Thermal-Aware Overclocking for Smartphones

Guru Prasad Srinivasa
Computer Science and Engineering Dept.
University at Buffalo
gurupras@buffalo.edu

David Werner, Mark Hempstead
Electrical and Computer Engineering Dept.
Tufts University
{david.werner,mark.hempstead}@tufts.edu

Geoffrey Challen
Computer Science Dept.
University of Illinois
challen@illinois.edu

***Abstract*—Heat dissipation and battery life continue to be major challenges for smartphones. Smartphones seldom spend time at their highest performance points due to thermal concerns and frequently undergo thermal-throttling, where performance is limited while the device cools. While overclocking and computational sprinting can be used to increase system performance, these techniques have not been evaluated on smartphones because they exacerbate both heat dissipation and battery life.**

In recent years, certain machine-learning workloads such as object detection and speech recognition have been moving away from the cloud and towards the edge. These workloads are short and user-facing making them excellent candidates for sprinting. To successfully overclock any workload however, any applied technique must ensure that it avoids forcing the system to throttle.

In this paper, we describe and evaluate a system that accurately predicts the impact workloads have on the thermal state of a smartphone. Thus, the system can determine whether overclocking a specific workload will result in thermal-throttling. We show that our system's careful application of overclocking can decrease the latency of certain user-facing workloads by up to 18%.

I. Introduction

Overclocking is the process of operating a chip at a frequency above its base frequency to temporarily improve performance at the cost of increased power consumption and heat generation. If the device gets too hot, it undergoes thermal-throttling, where performance is reduced in order to allow the device to cool down. While the effects of thermal-throttling in conventional computers can be mitigated through the use of active cooling, case design and controlled thermal environments, smartphones frustrate all of these strategies as they are operated in uncontrolled thermal environments and lack active cooling.

Although predicting ambient temperature, thermal modeling and overclocking have been extensively studied in the past, to the best of our knowledge, we are the first to investigate thermal-aware overclocking of smartphones. In this work, we develop 1) a mechanism to estimate the ambient temperature that a smartphone is in and 2) a predictive thermal model for the smartphone, without the need for additional hardware support. With these components, we develop a system called THERMACLOCK that enables the phone to preemptively determine if the phone has the thermal headroom required to overclock a given workload. We evaluate THERMACLOCK on machine-learning inference workloads, demonstrating the ability of the developed system to improve system performance while drastically reducing the risk of thermal throttling.

II. Motivation: Smartphone Thermal Properties

Due to the form factor and lack of active cooling in phones, they are very prone to overheating. Even at stock settings, a phone will eventually thermally-throttle when executing intense workloads. Thermal throttling is so pervasive that PhoneLab—a 150 device smartphone testbed [1]—throttling occurred in over 80% of all device-days.

For one workload, we found that throttling occurred within seconds when ambient temperature was 35°C and in just two minutes when ambient was 10°C. Critically, when ambient temperature was around room temperature, throttling occurred within just 20 seconds. Therefore, only short workloads are good candidates for overclocking. In this work, we evaluate a group of Machine-Learning workloads, as they are now appearing on edge devices and are only a few seconds in duration [2].

If a system is naively overclocked under the wrong conditions—e.g. high ambient temperature—device performance can actually *decrease* relative to the stock system, as demonstrated in Fig. 1. In this paper, situations where overclocking can be done without resulting in thermal events are referred to as *Overclock-Safe*.

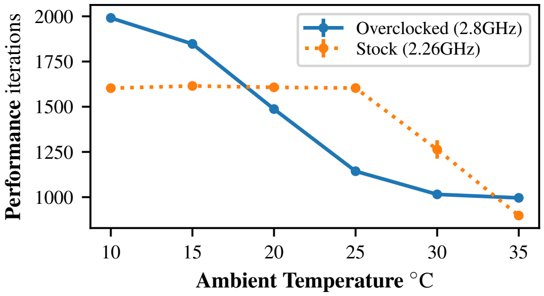

Fig. 1: **Performance Comparison of Overclock vs Stock**. The overclocked system *only* outperforms stock at low temperatures where there is increased thermal-headroom and at high temperatures when even the stock system encounters thermal events.

III. Design & Methodology

The system developed in this paper aims to be flexible, lightweight, and thermal-aware while predicting and managing the power and thermal state of smartphones. It is comprised of an ambient temperature estimator and a thermal model.

1) Estimating Ambient Temperature: The ambient temperature estimator runs a short CPU intensive workload to put the phone into a consistent thermal state and observes the rate at which the phone cools down. The correlation between the ambient temperature and the initial/final temperature of the phone is then used to estimate the ambient temperature; a clear linear relationship was observed. Under cross validation, our predictive model has an average accuracy of 90% with a standard deviation of 0.05.

2) Thermal Model: Our simple but powerful thermal model is designed to be general enough to be adapted to any smartphone that has a thermal sensor. Our model can be used in one of two modes: *thermal-estimator* mode, where the temperature is estimated for a given power trace, and *power-estimator* mode, where power is estimated for a given thermal trace. To minimize computational requirements of our on-line model, we constructed a simple 2-stage RC model, leveraging the well known mathematical similarity between the equations of heat diffusion and those of RC networks [3], [4].

The precise values of the *R* and *C* components in this model are parameterizable and need to be calibrated in order to achieve accurate results. As a case study, we developed a thermal model for a Google Nexus 5, which is built on the Qualcomm Snapdragon 800 SoC and has 4 cores capable of running at a maximum frequency of 2.26 GHz. We chose R and C values using a simple multivariate solver that aimed to minimize RMS errors between the predicted temperature and ground-truth temperature measurements, as shown in Fig. 2.

