

A discussion on Performance analysis for SMT

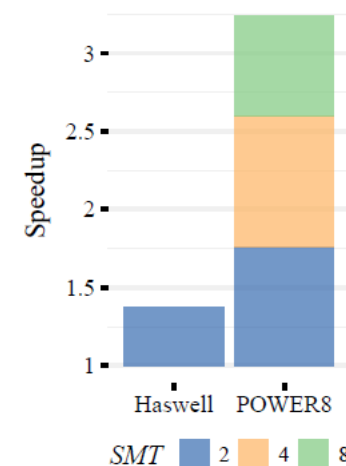
Moderators: Michael Chynoweth & Ahmad Yasin

Scalable Tools Workshop

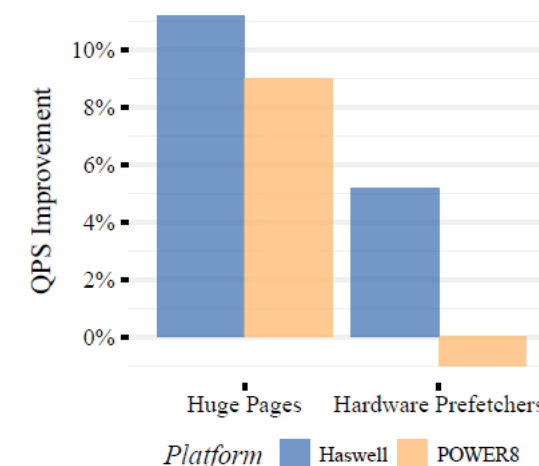
Solitude, Utah - July 12th, 2018

Motivation

	# threads per core
Intel Xeon - Skylake	2
AMD Zen	2
Intel Xeon Phi (KNL)	4
IBM Power 9	8
ARM, Intel Atom	1



(b) Simultaneous Multithreading (SMT)



(c) Huge Pages and Prefetching

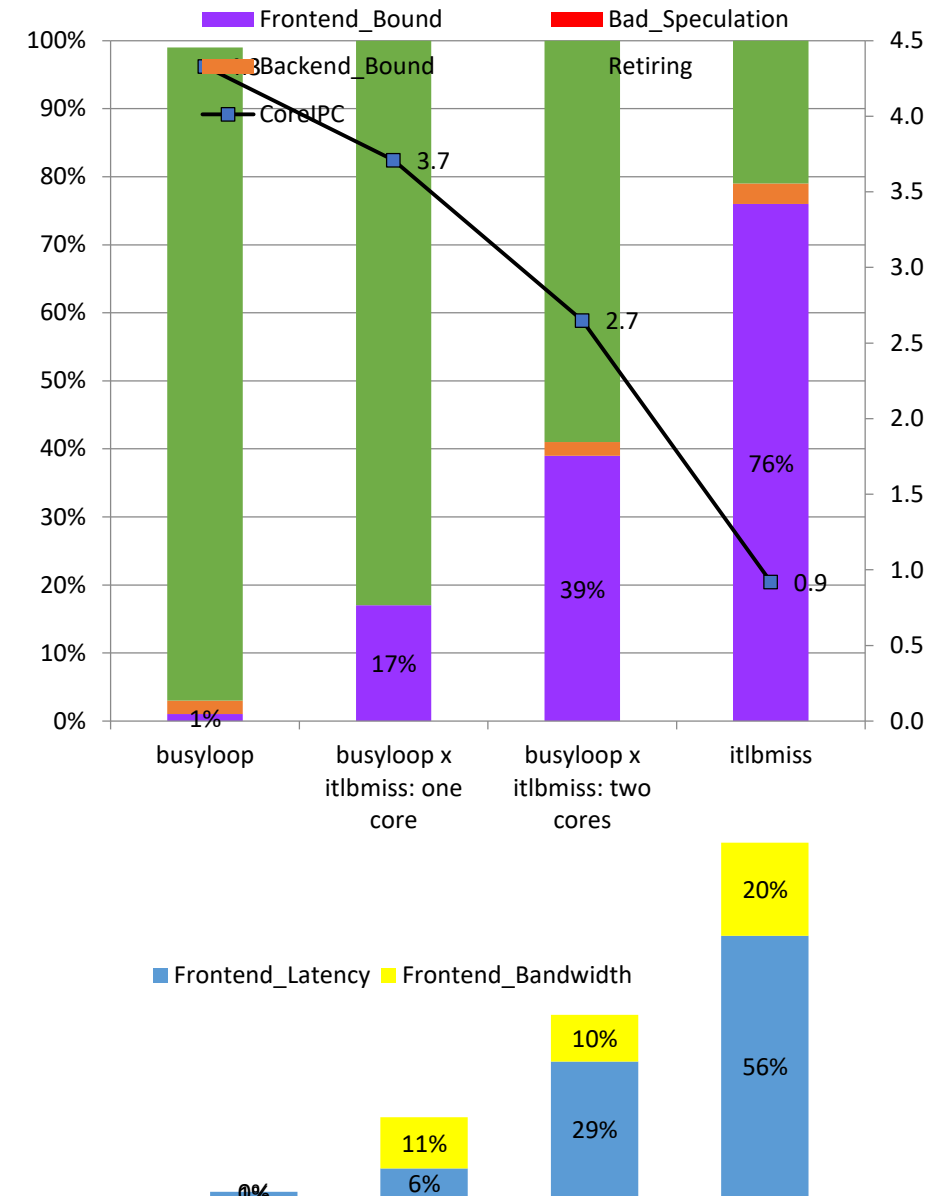
it, thread-level parallelism, large pages, and prefetching on search throughput.

