Extending Big Data Fuzz Testing with Coverage Exploration

Author: Bo van den Berg b.vandenberg-6@student.tudelft.nl
Supervisor: Burcu Özkan b.ozkan@tudelft.nl

02-07-2021

TU Delft

1 Background

DISC Systems - Often used for handling large data. Rare and erroneous corner cases are frequently encountered.

Fuzz Testing - An automated software testing technique: Automatically generate (malformed) inputs and see if breaks the program.

Big Data Testing - Hard to apply traditional fuzzing, because:
1. DISC systems have long latence
2. most code comes from the framework implementation
3. random mutations rarely generate valid data

BigFuzz – Mutation-based fuzzing tool for big data applications. First abstracts the DISC framework to create a smaller application that is suitable for fast test generation.

2 Aim

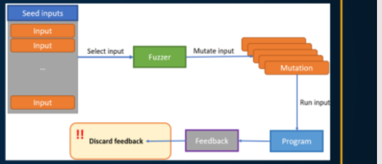
How does input selection based on coverage affect the performance of fuzz testing big data applications?

- How is the coverage information currently used by big data fuzzers?
- How can big data input selection be improved based on coverage information?
- How does the extended fuzzer compare to the current fuzzer?

3 Method

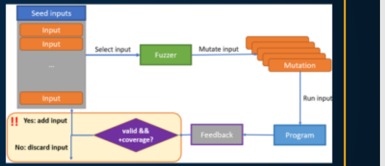
Black-box fuzzer

Does not use coverage information



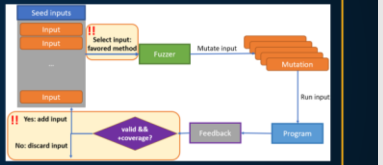
Grey-box fuzzer

Uses coverage information



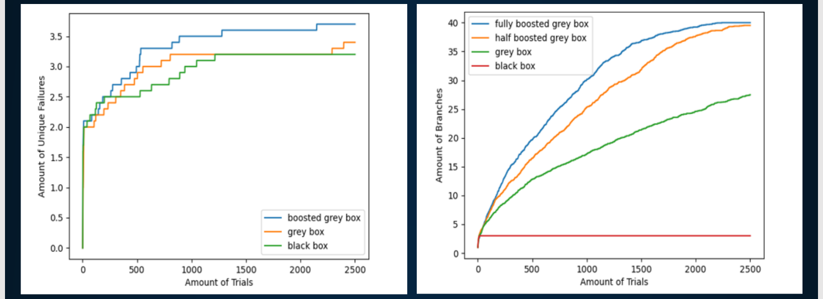
Boosted grey-box fuzzer

Uses coverage information
And favored input selection



4 Results

Comparison of black-box, grey-box and boosted grey-box fuzzer



Error detection

Branch Exploration

5 Conclusion

Both perform at least as good as black-box fuzzing on error detection

Both extensions allow coverage exploration

Boosted grey-box fuzzing is most efficient