HangFix: Automatically Fixing Software Hang Bugs for Production Cloud Systems

Jingzhu He[1], Ting Dai[2], Xiaohui Gu[1], Guoliang Jin[1]
[1]Department of Computer Science, North Carolina State University
[2] IBM Research

NC STATE UNIVERSITY

Motivation and Problem

- Software hang bugs cause unresponsive or frozen system instead of system crashing.
- Hang bugs are difficult to diagnose and fix due to the lack of debugging information.
- Previous work focuses on generic hang bug detection and little work explores how to fix hang bugs automatically.

Challenges:
- The root causes of hang bugs are diverse.
- Source node is often inaccessible, and it is essential to design application-agnostic bug fixing system.
- Tradeoff between design complexity and fixing coverage.

Contributions:
- We build a new **domain-agnostic**, **byte-code-based** software hang bug fixing system.
- We classify hang bugs into different likely root cause patterns and generate patches.
- We conduct an empirical study of **237** bugs to quantify the generality of root cause patterns and fixing coverage.
- We implement a prototype and conduct experiment on **42** real-world bugs.

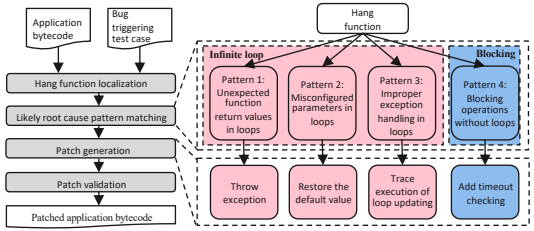
HangFix Framework

Hang function localization: We leverage stack traces to pinpoint the root cause hang function.

Likely root cause pattern matching: We leverage static code analysis to match commonly seen root cause patterns.

Patch generation: We produce patched bytecodes based on the identified root cause patterns.

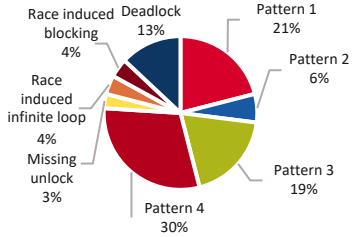
Patch validation: We validate the patches by re-running hang bug detection, hang function localization, and application's regression test suites.

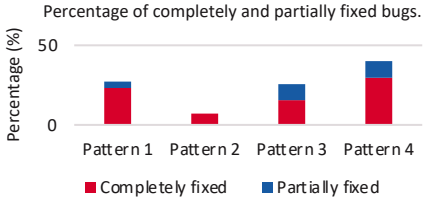


Results Analysis

RQ1: How many bugs fall into the HangFix's four root cause patterns?

- The empirical study results show that **76%** hang bugs fall into HangFix's four root cause patterns.
- For the hang bugs falling into the four root cause patterns, HangFix can fix **75%** of them completely.
- For the hang bugs that cannot be completely fixed, their manual patches contain application-specific functions or it is required to restore system state to fix the bug.





RQ2: How's HangFix's fixing performance, including fixing coverage, fixing time and additional overhead after adopting HangFix's patches?

- Only **14** of the 42 reproduced bugs are resolved with working manual patches.
- The experimental results show that **40** out of 42 reproduced bugs are completely fixed by HangFix.
- The fixing time ranges from 0.7 to 22 seconds.
- The additional performance overhead after adopting HangFix's patch is less than 1%.

Lessons

- HangFix leverages both **dynamic** and **static** analysis techniques.
- HangFix is a new **pattern driven** approach and the patch generation is based on the identified root cause patterns.
- HangFix's design principles and its performance make it **practical** to be applied in production cloud systems.
- HangFix focuses on hang bug **fixing** and it can be integrated with existing hang bug detection tools.

Supported by the NSF funding CNS1513942 and CNS1149445