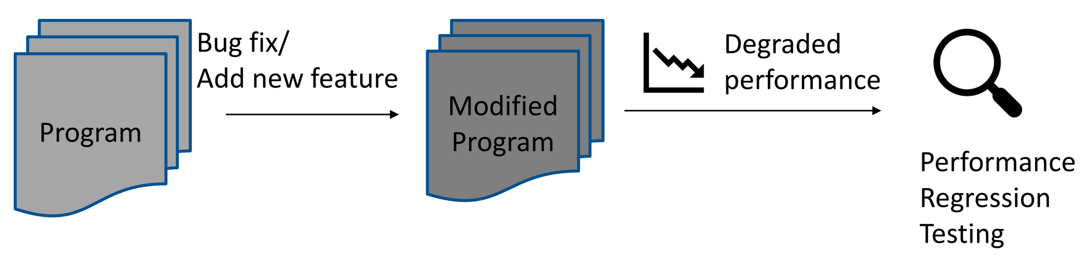
A ZERO-POSITIVE LEARNING APPROACH FOR DIAGNOSING SOFTWARE PERFORMANCE REGRESSIONS

Mejbah Alam, Justin Gottschlich, Nesime Tatbul, Javier Turek, Timothy Mattson, Abdullah Muzahid

[Intel Labs + Texas A&M]

PROBLEM

Automatic Performance Regression Testing



Detecting **performance anomaly** introduced by a change in software

Diagnosis of Parallel Software Performance Anomalies is Challenging

Real-world performance regressions are **diverse** and **complex**

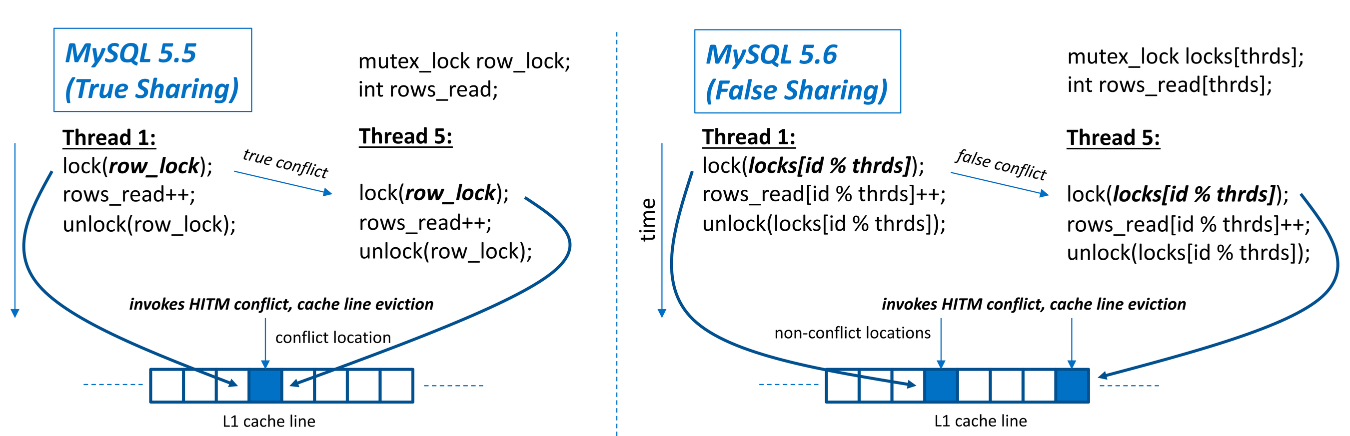
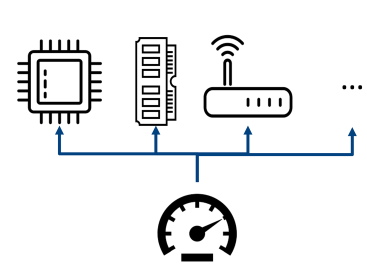


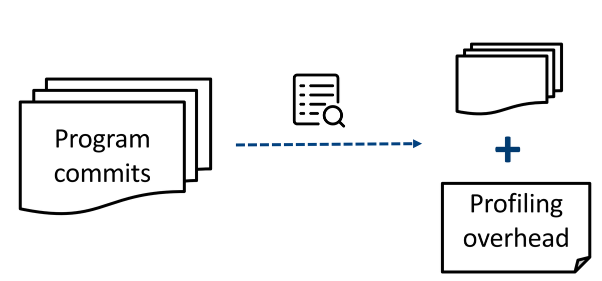
Figure: Example of performance regressions in parallel software

- **Key Challenges in Existing Tools:**

  1. **Generality:**
     - Detect root cause of diverse types software performance issues.