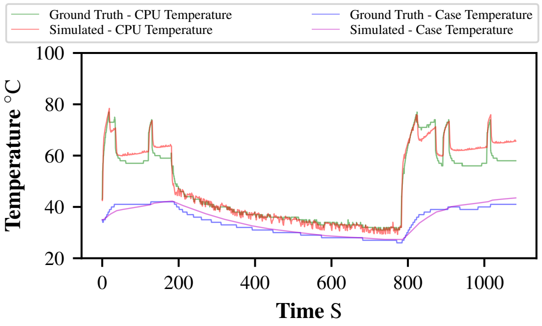

Fig. 2: Predicted Temperatures Using our simulated RC Model

3) Workload Power Traces: Smartphone devices lack the precise run-time power consumption information needed for our thermal model. In this work, we propose a novel method of obtaining representative power traces using only temperature sensors readily available on almost all smartphones. Given this thermal trace, the power is then estimated by leveraging the thermal model using the *power-estimator* mode, which allows power estimates to be derived from thermal traces. Furthermore, we account for the dependence of leakage power on temperature by scaling the total power based on observed power consumption at various temperatures. For example, we observed 30% more power is consumed at 40°C than at 10°C.

We use the benchmarking technique ACCUBENCH as well as the THERMABOX thermal environment as described in [5] to ensure that multiple iterations of our experiments performed consistently. In short, ACCUBENCH technique does the following: 1) warm up the CPU for fixed time, 2) sit idle until CPU reports target temperature, and 3) run the workload.

IV. Evaluation

We evaluate our thermal model by predicting whether a system is *Overclock-Safe* in the ideal conditions where power-data is obtained from a Monsoon power monitor [6] and ambient temperature was obtained from the THERMABOX controller. Under these conditions, overall accuracy was 87.1%. Table I shows the confusion matrix for the model, which breaks down the overall accuracy into its various components.

		Overclock-safe (predicted)	
		Yes	No
Overclock-safe	Yes	5,582	2,796
	No	1,835	25,675

TABLE I: Confusion matrix for the model that predicts if a workload is *overclock-safe* given ideal inputs. Optimal cases are shaded in green, missed overclocking opportunities in yellow, and performance degradation in red.

		Target Temp					
		10	15	20	25	30	35
Source Temp	10		76.5	99.5	100	100	100
	15	77		98.5	100	100	100
	20	58.2	72.2		100	100	100
	25	56.7	75.7	100		100	100
	30	55.9	66.1	89.0	100		100
	35	70	53	62.7	89.5	100	

TABLE II: **Accuracy of *Overclock-Safe* prediction for Image Classification Workload**. System has a higher accuracy of predicting from a lower temperature to a higher temperature for this workload.

1) End-to-End Evaluation: We evaluate THERMACLOCK using estimated ambient temperature and estimated power consumption on three edge-workloads inspired by existing AI Benchmarks [7]. The workloads include image classification [8], object detection [9], [10], and video upscaling [11]. We show the results when the power trace is obtained at a *different* ambient temperature (*target temperature*) than it was profiled under (*source temperature*).

Table II summarizes the results from the classification workload, with similar trends being observed for the others. When going from 25°C to 10°C, prediction accuracy for experiments that resulted in thermal-throttling is 100% (100 experiments) but only 56% for the remaining 300 *Overclock-Safe* experiments. As a result, accuracy was only 56.7% since many overclocking opportunities were missed. On average, overall performance was improved by 18% (image classification and object detection) and 12% (upscaling).

V. Conclusion

In this work, we introduced a thermal-aware approach for overclocking smartphones using a novel thermal model and ambient temperature estimator that together leverage existing CPU temperature sensors. We developed a system—THERMACLOCK—that can 1) estimate ambient temperature within 2°C, 2) profile workloads to obtain power estimates, and 3) accurately identify *Overclock-Safe* situations.

Across different edge-workloads, our system correctly identified the thermal conditions and makes the right choices 87.1% of the time when given accurate measurements. Of the remaining 12.9%, 7.8% were missed overclocking opportunities and 5.1% are wrong predictions. When only considering the *Overclock-Safe* opportunities, THERMACLOCK improved user-experience 66.6% of the time with the user experiencing no change in the remaining 33.4%, resulting in up to 18% performance increase.

References

[1] A. Nandugudi, A. Maiti, T. Ki, F. Bulut, M. Demirbas, T. Kosar, C. Qiao, S. Y. Ko, and G. Challen, "Phonelab: A large programmable smartphone testbed," in *Proceedings of First International Workshop on Sensing and Big Data Mining*. ACM, 2013, pp. 1–6.

[2] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia *et al.*, "Machine learning at facebook: Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 331–344.

[3] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, 2006.

[4] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3d-ice: A compact thermal model for early-stage design of liquid-cooled ics," *IEEE Transactions on Computers*, vol. 63, no. 10, pp. 2576–2589, 2014.

[5] G. P. Srinivasa, S. Haseley, M. Hempstead, and G. Challen, "Quantifying process variations in smartphones," in *2018 IEEE International Symposium on Performance Analysis of Systems and Software*. IEEE, 2018.

[6] "Monsoon power monitor," https://goo.gl/rlizsj.

[7] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on android smartphones," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[11] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.