Separated By Birth: Hidden Differences Between Seemingly-Identical Smartphone CPUs

Guru Prasad Srinivasa*, Rizwana Begum†, Scott Haseley*, Mark Hempstead‡, and Geoffrey Challen*

*: University at Buffalo, †: Drexel University, ‡: Tufts University

*: {gurupras,shaseley,challen}@buffalo.edu †: rb639@drexel.edu ‡: mark.hempstead@tufts.edu

ABSTRACT

Smartphone consumers, app developers, and even mobile systems researchers operate under the assumption that performance differences between identical smartphones should be small. Consumers pick a model to purchase and don't consider that the specific device they leave the store with may vary quite dramatically from the identical models it sat next to on the shelf. App rating systems typically collect the model from reviewers, but not more detailed information–again, assuming that all instances of a particular model perform similarly. Even mobile systems researchers will conduct studies using small numbers of devices that fail to account or control for inherent differences between identical phones.

Unfortunately seemingly-identical smartphones can in fact have very different performance characteristics. Note that we are not referring to differences in battery or Flash performance caused over time by wear. Inherent differences would separate two brand-new phones still in the original packaging. Our experiments show up to 20% performance and energy consumption differences between otherwise identical devices. These differences result from process variation in the manufacture of smartphone CPUs, which causes some CPUs to perform much more poorly than others. This paper explains the causes of this variation, measures its impacts, and discusses implications for smartphone researchers, software developers, and consumers.

1. INTRODUCTION

Two brand-new smartphones sit side-by-side. While they appear identical, one will consume 20% more energy than the other. It will get hotter, and run slower. A consumer that selects it may complain about poor battery lifetime and overheating. Or they may direct their negative reviews against apps that run on their device. A researcher experimenting with these devices may incorrectly conclude that a new approach saves (or does not) energy, or performs better (or worse). While the problematic device will happily identify itself, nobody is listening when it describes the problems it will inevitably face.

HotMobile '17, February 21-22, 2017, Sonoma, CA, USA

 ISBN 978-1-4503-4907-9/17/02…$15.00

DOI: http://dx.doi.org/10.1145/3032970.3032982

The seeds of this inherent variation were sown long ago in a factory far away. With increasing complexity and smaller process sizes, silicon variations are unavoidable. This makes it impossible to produce two identical smartphone CPUs. While process variation is well-known and the reasons for it are well-understood, we see little evidence that its effects are considered by smartphone consumers, app developers, and even mobile systems researchers. And, unlike desktop and server processors, where differences between good and bad chips are priced accordingly, in the mobile market manufacturers attempt to paper over these differences. Despite their effects on energy consumption and performance, good and bad CPUs are sold side-by-side in identically-priced devices. Despite the importance of battery lifetime, consumers purchase devices without testing them. And despite the differences between good and bad devices, app developers read reviews without this context and mobile systems researchers run experiments without controlling for this variation.

Note that we are *not* describing variations that occur over time due to usage. Degradations in battery and Flash quality as a result of long-term use are well-understood. Well-designed experiments will control for smartphone wear, and consumers have no choice but to replace aging components or their entire device. What makes inherent variation so surprising and important is that it is present and inescapable from the moment that the device is unboxed.

Our goal in this paper is to explain these inherent differences, measure their impact, and discuss implications for the various communities that work with smartphones, including consumers, app developers, and researchers. We begin by describing the underlying reasons for these variations in Section 2. We find it interesting that the same variations that produce price differences in non-mobile CPUs do not produce price differences in mobile CPUS. Instead, the differences in "cost" are passed on to the consumer in the form of differences in energy consumption and heat generation.

In Section 3, we describe our experimental setup and in Section 4 we perform a series of experiments to quantify the effect of CPU process variation. We observe significant differences in heat generation and energy efficiency, which we expected. However, due to the restricted thermal environment smartphones operate in, differences in heat generation quickly lead to performance differences. As a result, consumers (or researchers) stuck with poor-quality smartphone CPUs will experience both degraded energy consumption and performance. We discuss the impact of these differences in Section 5. Finally, we wrap up by discussing some of our future work in Section 6 and conclude in Section 7.

