Adaptive Execution Tracing in System Behavior Analysis: A Language Model-based Approach

Kasra Darvishi, Morteza Noferesti, Naser Ezzati-Jivan

Brock University

Outline

Tracing

Overview: A fundamental technique for capturing the complex behavior of computer systems

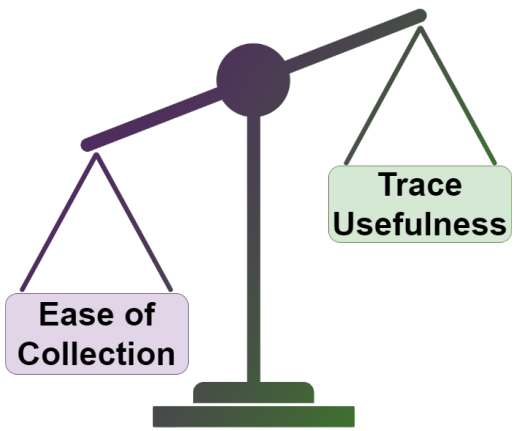
Kernel Tracing: Collects comprehensive data from the operating system

Tracing

Overview: A fundamental technique for capturing the complex behavior of computer systems

Kernel Tracing: Collects comprehensive data from the operating system

Data Management: Managing and analyzing the large volume of data from long-running traces is challenging.



Efforts to Reduce Tracing Burden

Automation of Trace Analysis: ML & DL for anomaly detection

- "Language Models for Novelty Detection in Kernel Traces" by Quentin Fournier et al.

Efforts to Reduce Tracing Burden

Automation of Trace Analysis: ML & DL for anomaly detection

- "Language Models for Novelty Detection in Kernel Traces" by Quentin Fournier et al.

Seeking Balance in Quality and Cost:

- Instrumentation Budgeting: Pythia caps tracing expenses [1]
- Adaptive Instrumentation: Zhang et al. adaptively instruments user space [14]

Addressing the Gap with Adaptive Tracing

Core idea is to trace the system with minimal overhead:

Addressing the Gap with Adaptive Tracing

Core idea is to trace the system with minimal overhead:

- Abstaining from detailed tracing when it is unnecessary

Addressing the Gap with Adaptive Tracing

Core idea is to trace the system with minimal overhead:

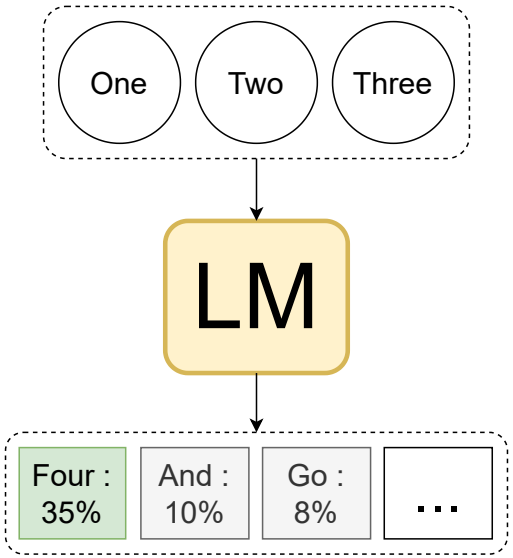
- Abstaining from detailed tracing when it is unnecessary
- Record traces for analysis only during significant shifts

Addressing the Gap with Adaptive Tracing

Core idea is to trace the system with minimal overhead:

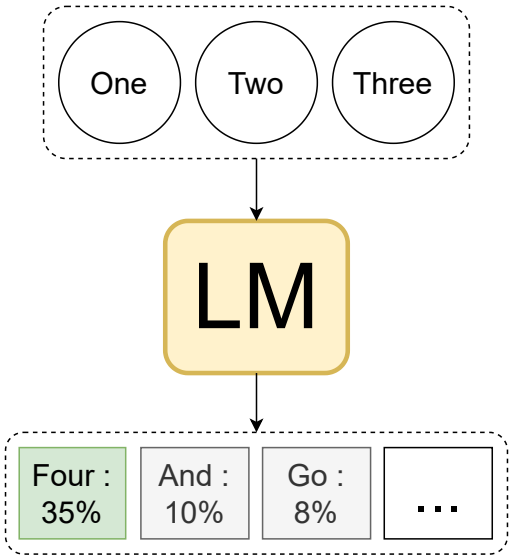
- Abstaining from detailed tracing when it is unnecessary
- Record traces for analysis only during significant shifts
- Focus on subsystems related to root-cause

