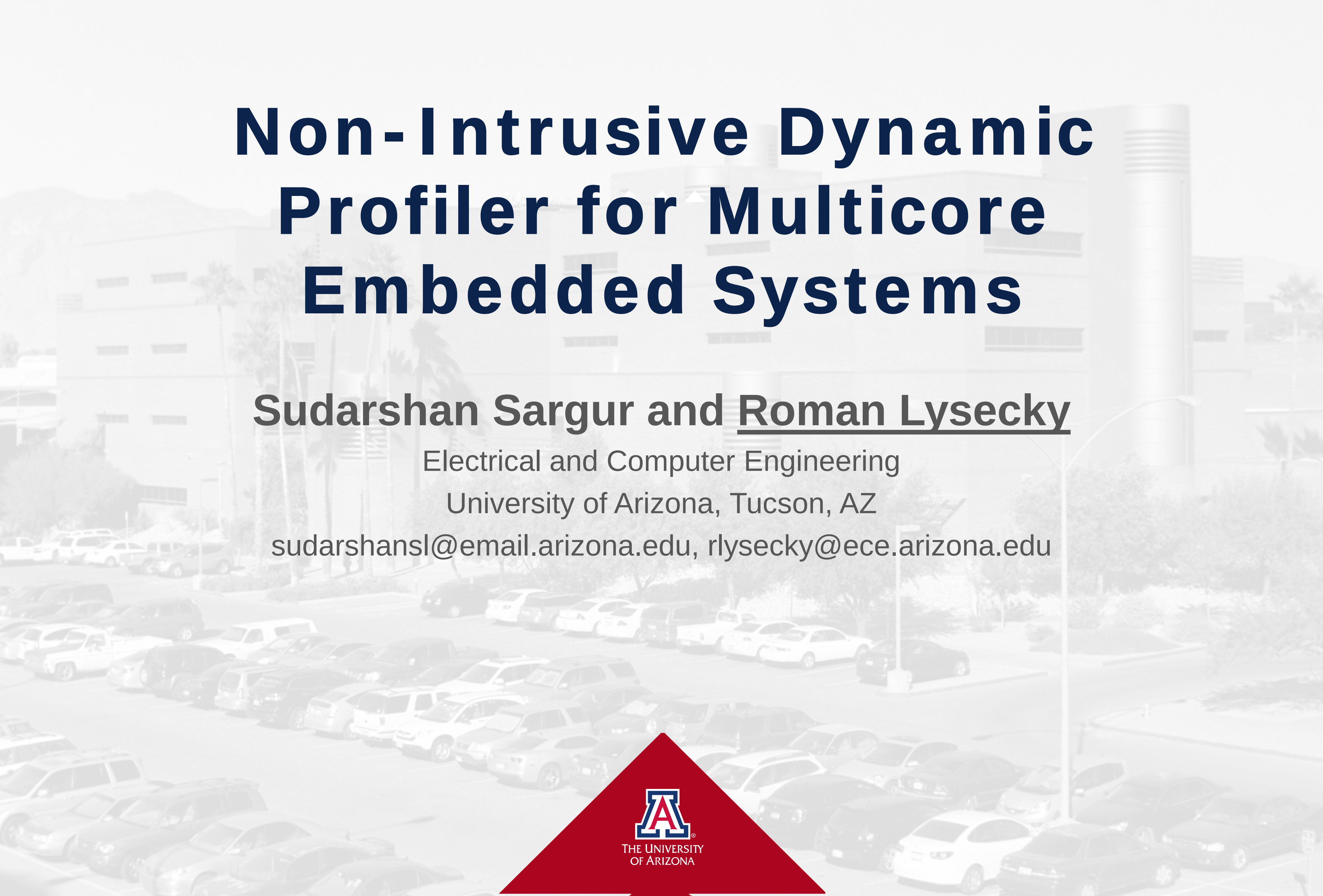
THE UNIVERSITY OF ARIZONA
College of Engineering