2. BACKGROUND

Our curiosity about variations between smartphones arose from our inability to reproduce performance results for a CPU intensive benchmark. By swapping the SoC while keeping the workload, device casing and battery constant, we confirmed that the SoC was the source of variation. Further experimentation revealed that the variations were caused due to SoCs belonging to different CPU bins. We explain the binning process and the reason for it below.

2.1 Process Variation

As transistor sizes have shrunk, process variation has made it virtually impossible to produce two chips that are exactly the same in all aspects, let alone millions of chips that are in our smartphones today. The manufacturing process of chips involves multiple steps such as wafer production, lithography-based masking, doping, diffusion and ion implantation—to alter electrical characteristics of silicon—followed by metallization to create contacts with silicon. This complex process naturally results in variations, called *process variation*, in transistor attributes such as length, width and bias. Differences in bias result in different threshold voltages for the transistors which in turn results in variable leakage current and switching speed. Thus, a single wafer design ends up producing transistors of varying speeds and leakage power—only some of which satisfy the design constraints while others are either too slow or too leaky thereby consuming high power [9].

As only a fraction of these chipsets will satisfy the design constraints such as expected timing or power, process variation tends to result in lower yield rates. To make matters worse, as the transistor sizes shrink, process variation becomes more pronounced, resulting in decreased yields. To improve yield rates and minimize the cost of a single chip, manufacturers find ways to sell even the "defective" pieces. Traditionally, manufacturers have dealt with defects by introducing some redundancy or, in some cases of failure, disabling certain features. For example, if the bad transistors are all co-located and appear in the processor block, that specific core is disabled and the chip is sold at a lower price. Similarly, if certain cache lines are faulty, then either the lines are rewired to use redundant cache lines—that were put in place for this very situation—or the entire cache block is disabled and the chip is sold with a lower cache size. These process variations are applicable to all components on smartphones. However, in this paper, we focus on studying the impacts of these variations on CPUs.

2.2 CPU Binning

In some cases, despite having no defects, differences in the semiconductor doping process can result in a chip requiring higher or lower voltages to operate. The process of segregating CPUs based on its manufacturing quality and electrical characteristics is known as *CPU binning*. Note that because multiple cores that are part of a single modern CPU are drawn from the same patch of silicon, differences are between entire CPUs and not between cores.

The two major techniques employed by manufacturers to improve yield rates are speed binning and voltage binning. When chips are manufactured, they are first tested to identify their stable operating frequencies. If a chip does not meet the necessary timing constraints or fails to operate at the expected frequency, the operating frequency is lowered until it passes the tests. The chips are then sorted into bins and labeled according to their speed. This process is called speed binning. They are then sold at price points proportional to their speed bin [13].

Voltage	Frequency (MHz)				
(mV)	300	729	960	1574	2265
Bin-0	800	835	865	965	1100
Bin-1	800	820	850	945	1075
Bin-2	775	805	835	925	1050
Bin-3	775	790	820	910	1025
Bin-4	775	780	810	895	1000
Bin-5	750	770	800	880	975
Bin-6	750	760	790	870	950

Table 1: **Voltage (mV) vs. Frequency (MHz) Across Bins.** Voltage set for various frequency levels across bins for Nexus 5.

Depending on the stable frequency, the CPU is categorized into one of the manufacturer's bins (e.g. i3, i5, i7) and priced accordingly. So, in reality, certain Intel i5 chips are in fact lower binned i7 chips with certain features, like hyper-threading, disabled. Using this approach, manufacturers are able to utilize a higher ratio of the silicon wafer—even the chips with minor defects.