Trace Sequence Modeling



Trace & Natural Language: Can be viewed as natural language, possessing specific patterns and grammar rules

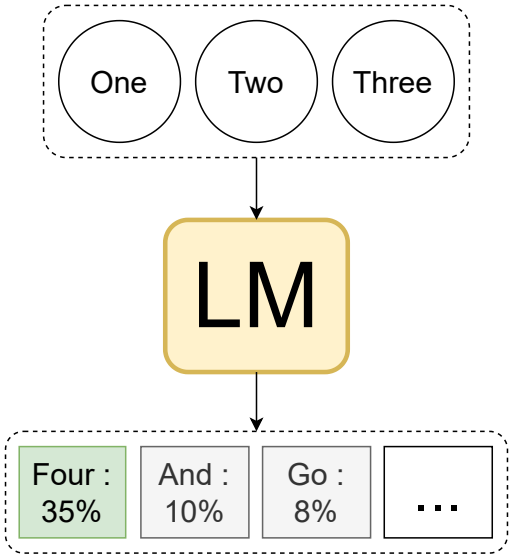
Trace Sequence Modeling



Trace & Natural Language: Can be viewed as natural language, possessing specific patterns and grammar rules

Language Model (LM): Frequently used for text generation

Trace Sequence Modeling

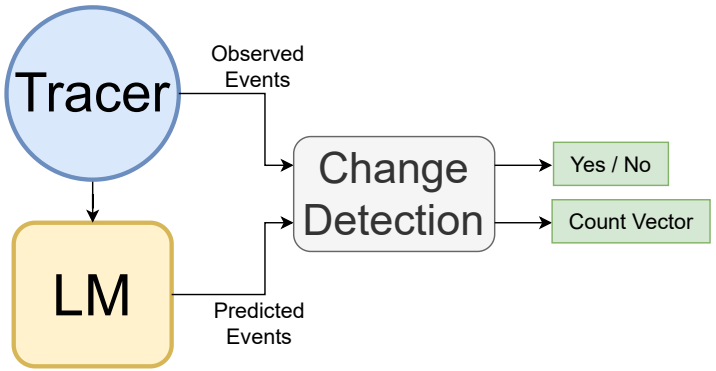


Trace & Natural Language: Can be viewed as natural language, possessing specific patterns and grammar rules

Language Model (LM): Frequently used for text generation

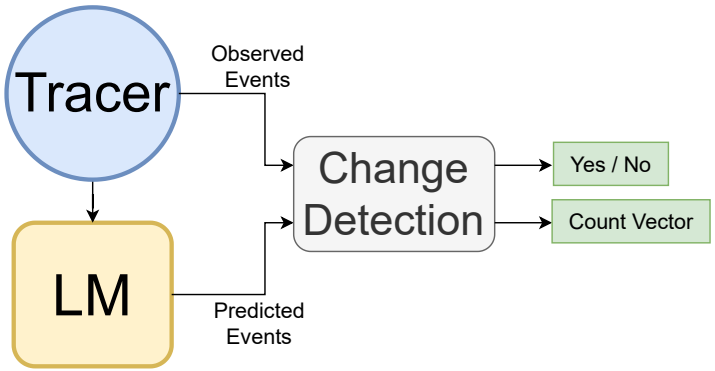
LM for System Behavior Modeling: Has been shown to be successful by several studies

Novelty Detection by LMs



Classification of Sequences: Each sequence is classified to normal or novel based on model's prediction