Non-Intrusive Dynamic Profiler for Multicore Embedded Systems

Sudarshan Sargur and Roman Lysecky

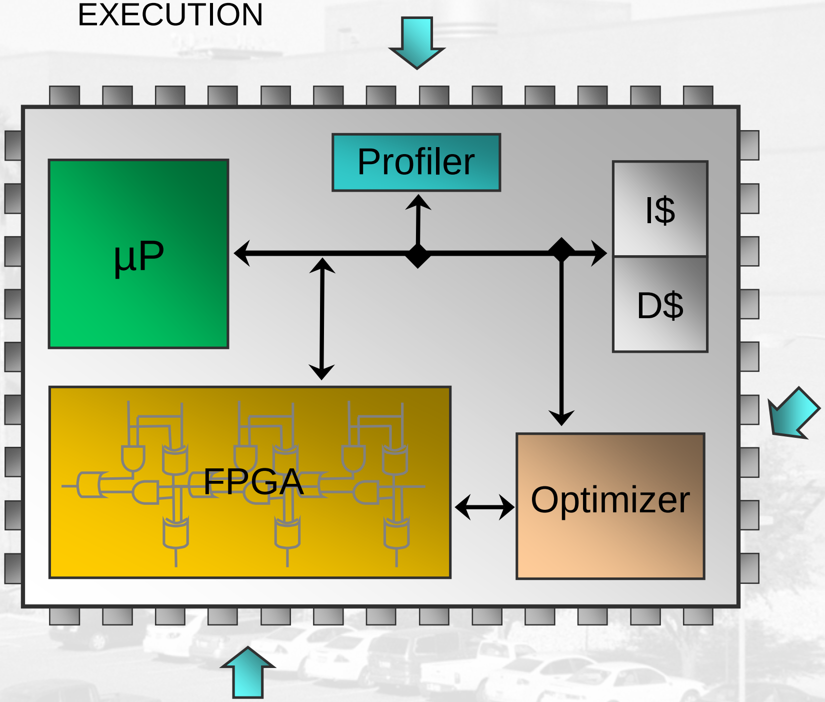
Electrical and Computer Engineering

University of Arizona, Tucson, AZ

sudarshansl@email.arizona.edu, rlysecky@ece.arizona.edu

THE UNIVERSITY
OF ARIZONA

The Past: Runtime-adaptive/Self-aware Systems



1 APPLICATION INITIALLY EXECUTES ON MICROPROCESSOR

2 PROFILER MONITORS APPLICATION EXECUTION

3 OPTIMIZER CHOOSES BETWEEN SOFTWARE AND HARDWARE ALTERNATIVES

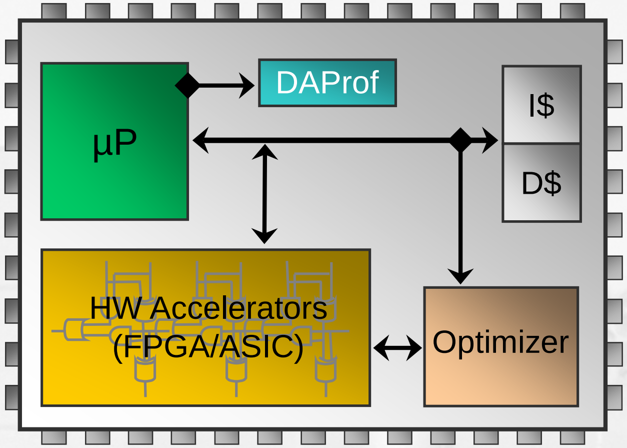
4 RECONFIGURE FPGA
UPDATE PROCESSOR & FPGA V/F

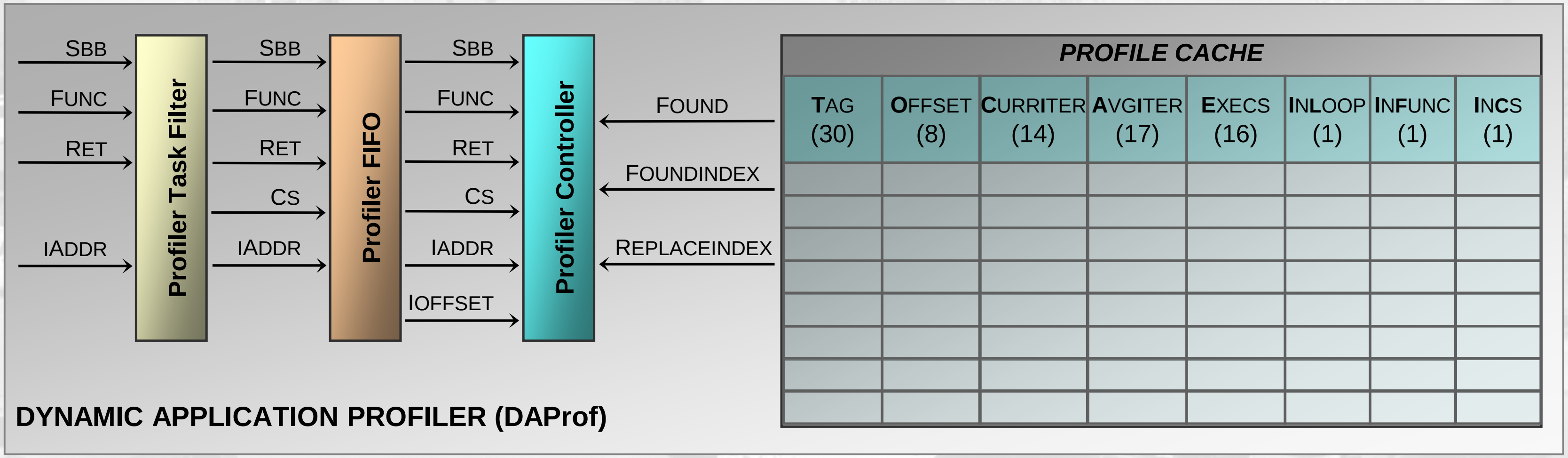
5 RINSE AND REPEAT

Needs accurate profile of application execution

The Past: Non-intrusive Profiling

- Dynamic Application Profiler (DAProf)
  - Provides loop/kernel-level profiling
  - Greater than 95% accuracy
  - **Interfaced to microprocessor trace port**
  - **Nonintrusive**
  - **5-10% area overhead**





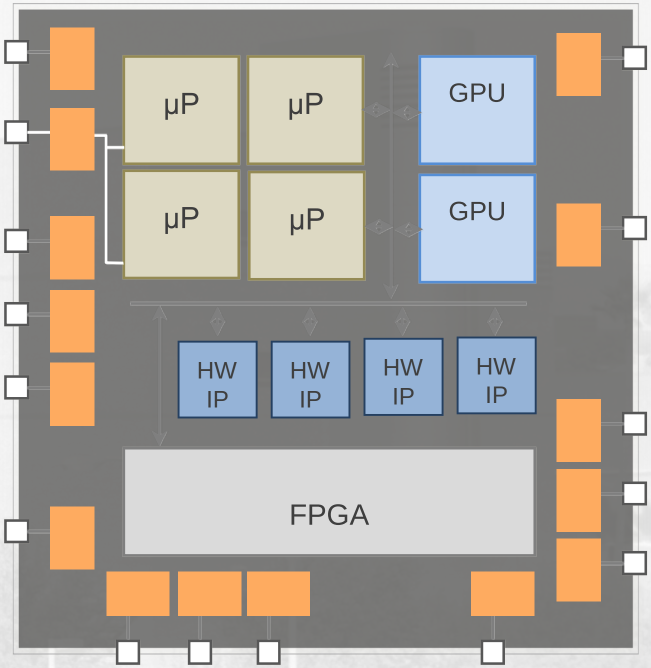
PROFILE CACHE

TAG (30)	OFFSET (8)	CURRITER (14)	AVGITER (17)	EXECS (16)	INLOOP (1)	INFUNC (1)	INCS (1)

DYNAMIC APPLICATION PROFILER (DAProf)

