Energy Efficiency Analysis of Heterogeneous Platforms: Early Experiences

Youhuizi Li*† , Weisong Shi†, Congfeng Jiang*, Jilin Zhang* and Jian Wan*‡
*Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, China
†Mobile and Internet Systems Laboratory, Wayne State University , USA
‡School of Information and Electronic engineering, Zhejiang University of Science & Technology, China
{huizi,cjiang,jilin.zhang,wanjian}@hdu.edu.cn, weisong@wayne.edu

***Abstract*—Heterogeneous multi-core platforms, e.g., ARM's big.LITTLE, are a promising trend to improve the performance and energy efficiency of future mobile systems. However, the immediate benefits and the challenges to take advantage of the heterogeneity are still not clear. In this paper, we present our early experiences about the energy efficiency of the two big.LITTLE heterogeneous platforms: ODROID XU+E and ODROID XU3. We quantified compared them with homogeneous platforms through multiple benchmarks which include popular mobile applications and high-performance parallel benchmarks. Besides, we analyzed the scheduling impact on the energy consumption of the heterogeneous platforms and the migration cost is also discussed. Based on the results, several insights, such as fine-granularity power control and thread level parallelism, related to hardware, application and system design are derived.**

***Index Terms*—Heterogeneous platform, Energy consumption, big.LITTLE**

I. Introduction

As the result of the dark silicon issue and increasing demand of specialized components, heterogeneity becomes more and more important and leads the trend of future devices' development, especially for mobile platforms. Heterogeneity is a general concept that may refer to CPU/GPU computing architecture, mixed types of accelerators and so on. The advantage of heterogeneous platforms is that they can improve energy efficiency while maintaining performance[1], [2]. To evaluate their benefits, previous work usually leverages DVFS to simulate different types of CPUs [3]. With the emerging of the ARM big.LITTLE processor [4] and Samsung Exynos 5 Octa system-on-chip (SoC) [5], we have the opportunity to explore and exploit real heterogeneous hardware platforms.

In this paper, we undertake the following questions:(1) *Compared with homogeneous platforms, how much energy can be saved in heterogeneous platforms?* We want to know the capability of heterogeneous platforms. (2) *what is the impact of scheduling from energy aspect?* As the different types of processor exist, the scheduling algorithm takes the responsibility of choosing proper processor for different workloads. Both of the benefits can be gained from the correct scheduling and penalty of the improper scheduling are important. And finally (3) *what is the migration overhead on performance and energy?* Applications have different phrases (e.g. loading content, waiting user input, etc.) and scheduler needs dynamically migrate workload to proper cores in runtime. In this situation, migration overhead is one of the key factors to decide when and which core to migrate. All of the three questions directly influence the benefits we can get from heterogeneous platforms.

To answer these questions, we picked the two ARM big.LITTLE platforms, ODROID XU+E (XUE) and ODROID XU3 (XU3) [6], as experimental platforms to study. Both XUE and XU3 have heterogeneous cores but provide different control strategies. XUE offers cluster switching and XU3 supports heterogeneous multi-processing [5]. The system and component level power information are collected to analyze their energy behavior under various cases. By analyzing the results, we can further understand the characteristics of the big.LITTLE platforms and wisely use them.

In this paper, we have three main contributions:

- We investigate the two generations of the same heterogeneous platforms and homogeneous platforms from the viewpoint of energy in a quantitative way. We found that most of the energy savings in heterogeneous platforms come from idle time and there is no much benefit for sustained heavy workloads.
- We analyze the impact of scheduling and migration overhead on the ARM big.LITTLE devices from the performance and energy aspects. The improper scheduling can consume 5% - 30% more energy.
- We derived a list of insights, such as fine-granularity power control and thread level parallelism, related to hardware, application and operating system design.

The remainder of the paper is organized as follows: We introduce the two heterogeneous platforms and illustrate our experiment setup in Section II. The detailed case studies are demonstrated in Section III, which compared the heterogeneous and homogeneous platforms and analyzed the impact of scheduling and migration cost. Following that, we discuss a list of insights in Section IV. The related work is presented in Section V and Section VI summarizes the paper.

