SimTrace: Capturing Over Time Program Phase Behavior

Steven Flolid – UT Austin
Emily Shriver – Intel Labs
Zachary Susskind – UT Austin
Benjamin Thorell – UT Austin
Lizy John – UT Austin

Proxy Motivation

- Pre-silicon design emulators are prohibitively slow

Comparing Run Time on Different Platforms		
Silicon	1 second	1 minute
Emulation	4.7 hours	11.8 days

- Current techniques create proxies that capture key performance and power metrics

- It is beneficial to use the same workload throughout all stages of system design

Prior Art: Limitations & Research

Limitations

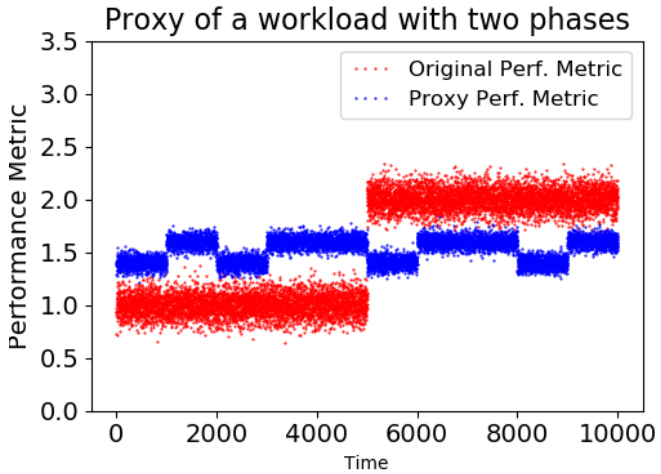
- CPU Centric
- Limited automation, requires hand-tuning
- Not modeling over-time behavior of metric (e.g. IPC, dynamic capacitance, cache miss)

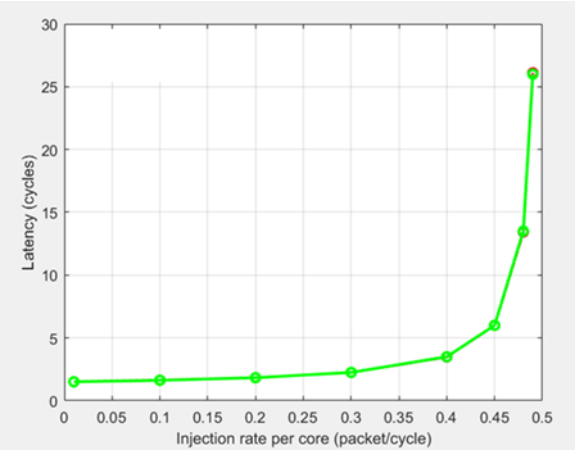
Open Research Questions

- Capture over time behavior
- Increase automation

Problem with Average Proxies

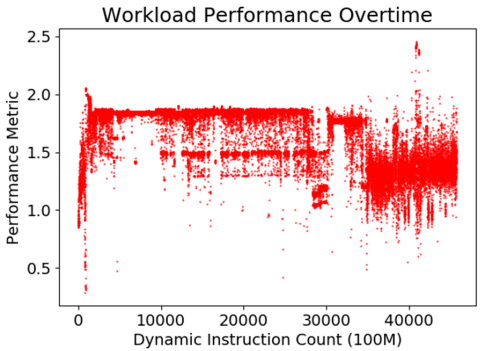
- Existing techniques do not capture over time variation within a program
- When a system runs multiple proxies, shared resources may not be utilized correctly
- Power management algorithms use over time behavior

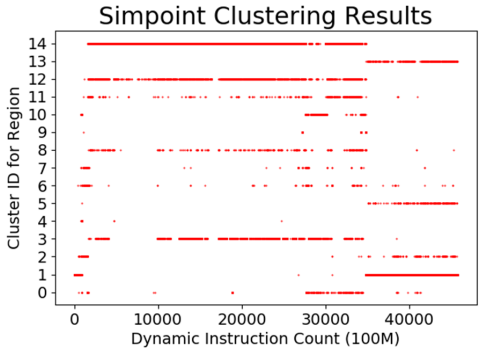




Overview of Simpoint

- Simpoint [1] breaks a workload into equal sized regions (100 Million Instructions)
- Regions are profiled based on micro-architecture independent Basic Block Vectors (BBV)
- Similar regions are clustered together based on the BBV using K-means
- A single region is simulated to represent each cluster