Speed binning labels chips according to their speed. Voltage binning keeps the frequency fixed across all chips and adjusts the voltage across bins. Voltage binning is based on the fact that both, speed and leakage power of a transistor, are a function of the supply voltage. Slow transistors—ones with larger gate lengths—leak less, while fast transistors—ones with shorter gate lengths—leak more. Manufacturers thus divide the chips into voltage bins where slower chips are binned at higher voltage so as to support the required operating frequency, while faster chips are binned at lower voltage in order to reduce their already high energy consumption. We believe that this is done in order to try and provide consistent performance (in terms of speed) across all devices using the same SoC.

Table 1 lists voltages used for multiple frequencies across bins on a Nexus 5 device. Bin-0 has the slowest transistors while Bin-6 transistors leak the most. Therefore, Bin-6 operates at lowest voltage while Bin-0 voltage is increased to enable equal performance as Bin-6. Manufacturers thus, use this technique to attempt to enable consistent performance and similar power consumption across all bins. Note that the process controls for speed, so even both the Bin-0 and Bin-6 CPUs provide the same set of operating frequencies.

It is important to note that while desktop manufacturers typically have different price tags associated with each of their bins, reflecting the difference between good and bad silicon, mobile SoCs do not. To the best of our knowledge, mobile SoC manufacturers appear to be assigning CPUs of all bins with the same label which are then eventually sold to the consumer at the same price.

3. EXPERIMENTAL METHODOLOGY

Before presenting our results, we first describe our experimental methodology. All experiments were performed on the LG Nexus 5 handset running Android 5.1 (Lollipop). The Nexus 5 was released in 2013 and has a 4 core 2.26 GHz ARM CPU and 2 GB of RAM. The CPU has 14 frequency steps ranging from 300 to 2265 MHz. Recent surveys of active Android devices and versions [7, 1] confirm that the Nexus 5 and Android 5.1 are still widely used, despite the availability of newer devices and Android versions.

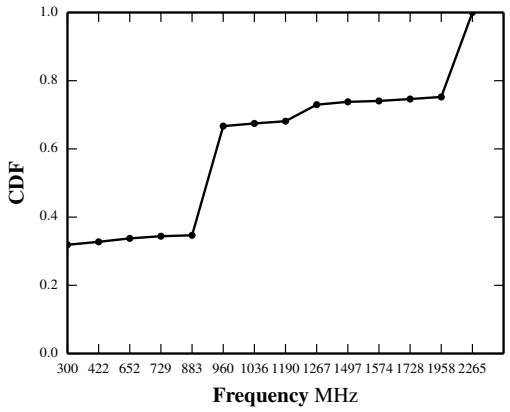


Figure 1: **CDF of Time Spent at Different Frequencies.** 300, 960 and 2265 MHz combined account for over 80%.

We used PhoneLab data [12] to determine that Nexus 5 CPUs spends most of their time at only 3 frequencies. Figure 1 shows the distribution of time spent at various frequencies on 150 devices over 30 days. We determined that both, 300 MHz and 960 MHz were resting frequencies when the CPU was idle and therefore focused our study on the max frequency, 2265 MHz.

By default, the Linux kernel has no power management logic to power cores on and off in response to load. Instead, Qualcomm distributes a closed-source binary, `mpdecision`, which is responsible for making these decisions based on CPU load. `mpdecision` is also responsible for the high usage of the 960 MHz frequency step. When the screen is on, `mpdecision` raises the resting frequency from 300 MHz to 960 MHz. This is done to provide better responsiveness and reduce jitter under the assumption that the user is likely to interact with the device while the screen is on.

Our workload was an Android app that computes the digits of π. The app can be configured to specify the number of threads to run in parallel and whether the goal is to compute a certain number of digits or to run for a specific time period. We refer to this app as PiBench. In our experiments, we configured PiBench to always use 4 threads to ensure that all cores are kept busy.

To measure energy, we used the Monsoon Power Monitor [4]. The Monsoon Power Monitor samples the power consumption at the rate of 5KHz while supplying stable voltage to the phone. Per the Nexus 5's battery specifications, we configured the Monsoon to use a 3.8 V supply voltage.