Novelty Detection by LMs

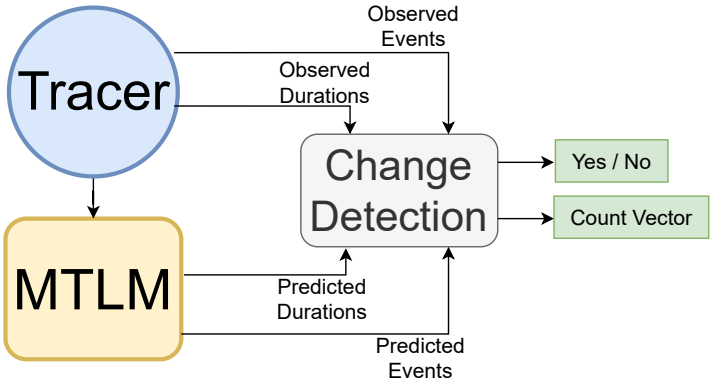


Classification of Sequences: Each sequence is classified to normal or novel based on model's prediction

Cross-Entropy loss: A metric to measure the difference between the expected and observed behavior

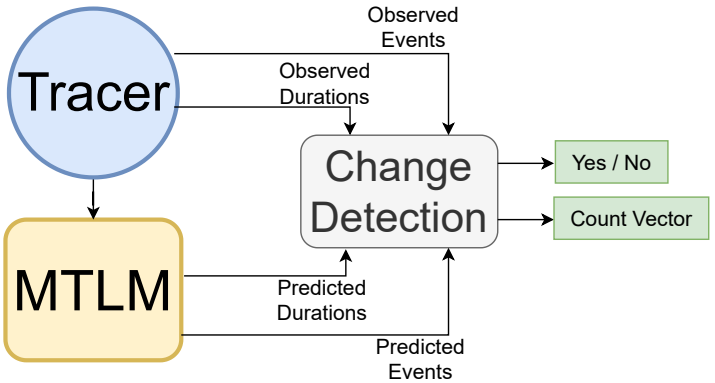
Threshold: Determines which changes are significant

Duration Modeling



Event Duration: Represents the time from when a system call starts until it finishes

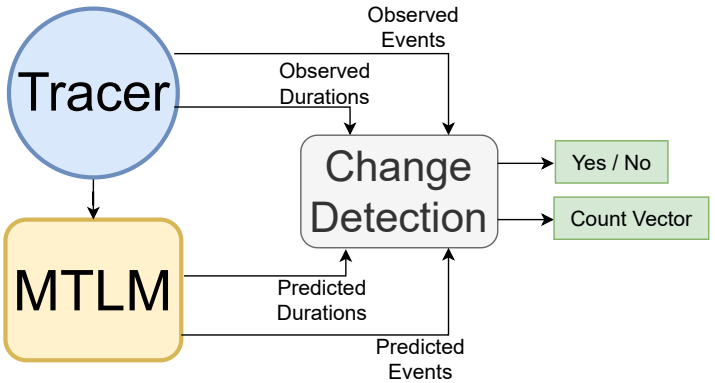
Duration Modeling



Event Duration: Represents the time from when a system call starts until it finishes

Categorization: $f(i) = \frac{k-i+1}{Total}$ & $Total = \frac{k \cdot (k+1)}{2}$.
For k=5: 33%, 26%, 20%, 13%, 6%

Duration Modeling



Event Duration: Represents the time from when a system call starts until it finishes

Categorization: $f(i) = \frac{k-i+1}{Total}$ & $Total = \frac{k \cdot (k+1)}{2}$.
For k=5: 33%, 26%, 20%, 13%, 6%