II. Experiment Setup

To practically investigate heterogeneous platforms, we did experiments on two generations of ARM big.LITTLE platforms produced by Hardkernel [6]. Their specifications are presented in Table I. ARM's big.LITTLE processor [4] is a single-ISA heterogeneous multi-core processor which contains

TABLE I: The specifications of the two platforms.

	ODROID XU+E (XUE)	**ODROID XU3 (XU3)**
Generation	The first generation	The second generation
SoC	Samsung Exynos 5410	Samsung Exynos 5422
big CPU	quad-core Cortex-A15 CPU (800 MHz to 1600 MHz)	quad-core Cortex-A15 CPU (1200 MHz to 2000 MHz)
LITTLE CPU	quad-core Cortex-A7 CPU (500 MHz to 1200 MHz)	quad-core Cortex-A7 CPU (1000 MHz to 1500 MHz)
L1 Cache	32 kB/ 32 kB	32 kB/ 32 kB
L2 Cache	2 MB on big CPU, 512 KB on LITTLE CPU	2 MB on big CPU , 512 KB on LITTLE CPU
Memory	2 GByte	2 GByte
Key Features	CPU hotplug, Cluster switching	CPU hotplug, Heterogeneous multi-processing (HMP)

two types of cores: Cortex-A15 and Cortex-A7. Data is shared between the clusters via the CCI-400 cache coherent interconnect to achieve seamless migration. The first generation, denoted as XUE, provides only cluster switching mode, that is, either the big cluster or the LITTLE cluster is active. The cluster switching is achieved by modifying the CPU frequency. On the contrary, XU3, the second generation, supports heterogeneous multi-processing (HMP) which is a core level migration and all the eight cores can be active at the same time. We can dynamically set the CPU affinity using *taskset* command to migrate thread between the different cores. For homogeneous platforms, we disable the cluster migration on XUE and control the platform runs on big (LITTLE) cores only to simulate A15 (A7) platform. Both of XUE and XU3 feature four separated current sensors to measure the power consumption of big core cluster, LITTLE core cluster, GPU and memory in realtime. To get the system power data, we use a BK Precision programmable power supply [7] to power up the platforms, which provides a constant voltage of 5 V and a maximum current of 5 A. It logs the current at a sample frequency of 4 Hz.

III. Case Studies

In this section, we investigated the two big.LITTLE platforms by comparing their power and energy consumption with homogeneous platforms' results under various applications. The experimental benchmarks include mobile applications and high-performance synthetic benchmarks. Moreover, we analyzed the scheduling impact and migration overhead of heterogeneous platforms to study how to get the benefits of heterogeneity.

A. Active Idle Power

The short battery life of mobile devices is one of the main motivations that lead to the development of big.LITTLE architecture which promise to deliver peak-performance capacity at significantly lower average power [4]. The energy is saved by running on LITTLE (energy efficient) cores when the system is in idle state. Most time mobile devices, like smartphones, are in sleep state, so we first investigate the active idle power difference between homogeneous and heterogeneous platforms to see how much we can save. Here the active idle represents the situation that system is awake but no application is running.

Figure 1 presents the active idle power of the big and LITTLE cores at each frequency in the Android OS. The power refers to the whole device's power, not the core level power. The LITTLE core power of XU3 is measured when

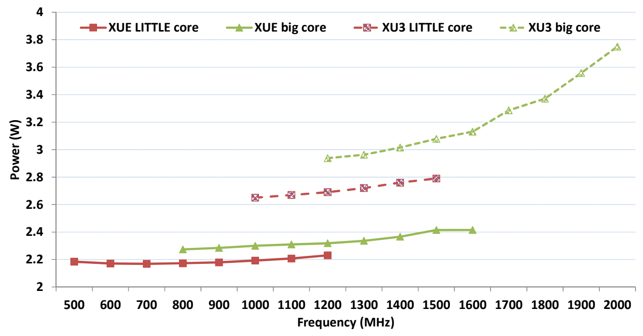

Fig. 1: The active idle power of the two platforms under each frequency.