The Present: Runtime-adaptive/Self-aware Systems

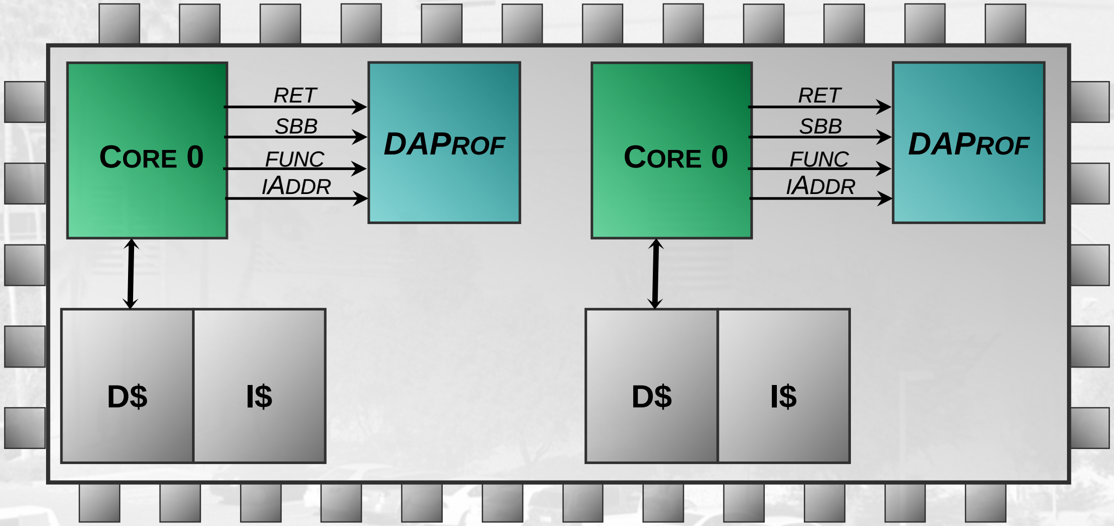
- **Multicore and many core systems**
  - Thermal and power management
  - Dynamic data-driven application systems
  - Thermal/aging aware dynamic task allocation and scheduling
  - Dependable systems
  - Malware detection
  - Energy optimization using heterogeneous/asymmetric processors
  - Reconfigurable computing
  - Runtime SOC tuning (i.e., tuning cache, memory, NOC, priority encoders, DAM, etc.)



Needs accurate profile of *system* execution

The Present: Profiling Inaccuracy

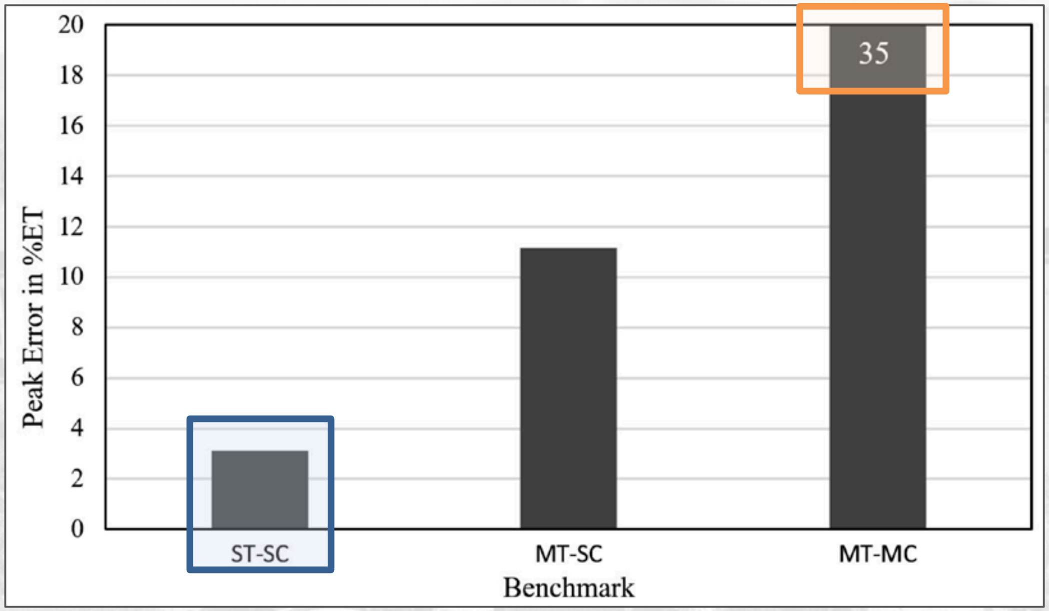
- DAProf profiler in multicore systems



Direct application of DAProf to individual processor cores can lead to inaccurate *system* profile

The Present: Profiling Inaccuracy

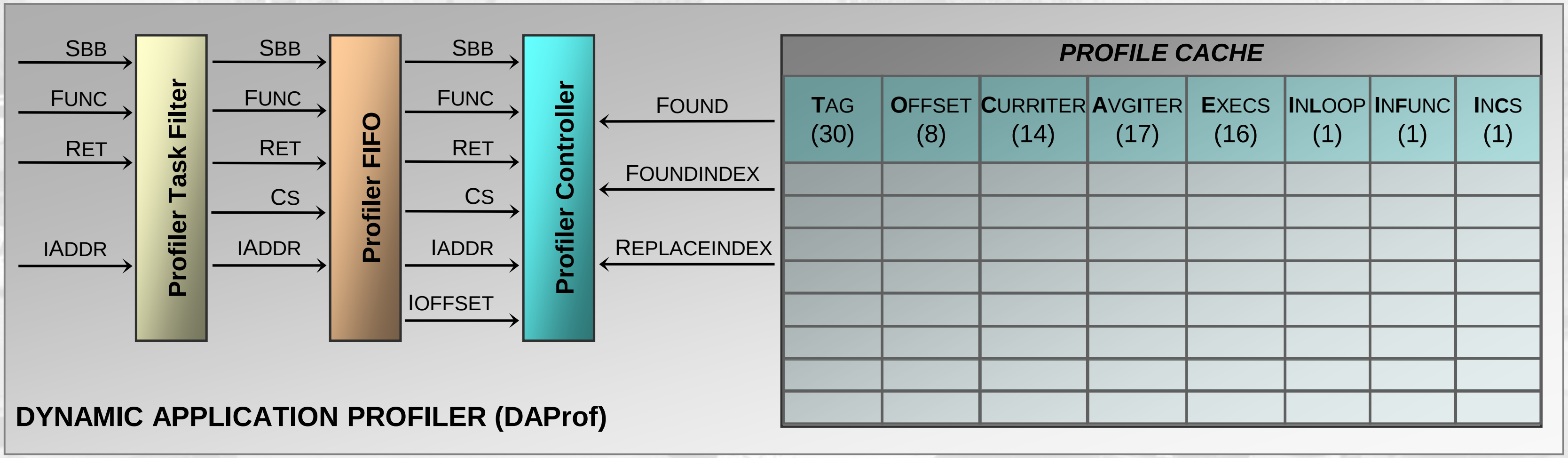
- **DAProf profiler in multicore systems**