Before starting the actual workload, PiBench introduces a unique power signal that allows the Monsoon's and the smartphone's time-bases to be synchronized. This was necessary given that we were attempting to attribute Monsoon's energy measurements to the phone's logs denoting the beginning and end of experiments. To generate this unique signal, we turned on the device's flashlight for 20 s. This creates a period of high energy consumption over a known duration which is easily identifiable during post-processing.

Differences in heat generation between CPU bins affect energy efficiency and performance. To ensure that we controlled for the ambient temperature, all our experiments were performed in a controlled thermal environment which we refer to as thermabox. Figure 2 shows this setup. The temperature inside our thermabox is controlled by an STC-1000 temperature controller which has a range of -50°C to

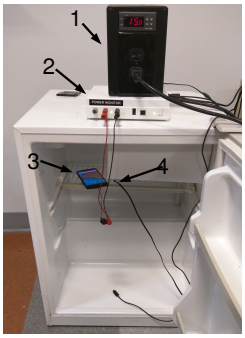


Figure 2: **Controlled thermal environment**. All our experiments were run inside a controlled thermal environment with an ambient temperature of 10±1°C. 1) Temperature Controller, 2) Monsoon, 3) Device, 4) Temperature Probe.

99°C with a resolution of 0.1°C. The temperature controller cools the thermabox by power cycling the small refrigerator as needed. We configured the controller to maintain the temperature of the thermabox at 10±1°C or 50°F. This temperature was chosen because it represents a reasonable outdoor ambient temperature that a smartphone might encounter—at least in Buffalo. It also allowed the smartphone to cool more rapidly between runs so that we could repeat experiments more quickly. This setup was necessary to be able to produce reproducible results. Further discussion on this is continued in Section 4.

To control for differences in the device packaging, we used a single case for all experiments. Only the SoC itself was removed from its case and swapped into our harness in each experiment. On the Nexus 5, the SoC can be easily removed and replaced without requiring any soldering.

All device parts were kept in the thermabox between experiments so that they began runs as close as possible to the target temperature. Otherwise we noticed cooling artifacts over the first few experiments with a new device as the SoC gradually cooled down to the thermabox set temperature.

At the start of each experiment, the device first does a short warm-up run for 10 seconds at the max frequency to heat the device up. It then waits until the CPU temperature sensor reports a stable temperature of 25°C or lower. Note that the CPU temperature sensor is on the CPU die and so reports a temperature that is much higher at all times when the CPU is running than the case or ambient temperature. After the temperature sensor stabilizes, the workload is started. When the workload completes, logs are uploaded to a server and combined with data from the Monsoon.

All of our experiments were repeated up to 10 times from which we consider the last 5 iterations for the data presented here. This is done to ensure that the phone reaches some steady state over the time taken to complete the first few iterations. When applicable, we provide the standard deviation across the last 5 iterations to validate our approach.

4. RESULTS

Our results are broadly classified into three parts. First, we confirm the inter-bin energy differences that result from voltage binning. Second, we show that despite manufacturer's best efforts, differences in heat generation between bins quickly create performance differences as well. Finally,

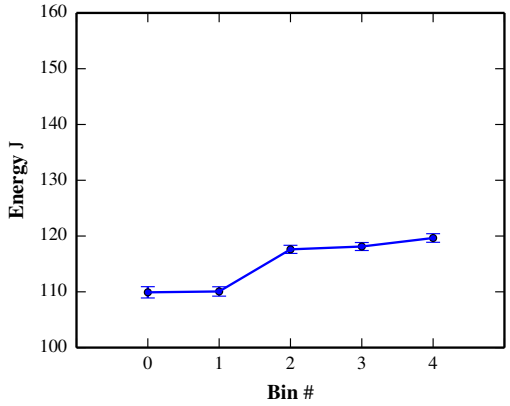


Figure 3: **Nexus 5 inter-bin energy variation**. Bin-4 ends up burning ≈ 9% more energy over the same time period while also doing the same amount of work. Can Bin-4 be considered similar to Bin-0?

