ECE5658: Operating Systems Design

Paper Reading Assignment

2019712410 Jeon Hyung-Joon

Fine Grained Energy Accounting on Smartphones with Eprof

Summary:

The objective of this paper is to introduce Eprof, the first fine-grained energy profiler for smartphone applications. It analyzes internal energy dissipation of various applications in mobile phones, and it diagnoses where in the apps are causing excessive power consumption. This tool also helps developers to pinpoint source code positions that are causing wakelock bugs. From the actual use of the tool in their case study, the paper proposes that unoptimized I/O operations in smartphone apps are the main cause of excessive power consumption. Then, it suggests refactoring the source code of the applications, which in turn reduce energy costs as much as 20% to 65%.

Strengths and Weaknesses:

The idea of fetching and presenting I/O bundles in internal application execution to the user seems quite innovative. This will certainly help developers in optimizing their code in terms of power efficiency. According to statistics of I/O bundles for a certain application, the developer can refactor the code to reduce power consumption.

Currently, however, Eprof can mostly provide different power consumption scales on the basis of four levels of actions: processes, sub-routines, system calls, and threads. It cannot give out results in terms of exactly which application or module is causing the problem. This makes it difficult to diagnose power consumption issues when multiple modules or applications are executed, especially when deep interactions between the processes are technically mandatory.

Suggestions for Improvement:

Eprof would be a better application when it is possible to filter profiling according to application requirements and context. A typical application can have multiple threads related to it, each of which can deal with various devices within a machine, and the developer may gain convenience when being able to view the inner details on this. For instance, by specifically being able to view the system call power consumption in terms of GPU I/O operation, developers can get the gist on how to improve their code for rendering graphic components needed in their application. By adding such filtering options, I convince the Eprof can be improved to the needs of application developers craving for power efficiency.