  - 3% peak error for single-tasked application executing on single processor core (ST-SC)
  - 35% peak error for multi-tasked application executing on multicore processor (MT-MC)

Dynamic Application Profiler (DAProf)

- **Profile Task Filter**
  - Programmable component storing start and end address of each task (or region) to be profiled
  - Monitors trace port bus to detect context switches
    - Asserts CS signal if iAddr falls outside of the current task's address range



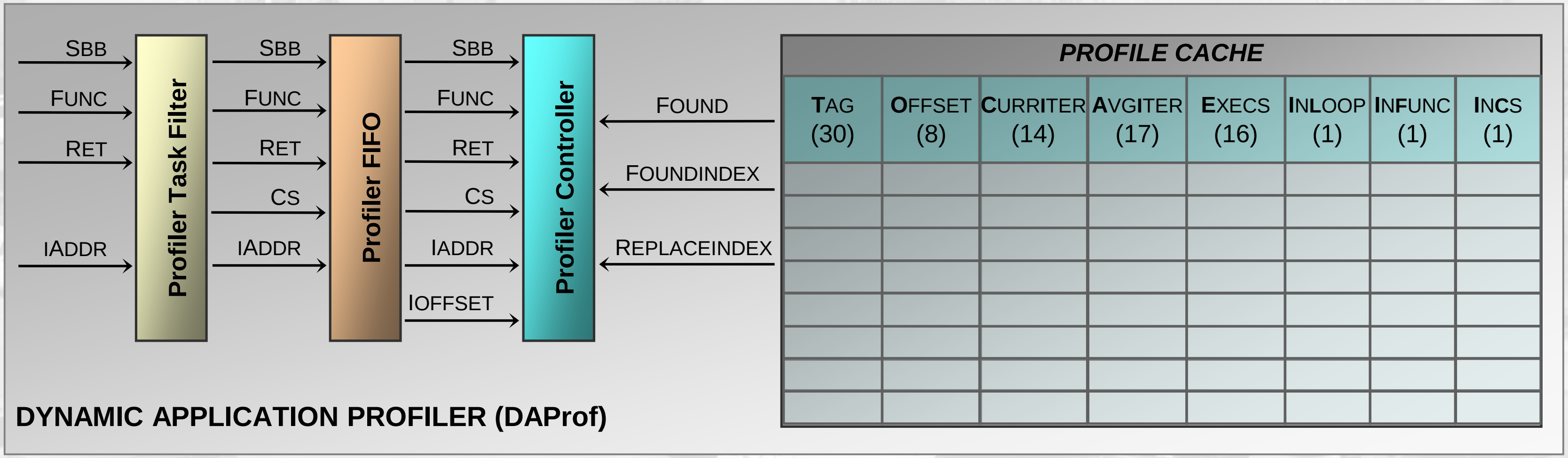
PROFILE CACHE

TAG (30)	OFFSET (8)	CURRITER (14)	AVGITER (17)	EXECS (16)	INLOOP (1)	INFUNC (1)	INCS (1)

DYNAMIC APPLICATION PROFILER (DAProf)

Dynamic Application Profiler (DAProf)

- **Profile Cache**
  - ***Tag***: Address of the short backwards branch
  - ***Offset***: Negative branch offset, corresponding to loop size
  - ***CurrIter***: Number of iterations for the current loop execution
  - ***AvgIter***: Average Iterations per execution of the loop
    - 17-bit fixed point representation with 14 bits integer and 3 bits fractional

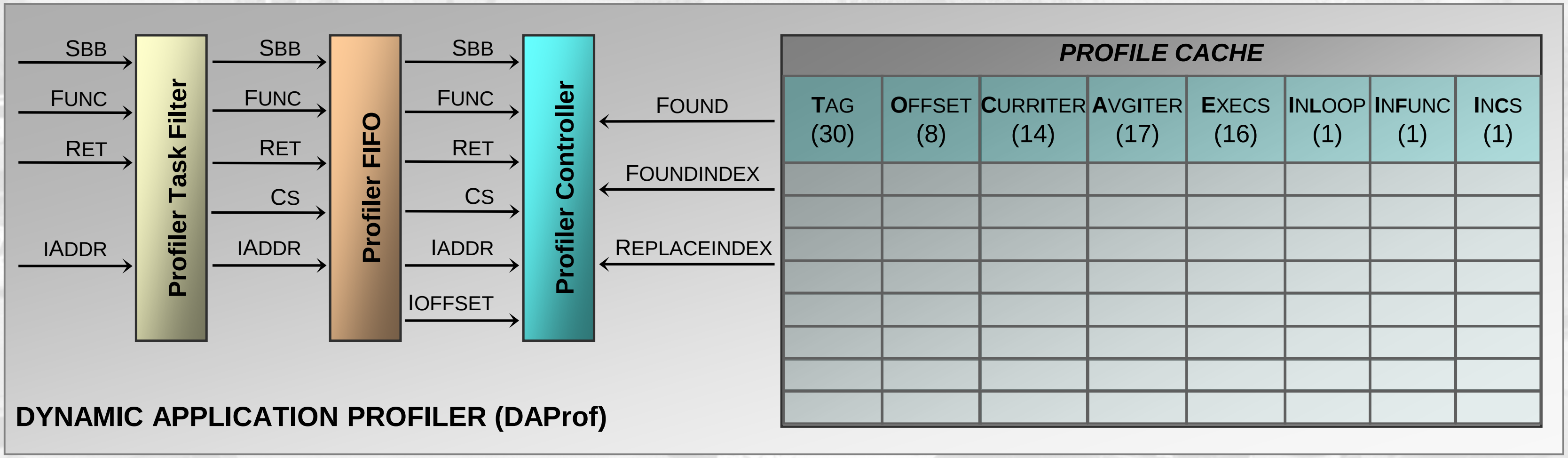


PROFILE CACHE

TAG (30)	OFFSET (8)	CURRITER (14)	AVGITER (17)	EXECS (16)	INLOOP (1)	INFUNC (1)	INCS (1)