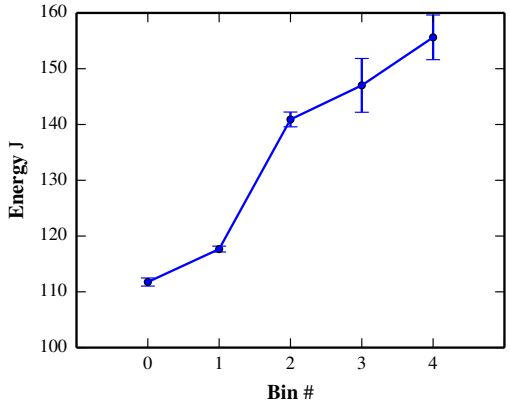


Figure 4: **Nexus 5 inter-bin energy variation under uniform voltage**. Without voltage binning, phones in Bin-4 consume ≈ 40% more energy than Bin-0.

we use more PhoneLab data to show that these effects occur on real devices and not just in a lab setting.

4.1 Energy Variation

Since manufacturers have focused their efforts on providing consistent performance, we will first quickly confirm that the performance of all bins is equivalent. We also evaluate the amount of energy that different bins sacrifice to achieve this performance. To do this, we configure PiBench to run for a fixed time period of 30 seconds. This workload suits our purposes as it allows us to measure relative performance in terms of progress—the number of digits computed over the specified time period.

As expected, we observed negligible difference in performance across various bins. Over the fixed time period of 30 s, on average, CPUs from bins 0 through 4 all computed the first 3200 digits of π with a standard deviation of 0.1%, showing that CPUs of various bins have similar performance capabilities.

We now look at the energy consumption of the chips while achieving consistent performance. Figure 3 plots the energy consumption of various Nexus 5 bins to run the same workload for 30 s. From Figure 3, we see that bad bins end up consuming as much as ≈ 9% more energy compared to the good ones while performing similarly.

In Figure 3, two variables contribute to differences in energy consumption—operating voltage and transistor variations. To understand how different the underlying transistors are, we removed voltage variations by running all bins at the same voltage using over-volting. Figure 4 plots the energy consumption for various bins of a Nexus 5 smartphone when configured to run at the same voltage. This figure shows that transistor differences can account for as much as ≈ 40% difference in energy. Thanks to voltage binning, that energy difference drops down to 9% across bins.

Note that Figure 3 establishes a trend that continues across the rest of our experiments. Despite testing 5 CPU bins, we see pronounced differences only between two groups: Bins 0, 1 and Bins 2, 3, 4. We are unsure why this is the case. When we control for voltage differences in Figure 4, we do see differences in all 5 bins. But at their standard operating voltages only two metabins seem to exist. This may be because manufacturer's efforts to equalize performance across all bins can only go so far. Adjusting the supply voltage can reduce the number of bins from 5 to 2, but not any farther.

4.2 Performance Variation

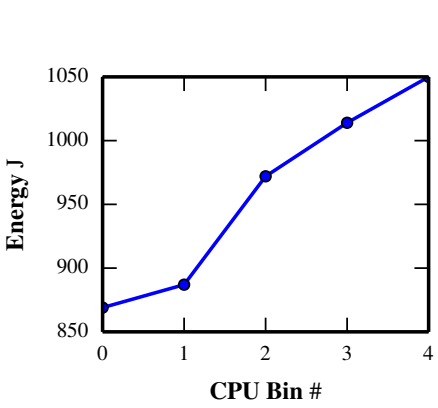
Our short PiBench experiments confirm the inter-bin energy differences we expected. In the following experiments, we configured PiBench to do fixed work. Specifically, to perform the first 400,000 iterations of computing the digits of π which generates the first 114,290 digits.