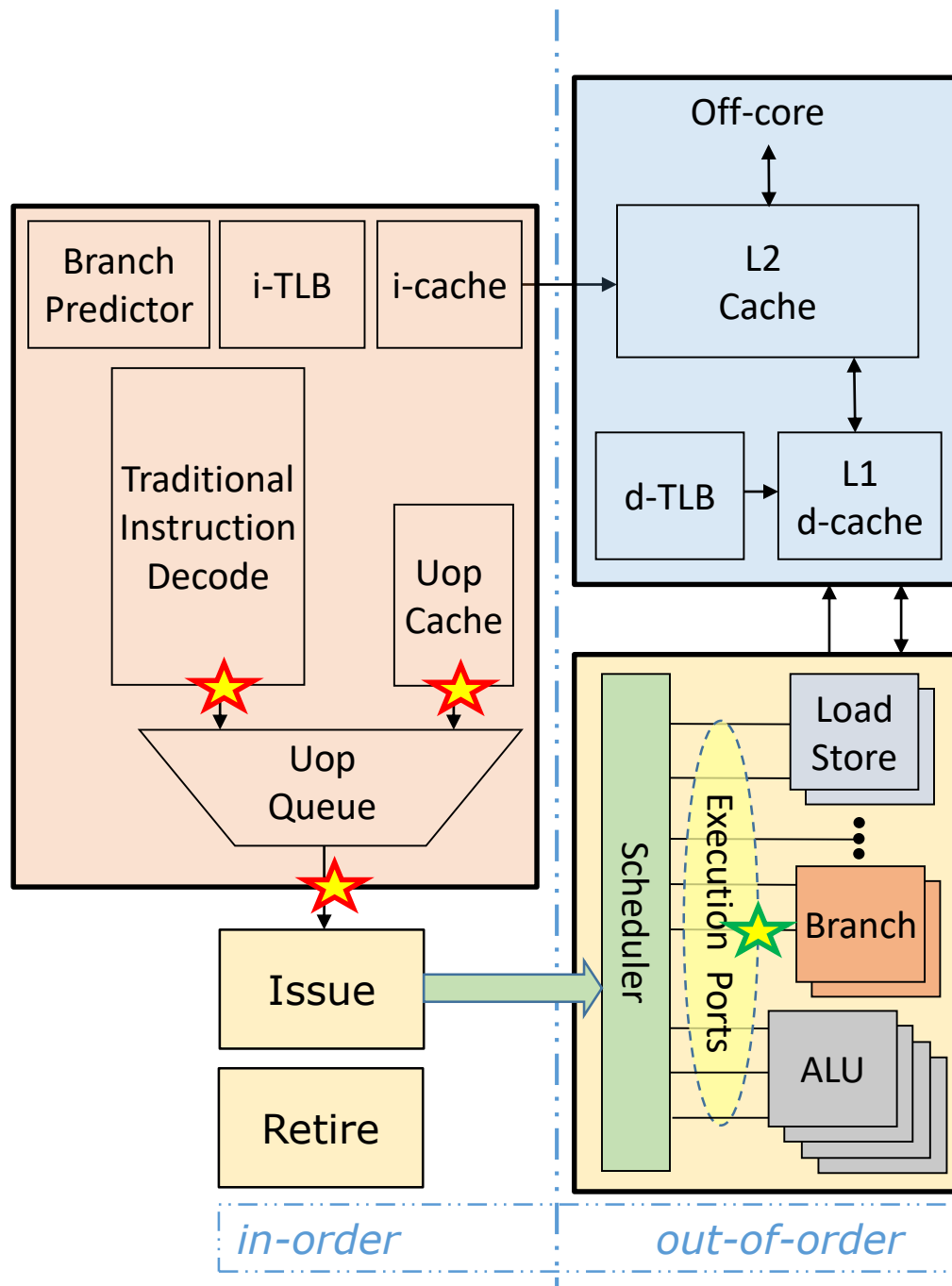
Figure source: Ayers, G., Ahn, J.H., Kozyrakis, C. and Ranganathan, P., 2018, February. Memory Hierarchy for Web Search. In High Performance Computer Architecture (HPCA), 2018 IEEE International Symposium on (pp. 643-656). IEEE.