Cluster ID	Cluster Weight	Representative Perf. Metric
0	0.87%	1.60
1	23.62%	1.21
2	1.36%	1.91
3	12.00%	1.47
4	0.09%	1.04
5	0.44%	1.47
6	0.69%	1.61
7	0.36%	1.46
8	0.84%	1.54
9	0.17%	1.33
10	3.60%	1.16
11	0.82%	1.74
12	5.28%	1.77
13	1.21%	1.41
14	48.65%	1.85

Weighted Sum

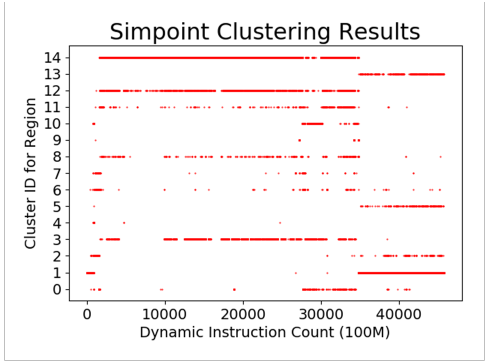
Simpoint Estimate
1.61

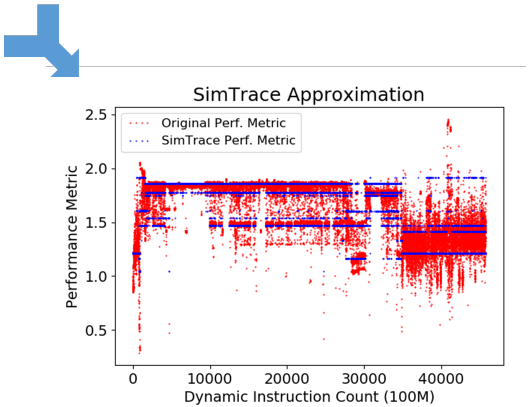
[1] Automatically Characterizing Large Scale Program Behavior: https://dl.acm.org/citation.cfm?id=605403

The SimTrace Technique

- Simpoint possesses an over time cluster trace but does not use it
- A single representative proxy could be created for each cluster
- Replaying the proxies in the cluster trace order results with a SimTrace
- This technique could serve as a baseline for future over time proxies

Cluster ID	Proxy Metric
0	1.60
1	1.21
2	1.91
3	1.47
4	1.04
5	1.47
6	1.61
7	1.46
8	1.54
9	1.33
10	1.16
11	1.74
12	1.77
13	1.41
14	1.85