Dynamic Application Profiler (DAProf)

- **Profile Cache**
  - ***Execs***: 16-entry storing the number of times a loop executes
  - Maintains relative execution count
    - If Execs counter saturates, profiler controller adjusts all loop execution counts by dividing by 2 (implemented as right shift)
    - **Applications may saturate at different times on different processor cores**



SBB
FUNC
RET
IADDR
Profiler Task Filter
SBB
FUNC
RET
CS
IADDR
Profiler FIFO
SBB
FUNC
RET
CS
IADDR
IOFFSET
Profiler Controller
FOUND
FOUNDINDEX
REPLACEINDEX

PROFILE CACHE

TAG (30)	OFFSET (8)	CURRITER (14)	AVGITER (17)	EXECS (16)	INLOOP (1)	INFUNC (1)	INCS (1)

DYNAMIC APPLICATION PROFILER (DAProf)

Dynamic Application Profiler (DAProf)

- **Profile Cache**
  - ***Execs***: 16-entry storing the number of times a loop executes
  - Maintains relative execution count
    - If Execs counter saturates, profiler controller adjusts all loop execution counts by dividing by 2 (implemented as right shift)
    - **Applications may saturate at different times on different processor cores**

Local Profile (Core 0)	
Loop ID	**Execs**
A	42500
B	22000
C	34000

Local Profile (Core 1)	
Loop ID	**Execs**
X	34196
Y	800
Z	10500

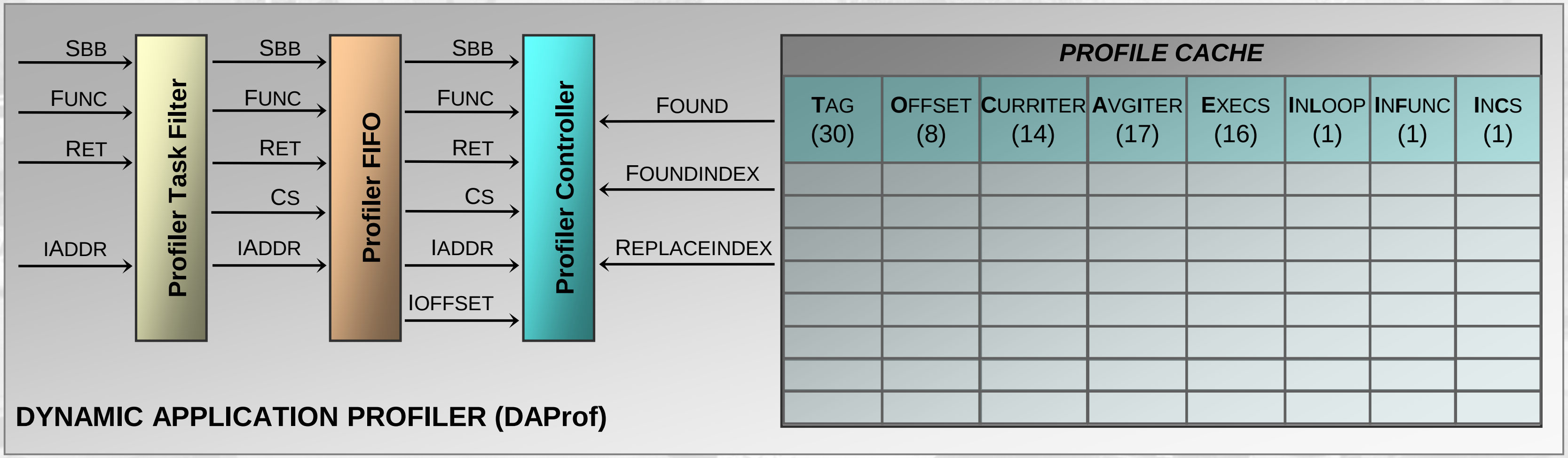
Profile Results from Individual Cores

Global Profile					
Loop ID	**Execs (Actual)**	**Execs (DAProf)**	**% ET (Actual)**	**%ET (DAProf)**	**Error**
A	42500	42500	17	29	12
B	22000	22000	9	15	6
C	34000	34000	14	24	10
X	132500	34196	53	24	29
Y	1100	800	0	1	1
Z	17500	10500	7	7	0

Naively Combined System Profile

Dynamic Application Profiler (DAProf)

- **Options for Improving Multicore Profiling using DAProf**
  - Larger DAProf entry for executions
    - Requires large area increase, and loops saturation are still not be synced across cores
  - Global loops saturations
    - Requires increase in both execution entry and number of loops to maintain profile accuracy
    - Introduce timing and layout challenges

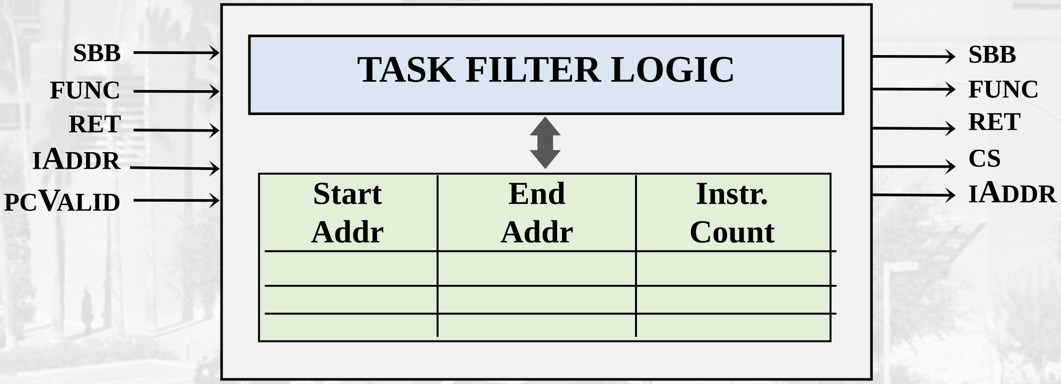


PROFILE CACHE

TAG (30)	OFFSET (8)	CURRITER (14)	AVGITER (17)	EXECS (16)	INLOOP (1)	INFUNC (1)	INCS (1)

