

SMARTSTORAGE: STORAGE-AWARE SMARTPHONE ENERGY SAVINGS

WILLIAM & MARY

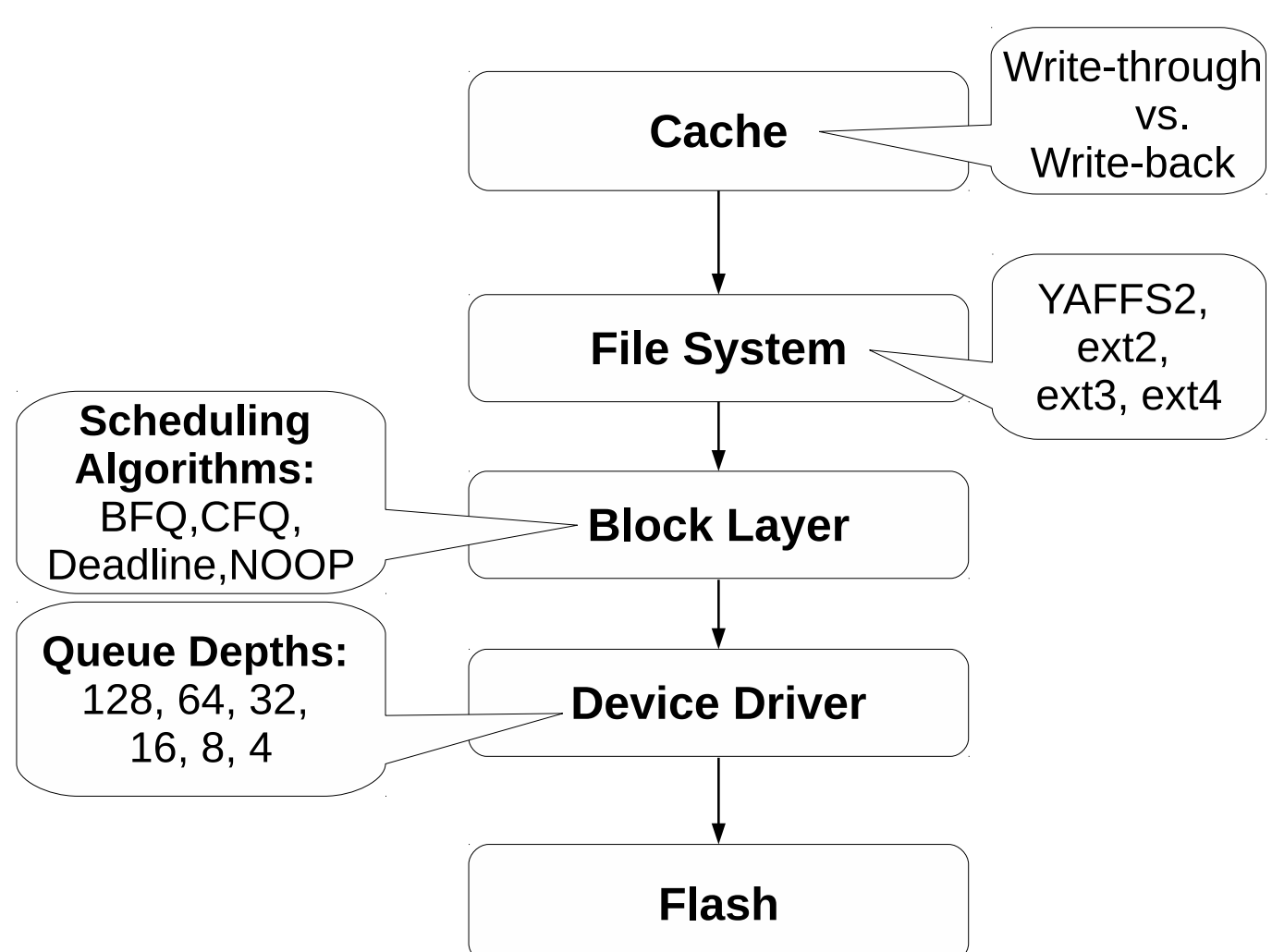
DAVID T. NGUYEN. COLLEGE OF WILLIAM & MARY. {DNGUYEN}@CS.WM.EDU

INTRODUCTION

Continual advancements in the technology of smartphones have become an important, if not essential, aspect of our daily life. However, a common complaint among smartphone owners is the poor battery life. To many such users, being required to charge the smartphone after a single day of moderate usage is unacceptable.