all the big cores are disabled. At the same frequency, the big core active idle power is always greater than LITTLE core power, and the difference is around 0.1 W for XUE and 0.25 W for XU3. The range of LITTLE cores' power is very small, less than 0.2 W. While the big cores are sensitive to frequency change, so that it can provide corresponding high performance. Assume a smartphone's idle time is 8h, the battery voltage is 4 V. Compared with *A15 Only* homogeneous platforms, the heterogeneous platform with LITTLE core can save as much as 200 mAh. The system power difference of XUE and XU3 mainly caused by other components on the boards, not the result of the heterogeneous cores, GPU or memory which we can measure power directly.

One of the well-known power saving approaches in multicore systems is core offlining [8]. We evaluated the influence of disabling CPU cores on the two platforms. Due to the page limit, we illustrated XU3's results here and the behavior of XUE is similar. *The offlining did not work well within CPU clusters.* The power was always 3.18 W when we disabled 1 to 3 big cores. Then, the power decreased sharply from 3.18 W to 2.68 W since all the big cores were disabled and the package was idle. For LITTLE core cluster, the system power decreased slightly (from 2.68 W to 2.64 W) when the number of active cores decreased.

B. Benchmarks

As we presented previously, heterogeneous platforms improve system energy efficiency in the active idle case. Next we will investigate their performance and energy information under various workloads.

We first analyze mobile applications since heterogeneous platforms already appear in the mobile market. Six mobile applications/benchmarks are chosen: *YouTube* and *Castle Master* represent the two popular application categories: Video Player

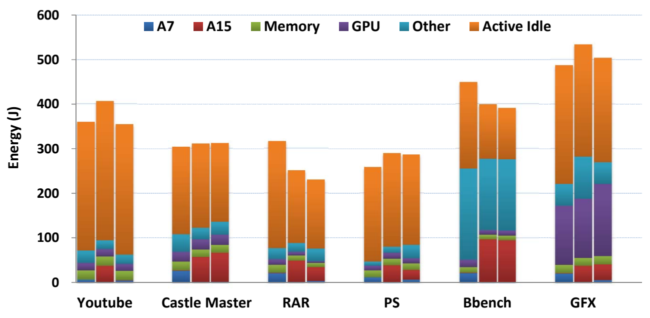

Fig. 2: The component level energy information of mobile applications on A7 Only, A15 Only and XUE platforms (left to right).

and Game. *RAR* is a compression program and *PhotoShop (PS)* is a photo editing tool. They are "heavy" workloads that appear on mobile devices. The last two applications are *BBench* [9] and *GFXBench*, they are browser and GPU benchmarks used to stress the system and are not commonly used in daily life.

Figure 2 shows their component level energy consumption. "Other" refers to the power used by components rather than CPU, GPU and memory. Its value is total system power minus the measured five parts (A7, A15, memory, GPU, active idle). The heterogeneous platform data were collected from XUE, because XU3 has higher CPU frequency which may interfere comparison results. To keep the figure readable, we proportionally adjusted the time within each application. From the viewpoint of energy, *A7 Only* platform consumes the least energy in most cases. However, with the consideration of user experience, which is an important factor for mobile devices, not all the energy optimal configurations are suitable. There is an obvious delay when *PhotoShop* and *Castle Master* run on A7 cores. Except *YouTube*, XUE leveraged A15 cores in all the applications. The A7 energy in XUE is not obvious due to its low power and less active time, but compare the total system energy of XUE and *A15 Only* platform, the contribution of A7 is identifiable. From the results, *we argue that performance sensitive applications merely contribute to the energy efficiency improving due to performance constrains, while light workloads benefit most on heterogeneous platform.* For applications that stress a specific component, like *GFX* which GPU power consumptions are similar in the three platforms, the heterogeneous CPU cores are not very helpful. However, as most of mobile workloads are periodical [2], there is still considerable energy saving potential on heterogeneous platforms.

