LTmeter: An App Launching Time Analyzer for Personal Smart Devices

Jehun Lim, Hakbong Kim, Wook Song and Jihong Kim
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {brownrabbit, haknalgae, answer03, jihong}@davinci.snu.ac.kr

***Abstract*—The application launching time is an important design issue for optimizing interaction-oriented devices such as smartphones and tablet PCs. In order to improve the application launching time, it is necessary to define and measure the launching time from the user's perspective. In this paper, we propose a new definition of the application launching time from the user experience perspective. Based on the new definition, we develop an launching time analyzer for Android-based smart devices, called LTmeter. Our measurement results using LTmeter show that many Android applications respond to user inputs before entire user interface components appear on the screen. Our findings also suggest that there is a large opportunity for optimizing app launching times when designing apps because many apps exhibit large differences between the minimum and maximum app launching times analyzed by LTmeter.**

I. Introduction

Personal smart devices such as smartphones and tablet PCs are interaction-oriented devices. For such personal smart devices, the quick response to user inputs is an important design goal, because it often determines the quality of user experience. Among several responsiveness-related performance metrics, the application launching time is one of the most important metrics. For example, a recent user study [1] found that if the response time ranges from 150 milliseconds to 1 second, users can notice a delay. Furthermore, if the response time exceeds 1 second, users feel significant inconvenience in interacting with the apps[1]. Therefore, it is important to reduce the app launching time below these quality threshold values for better user experience.

In order to optimize the app launching time, the first step is to define the launching time adequately and measure the launching time accurately based on the definition. Although the concept of the app launching time is rather straightforward, it is difficult to define it precisely from the user experience view. For example, some researches [2] assumed that the app launching is completed when all the I/O requests for the app launching are serviced. From the user experience perspective, however, this definition might be misleading because many apps will respond to user inputs once a part of its executables are ready to accept user inputs. In order to improve the user experience, we strongly believe that the app launching time should be defined from the user experience perspective. As an alternative definition of the app launching time, we may

[1]In the rest of this paper, we call an "application" by a (more popular) "app".

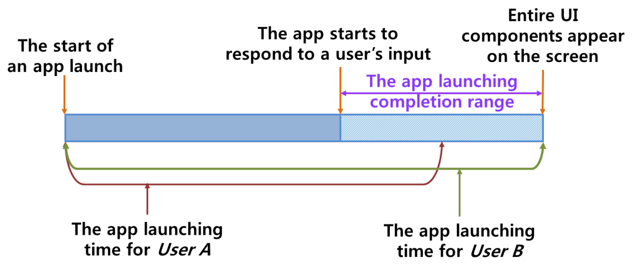
The start of an app launch
The app starts to respond to a user's input
Entire UI components appear on the screen
The app launching completion range
The app launching time for *User A*
The app launching time for *User B*

Fig. 1. Relationship between the app launching completion range and true app launching times

assume that an app was completely launched when all the user interface components are completely shown to the user. Since the user can often interact with the app even though all the components of the user interface are not shown to the user, this definition is still not appropriate.

In this paper, we propose a range that can be regarded as the app launch completion time, which takes account of the user experience perspective. In order to overcome the difficulties of identifying the launching completion from the user experience perspective, we introduce a period concept to recognize the completion of the app launch. We developed an app launching time analyzer for Android-based smart devices, called LT-meter, in order to measure the app launching completion time by exploiting our new definition. Our measurement results show that the user can interact with the app before entire user interface components are drawn on the screen. This means that the app launching completion time can varies within a range between above two times according to various parameters such as the user's app usage characteristics. The wide range implies that there is a large opportunity for optimizing the app launching time.

II. Launching Times for Android Apps

As mentioned above, it is difficult to define clear criteria to identify the completion of the app launching from the user experience perspective. A true launching time is very subjective when it is defined from user's perspective. For example, different users may have different views on when an app is ready for user inputs. Therefore, we introduce two candidate criteria for recognizing the launching completion. One is the minimum elapsed time between the moment the

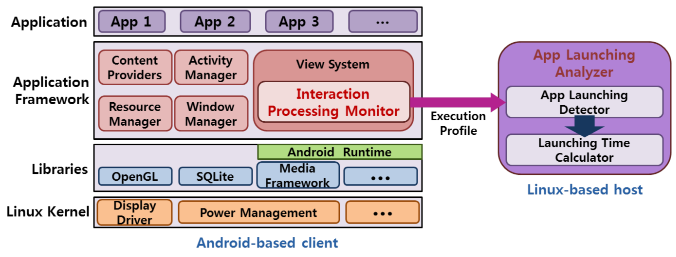

Fig. 2. An overview of LTmeter.

user requests to launch an app and the moment the app can respond to the user input. This criteria can somewhat underestimate the app launching time, because the app can react to the user request though no meaningful user interface components is shown to the user. Therefore, the app launching time based on the response to the user input can be shorter than the true launching time experienced by the user. Another is the elapsed time until entire components of the user interface appear on the screen. This criteria, on the other hand, can overestimate the app launching time, because it is possible for the user to interact with the app in case that the partial user interface which the user is interested in is shown. For these reasons, we do not define the app launching time as a single time, but propose an interval that is guaranteed to include the true app launching time from the user's perspective. We call this interval as the app launching completion range (ALCR).