SMT is commonplace for general-purpose high-performance

Background: Tasting SMT

- SMT: two threads sharing a physical core
- Hardware increases core's net efficiency
 - Example: iTLB miss stalls are turned into useful slots for high IPC code (busy-loop)
 - CoreIPC of 3.7 in one core vs 2.7 in two cores
 - See top chart - Measured on Broadwell.
- But.. complicates performance analysis: SMT interference
 - Scheduling iTLB-miss kernel induce Frontend (BW) stalls on busy-loop
 - These induced stalls do not exist when busy-loop is alone.
 - And thus cannot be detected by its own (bottom-up) miss events





SMT
thread-
arbitration
points

Frontend

Out Of Order

Memory Subsystem



SMT-off

SMT-on two threads


SMT-on single-thread



A new solution: SMT-aware events

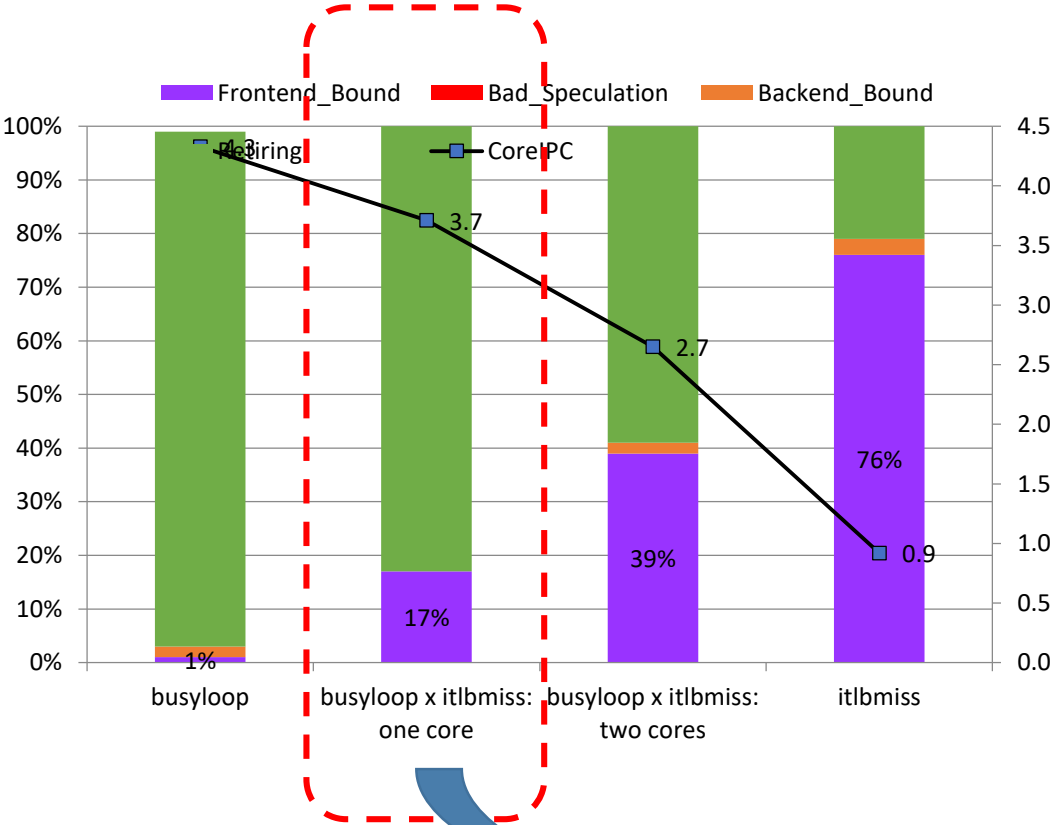
- Idea: distribute count among active threads in overlapping periods.
 - For events with threads contention
 - Aggregate on all threads gives a “core count”.
- Key advantages
 - ✓ Per-thread cycle accounting
 - ✓ Virtualization friendlier
 - ✓ Sampling mode
- Example events
 - Core Clockticks (see chart)
 - TOPDOWN.SLOTS
 - Total number of available slots for an unhalted logical processor.
 - TOPDOWN.BACKEND_BOUND_SLOTS
- Introduced in Icelake

clock	1	2	3	4	5	6	7	8	9	Sum
CPU_CLK_UNHALTED.THREAD: T0	1	1	1	1	1	1	1			7
CPU_CLK_UNHALTED.THREAD: T1				1	1	1	1	1	1	6
										13!

	CPU_CLK_UNHALTED.CORE: T0	1	1	1	1	0	1	0	-	-	5
	CPU_CLK_UNHALTED.CORE: T1	-	-	-	0	1	0	1	1	1	4
											9

New reality

TMA per-Core



TMA per-thread

SMT run + per-thread topdown		
	Thread0	Thread1