Nowadays, multi-core architecture is widely-used in computer systems and it improves parallel applications' performance a lot. Both XUE and XU3 contain quad-core CPU, so we focus on the parallel benchmarks in the following examples, such as sysbench and NAS Parallel Benchmarks (NPB) [10], to evaluate their performance and energy behaviors.

As we mentioned above, the XUE platform allows four cores (big or LITTLE cores cluster) to be active, while the XU3 platform supports all eight cores working concurrently. Hence, we tested the energy consumption of the sysbench CPU benchmark which calculates the prime numbers that are smaller than 10000 on the two platforms with 1 to 8 threads. The core frequency is fixed (big:1200 MHz and LITTLE:1000 MHz) to eliminate the potential interference from DVFS. From the perspective of the total system energy consumption, XUE consumed less energy than XU3 in 1 to 4 threads cases. As the number of threads increased, the XU3 platform became more energy efficient because it can leverage the four more LITTLE cores. The execution time on the two platforms is almost the same and the average difference is 0.6 s when we ran 1 to 4 threads. After that, XU3's time continually decreased until reaching eight threads. Although the active idle power of XU3 is greater than XUE's, its system energy becomes smaller as the result of the decreased time. *Hence, similar with homogeneous multi-core systems, the number of active threads should match the number of cores to take advantage of multi-core architecture.*

Parallel benchmarks usually run on highest frequency to get better performance, but the high frequency usually is not the energy optimal configuration. To analyze the parallel benchmarks' energy behavior on heterogeneous cores, we measured the energy consumption of the NPB benchmarks under several frequency settings. Figure 3 lists three representative benchmarks' results. During the benchmarks' running time, there is no migration exist in XUE, so we leverage XUE to get homogeneous platforms' results and the compared heterogeneous platform in the examples is XU3. We used 4 threads to run on XUE and 8 threads to run on XU3 so that we can leverage all the available cores. On the homogeneous platforms, the energy optimal setting is 800 MHz in *A15 Only* platform and 1200 MHz in *A7 Only* platform. For the heterogeneous platform, the optimal configuration is always the smallest frequency on the big core, while the optimal LITTLE core frequency depends on the programs (LU.A prefers high frequency, UA.A has no obvious preference). *Since the active idle power is varied a lot based on the big core frequency, which can not be compensated by the speedup, the smallest frequency on the big core is always the optimal energy choice in both platforms.* For component level energy consumption, the "Other" part increased with the increase of frequency, especially on big cores. Besides, we can see that there is a trade off between CPU energy and memory energy, the memory energy consumption increases with the decreasing of CPU frequency.

From the perspective of performance, all benchmarks' execution time decreased with the frequency increase on the homogeneous platforms, while the results varies on XU3. For LU.A and MG.B cases, their running time mainly depended on the LITTLE core frequency. The time of UA.A was decided by both big and LITTLE cores. Hence, there is an obvious energy difference caused by the LITTLE core when the big core frequency is fixed for LU.A and MG.B, while the same situation was not found in UA.A. *Compared with the homogeneous platforms, the performance on heterogeneous platform is not directly proportional to the frequency since it has two types of cores working at the same time which makes the synchronization issue becomes more significant.*