  2. **Scalability:**
     - Fine-grained diagnosis of program execution with reduced perturbation.



- **General Anomaly Detection Challenges:**

  Learning from "normal" programs:
  - Anomalies are rare
  - Leverage non-anomalous programs to detect anomalous ones.

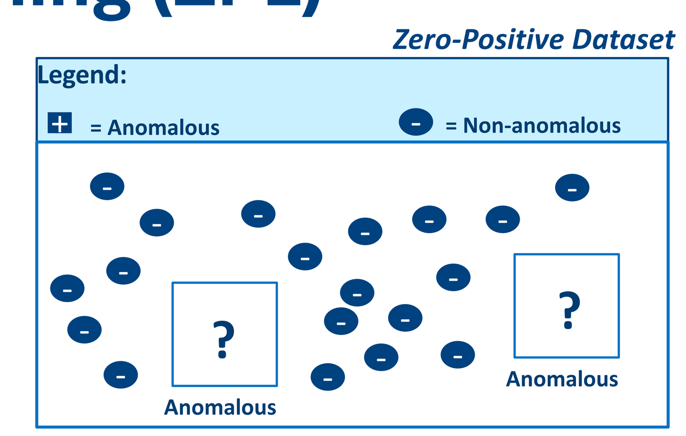
SOLUTION

AutoPerf

Zero-Positive Learning + Auotencoders + Hardware Telemetry

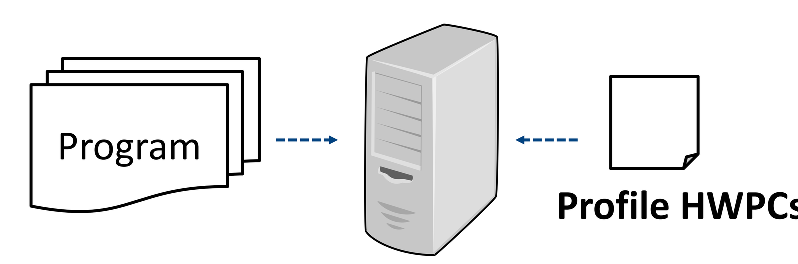
Zero-Positive Learning (ZPL)

- Train only on non-anomalous data
- Why ZPL for performance regressions?
  - Does not rely on training data that includes performance regressions



Hardware Telemetry for Perf Regressions

- **Hardware Performance Counters (HWPCs):**
  - Special purpose registers in modern CPUs
  - Store counts of wide-range of hardware-related activities



✓Low overhead
✓Reduced perturbation

ZPL of Performance Regressions

- Autoencoder to learn HWPC data distribution of normal (non-anomalous) program executions

