Deterministic Replay for User-Interactive Android Applications

Vanessa Lin[1,2], Cyril Saade[2], Dr. Ayse K. Coskun[2]

Dougherty Valley High School, 10550 Albion Rd, San Ramon, CA 94582[1]

Boston University, Electrical & Computer Engineering Dept, 8 St. Mary's St, Boston, MA 02215[2]

ABSTRACT

Record-and-replay tools for the Android smartphone platform, such as RERAN and Mosaic, have emerged to enable repeatable execution of real-life applications, which typically include touchscreen input actions or network accesses that easily change application behavior. This replay capability is essential when conducting mobile systems research or designing new apps or devices. Despite the benefits of these techniques in helping with consistent, predictable experimental work on smartphones, such tools often suffer from issues that hinder their ability in successfully replaying the exact actions as inputted. This is because of the non-deterministic behavior and complexity of the behavior of real-life applications. In this project, in order to evaluate RERAN's playback capabilities under different conditions, we installed a set of applications such as games, utilities, and online apps on a Dragon mobile development board. We tested RERAN on the installed applications while filming the application progress for comparison, and replayed the execution sequence of events in each recorded application 3 times. Experiments show that RERAN often fails to replay the original event sequence due to challenges in precisely capturing UI timing, networking, or other application events. We then used this analysis of the RERAN tool to design an Android app and then build a deterministic version of the same app. The longer term aim of this research is to help optimize replay tools so that researchers and developers are able to run the same real-life application trace deterministically on mobile platforms.

INTRODUCTION

- RERAN, a record-and-replay tool, is able to replay 86 of the top rated 100 applications in the Google Play store and is adaptable to any type or model of Android mobile platforms. [1]
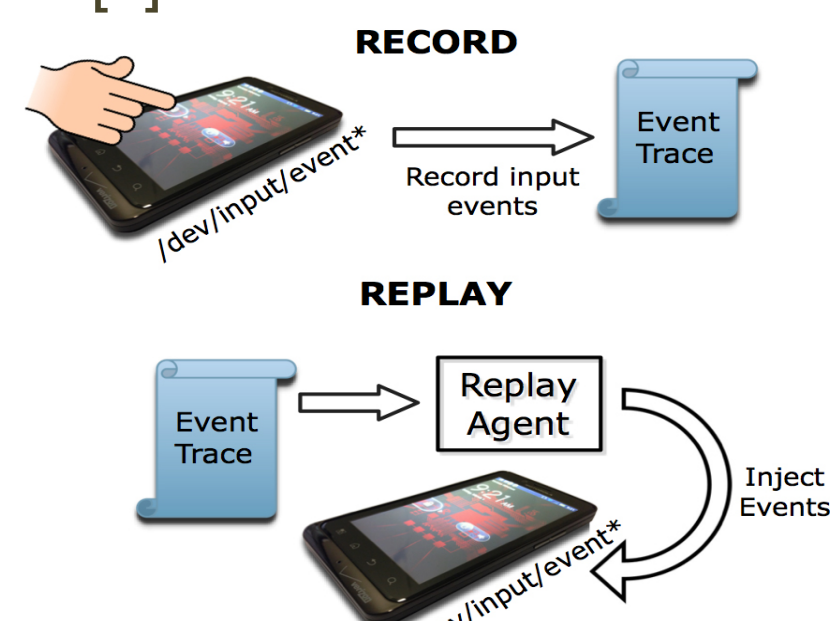
Figure 1. Overview of RERAN [1]

- Designed for developers to test their applications for any bugs; however, RERAN unfortunately fails while running playback depending on certain applications.