From Figure 5a, we see that energy varies from bin to bin while they all perform a fixed amount of work. Quantitatively, the difference in total energy consumption between Bin-0 and Bin-4 is ≈ 20%. Recall that in our earlier 30 s experiment summarized in Figure 3, we observed a different of 9% between the two bins.

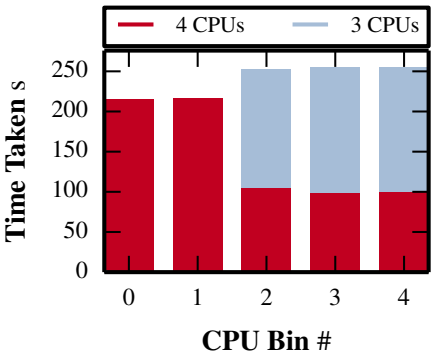
To paint the full picture, we also look at relative performance before delving into the reasons for increased energy drain. Contrary to our earlier findings, Figure 5b shows that performance also begins to vary over time. Bin-4 consistently took 18% longer time to complete the workload than Bin-0. Combined, these results show that the duration of the workload has varying impacts on both performance and energy across CPU bins.

Given that manufacturers have controlled for speed in the voltage binning process, the goal is clearly to eliminate performance variation. But performance variation clearly still exists. Before describing the underlying cause, we first need to establish a few facts. Leakage current of transistors is proportional to temperature [11]. Transistors that leak more also generate heat at faster rate compared to those that have lower leakage current. To make matters worse, in cases where the cooling rate is not increased, the higher heat dissipation increases the temperature of the device. This in turn creates a feedback loop wherein the leakage current further increases. The inter dependencies of temperature and leakage current cause the faster, leaky transistors to hit high temperatures sooner and more often.

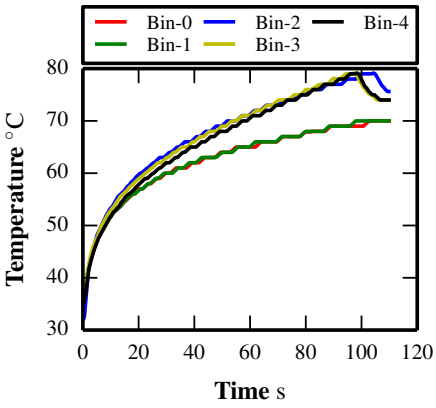
Although temperature accounts for changes in energy, it still does not explain the performance variation. This arises from the kernel's thermal management policies. On the Nexus 5, the kernel begins to throttle the CPU when the temperature reaches 80°C. The leakage current of Bin-2, 3 and 4 was sufficiently high to cause the chip to hit 80°C and trigger the kernel's thermal throttling policies. Note that while the policies about how to perform thermal management are up to the kernel, overheating cores *must* be



(a) **Energy Comparison Across Bins**. Bin-4 consumes ≈ 20% more energy than Bin-0.



(b) **Inter-bin performance comparison**. Bin-4 consumes 20% more energy while also taking 18% longer—lose-lose.



(c) **Inter-bin temperature variation**. Bins 2, 3, 4 encounter thermal throttling which impacts their overall performance.

Figure 5: **Energy, Performance and Temperature Variation Across Bins**

throttled or shut down to cool. Regardless of how it is done, it will disproportionately affect the performance of CPUs that generate more heat.

From Figure 5b, we see that bins 2, 3 and 4 ran the workload under the requested 4-CPU configuration for a total of 100 seconds. Combining this with Figure 5c, we see that approximately the first 100 seconds of the workload was when all 4 CPUs were active. Beyond this point, kernel throttling forces one CPU to go offline. From this figure, we also see that bins 0 and 1 do not raise the temperature high enough for the thermal throttling to ever trigger—thereby allowing these CPUs to finish the workload faster.