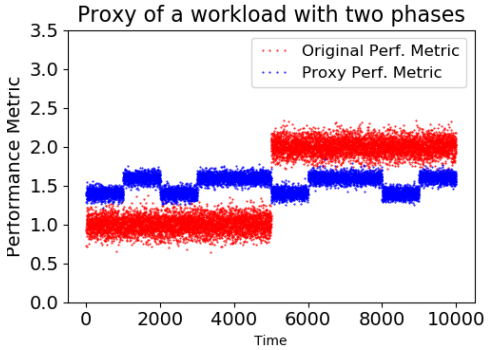


Measuring Over Time Similarity

- A Program's average error is often used for accuracy measures

$$Average\ Error = \frac{|x_{experimental} - x_{reference}|}{x_{reference}}$$

- Over time comparisons require more powerful techniques

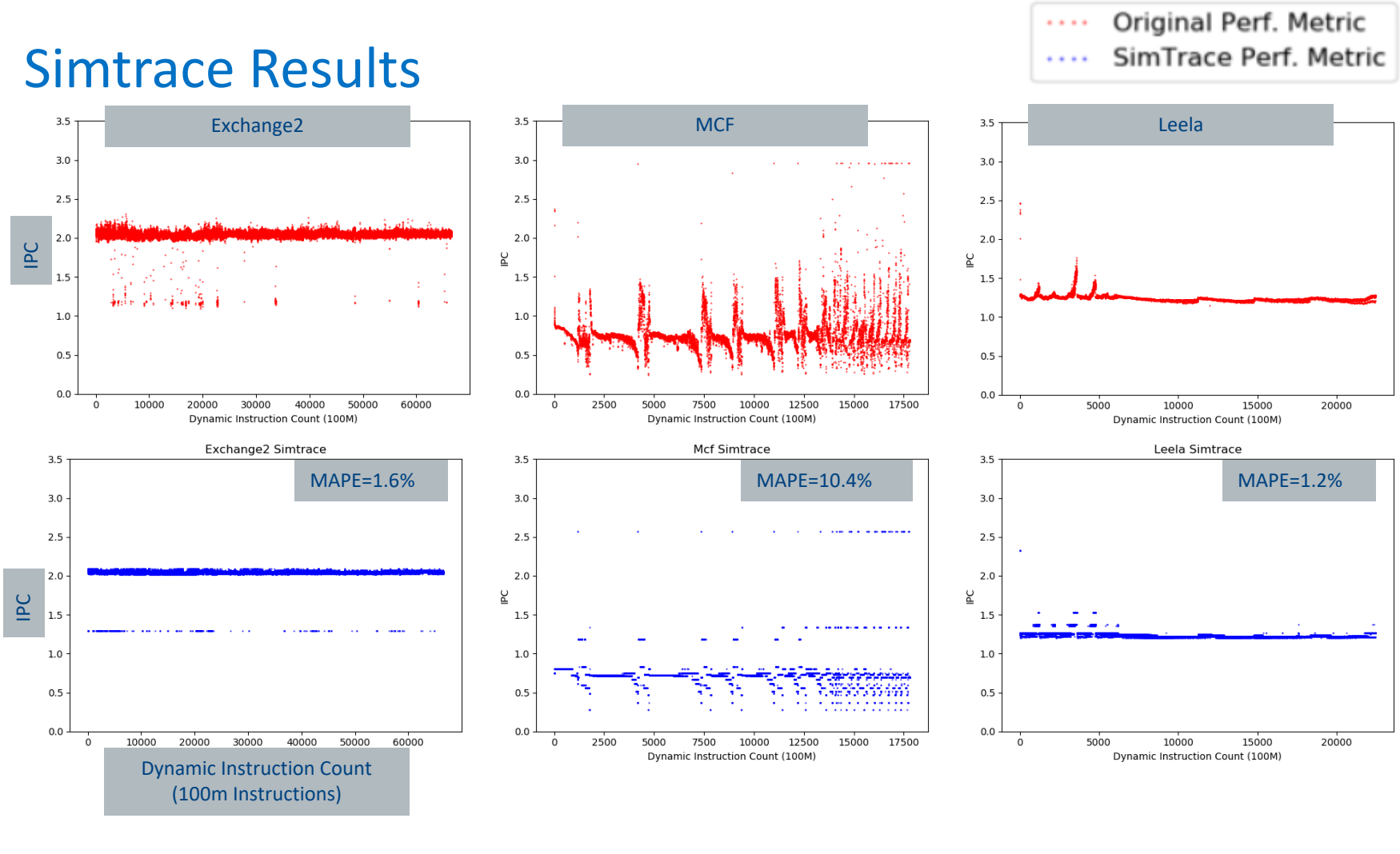


- Various options in literature to measure time series similarity[1]
  - Point by point Mean Abs Error
  - Euclidean Distance
  - Correlation
  - Dynamic Time Warping
  - Kolmogorov-Smirnov Test
  - Many Others

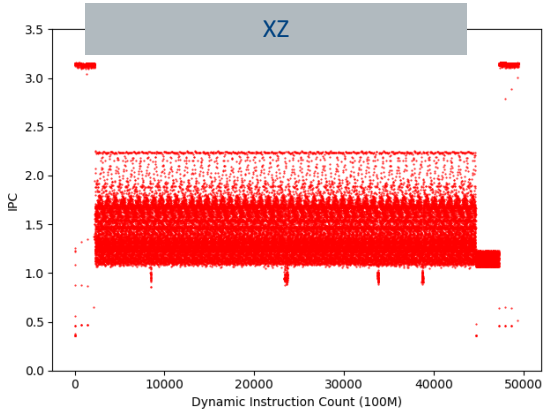
We used metrics recommended by [1]

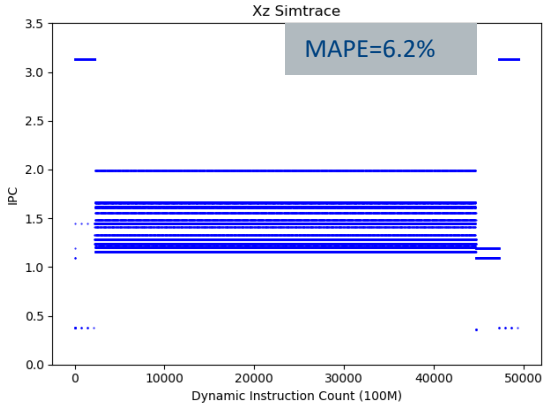
[1] An Empirical Evaluation of Similarity Measures for Time Series Classification: https://arxiv.org/abs/1401.3973