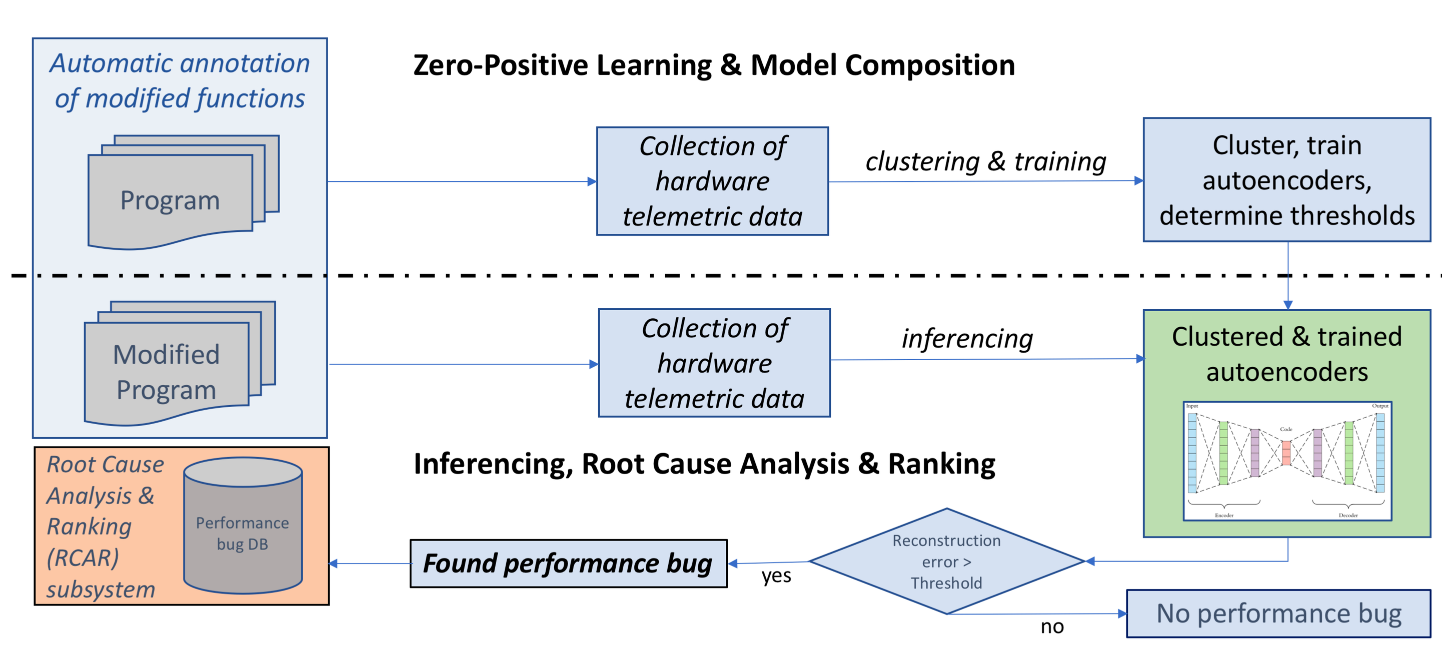


Figure: Overview of AutoPerf

Reconstruction error threshold: $\gamma(t) = \mu_\epsilon + t\sigma_\epsilon$

μ_ϵ : *mean reconstruction error*
σ_ϵ : *standard deviation*
t : *controls threshold level*

More Information

Watch: https://www.youtube.com/watch?v=FkT1aNoKbG4&feature=youtu.be

Read: https://arxiv.org/abs/1709.07536 **Use:** https://github.com/mejbah/AutoPerf

RESULTS

Generality

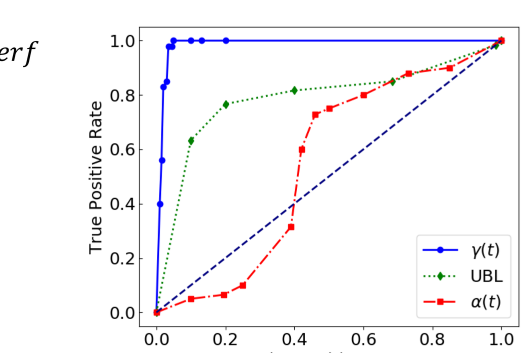
- Detects **10** real perf bugs in **7** benchmark and open-source programs
- Different types of bugs in parallel software: True Sharing (TS), False Sharing (FS), NUMA Latency (NL)
- Better accuracy than state-of-the-art approaches DT[1] and UBL[2]

