

Detected non-deterministic results under various configurations

Open

...

Hi, I have recently been using FlowDroid for an empirical study to detect non-deterministic behaviors in static analyzers. The experiments resulted in discovering some nondeterministic analysis results across multiple runs under various configurations of Flowdroid.

The details of the experimental setup are as below:

- The experiments were conducted on the micro-benchmark DroidBench and a real-world benchmark FossDroid.
- The experiments were conducted under 19 sample configurations which were generated using a 2-way covering array from the configuration space.
- The timeout set for FlowDroid running on each DroidBench program was 5 minutes. For running on each FossDroid program, the timeout was set to 1 hour.
- We ran FlowDroid on each program-configuration combination 5 times and compared the results across 5 runs for detecting non-deterministic behaviors.
- All experiments were conducted in docker containers. The hardware environment is a server with 376GB of RAM and 2 Intel Xeon Gold 5218 16-core [CPUs@2.30GHz](#) running Ubuntu 18.04.

In the end, the experiments detected non-deterministic results on 21 programs. 2 out of these 21 programs were from the DroidBench, and 19 out of 21 were from the FossDroid programs. These results were detected under 16 configurations out of a total of 19 sample configurations.

The attached data is the [detected nondeterministic results from DroidBench and FossDroid](#) and [configuration files](#) (note: the configurations are hash-coded in the detected results, but the actual configuration options and values that each hash code stands for can be found in the attached configuration files.)

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

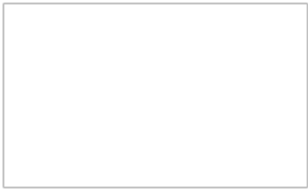
No branches or pull requests

Notifications

Customize

Unsubscribe

You're receiving notifications because you authored the thread.



Contributor

...

Hi,

Thanks for reporting non-deterministic behavior. That is always hard to spot and even harder to find the cause.

Which version did you use? Specifically, did you use a commit after ,

Did you try (or are you planning) bisecting the configuration flags to reduce the configuration to minimal reproducer?

The timeout set for FlowDroid running on each DroidBench program was 5 minutes. For running on each FossDroid program, the timeout was set to 1 hour.

Did you remove all apps with timeouts (either memory or time) from the dataset? FlowDroid reports the intermediate results as soon as one of the timeouts occurs. Due to thread scheduling and load from other processes on a non-isolated system, the number of edges can vary by quite a bit such that results with timeouts should always be regarded as non-deterministic.

Hi, thank you for the prompt reply!

Which version did you use? Specifically, did you use a commit after ?

We were using v2.111.1 for testing. And we are currently rerunning the experiments to see if the nondeterminism still occurs after the fix. We will provide updates once the experiments are done.

Did you remove all apps with timeouts (either memory or time) from the dataset?

Yes, the timeout we chose for both benchmarks allows all runs to complete, so no runs timed out in the previous experiments.



Hi Tim [@timll](#), we have finished the new experiments on the latest version of FlowDroid. The new experiments detected non-deterministic results under 18 configurations out of a total of 19 sample configurations. The new results are attached [here](#)

The new experiments detected nondeterminisms under two more configurations which were not detected in the previous experiments:

config hash `3d42058705611ed0e83612b6dff38a35`

```
--aplength 1 --cgalgo SPARK --dataflowsolver CONTEXTFLOWSENSITIVE --staticmode  
CONTEXTFLOWINSENSITIVE --nostatic --codeelimination PROPAGATECONSTS --implicit  
NONE --callbackanalyzer DEFAULT --maxcallbackspercomponent 5 --maxcallbacksdepth  
1 --enablereflection --pathalgo SOURCESONLY --noexceptions --taintwrapper NONE
```

config hash `d4bda2f41ac052fdcd0890aec02ef8cb`

```
--aplength 5 --cgalgo RTA --nothischainreduction --dataflowsolver  
CONTEXTFLOWSENSITIVE --staticmode CONTEXTFLOWINSENSITIVE --nostatic --  
codeelimination PROPAGATECONSTS --implicit ARRAYONLY --nocallbacks --  
callbackanalyzer FAST --maxcallbackspercomponent 10 --maxcallbacksdepth 1 --  
pathalgo SOURCESONLY --noexceptions --taintwrapper NONE
```