4.3 Overheating in the Wild

One question that remains to be answered is whether thermal throttling occurs in the wild. To answer this question, we processed PhoneLab data looking for kernel logs pertaining to thermal throttling. Each device's logs was broken up by day and the number of thermal throttling events were totaled. Thus, we have one data point for each device-day. Our findings are summarized in Figure 6. 80% of all device-days report one or more throttling events. 20% of all device-days report 50 or more throttling events. So this is not an artifact of the lab environment.

In fact, our low thermabox operating temperature of 10°C was chosen to produce less overheating. Indoor building set points are higher, as are outdoor temperatures during warm parts of the year in most parts of the world. Because smartphone CPUs rely on the ambient thermal environment for cooling, higher ambient temperatures will cause more overheating, reduced energy efficiency, and performance loss. This effect will be felt across all CPUs, but be worse on CPU bins that naturally generate more heat.

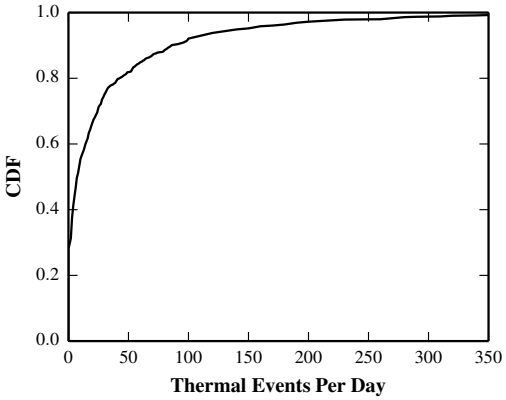


Figure 6: **CDF of number of thermal events observed per-device per-day in PhoneLab**.

5. IMPACT

In this section, we describe how the manufacturer's binning policies affect research and end-users.

5.1 Research

A significant portion of prior research assumes cross-device similarity within the same device models. To the best of our knowledge, no prior work has accounted for variations occurring due to CPU bins while modeling system energy consumption. All prior work that attempts to generate a model from a set of devices and apply this model to a larger, more general set are also affected. Table 2 lists some of the work that can greatly benefit from accounting for CPU bins, including several of our group's own previous projects.

Given that the CPU is among the largest consumer of energy on smartphones [10], future systems that model energy consumption need to take CPU bin information into account while modeling. In other words, selecting a smartphone make and model is no longer sufficient to perform studies. One must also ensure that either that all the devices belong to the same bin or that the studies account for bin differences. It is also crucial to run experiments within controlled thermal environments. As shown earlier, different bins have significantly different thermal characteristics which can very quickly lead to performance variations. This effect can persist from run to run. Without proper controls and time to cool, a device that gets hotter will stay hotter, and that heat will cause subsequent experiments to consume more energy.

5.2 End-Users

End users are perhaps the worst affected by the manufacturer's practices. Most online reviews of smartphones do not discuss CPU binning or the bin to which the particular device they tested belongs. This in turn causes users

Table 2: Research Systems Impacted by CPU Bins

Research Work	How is it impacted?
Carat	Compares energy measurements obtained on one device with another
Mote is Dead!	All energy measurements from one device
Jouler	Uses power model that does not account for bin differences
E^3	Savings reported would vary depending on CPU bin
WattsOn	Attempts to apply energy measurements from one device to another

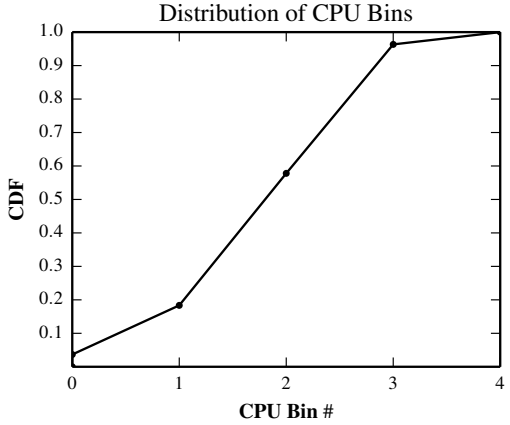


Figure 7: **Bins of 109 PhoneLab Nexus 5 Devices**

to blindly believe that the phone they purchase will end up performing similar to the device that was reviewed.