In this project, we investigate the direct impact of smartphone storage techniques on total energy consumption and we answer two key research questions: *How does storage affect smartphone power efficiency?* and *How can we optimize smartphone storage in order to save more energy?*

BACKGROUND



The figure illustrates the main kernel components on the I/O path. We focus on the block layer and device driver.

At the block layer, the main work is scheduling I/O requests from above and sending them down to the device driver. The Linux kernels on Android smartphones offer 4 scheduling algorithms: BFQ, CFQ, Deadline, and Noop. In some Android phones, the default fixed algorithm is BFQ (e.g., Nexus 1), others use CFQ (e.g., Nexus 4). The device driver has a parameter called *queue depth*, defined as the number of pending I/O requests for storage. The queue depth is fixed to different values depending on vendors, usually 128.

REFERENCES

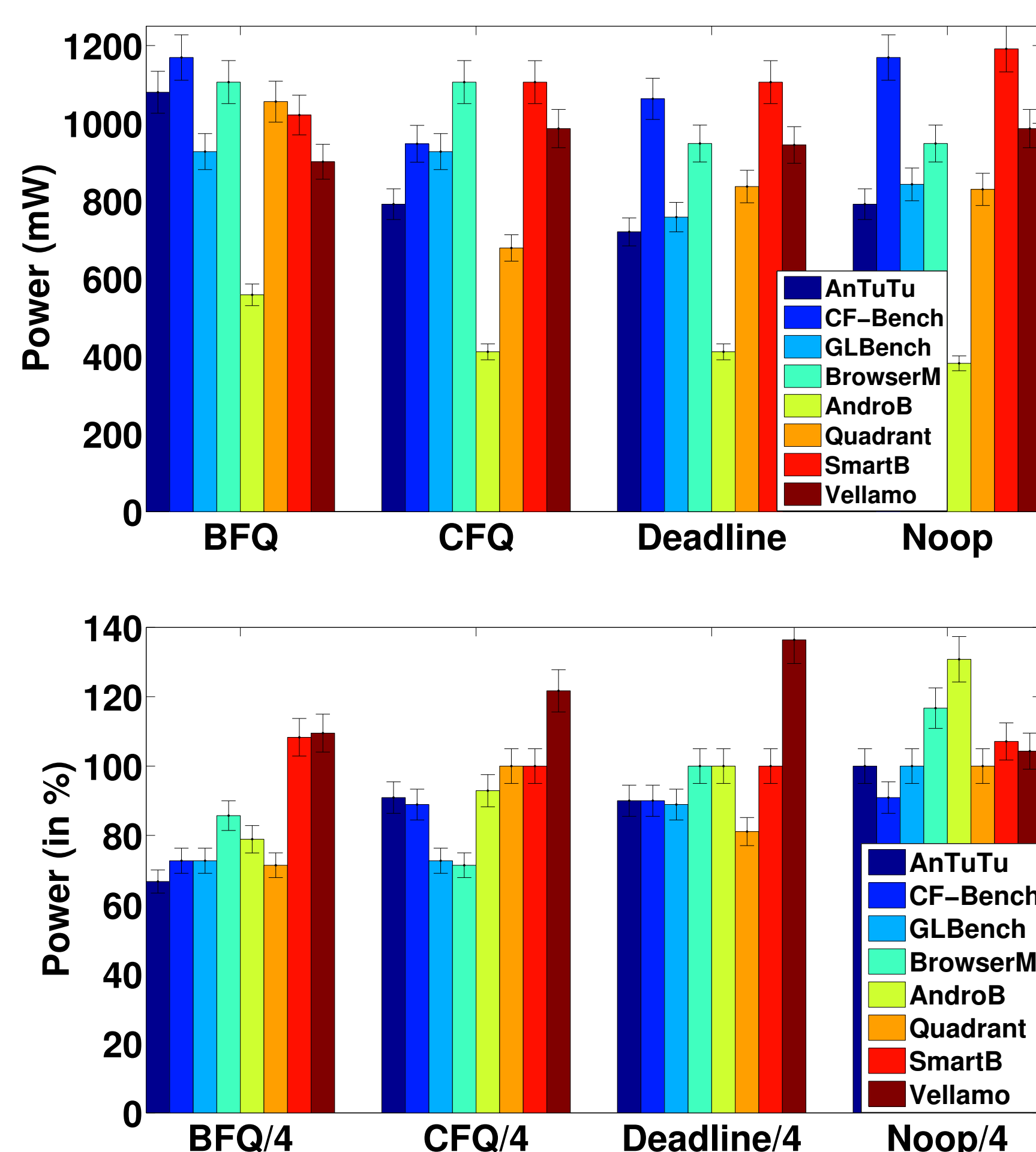
- [1] D. Nguyen, G. Zhou, X. Qi, G. Peng, J. Zhao, T. Nguyen, D. Le. Storage-aware Smartphone Energy Savings. In *Proceedings of ACM UbiComp '13, Zurich, Switzerland.*