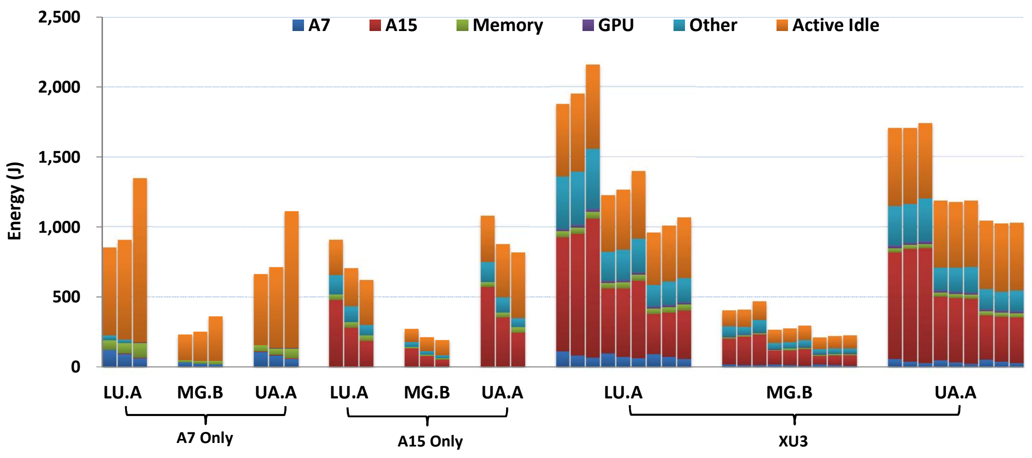

Fig. 3: The component level energy information of NPB benchmarks on the platforms. The frequencies (left to right) are 1200, 1000, 500 MHz on A7 Only platform and 1600, 1200, 800 MHz on A15 Only platform. The frequencies (left to right) for big core and LITTLE core on XU3 are 2000&1400, 2000&1200, 2000&1000, 1600&1400, 1600&1200, 1600&1000, 1200&1400, 1200&1200, 1200&1000 MHz.

C. Impact of Scheduling

With the more components we can control, the potential to achieve better energy efficiency becomes higher, while the complexity to find the optimal configurations also increases. Multiple applications can run concurrently, the scheduling becomes the crucial part that directly influences the performance and energy efficiency. Assuming that there are two processes running in the system, is it better to put them on the same core or different cores?

We took LU.A and UA.A as examples. The benchmarks were compiled as one thread program, and each time we run two instances (e.g. LU.A.1 and LU.A.2) under different core configurations. The results are shown in Figure 4, the most energy efficient choice is putting the two processes on two big cores and the second optimal option is using one big core. It is reasonable as the big core consumes less energy than the LITTLE core when there is only one process. The energy consumption of leveraging both big and LITTLE core is similar with running on two LITTLE cores. The energy difference between the two settings and the optimal configuration is as high as 30% of the *1b+1L* case. From application's viewpoint, the LU.A benchmark is more sensitive to the LITTLE core. Compared with UA.A, the energy consumption of LU.A is very different under cases with and without LITTLE core. One of the reasons is that LU.A uses the CPU more intensively.

From previous synthetic benchmark examples, we can see that the key factors of scheduling are the number of threads, the number of cores as well as workload preferred core characteristics. Next, we analyze the scheduling impacts of application benchmarks with different phrases.

The scenario included *BBench* which simulates user web browser behavior and music player that run on the background. There was a two-second pause after each page loaded to mimic reading behavior. Hence, two phrases exist in the example: the loading phrase which requires big core and the pausing phrase which prefers run on LITTLE core. The music player should always run on LITTLE core based on its requirement. The default scheduling method was indirectly controlled by

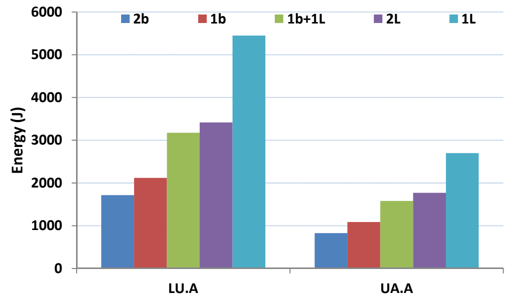

Fig. 4: The energy consumption of LU.A and UA.A under different configurations, big core frequency is 1600 MHz and LITTLE core frequency is 1200 MHz. b and L represent big core and LITTLE core respectively.