DAProf with multitask and multicore scaling (DAProf-MT-MC)

- **DAProf with multitask and multicore scaling (DAProf-MT-MC)**
  - Extend profile task filter with *per task* instruction counts
    - ***pcValid***: Processor trace port indicating current PC is a valid instruction
    - ***Instruction Count***: 64-bit register storing per task count of instructions executed
    - Instructions counts used to scale the estimated percentage of execution (%ET) across tasks



SBB
FUNC
RET
IADDR
PCVALID

TASK FILTER LOGIC

Start Addr	End Addr	Instr. Count

SBB
FUNC
RET
CS
IADDR

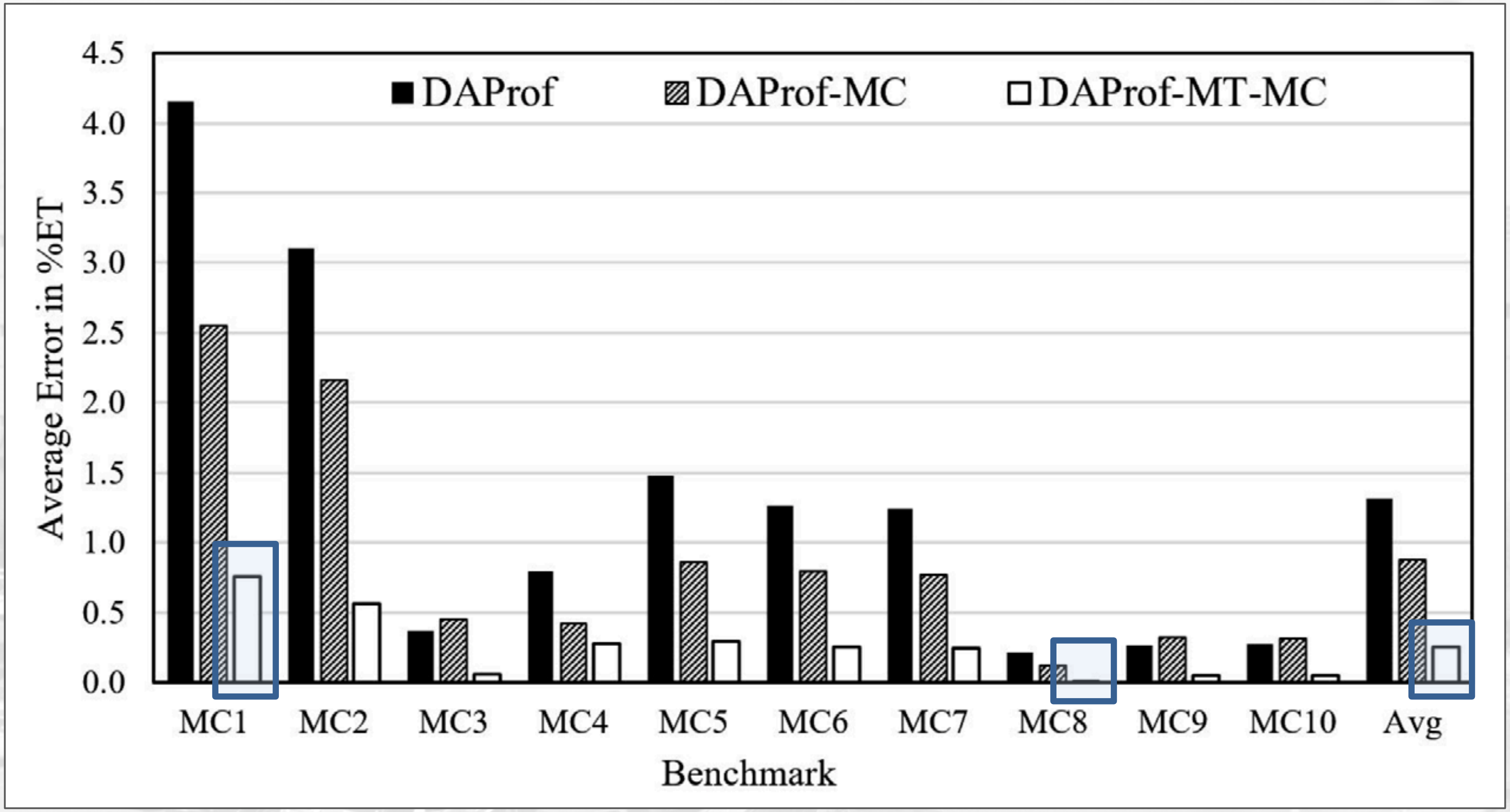
- **DAProf with multicore scaling (DAProf-MC)**
  - Extend profile task filter with *per core* instruction counts
    - Instructions counts used to scale the estimated percentage of execution (%ET) across tasks

Experimental Setup

- Multitasked benchmarks
  - Individual application tasks taken from the MiBench benchmark suite
  - Labeled *MTx.y*, where *x* indicates the number of tasks within the benchmark and *y* is a unique ID
- Multicore benchmarks
  - 10 multicore benchmarks
  - Two multitasked application mapped onto separate processor core
- Evaluate the accuracy of DAProf, DAProf-MC, and DAProf-MT-MC