Training Objective: Prediction of duration category for exit system calls

Root-cause Analysis

Goals: Guide adaptive tracing procedure and admin

Root-cause Analysis

Goals: Guide adaptive tracing procedure and admin

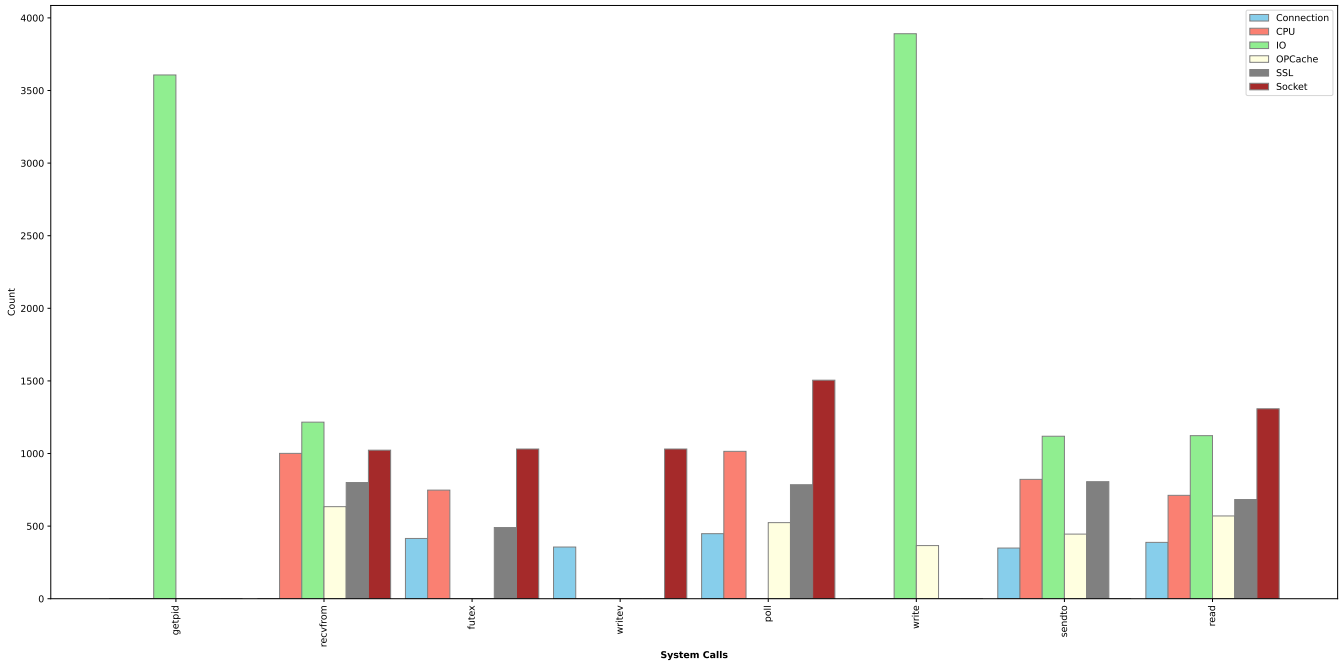
Count Vectorization: Events contributing to the sequence's abnormality

Root-cause Analysis

Goals: Guide adaptive tracing procedure and admin

Count Vectorization: Events contributing to the sequence's abnormality

Matching the Vector: Centroids represent each subset of previously tagged traces



Trace event reduction

Core idea is to use a subset of possible events for constant monitoring to further reduce the overhead

Recording Events: A separate subset of events are recorded based on root-cause

Trace event reduction

Core idea is to use a subset of possible events for constant monitoring to further reduce the overhead

