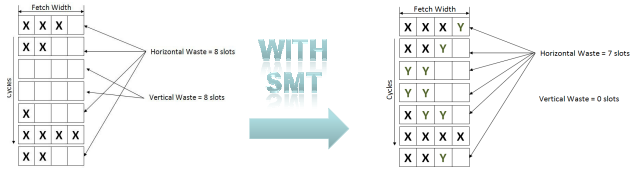
FLORIDA INTERNATIONAL UNIVERSITY

Hardware-aided Monitoring of L1 and L2 D-Cache Misses in SMT

Lichen Weng and Chen Liu
Electrical and Computer Engineering Department
{lichen.weng, chen.liu}@fiu.edu

Where are we?

Simultaneous Multithreading (SMT) architectures are defined as fully shared execution resources among several concurrently running threads in the same core [1].



Long-latency load is one of the major obstacle to better performance as the expression of *Memory Wall* in the SMT architectures[2]:

I. • A load misses in the Level 2 Data Cache

II. • It has to fetch data from lower memory architectures

III. • It still holds the shared resources, e.g., ReOrder Buffer, for hundreds of cycles during such fetching

IV. • Resource efficiency is harmed because the shared resources are held without throughput

V. • Task Level Parallelism (TLP) is reduced because other threads cannot utilize such shared resources

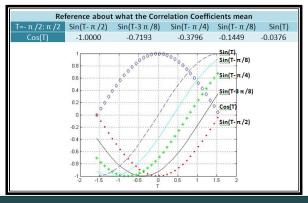
VI. • Therefore, system performance is decreased

Fetch policy, which assigns the priority in fetch stage is used to manage the shared resources and handle long-latency load issue.

	STALL[2]	DG[3]	DWarn[4]
Action Timing	L2 D-Cache Miss	L1 D-Cache Miss	L1 D-Cache Miss
Action	Suspend the thread	Suspend the thread	Reduce the thread priority

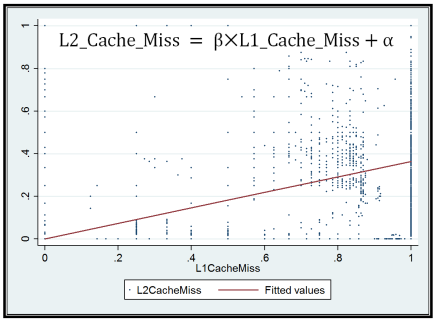
The relationship between L1 and L2 cache misses is more complicated than it is assumed.

gzip	L1 cache miss rate	L2 cache miss rate
L1 cache miss rate	1.0000	-0.1792
L2 cache miss rate	-0.1792	1.0000



What do we propose?

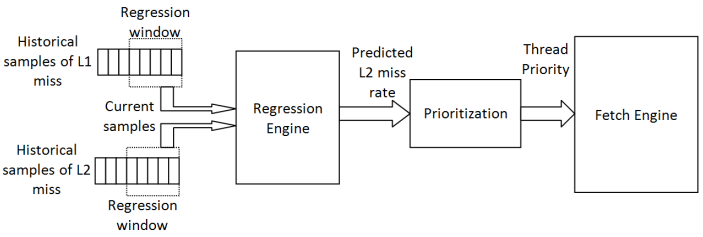
During an interval, the Ordinary Least Square (OLS) regression can be employed to describe the relationship, considering knowledge about L2 miss in advance will benefit the system.



The β = 0.365134 and α=0.0003677 are from the OLS regression for the benchmark *apsi*.

The linearity between L1 and L2 cache miss is statistically modeled.

The OLS regression can be implemented for every thread and then conduct instruction fetching (single thread to illustrate in the figure).



Regression

Two-level cache misses are sampled in Sampling Period, i.e., certain CPU cycles

Certain samples (Window Size) are utilized for OLS regression

Prioritization

The model evaluates future L2 cache miss based on immediate L1 cache miss rate for every thread

The priority descends as evaluated L2 miss rate grows

Fetching

Fetch from the thread with highest priority then the second, and so on so forth

What did we achieve?

Linearity confirmation

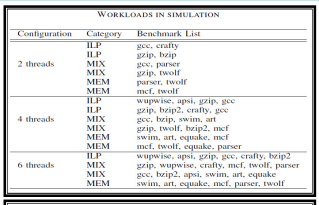
- F values are used to test the linearity between L1 and L2 cache miss rate for various benchmarks, which confirms its significance

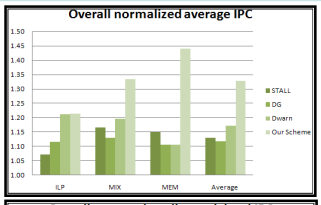
Performance improvement

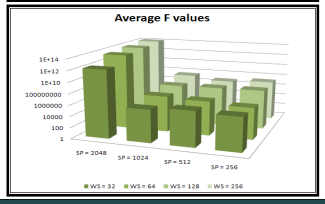
- It adaptively minimizes the influence of long-latency load, because it utilizes updated statistical model
- It achieves higher resource efficiency, because it reduces priority rather than gates threads

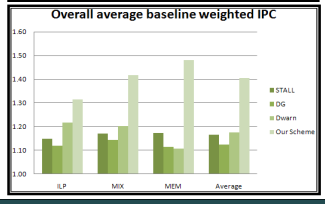
Sensitivity analysis

- Larger sampling period leads to better performance
- Larger L2 cache size means more throughput









Who did we reference?

[1] D.M. Tullsen, S.J. Eggers, J.S. Emer, H.M. Levy, J.L. Lo and R.L. Stamm, "Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor". *ISCA*, 1996.

[2] D.M. Tullsen and J.A. Brown, "Handling long-latency loads in a simultaneous multithreading processor". *ISCA*, 2001.

[3] A. El-Moursy and D.H. Albonesi, "Front-end policies for improved issue efficiency in SMT processors". *HPCA*, 2003.

[4] F.J. Cazorla, A. Ramirez, M. Valero and E. Fernandez, "DCache warn: an I-fetch policy to increase SMT efficiency". *IPDPS*, 2004.

[5] T.T. Soong, "Fundamentals of probability and statistics for engineers". *John Wiley and Sons, Ltd*, 2004