Normal Program	False Positive Rate			Anomalous Program	Defect Type	False Negative Rate		
	AutoPerf	DT	UBL			AutoPerf	DT	UBL
blackscholes_L	0.0	N/A	0.2	blackscholes_K	*NL*	0.0	N/A	0.0
bodytrack_L	0.0	0.7	0.8	bodytrack_K	*TS*	0.0	0.17	0.1
dedup_L	0.0	1.0	0.2	dedup_K	*TS*	0.0	0.0	0.0
histogram_M	0.0	0.0	0.0	histogram_M	*FS*	0.0	0.1	1.0
$\text{linear\_regression}_M$	0.0	0.3	0.0	$\text{linear\_regression}_M$	*FS*	0.0	0.4	0.35
reverse_index_M	0.0	0.4	0.15	reverse_index_M	*FS*	0.0	0.1	0.05
streamcluster_L	0.0	N/A	0.6	streamcluster_K	*NL*	0.0	N/A	0.1
Boost_L	**0.3**	1.0	0.4	Boost_L	*FS*	0.0	0.2	0.2
Memcached_L	0.0	1.0	0.4	Memcached_L	*TS*	0.0	0.4	0.3
MySQL_L	**0.2**	1.0	0.1	MySQL_L	*FS*	0.0	0.5	0.8

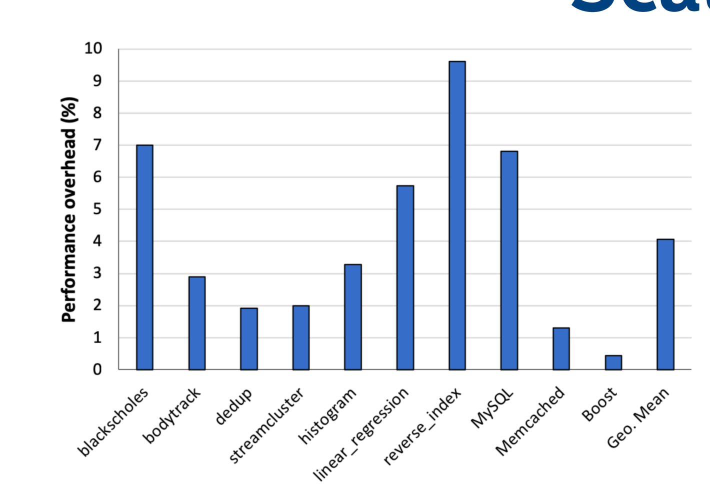
Figure: Diagnosis ability of AutoPerf vs DT[1] and UBL[2] in candidate programs. K, L, M are # of executions used for experiments (K=6, L=10, M=20).

$\gamma(t)$: *Threshold of AutoPerf*
UBL : State-of-the-art [1]
$\alpha(t)$: *Arbitrary threshold*

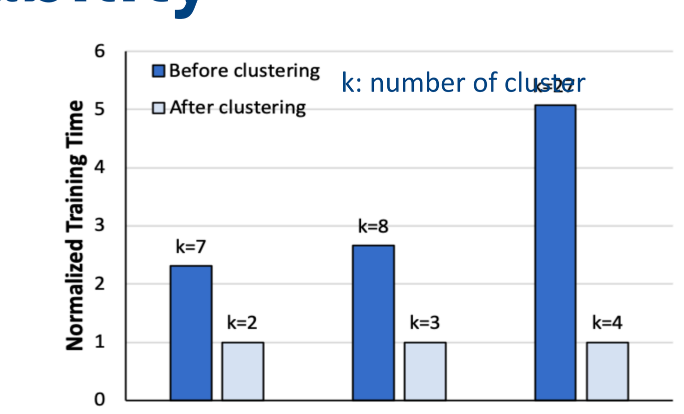
No false negatives found in our tests
(no missed performance bugs)



Scalability



Profiling overhead (< 4%)



Reduced training time using clustering

Conclusion & Future Work

- AutoPerf makes software performance analysis with hardware telemetry more **general** and **scalable** with zero-positive learning.
- **Limitations:**
  - Diagnoses performance defects if explainable by HWPC
  - Availability of clean data, effective test cases for execution profiles

References

1. S. Jayasena, S. Amarasinghe, A. Abeyweera, G. Amarasinghe, H. D. Silva, S. Rathnayake, X. Meng, and Y. Liu. Detection of False Sharing Using Machine Learning. In 2013 SC -International Conference for High Performance Computing, Networking, Storage and Analysis(SC)
2. D. J. Dean, H. Nguyen, and X. Gu. UBL: Unsupervised Behavior Learning for PredictingPerformance Anomalies in Virtualized Cloud Systems. In Proceedings of the 9th InternationalConference on Autonomic Computing, ICAC '12