Figure 1 shows the relationship between true app launching times and the app launching completion range. For example, if *User A* generates user inputs such as touch events when only parts of the user interface with which the user want to interact are visible, the launching time for *User A* is the elapsed time from the app launch to the moment when the app responds to the *User A*'s request (which falls with ALCR). On the other hand, if *User B* waits until all the components of the user interface appear on the screen, the launching time for *User B* is the elapsed time to the moment the entire components are shown to the user. As shown in this example, the launching time perceived by the user can be ranging according to various parameters such as subjective app usage characteristics.

III. Overview of LTmeter

The proposed LTmeter consists of two main components, the interaction processing monitor and the app launching analyzer. Figure 2 shows an overview of LTmeter. The interaction processing monitor in the Android-based client provides meaningful information, which indicates changes of the user interface occurred during processing of the user input, to the app launching analyzer. The app launching analyzer which resides in the Linux-based host detects the completion of the app launching based on the information provided by the monitor and analyzes the launching time. For two components to communicate, the ADB (Android Debug Bridge) via the USB interface is used.

A. Interaction Processing Monitor

As mentioned in Section I, the user interfaces are composed of different responsive interface components and each component is shown to the user at different times. For these reasons, it is required to monitor the changes for each responsive interface component and to provide the execution profile to the app launching analyzer. In order to monitor the changes of the user interface, tracking functions are implemented to the Android platform. In detail, the functions are inserted into the touch event handlers and the draw functions of each component. The tracking functions detect all changes of the responsive user interface and collect the execution profile of each user interface in the foreground apps. The execution profile contains the sequence of all the changes in execution order. The interaction processing monitor provides the execution profile to the app launching analyzer.

B. App Launching Analyzer

The app launching analyzer detects two end points of ALCR as described in Section II using the execution profile from the interaction processing monitor. In order to measure the point when the app starts to respond to a user's input, the analyzer automatically launches an app and generates the user input after waiting certain time from the start of the app launch. Then it identifies whether the app responds to the input by analyzing the information provided by the interaction processing monitor. If the meaningful response is detected, the analyzer then estimates the app's response start time. On the contrary, if the analyzer determines that the app does not respond to the input yet, it repeats above process by increasing the waiting time until the response to the input event occurs. For measurement of the other point when all the components of the user interface are shown, the analyzer automatically launches an app and waits a sufficient time until the app launching completion. Then, based on the execution profile of the user interface, the analyzer classifies whether each user interface corresponds with the user experience, such as responsive buttons, or not, such as advertisings. Finally, the analyzer decides the point when all user interfaces related with the user experience are shown.

IV. Measurement Results

We implemented the interaction processing monitor on the Nexus S smartphone and the app launching analyzer on the Linux-based PC. For experiments, we selected twenty apps as benchmark programs which include apps in groups like browser, communication, system, etc.

Figure 3 shows the measured ALCR values of twenty apps. Our results show that all twenty apps respond to user inputs before entire user interface components appear on the screen. The results also represent that apps vary significantly in their respective launching times from user experience perspective. For example, the ALCR of *app 3* is [273 ms, 3379 ms]. This means that the user can interact with *app 3* after only 273 ms from the start of the app launch. However, the true launching time is longer than this value, which can be up to 3379 ms,

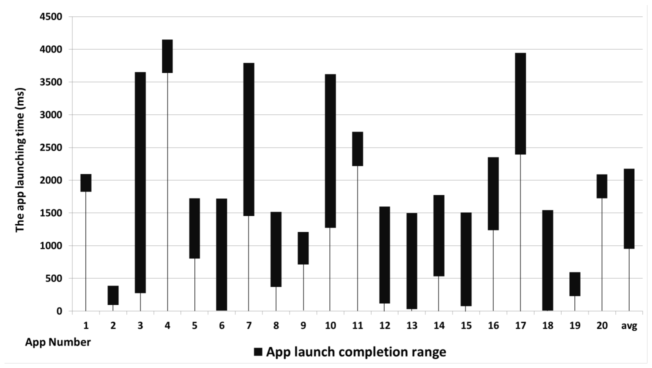

Fig. 3. An app launching completion range of twenty apps

because the user will wait until the meaningful user interface components appear on the screen. A wide interval length suggests that when designing *app 3* from the user experience perspective, an intelligent user interface layout is important so that *app 3* can respond much earlier than the entire user interface is ready. In other words, ALCR value measured by LTmeter can provide meaningful information to optimize the true launching time by adjusting the user interface layout.

V. Conclusion

We have described a design and implementation of LTmeter, an app launching time analyzer. LTmeter automatically analyzes the app launching time from user experience perspective. Experimental results using 20 apps on the Nexus S show that apps differ significantly on their launching time ranges.

The analysis results of LTmeter can be used in improving apps' launching times in several directions. For example, we plan to extend the current LTmeter for computing a true app launch time (which is within the ALCR interval) based on app developer's input on the initial user interface.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 20110020426, No. R33-10095, and No. 2011-0020514). This work was also supported by the Brain Korea 21 Project in 2011. The ICT at Seoul National University and IDEC provided research facilities for this study.

References

[1] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying Interactive User Experience on Thin Clients," *IEEE Computer*, Vol. 39, No. 3, pp. 46-52, Mar. 2006.

[2] Y. Joo, J. Ryu, S. Park, and K. G. Shin, "FAST: Quick Application Launch on Solid-State Drives," *In Proceedings of the 9th USENIX Conference on File and Storage Technologies*, pp. 259-272, Feb. 2011.