Recording Events: A separate subset of events are recorded based on root-cause

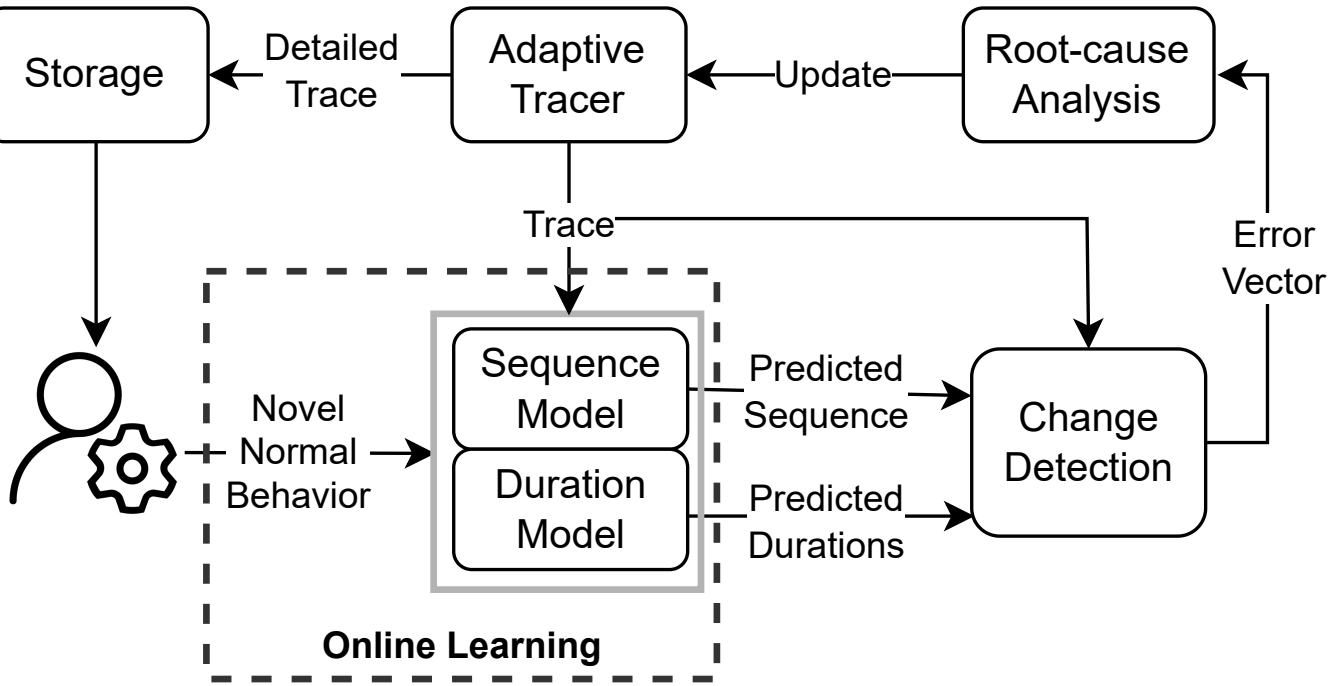
Performance: $\sim 5\%$ decrease in performance for $\sim 30\%$ decrease in trace volume

Trace event reduction

Core idea is to use a subset of possible events for constant monitoring to further reduce the overhead

Recording Events: A separate subset of events are recorded based on root-cause

Performance: $\sim 5\%$ decrease in performance for $\sim 30\%$ decrease in trace volume



Change Detection by Duration

Threshold: A single value selected for all the noise sets from Fournier et al. (2023) and our collected dataset

Change Detection by Duration

Threshold: A single value selected for all the noise sets from Fournier et al. (2023) and our collected dataset

Performance: $\sim 10\%$ increase in F1-score

Reliability: The choice between these models is context-dependent

Noise Type	Duration	Multi-Task	Fournier et al.
Connection	99.1	96.7	67.9
CPU	98.3	93.6	88.6
IO	99.2	98.5	93.4
OPCache	96.6	98	93.3
Socket	99.2	98.5	93
SSL	99.2	98.2	86
Total	98.6	97.2	87.0

Noise Type	Duration	Multi-Task	Event
Sysbench	82.2	87.3	88.7
Bandwidth	79.3	78.8	60.1

Root-cause Analysis

Event Vs Duration Sequence: Events are more effective for root-cause identification, showing the need for both models

Noise Type	Event	Duration	Multi-Task
Connection	94.2	82.8	85.5
CPU	99.2	99.5	99.1
IO	99.9	99.9	99.9
OPCache	99.1	97.4	98.4
Socket	99.5	99.7	99.6
SSL	91.1	81.4	84.2
Total	97.17	93.45	94.45