CONTRIBUTIONS

We evaluate power degradation at several layers of block I/O, focusing on the block layer and device driver. At each layer, we investigate the amount of energy that can be saved, and use that to design and implement a prototype with optimal energy savings named SmartStorage.

The system tracks the run-time I/O pattern of a smartphone that is then matched with the closest pattern from the benchmark table. After having obtained the optimal parameters, it dynamically configures storage parameters to reduce energy consumption. The evaluation on the 20 top applications shows 23% to 52% energy savings.

MEASUREMENTS

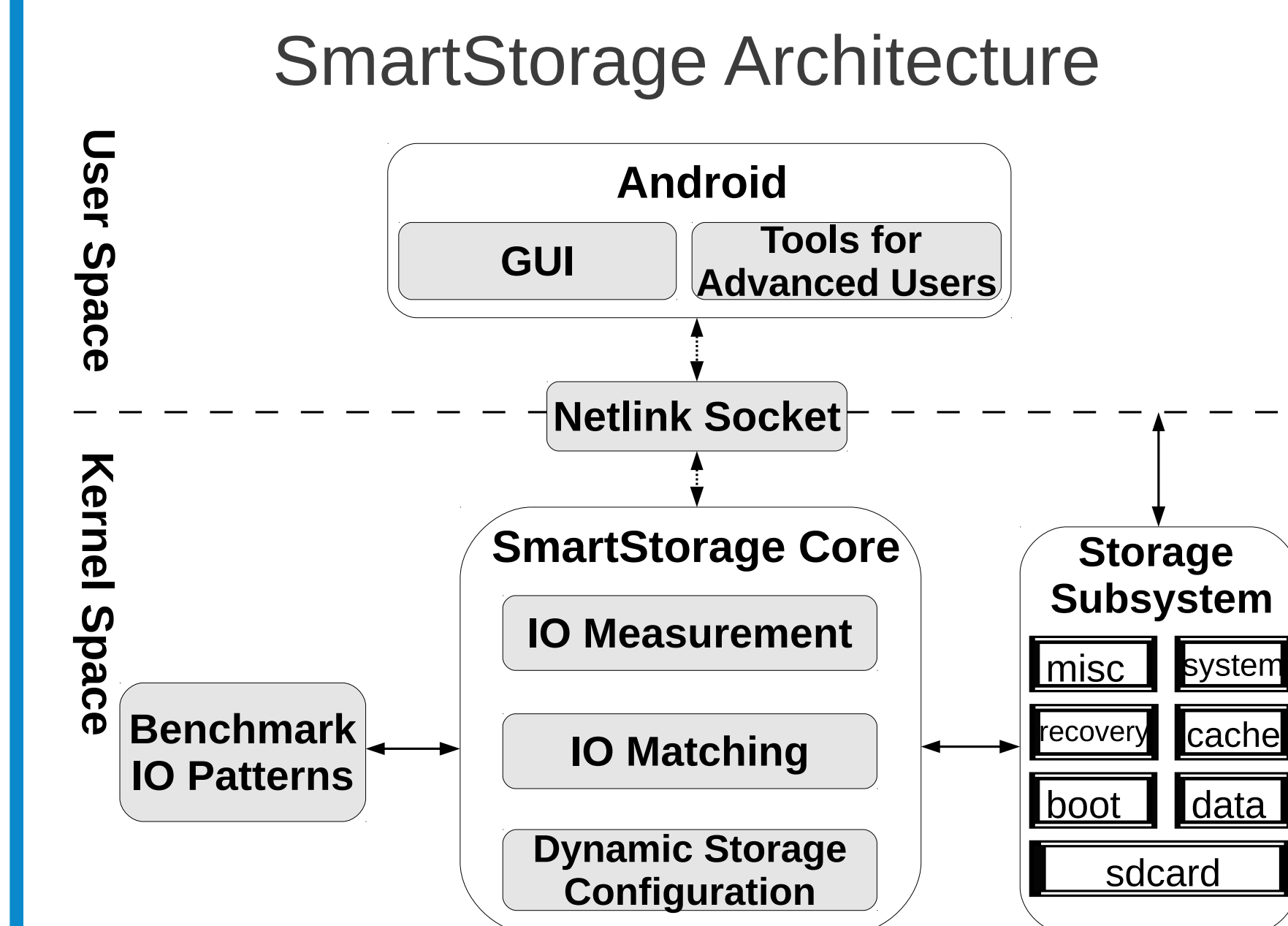


Eight benchmarks are run on Nexus 1 with different configurations, and power levels are recorded. The top figure is for the default depth (128), the bottom one compares results with those of the default configuration. Benchmarks achieve significant savings with optimal configurations!

FUTURE WORK

We plan to investigate the remaining components on the I/O path to understand and explore additional energy savings. Additionally, we want

SMARTSTORAGE DESIGN



SmartStorage Core has three main functionalities. First, it obtains phone's run-time I/O pattern. Next, it gets a combination of a scheduling algorithm and queue depth with optimal power efficiency. Finally, it configures this combination in the block layer and device driver.

The **I/O Measurement** module obtains the phone's run-time I/O pattern via *blktrace*, a block layer tracing utility. After gathering I/Os for a predefined time period, it calculates the run-time I/O pattern that is later used for matching with benchmark I/O patterns. The I/O pattern consists