- Non-determinism: A property that causes an application to produce different outputs every time it runs

- Determinism: A property in which the application will be constant in producing the same output for every run

- Through testing RERAN with a number of applications, we can better understand the different failures that causes RERAN to fail and the distinctions between deterministic and nondeterministic applications
  - UI Timing: Time intervals between the condition changes and the data transmission signal
  - Networking: Connection between devices, its peripherals through the internet, and application's server
  - Other Application Events: An identifiable occurrence produced by back end implementation that allows the application to perform different functions

- After accumulating data of discrete characteristics, we categorized the comparisons under a criteria of success, pass, or fail

- Pulling from the data of characteristics that make an application deterministic, we built an Android application, which RERAN can test its deterministic and nondeterministic versions

METHODOLOGY

Figure 2. Dragon Board with Snap Dragon [3]

Equipment and Tools
- Qualcomm Dragon Board Development Platform with Snapdragon 800 (APQ 8074)
- RERAN – record and replay program
- Android Studio
- Programming Language : Java

Application Set
- 35 Applications from the Google Play Store: Games (Angry Birds, 2048, etc.), Utilities (Calculator, Calendar, etc.), Online (Facebook, Twitter, etc.)
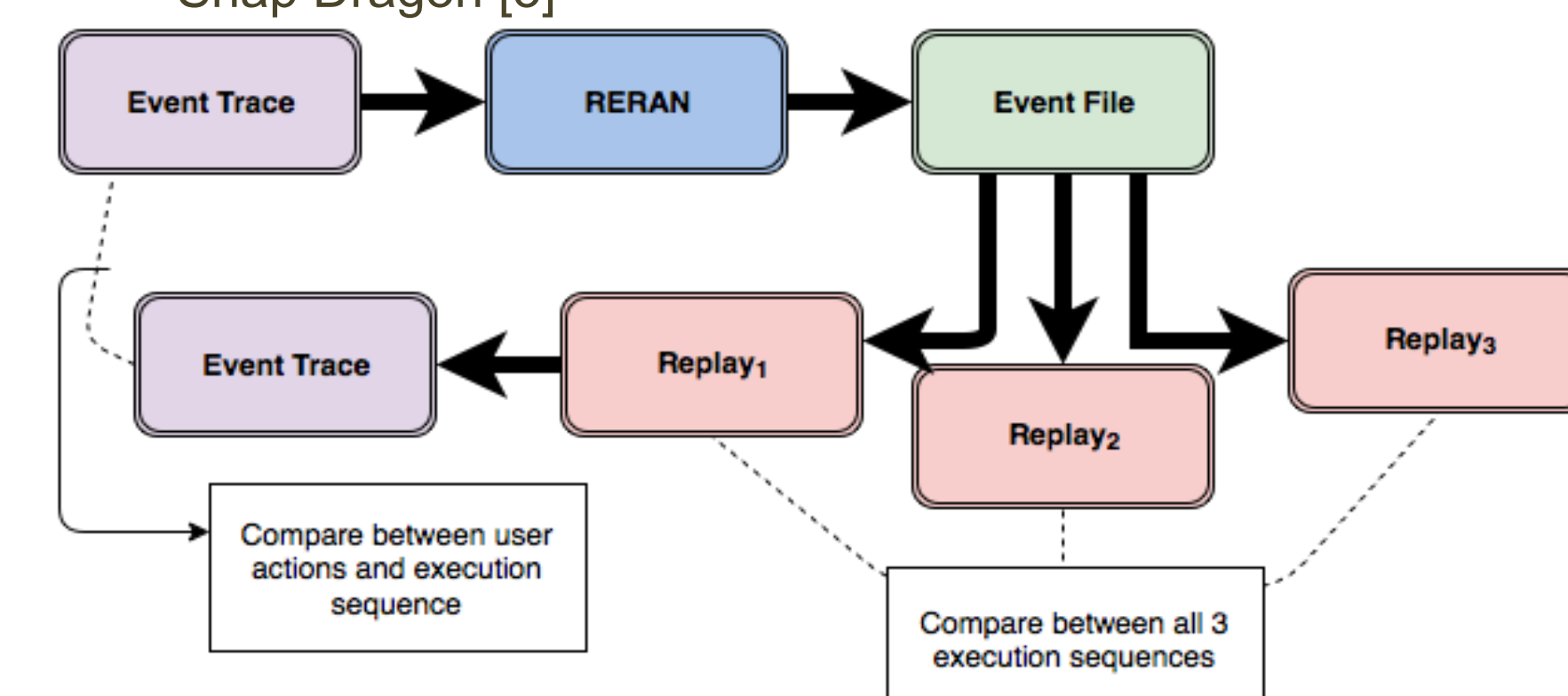

