Benchmarking Smart Phones

Daniel T. Wagner

Modern smart phones are complex communication and computing devices that compete in a heterogeneous and constantly changing market. Keeping an overview and objectively comparing devices is more difficult than ever. Benchmarks establish a common ground on which it is possible to compare systems with one another. In the past we have seen benchmarks act as drivers of innovation [1]. For both software and hardware benchmarks it is of crucial importance to reflect real-world workloads [2]. Today we can see many specialized benchmarks for different areas such as SPECweb [5] and MediaBench [3]. Some benchmarks for Android smart phones have already begun to emerge, such as Quadrant [6] and the Java-based CaffeineMark [7].

Many system characteristics of smart phones can be measured by a variety of benchmarks. Examples include processing power, storage architecture, battery capacity or radio sensitivity. It would be easy to borrow an existing benchmark from a different domain and apply it to smart phones. However, benchmarks have previously been shown to measure unrepresentative workloads when employed across domains [4].

Relevance Standby time and call time seem to have established themselves as the industry standard for benchmarking mobile phone battery performance across devices. This was appropriate some years ago when mobile phones could only make phone calls and send text messages but seems largely obsolete now. Current smart phones feature powerful processors, large displays and lots of storage space and are marketed as all-rounders that run 3D games, store our entire music collection and stream videos off the internet.

Technology enthusiasts rely on technical specifications to estimate if a smart phone suits their needs. Most customers, however, will not understand the details of operating systems, processor types and speeds or know about memory requirements. A good benchmark could serve as a guide to users who are shopping for a new device as well as the industry by highlighting areas that could benefit from further development.

If we want to build *relevant* benchmarks that are representative of how people use smart phones in the real world we need to first gather appropriate ground truth.

Characterizing a representative workload To discover how people use their phones in detail I have developed *Device Analyzer*, an application that runs on Android smart phones and unobtrusively collects detailed usage statistics. Periodic uploads of this anonymized data to a central server allow us to collect a large corpus of detailed anonymous usage data.

Knowing how a broad range of people use their smart phones will allow us to extract prototypical behavior for different groups of users with their idiosyncratic usage patterns. Building on this dataset it will be possible to create a benchmark that reflects the real world usage of mobile phones today. By identifying trendsetters in our population it might also be possible to replace speculations about future use with concrete data based on real people.

Further Questions The dataset that *Device Analyzer* gathers may also be helpful in addressing the following problems that extend beyond benchmarking of smart phones, some of which can be thought of as *personal benchmarks* that measure user rather than system behavior.

Users could rely on their own gathered data to guide the purchase of their next phone or when changing phone tariffs. Due to its embedded tacit information this dataset is also a good candidate for exploratory analysis of large user-centered datasets . Automatic inference of relationship or other personally relevant data could help us understand individual usage patterns and eleminate the need for error-prone self-reporting. It would also be possible to verify existing models of human behavior such as movement and communication patterns as well as interruptibility based on sensed context. Furthermore, it would be possible to answer privacy questions based on a big dataset of real users that care about real data such as their phone logs or their GSM cell location.

Summary Device Analyzer collects a large corpus of detailed anonymous usage data from Android smart phones. Based on this data I can then devise benchmarks for smart phones that are grounded in real user behavior. Such benchmarks could benefit users as a decision helper when shopping for a new mobile phone and allow applications to specify a minimum required rating to provide a good user experience.

One mechanism for evaluating my benchmarks is to have independent experts analyze and rank a range of smart phones according to suitability for different tasks, comparing their ranking with the benchmark results. Additionally, a user study where participants compare smart phones could show whether their experience correlates with benchmark results.

Acknowledgements

I want to thank Andrew Rice, Alastair Beresford and Andy Hopper for their support, insight and helpful guidance. This work was supported by the University of Cambridge Computer Laboratory Premium Studentship scheme, a Google focussed research award and the EPSRC Standard Research Grant EP/P505445/1.

References

[1] A. B. Brown and M. I. Seltzer. Operating system benchmarking in the wake of lmbench. *ACM SIGMETRICS Performance Evaluation Review*, 25(1):214–224, June 1997.

[2] J. Dongarra, J. Martin, and J. Vorlton. Computer benchmarking: paths and pitfalls. *IEEE Spectrum*, 24(7):38–43, July 1987.

[3] C. Lee, M. Potkonjak, and W. H. Mangione-Smith. MediaBench: a tool for evaluating and synthesizing multimedia and communicatons systems. *MICRO 30 Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture*, pages 330–335, Dec. 1997.

[4] A. M. G. Maynard, C. M. Donnelly, and B. R. Olszewski. Contrasting characteristics and cache performance of technical and multi-user commercial workloads. *ACM SIGOPS Operating Systems Review*, 28(5):145–156, Dec. 1994.

[5] http://www.spec.org/web2005

[6] http://www.aurorasoftworks.com/products/quadrant

[7] http://flexycore.com/benchmark-database-access.html