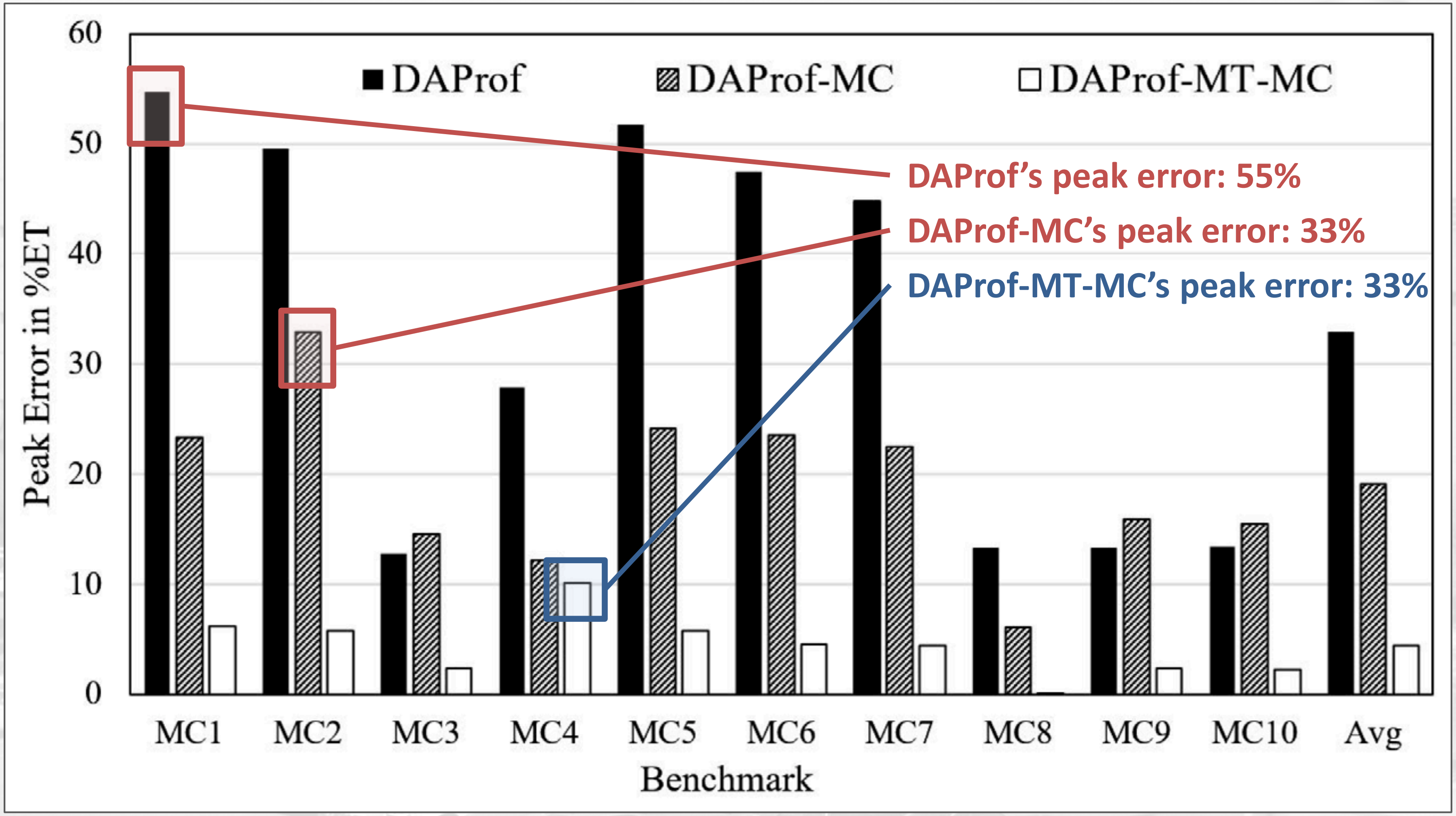
CJPEG				✓	✓		✓
DJPEG						✓	
FFT							✓
TIFF2BW						✓	
SUSAN	✓			✓	✓		✓
DIJKSTRA	✓				✓		✓
BIT COUNT		✓	✓				
STRINGSEARCH		✓	✓			✓	
QSORT				✓	✓		✓
RAWCAUDIO						✓	
RAWDAUDIO			✓				
	MT2.3	***MT2.5***	***MT3.1***	***MT3.3***	***MT4.1***	***MT4.2***	***MT5.1***
MC1	✓	✓					
MC2	✓		✓				
MC3	✓					✓	
MC4		✓		✓			
MC5		✓					✓
MC6			✓		✓		
MC7			✓				✓
MC8				✓		✓	
MC9					✓	✓	
MC10						✓	✓