Figure 3. Method of Testing Applications through RERAN

Procedure
- Used scripts to record user actions and push file onto the Dragon Board device
- Used RERAN to produce event files for replays

Android Application
- Created a main activity that sets a button to move to 1 of the 4 corners of the device's screen
- Intent: Button moves to a corner every time user clicks
- Set up a UI timer to create time delay response of button
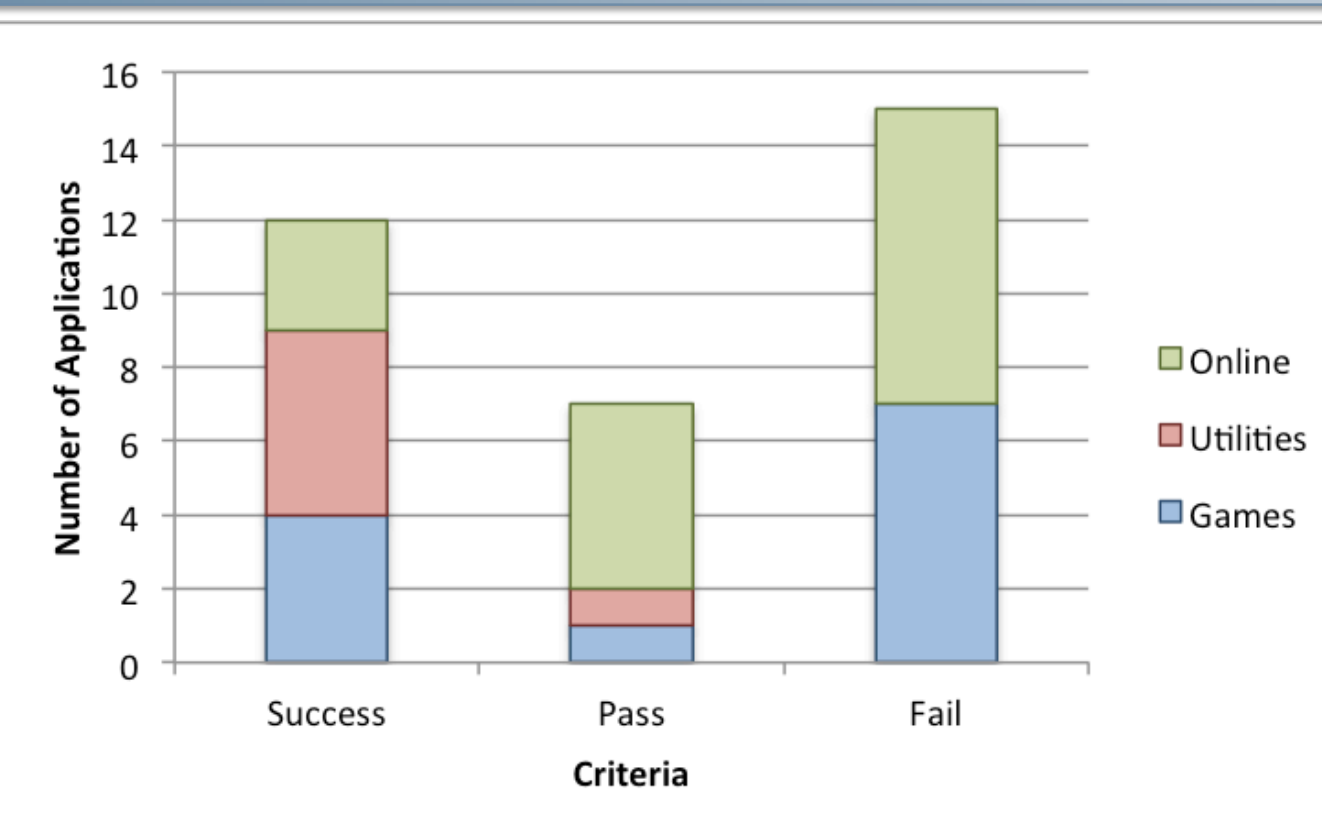
RESULTS