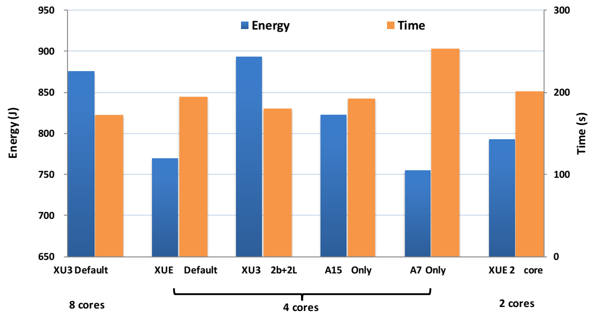

Fig. 5: The energy consumption of BBench under different configurations with default scheduling. b and L represent big core and LITTLE core respectively.

offline CPU cores, so the workload is forced to run on the available cores. Figure 5 presents the energy consumption under different configurations. The homogeneous *A7 Only* case consumes the least energy since its power consumption was small and there was no much speedup in other cases. In the heterogeneous cases, the energy difference of *XU3 Default* and *XUE Default* are mainly caused by the difference in their idle power. The 5% difference of *XU3 Default* and *XU3 2b+2L* is the results of the workload's demands on big cores. We omited the *XU3 1b+1L* case in the figure, because its time was twice of others which leads to the highest energy consumption and

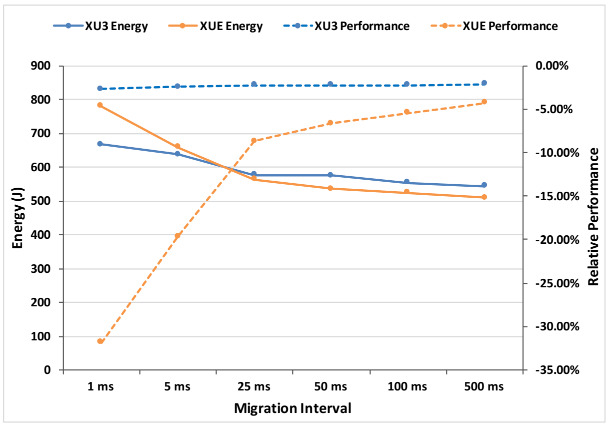

Fig. 6: The energy and performance information of sysbench CPU benchmark in different migration interval cases.

no comparability.

The principle of scheduling is providing enough resources to workloads. The consequence of improper scheduling depends on workload's characteristics. Compared with synthetic benchmarks, the phrase-based applications are more tolerant.

D. Migration Cost

The advantage of the ARM big.LITTLE architecture is that we can choose the proper type of cores to use in the runtime to save energy. In this section, we investigated the migration cost from the aspects of performance and energy.

To evaluate the migration cost, the basic approach is to compare the results with the ground truth which does not include migration overhead. However, the ground truth is almost impossible to measure on real devices. Hence, we take the theoretical value as the baseline, which is calculated as follows: Assume migration overhead is zero, the whole workload W running only on big (LITTLE) core requires T_{bW} (T_{LW}) time and E_{bW} (E_{LW}) energy. Then the workload completed in the unit time is: $w_b = \frac{W}{T_{bW}}$ for big core, and $w_L = \frac{W}{T_{LW}}$ for LITTLE core. Suppose in the migration enable case, the workload is migrated after running on big core for t_b seconds or running on LITTLE core for t_L seconds. So the time ratio of running on big core and LITTLE core is $t_b : t_L$. Assume the total big core running time is $t_b * k, k \in \mathbb{R}$, then

$$W = w_b * t_b * k + w_L * t_L * k = \frac{W}{T_{bW}} * t_b * k + \frac{W}{T_{LW}} * t_L * k$$

So $k = \frac{T_{bW} * T_{LW}}{t_b * T_{LW} + t_L * T_{bW}}$, and the theoretical running time is:

$$t_{Th} = t_b * k + t_L * k = \frac{(t_b + t_L) * T_{bW} * T_{LW}}{t_b * T_{LW} + t_L * T_{bW}}$$

Similarly, the theoretical energy is:

$$\begin{aligned} E_{Th} &= \frac{E_{bW}}{T_{bW}} * t_b * k + \frac{E_{LW}}{T_{LW}} * t_L * k \\ &= \frac{E_{bW} * t_b * T_{LW} + E_{LW} * t_L * T_{bW}}{t_b * T_{LW} + t_L * T_{bW}} \end{aligned}$$