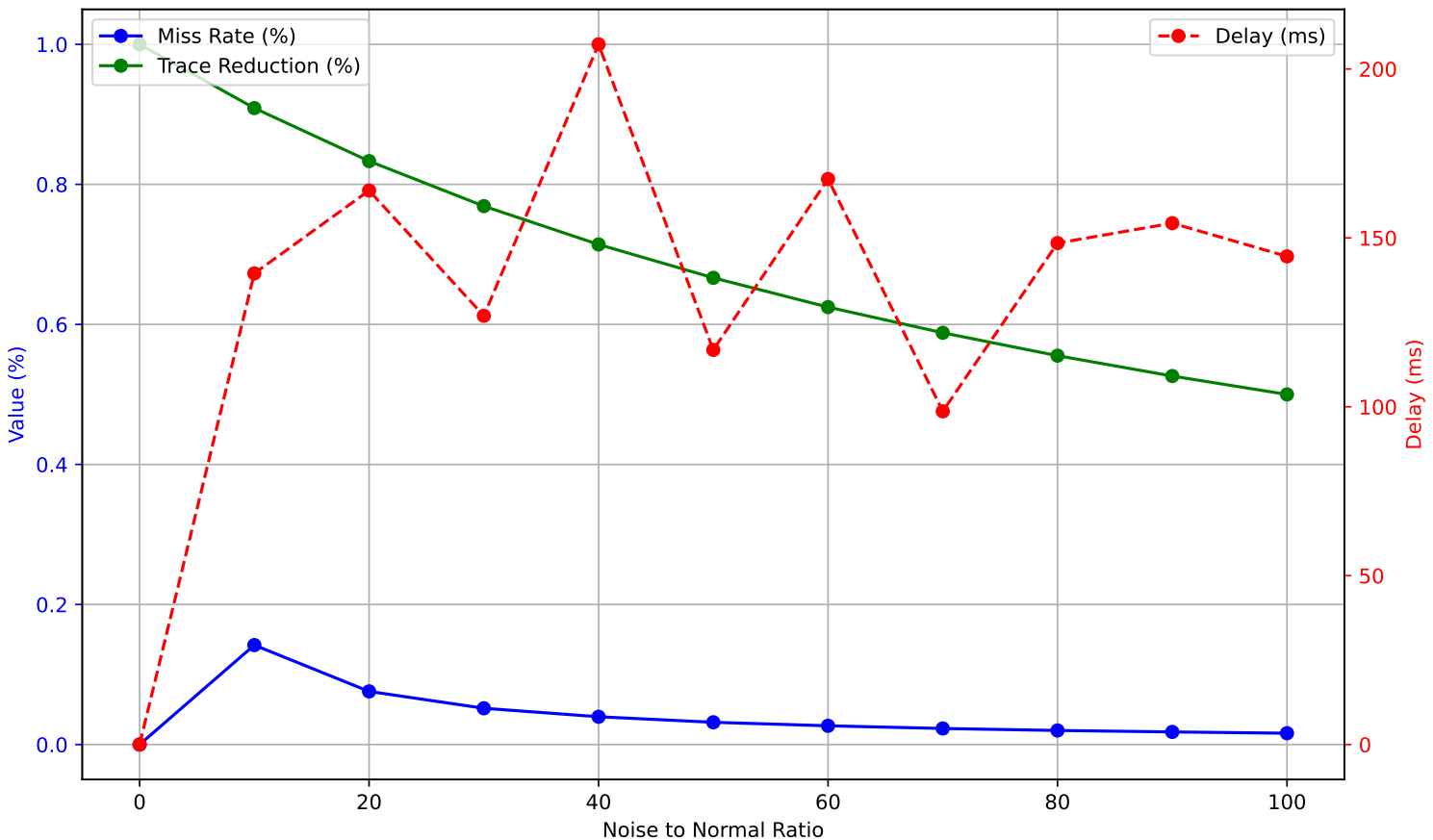
Adaptive Tracing

Test scenarios: Consisting of 0% to 50% noisy events

Adaptive Tracing

Test scenarios: Consisting of 0% to 50% noisy events

- 77% reduction of recorded events
- Average miss rate of 5.8%
- Detection of significant changes within an average of 170ms



Conclusion & Future Works

Conclusion & Future Works

Improved System Modeling and Analysis Through Kernel Trace

Conclusion & Future Works

Improved System Modeling and Analysis Through Kernel Trace

Significant Reduction in Tracing Overhead

Conclusion & Future Works

Improved System Modeling and Analysis Through Kernel Trace

Significant Reduction in Tracing Overhead

Accepted at ICSE - NIER

Next Steps:

- Extension to Other Data
- Online Learning

Thank You!

Any Questions?

Email: kdarvishi@brocku.ca