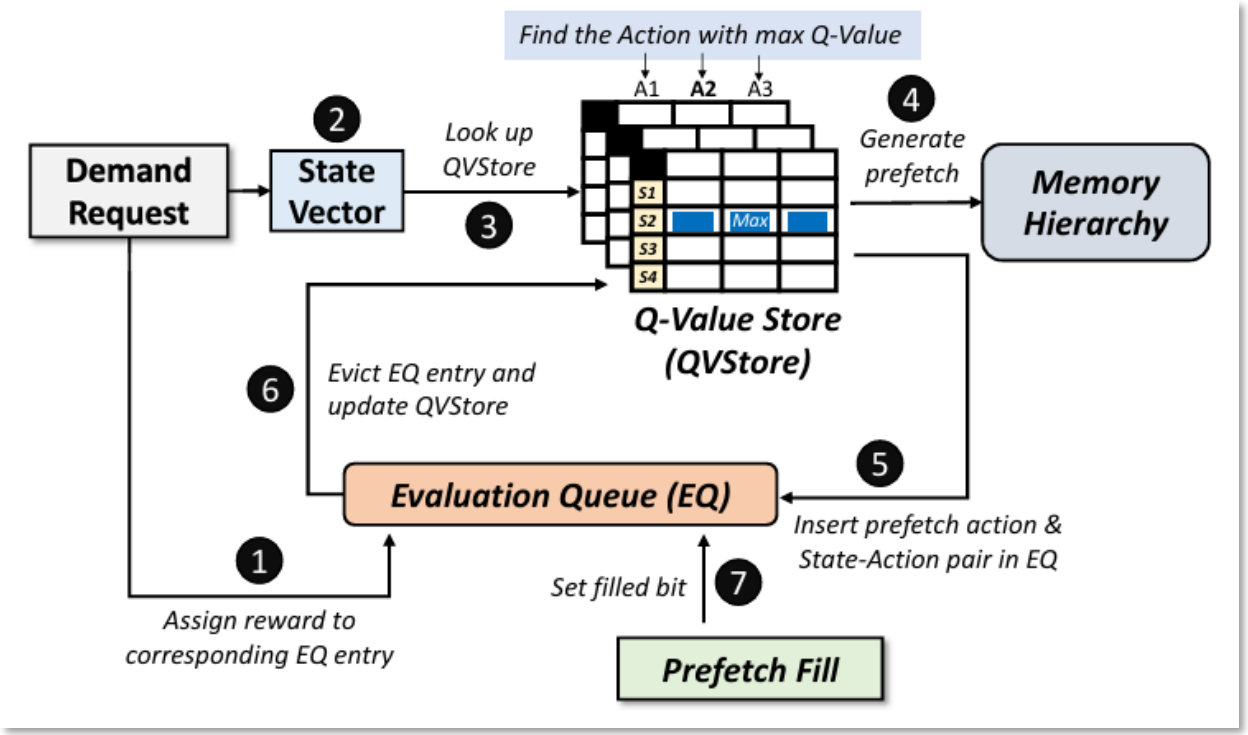




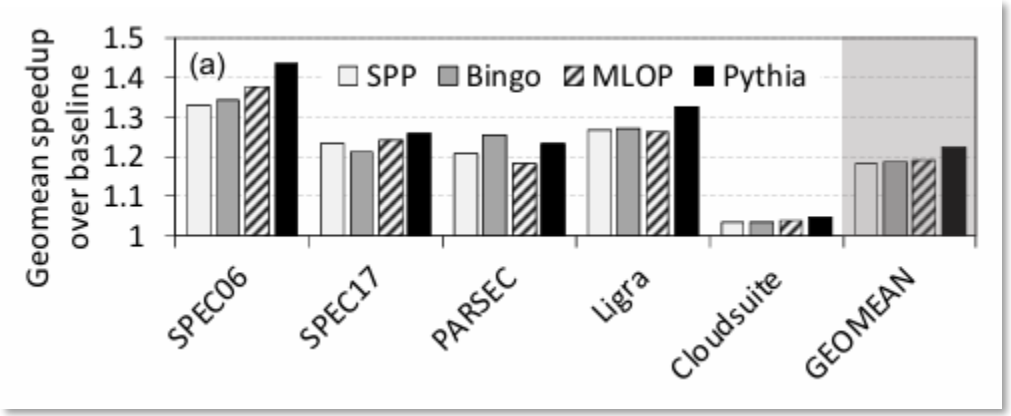
22307130436@m.fudan.edu.cn

复旦大学 微电子学院
SCHOOL OF MICROELECTRONICS
FUDAN UNIVERSITY

Hardware Evaluation and Performance



Structure	Description	Size
QVStore	• # vaults = 2 • # planes in each vault = 3 • # entries in each plane = feature dimension (128) × action dimension (16) • Entry size = Q-value width (16b)	24 KB
EQ	• # entries = 256 • Entry size = state (21b) + action index (5b) + reward (5b) + filled-bit (1b) + address (16b)	1.5 KB
Total		25.5 KB



22307130436@m.fudan.edu.cn

Reference

[1] P. Michaud, "Best-offset hardware prefetching," 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), Barcelona, Spain, 2016, pp. 469-480, doi: 10.1109/HPCA.2016.7446087.

[2] M. Bakhshalipour, M. Shakerinava, P. Lotfi-Kamran and H. Sarbazi-Azad, "Bingo Spatial Data Prefetcher," 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), Washington, DC, USA, 2019, pp. 399-411, doi: 10.1109/HPCA.2019.00053.

[3] Rahul Bera, Konstantinos Kanellopoulos, Anant Nori, Taha Shahroodi, Sreenivas Subramoney, and Onur Mutlu. 2021. Pythia: A Customizable Hardware Prefetching Framework Using Online Reinforcement Learning. In MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '21). Association for Computing Machinery, New York, NY, USA, 1121–1137. https://doi.org/10.1145/3466752.3480114

Thanks!

高考加油