Among popular reviews of the Nexus 5 [2] [6] [5] [8], we found only one [8] that talks about CPU bins. However, contrary to our results, this review concludes that higher bins are better and that Bin-1 is the slowest bin. Given our knowledge that Bin-1 is among the best performing bins, this raises an important question of whether the reviewer picked up a Bin-1 device by chance or whether the manufacturer knowingly sent in a Bin-1 device for the review.

Figure 7 shows the bin distribution of 109 PhoneLab phones for which we were able to identify CPU bin information. Given how bins 2, 3 and 4 account for 80% of the total phones, the rational approach would be to establish the characteristics of these bins as the baseline. In this scenario, a user has a 20% chance of buying a smartphone that has $\approx$ 20% better energy and performance characteristics—devices from bins 0, 1. This would also imply that manufacturers knowingly sold 20% of their CPUs which were superior at the same cost as the baseline ones.

Although all of our studies were performed on a single model, Nexus 5, we are aware of CPU binning procedures being applied on several others including the Nexus 6, Nexus 5X and Nexus 6P. While the energy and performance variations observed in this research work may not directly apply to other devices, we hope that the methodology and techniques used in this paper will be of use to future researchers looking to study smartphone energy and performance.

6. FUTURE WORK

Due to manufacturing process variations, there are variations in static and dynamic power consumptions across CPUs. Theoretically, this means that even though some CPUs are more inefficient than others at doing work, they could be better at idling. We were unable to conclusively verify this trend during our experiments. But if it indeed exist, then it results in some very interesting implications for smartphone users. Users can match the device they buy to their usage patterns. A heavy smartphone user may want a device that is more efficient when busy while a light user may want a device that is more efficient at idling.

We also plan to look into how bins are defined on other devices with heterogeneous cores. These devices spend a lot of time running the little cores [3]. Thus, if the little cores are from a bin that is inefficient, this can negatively impact the battery life and by extension, the user's experience.

7. CONCLUSIONS

To conclude, in this paper, we quantify the impact of process variations on both performance and energy in today's smartphones. We describe some new challenges and difficulties in conducting studies on smartphone performance and energy consumption. Importantly, we question the current naming practices adopted by mobile SoC manufacturers wherein chips under the same label exhibit up to 18% performance and 20% energy variations.

8. REFERENCES

[1] Apteligent data report may'16. goo.gl/5qTaqX.

[2] Google nexus 5 review. https://goo.gl/OnMf9c.

[3] Here's how the galaxy s6 uses its octa-core processor. https://goo.gl/sKZJC3.

[4] Monsoon power monitor. https://goo.gl/rlizsj.

[5] Nexus 5 - trustedreviews. https://goo.gl/8GizMB.

[6] Nexus 5 review: the best phone $350 can buy. https://goo.gl/hjqbLt.

[7] Opensignal android fragmentation report. https://goo.gl/2h8l4O.

[8] Redefining the android experience with google's nexus 5. https://goo.gl/1yaiS6.

[9] Bowman, K. A., Duvall, S. G., and Meindl, J. D. Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. IEEE Journal of solid-state circuits 37, 2 (2002), 183–190.

[10] Carroll, A., and Heiser, G. An analysis of power consumption in a smartphone. In USENIX annual technical conference (2010), vol. 14, Boston, MA.

[11] Kim, N. S., Austin, T., Baauw, D., Mudge, T., Flautner, K., Hu, J. S., Irwin, M. J., Kandemir, M., and Narayanan, V. Leakage current: Moore's law meets static power. computer 36, 12 (2003), 68–75.

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

[13] Zolotov, V., Visweswariah, C., and Xiong, J. Voltage binning under process variation. In Proceedings of the 2009 International Conference on Computer-Aided Design (2009), ACM, pp. 425–432.