# Mobile App Optimizations Using User-Centric Response Time Analysis

Minwoo Kwak Nosub Sung Wook Song Jihong Kim
*Department of Computer Science and Engineering*
*Seoul National University*
{*mwkwak, nssung, wooksong, jihong*}*@davinci.snu.ac.kr*

Smartphones are highly *interaction-oriented* devices [1]. Unlike traditional computers, most of usage scenarios on smartphones involve frequent user interactions in spite of their short time usage. For example, when searching for web pages with an interested keyword, a user first taps a web browser icon and waits for the application to be launched. Then, the user types the keyword into a search box and waits again for the result before starting the next interaction (i.e., selecting one page) with the device. In this case, the user expects the device to react immediately so that these series of processes are carried out with the user's deep concentration. For this reason, the quality of the user experience in the smartphone significantly depends on how *smoothly* and *quickly* the device can react to user's various requests. Therefore, if we can measure response time of each application to handle user's certain request, this information has the potential to be used as a key for improving the user experience. Unfortunately, the existing tools for developers do not provide any information about the response time in user's perspective.

**User-Centric Response Time** In traditional computing systems such as desktop PCs, the response time of a task represents the length of the time interval between the start and the end of the task execution, which we call *computation-centric response time*. However, since smartphone users tend to interact with their devices as soon as they feel that they can without waiting for the completion of the closet computation, in smartphones, the computation-centric response time is not suitable for the user's perspective [2]. Furthurmore, the computation-centric response time of the *user-interactive* application is very hard to be identified because this type of application usually repeats the loops to handle user's input and refreshes the visible interfaces. We propose such a definition of the response time which well reflects the user-perceived response time, a *display-centric response time* which is defined as the time interval from the beginning of new tasks triggered by a user input to the time when all of visible interfaces are drawn.

**Design and Implementation of UROPT** We have implemented an online optimization supporting tool for developers, called **uropt** (**u**ser-centric **r**esponse time **opt**imizer) on Galaxy Nexus smartphone running Android 4.0 (Ice Cream Sandwich). **uropt** is an extend version of **ura** (**u**ser-centric **r**esponse time **a**nalyzer), which is an online measurement framework of user-perceived response time for Android smartphones [3]. In order to give more hints for optimizations, **uropt** provides the longest execution path related to the user experience, *the critical path*. The critical path is an important key which determines the response time estimated by **ura**. The first part of **uropt**, which is called *uropt-target* is implemented in the Dalvik VM interpreter of smartphone and the second part, *uropt-host* is implemented in the host computer. *uropt-target* takes charge of collecting the method call trace to construct the critical path. By tracking spawned threads caused from the user input, *uropt-target* can collects all the method invocations caused by the user input handler (i.e., *onClick()*). It checks whether all the display-related requests have been processed so that the end of the response time can be decided. Then, *uropt-target* sends the gathered data to *uropt-host*, the analyzer. *uropt-host* constructs the visualized critical path and tests if there is any exception that violates the application design guidelines [4]. It also generates a checklist the application developers should verify.

**User-Centric Mobile App Optimizations** The main role of **uropt** is to make the user-centric application optimizations possible. Since **uropt** can clearly identify which part of a task execution affects the user-perceived response time by using a critical path, developers can easily improve the quality of user experience by improving the performance of the critical path. From our experimental results, the response times estimated by **uropt** show 93% of accuracy over manually measured times with about 1% performance degradation. In our experimental evaluations, we observed that many applications did not consider the user-perceived time in a proper fashion. Many popular applications (with more than one million downloads) had the same problem. For example, several applications renew the cached data before the end of response time and others harm the user experience by lazy loading of advertisements. We reproduced these applications using open-source applications and optimized their response times using **uropt**. We were able to reduce the response times of these applications by up to 33%. We plan to extend the current **uropt** in several directions including an automatic response time optimization functions.

## References

[1] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in Smartphone Usage," *In Proceedings of the International Conference on Mobile Systems, Applications, and Services*, 2010.

[2] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying Interactive User Experience on Thin Clients," *IEEE Transactions on Computers*, 2006.

[3] W. Song, N. Sung, B. Chun, and J. Kim, "Reducing Energy Consumption of Smartphones Using User-Perceived Response Time Analysis," *In Proceedings of the International Workshop on Mobile Computing Systems and Applications*, 2014.

[4] Android Open Source Project, "Performance Tips," http://developer.android.com/training/articles/perf-tips.html, 2013.