Name: Prashanth Mallyampatti
Unity ID: pmallya
Student ID: 200250501

CSC724- Advanced Distributed Systems
Paper Review

X-ray: Automating Root-Cause Diagnosis of Performance Anomalies in Production Software

Summary:

This paper addresses on how to identify the root causes for a problem, and costs of each root cause in production softwares. Performance hindrance in softwares are due to various reasons and can be addressed using profiling, log parsing, call tracing, etc., which all identify the cause but most of them do not infer why these events occurred. This paper focuses on this aspect and proposes a tool- X-ray which focuses on attributing performance issues to root causes.

X-ray pinpoints why a performance anomaly has occurred using instrument x86 binaries from Pin. For dynamic information flow analysis it modifies ConfAid in terms of taints being applied to every basic block, change the read source to configuration files, weights for taint propagation and mode of execution (live or recorded). Deterministic record and replay is used by X-ray to overcome the overheads of live execution. Custom replay implementations have been proposed addressing various known failures (deadlocks, traps, area of recording). X-ray's design has two phases: Online and Offline. Record and Replay is used by X-ray in online phase to gather performance data, system calls, timestamps, synchronization operations and data traces. Offline phase has two passes namely, Scoping & Performance summarization and a Fast forward heuristic. Scoping is the pass wherein, X-ray extracts basic blocks for each request when processed. It outputs a list of basic blocks denoted by <id, address, count> tuple. Inter-process communication is established using side channels instead of actual input sources. Performance costs are also attributed to all the events through various metrics. In performance summarization pass, costs to each root cause is attributed using Basic, Differential and Multi-Input summarization. Each of which is used and applied as per the users requirements, with all serving different kind of purpose. Fast forward heuristic is used to overcome long replaying delays by checkpointing mechanism- until n% of the taint values have been affected, X-ray remains quiesced.

X-ray has been evaluated on various applications and was able to infer root causes with very less overheads. Examples of some of the root causes in the test application have been explained in detail, along with an explanation how did X-ray infer them. Accuracies for use of different weights for sensitivity analysis, True/False positive rates for differential analysis and X-ray's online overheads have been discussed in detail. The overall overhead of 2.3% is very promising, although X-ray comes with some limitations which have discussed.

Strong Points:

1) X-ray troubleshooting problems without source code, developer support, log messages and traces, and with promising results discussed, makes this paper strong in this aspect.
2) Giving users an option to select time intervals, number of inputs, and methods for request extraction based on their application, greatly increases the scope of this tool in analyzing problems as a common tool for every type of application compromises its goal (here performance summarization).
3) Overheads have been discussed clearly and are considerably less which helps users to consider X-ray for their applications.
4) Leverages both online and offline perks for analyzing software problems.

Weak Points:

1) Without the use system call trace, many problems/errors cannot be inferred, such as in, kernel level calls, multithreading, etc.
2) In Fast forward heuristics, the paper says that it fast forwards execution to a point that is at-least n requests prior, this “n” – how is calculated or determined, what is its ideal value (obviously depending on the application but could have given a range or discussed for a few of them) hasn’t been discussed.
3) X-ray allowing the users to target the analysis scope (for nuanced and severe performance issues) could have been automated, as users might not know what their application could possibly have.