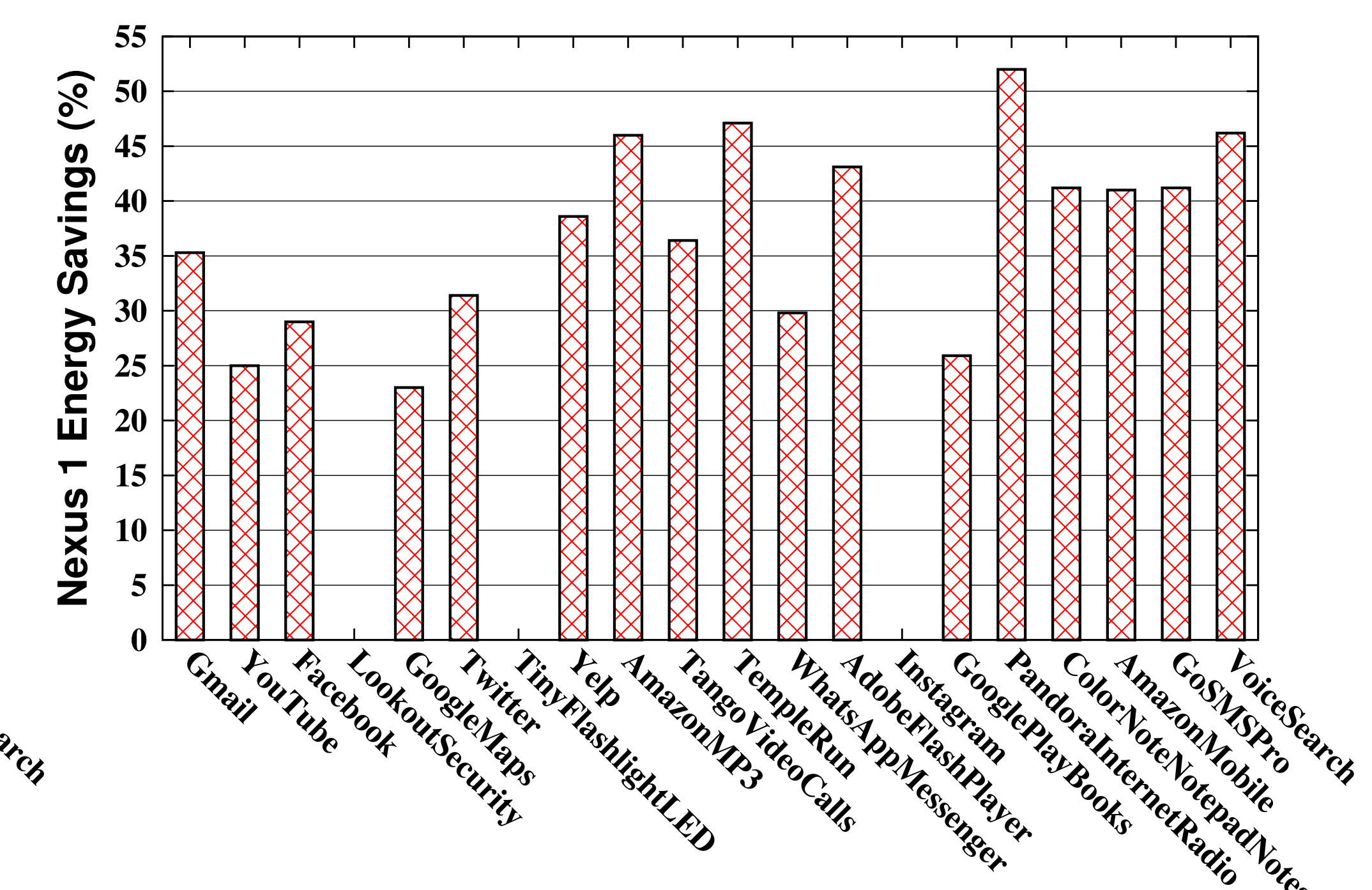
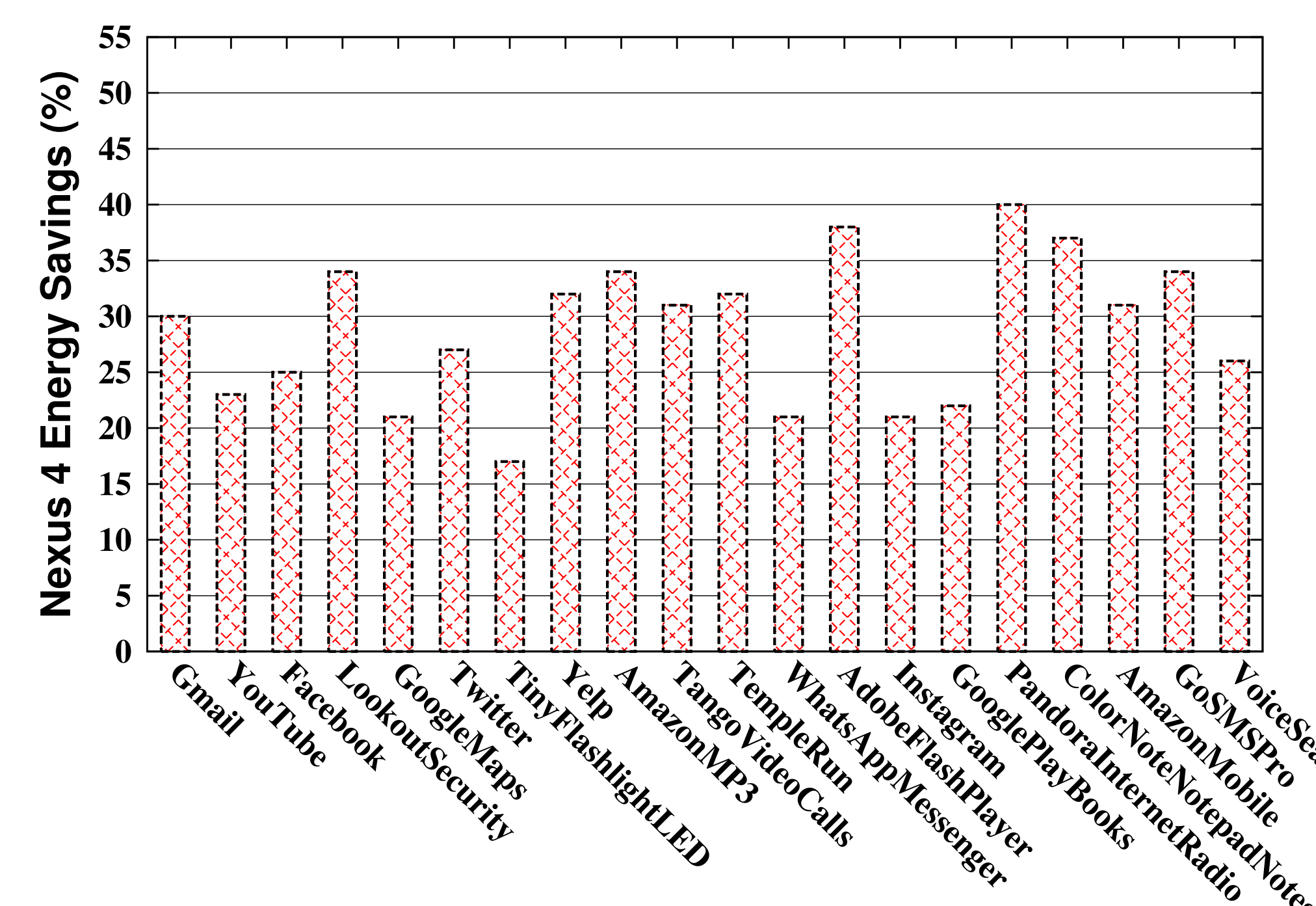
of rates of each I/O type per second.

I/O Matching is performed after having acquired the phone's run-time I/O pattern. The phone's pattern is matched to a benchmark with the most similar I/O pattern. Since each benchmark has a combination of a scheduling algorithm and queue depth with optimal consumption, that combination is returned as a result of this phase. Having all types of I/Os coming to storage, simple intuition says that what matters most are the total number of completed reads and number of completed writes in a given interval. We expand the simple intuition, and perform matching based on proportions of rates of completed reads and completed writes.

The **Dynamic Storage Configuration** module sets the optimal scheduling algorithm in the block layer scheduler, and the optimal queue depth in the device driver.

We implement our system on the Nexus 1 and Nexus 4 Android-based smartphones. SmartStorage works naturally as a background service to save energy, without any need of interaction from users.

PERFORMANCE EVALUATION



ACKNOWLEDGEMENT

The author would like to thank Dr. Gang Zhou, Xin Qi, Ge Peng, Jianing Zhao, Tommy Nguyen, and Dr. Duy Le for their contributions.

to research a machine learning method for matching I/O patterns.