Figure 4. Comparisons under different types of applications
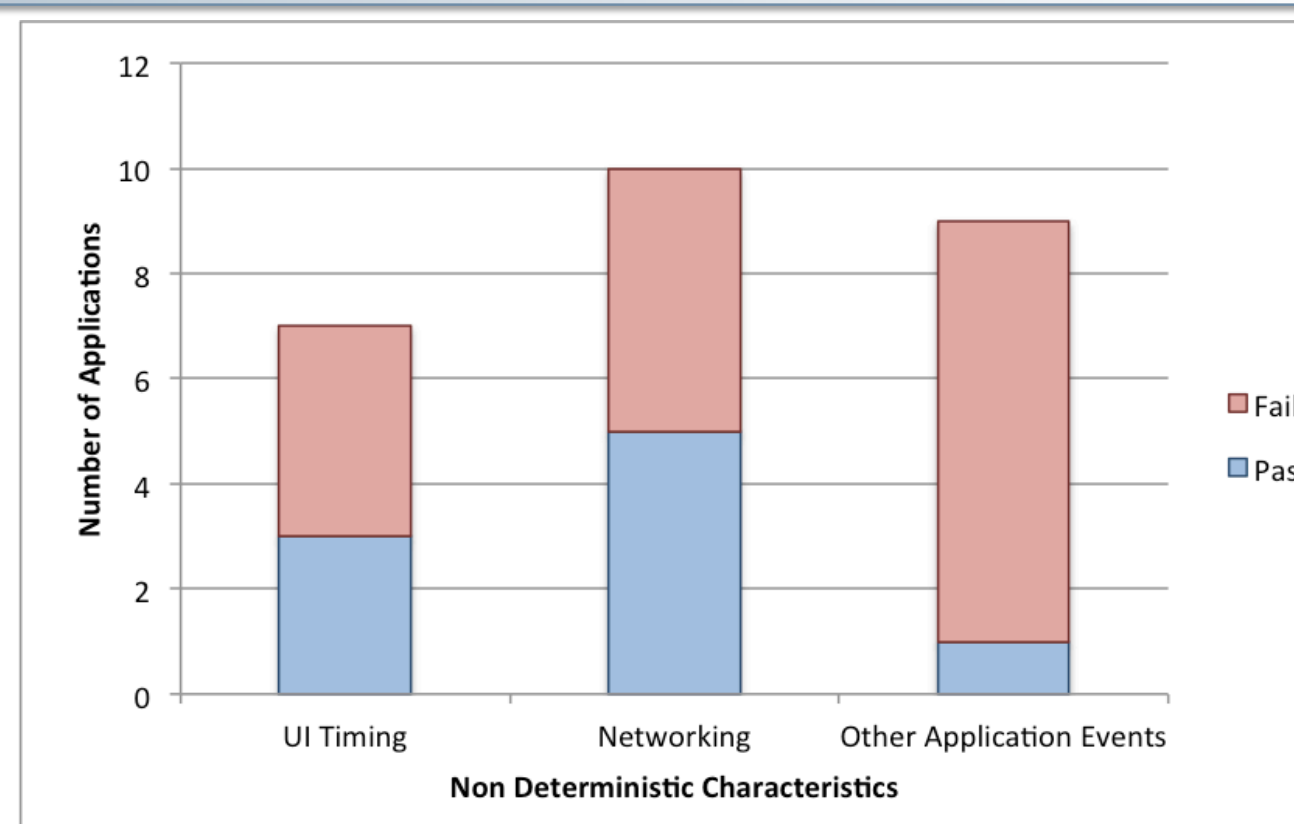

Figure 5. Non-deterministic characteristic divisions over pass and fail

- Large portion of the applications tested failed to perform the exact or similar actions as the original event sequence
- Networking accounts for the most of failures while applications undergo through RERAN
- Networking and UI timing are evenly split between passes and failures accounted among the dataset
- Of other application events, most of the applications associated with the characteristic failed except for 1 application, Angry Birds