Experimental Results – Profiling Accuracy



DAProf-MT-MC's average error ranges from 0% to 0.7%

Experimental Results – Profiling Accuracy



DAProf's peak error: 55%

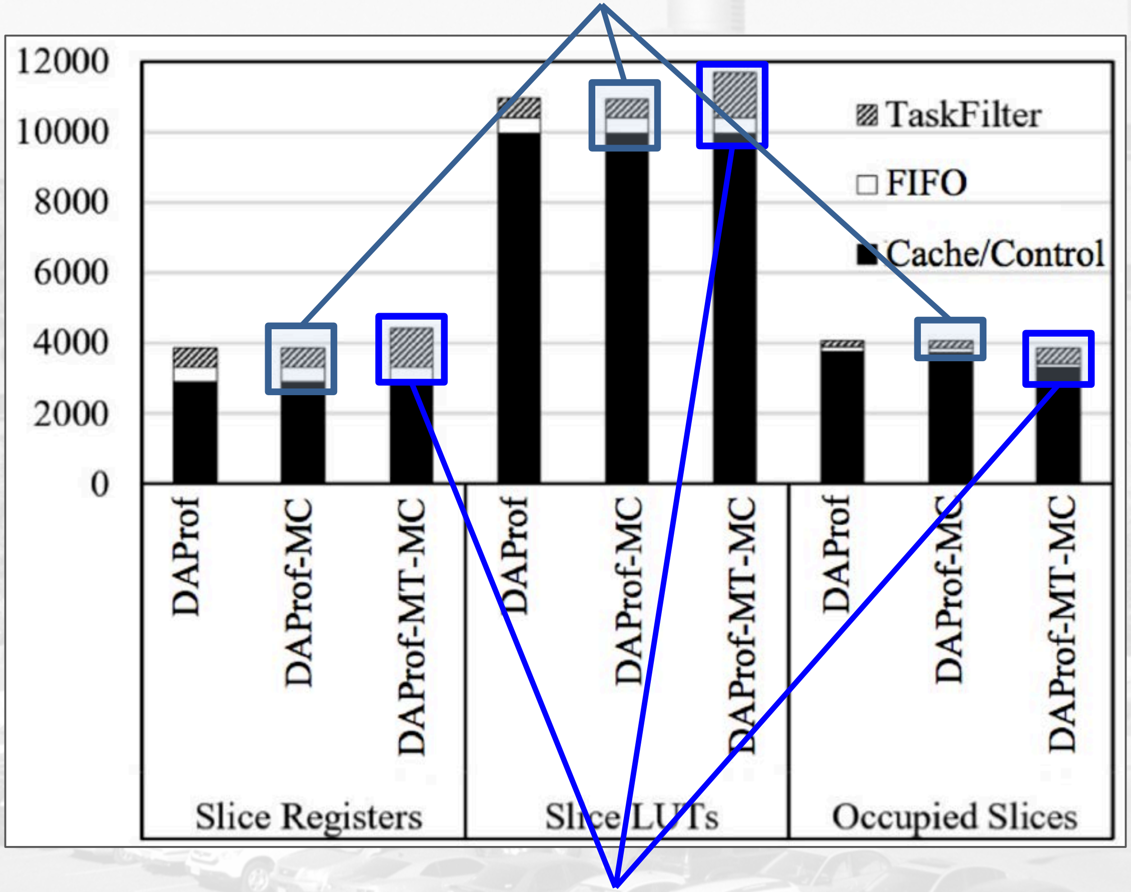
DAProf-MC's peak error: 33%

DAProf-MT-MC's peak error: 33%

DAProf-MT-MC's reduces peak error by as much as 33%

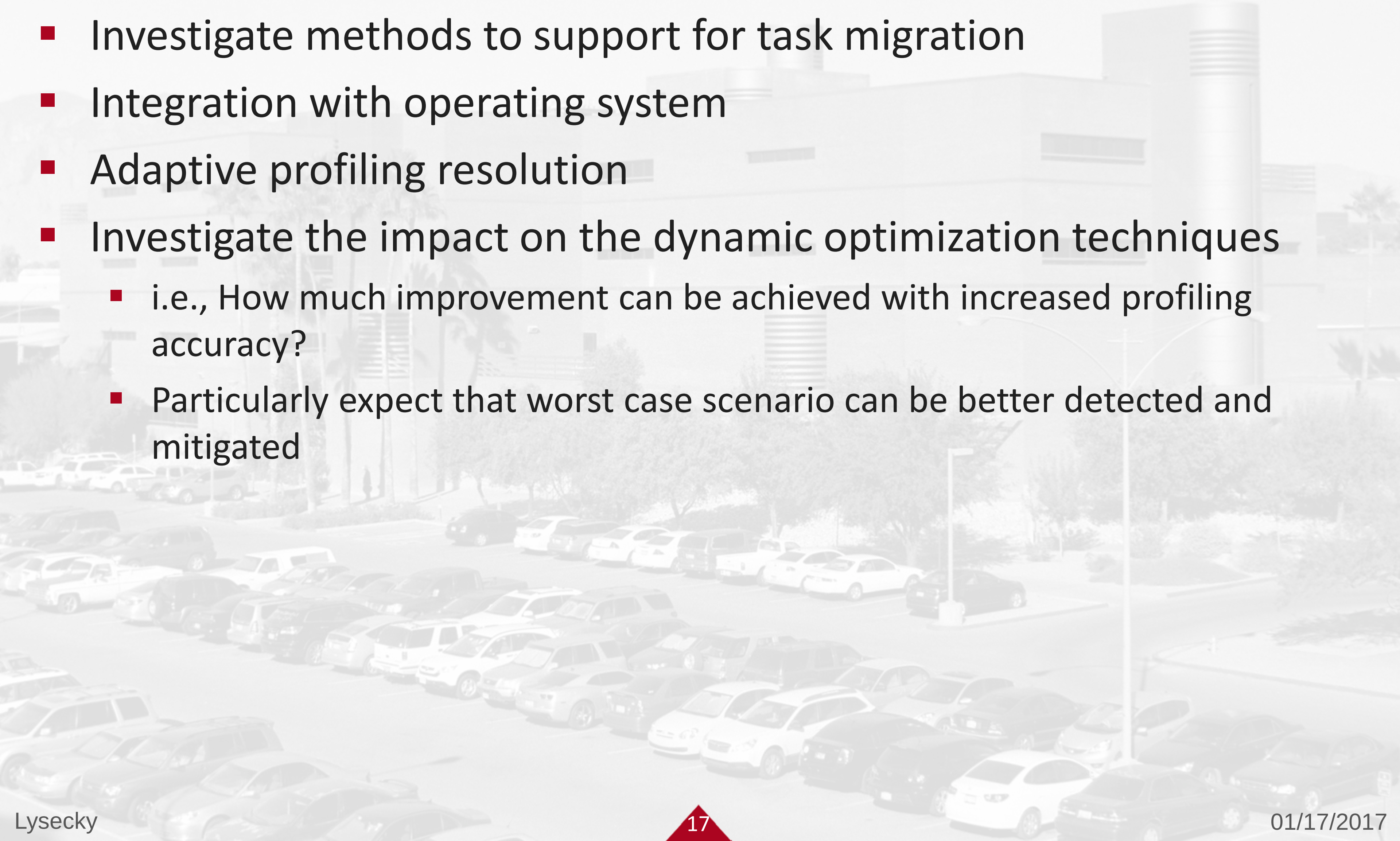
Experimental Results – Area Requirements

DAProf-MC's requires only 0.5% larger Profiler Task Filter



DAProf-MT-MC's requires 9.1% larger Profiler Task Filter

The Distant Future (The Year 2017)



- Investigate methods to support for task migration
- Integration with operating system
- Adaptive profiling resolution
- Investigate the impact on the dynamic optimization techniques
  - i.e., How much improvement can be achieved with increased profiling accuracy?
  - Particularly expect that worst case scenario can be better detected and mitigated

Thank You, Gracias, Danke, Merci, Tak, ありがとう, Kittos