Frontend Bound	80%	28%
Bad Speculation'	0	0%
Retiring	20%	72%
Backend Bound	0%	0%
Threads share out of core slots	0.11	0.89

SMT Performance Analysis Summary

- SMT is (the) key challenge for performance analysis
 - An overly complex technical area
- SMT-aware events
 - Universities (William & Mary, Versailles), PNNL Lab and Google liked the approach
- Latency vs. Throughput (likely homogenous work)
 - How to account when HW prioritizes a non-stalled thread?
 - For HPC and “native datacenter” users Throughput matters.
- Security remains a concern (Stephane)
 - The vulnerability is reduced to the shared interval (improved over AnyThread)
- Misc. other areas of interest
 - Xu: Extended PEBS support non-precise events including SMT-aware new ones
 - Nathan: Bias in precise load retired samples; Load Latency, SMT-off with retired loads as backup
 - Michael and Emmanuel: detailed attributes of various pipelines behavior with SMT

SMT Performance Analysis

- SMT interference example
- SMT-aware events
 - SE and Nathan (Pacific Northwest Lab) and Emmanuel (U of Versailles) and Xu (U of) like it.
- Data-centric profiling paper by Xu HPDC?
 - -> want load latency with SMT enabled.
 - SW need to increase data-sharing among the two threads.
- Latency vs. Throughput (likely homogenous work)
 - For HPC users Throughput matters.
- SE: Security
 - AY: The vulnerability is reduced to the shared interval (instead of full interval with AnyThread when the spy thread is sleeping)
 - MC: Can be detected with self-monitoring thread too.
 - SGX has pretty robust protection of the PMU