Application Names	Type of Application	Success/Pass/Fail	Causes
Unicorn Dash	Game	Fail	Application Events
Evernote	Utilities	Pass	UI Timing
Unblock Me Free	Game	Success	Deterministic
Google Translate	Online	Success	Deterministic (Networking)
Wikipedia	Online	Pass	Networking

Figure 6. Sample Set of 5 Specific Applications

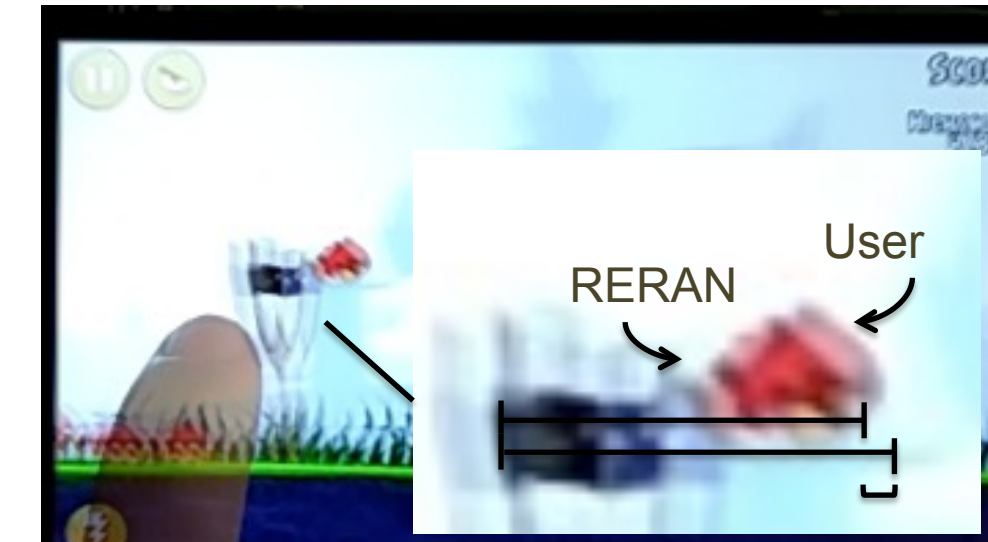Figure 7. Screenshot of User Input at a frame

Figure 8. Screenshot of RERAN at a frame

Figure 9. Overlay of RERAN and User Input

ANDROID APPLICATION

- A simple application in which one can change UI timing from turn to turn from nondeterministic to deterministic
- Upon clicking the button, a time delay of x seconds occurs until the button changes its location

Figure 10. Application

Time Delay (seconds)	Success/Pass/Fail	# of simulated cycles
0.1	Success	4
0.2	Success	4
0.3	Success	4
0.4	Pass	4
0.5	Pass	4
0.6	Fail	3
0.7	Fail	2
0.8	Fail	1.75

Figure 11. Table with Criteria and Increasing Time Delay

*cycle = 1 round of button moving through all 4 corners

DISCUSSION

- RERAN works very well with applications that allow user to hand write words (in contrast to typing) and executed the event sequences exactly like user input actions did

- DragonBoard presented difficulties in using RERAN as it often fell into a boot loop or crashed while recording user input for RERAN or replaying the event sequences
  - Needed to factory reset the device → took some time away – not as large of a dataset as wished

CONCLUSIONS

Through analyzing the causes of failures for RERAN, deterministic applications passed easily through while non-deterministic were largely affected by slow speeds in either the server response or randomization in the implementation of the application itself. By using RERAN and Linux OS, I was able to gain experience in debugging the DragonBoard device and terminal commands. By developing an Android Application, I also learned how to use the Android SDK to create user applications. In the future, researchers can use this analysis and application to further examine ways to optimize record-and-replay tools to account for flops in RERAN due to non-determinism.

REFERENCES

[1] Gomez, L.; Neamtiu, I.; Azim, T.; Millstein, T. *2013 35th International Conference on Software Engineering (ICSE)* **2013**.
[2] Halpern, M.; Zhu, Y.; Peri, R.; Reddi, V. J. *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* **2015**.
[3] *Dragonboard 800 Development Kit*; Intrinsyc.

ACKNOWLEDGMENTS