Track 3 update: Tracing the Trail, Investigating the Influence of Logging on Bug Resolution

Amir Haghshenas

Naser Ezzati-Jivan
Michel Dagenais

Summer 2025

POLYTECHNIQUE MONTRÉAL

Agenda

- TraceLense paper with Masoumeh.
- Update on third track paper.
- Final steps.

TraceLens: Early Detection of Software Anomalies Using Critical Path Analysis

POLYTECHNIQUE
MONTRÉAL

Problem statement

- Runtime anomaly detection in software systems is critical but faces challenges with traditional system call analysis methods.
- Existing approaches generate large volumes of trace data, making real-time detection impractical for large-scale systems
- Current methods often fail to capture execution dependencies, making it difficult to correlate anomalies with root causes

Objective

- Develop an efficient method for early detection of software performance anomalies in real-time
- Reduce data collection overhead while maintaining high detection accuracy
- Create an approach that enables adaptive tracing for large-scale systems

Key Achievements

- Achieved similar accuracy in anomaly detection using critical path analysis, comparable to system call methods.
- Reduced data collection overhead compared to for system call methods with 1/5 overhead.
- Decreased data size significantly (%0.0004 ratio) while maintaining similar precision and recall

Tracing the Trail: Investigating the Influence of Logging on Bug Resolution

POLYTECHNIQUE MONTRÉAL

Problem statement

- **Narrow Focus:** Most research looks only at immediate benefits of logging (debugging, monitoring).
- **Missing the Big Picture:** We don't understand how logging impacts software quality over time.
- **Unexplored Territory:** The efficiency of logging for bug-fixing across a project's lifespan remains a mystery.

Objective

- **Main research objective:** To find how efficient logging is for software bug-fixing.
- **Analyze Development History:** Examine how logging practices appear throughout a project's timeline.
- **Study Logging Evolution:** Observe how logging practices change and adapt over the project's lifetime.

Research Questions

1. What is the relationship between bug-fixing commits and logging in terms of the quantity of logs, the levels of logs used, and modifications made to logs within those commits?

2. What proportion of files that contain logging statements are involved in bug-fixing commits compared to those that are not? Does this suggest that logged files are more prone to bugs or more actively maintained?

3. How developers approach the logging activity during the development of a software? Can we identify a pattern?

Preliminary Results

Only a small percentage of bug-fixing commits are logged

Changes in logging decision is more frequent in bug-fixing commit.

Bug-fixing commits on average have more severe log levels.

Preliminary results

```
6a2ace739fa97dc25b41014944343d0c65e17c3a(10-2-2013)/castor/after_CastorMarshaller.java": {
  "description": "The function createXMLContext initializes a Castor XMLContext with mapping files, target classes, and target packages.",
  "features": {
    "before": {
      "number_of_loops": 3,
      "number_of_branches": 3,
      "number_of_lines_of_code": 27,
      "number_of_function_calls": 7,
      "logging_info": {
        "number_of_log_lines": 4,
        "log_details":[

          {
            "log_level": "info",
            "log_message": "Configured using [<mapping locations>]"
          },
          {
            "log_level": "info",
            "log_message": "Configured for target classes [<target classes>]"
          },
          {
            "log_level": "info",
            "log_message": "Configured for target packages [<target packages>]"
          },
          {
            "log_level": "info",
            "log_message": "Using default configuration"
          }
        ]
      }
    },
    "after": {
      "number_of_loops": 3,
      "number_of_branches": 5,
      "number_of_lines_of_code": 37,
      "number_of_calls_to_other_functions": 7,
      "logging_info": {
        "number_of_log_lines": 0,
        "log_details": []
      }
    }
  }
```

```
"e1720d89fcf65fca6b244df1696c1df67fc0c808(21-4-2014)/type/after_AnnotationMetadataTests.java": {
    "description": "The function asserts that the meta-annotations on a given AnnotationMetadata object have correctly overridden specified attributes.",
    "features": {
        "before": {
            "number_of_loops": 0,
            "number_of_branches": 5,
            "number_of_lines_of_code": 14,
            "number_of_calls_to_other_functions": 6,
            "logging_information": {
                "log_lines": 0,
                "log_details": []
            }
        },
        "after": {
            "number_of_loops": 0,
            "number_of_branches": 5,
            "number_of_lines_of_code": 22,
            "number_of_calls_to_other_functions": 5,
            "logging_information": {
                "number_of_log_lines": 5,
                "log_details": [
                    {
                        "log_level": "INFO",
                        "log_message": "length of basePackages list"
                    },
                    {
                        "log_level": "INFO",
                        "log_message": "basePackages[0]"
                    },
                    {
                        "log_level": "INFO",
                        "log_message": "length of value list"
                    },
                    {
                        "log_level": "INFO",
                        "log_message": "length of 0th value array"
                    },
                    {
                        "log_level": "INFO",
                        "log_message": "length of basePackageClasses list"
                    }
```

Next Steps

1. To answer the third research question, we have created a survey to collect developers' feedback on their logging approach.
2. Ongoing discussion with Ericsson to present the survey to a large audience.
3. Combination of source code analysis and survey results will be used to identify patterns in log maintenance.

Thank you

Amir Haghshenas

Amir.Haghshenas@polymtl.ca

POLYTECHNIQUE MONTRÉAL