Simtrace Results

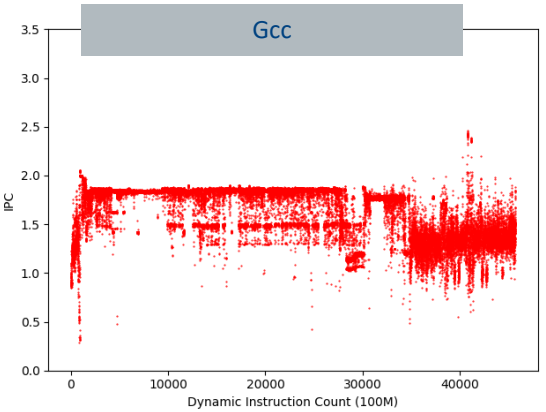


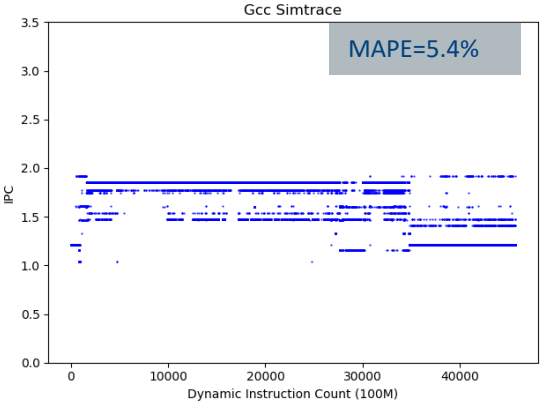
Simtrace Results (Cont..)





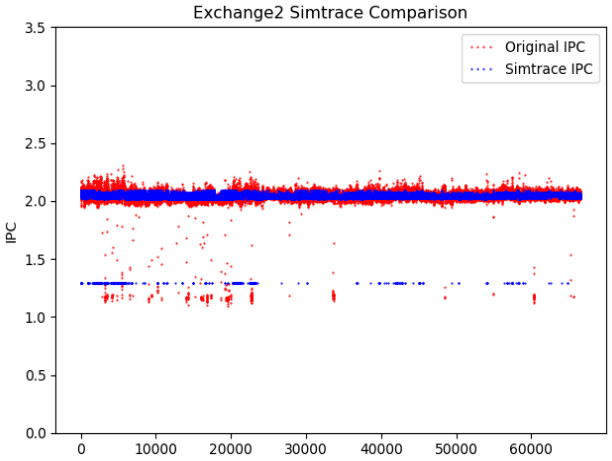
- Program's with similar average IPC can have vastly different over time behavior
- Simtrace naturally removes complexity from a program's performance
- Simtrace follows regular trends more accurately than irregular trends

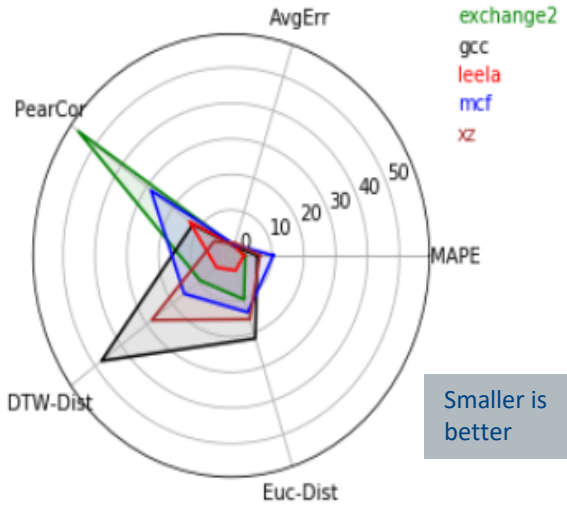




Similarity Results

IPC	Trace Length # simpoints	MAPE	Avg Err	Pear Cor	Euc Dist	DTW / #simpoints
leela	49,459	1.2%	-0.0015	0.87	3.65	0.01
exchange2	66,589	1.6%	-0.0006	-0.07	26.35	0.04
gcc	17,817	5.4%	-0.0005	0.88	28.99	0.36
xz	45,718	6.2%	-0.0094	0.96	34.29	0.13
mcf	22,460	10.4%	0.0044	0.72	20.92	0.10





- SimTrace performs well for Point by Point error metrics (MAPE, Avg Err)
- Each technique captures some characteristics of over time performance

Smaller is better

Conclusions and Next Steps

Conclusions:

- Promising initial results, but more investigation is needed
- DTW and Euclidean are useful for comparison but are difficult to interpret without normalization

Next Steps:

- Create Simtraces for other benchmarks of SPEC CPU 2017
- Explore Simtrace's ability to capture over time behavior of micro-architecture independent metrics (Imix, branches, footprint, etc)
- Normalizing Euclidian distance and Dynamic Time Warping