Figure 6 illustrates the migration impact on energy consumption and performance for sysbench CPU benchmark. The workload is migrated every x milliseconds, with x varying from 1 ms to 500 ms. It belongs to the $t_b : t_L = 1 : 1$ case. We can see that there is obvious energy and performance cost for XUE and energy cost for XU3 with the migration interval decreases. The potential reason for the different performance trend of XUE and XU3 is the cache coherence issue on XUE which leads to cache miss after cluster migration. Hence, we make the big cores offline/online after each migration on XU3 to redo the experiments. This time the relative performance is similar with the XUE's results. With the migration interval becomes longer, the energy and performance of the two platforms are more and more stable.

TABLE II: Migration cost for CPU benchmark.

Type	$t_b : t_L$	XUE T (s)	XUE E (J)	XU3 T (s)	XU3 E (J)
Typ$_M$	1:0	94.89	379.56	94.31	386.67
Typ$_M$	0:1	219.41	548.52	219.16	727.61
Typ$_{Th}$	1:1	132.48	430.57	131.87	489.24
Typ$_M$	2:2	132.67	443.12	132.11	507.3
Typ$_M$	6:6	133	434.91	132.25	493.29
Typ$_{Th}$	1:2	152.64	457.92	152.06	544.37
Typ$_M$	2:4	153.52	463.63	152.54	562.87
Typ$_{Th}$	2:1	117.03	409.6	116.42	447.04
Typ$_M$	4:2	117.56	420.86	116.91	460.62

Table II presents the results in different $t_b : t_L$ cases with larger migration intervals. Typ$_M$ stands for measured value and Typ$_{Th}$ represents theoretical value. The first two rows represents the situations that we only run the benchmark on big core (first row) and LITTLE core (second row). As Table II reports, the time delay is negligible since the theoretical time and measured time are very close, while the energy is different to some extent. Compare the two platforms, the energy consumptions of XU3 are greater than XUE's as the result of high active idle power. The same situation on the energy overhead aspect, take the 4:2 case as an example, the cost for one migration is 0.56 J in XUE and 0.68 J in XU3. In the two cases that the time ratio is 1:1, the results indicate the 6:6 case consumed less energy than the 2:2 case since the migration times is less in the 6:6 case. In a word, there is an energy cost to migrate between big and LITTLE cores. It is not always good to choose the optimal setting since the migration may not worth the price if the workload changes frequently.

IV. Insights

Based on the results, we derived a list of implications and grouped them into the following three categories:

Hardware Design: *Provide fine-granularity power control to further decrease idle power.* Based on the comparison of homogeneous and heterogeneous platforms, low idle power is the key factor that improves the energy efficiency on heterogeneous platforms. To save more energy, devices should provide fine-granularity power control and decrease components' power coupling so that each component can stay in low power mode freely. For example, CPU offlining in the platforms does not impact the system power very much unless all the cores in the package are idle. We can provide independent power supply to each core to effectively reduce CPU power.

Extend heterogeneity to multiple components. ARM big.LITTLE provides heterogeneity in CPU level, while the power dissipation of CPU only occupies part of the mobile system's power. The heterogeneity can be applied to other components to better serve users' requirements for different applications and save the energy at the same time.

Application Design: *Increase the usage of thread level parallelism while pay attention to synchronization.* Similar with homogeneous multi-core platforms, to benefit from the increasing number of cores, applications should improve their thread level parallelism. Most of the popular mobile applications only use one or two cores and there is little chance that they can leverage four cores. Comparing the XU3 platform with XUE, there is no obvious performance improvement in CPU intensive phrase, so as energy since the speedup cannot compensate the increased power. Compared with homogeneous platforms, the synchronization issue in multi-thread applications are more significant since the heterogeneous cores have different capabilities and the performance may easily be influenced by the workload runs on the LITTLE core. Take LU.A as an example, the execution time of a four-thread case on the *A15 Only* platform with big cores at 1200 MHz is smaller than a eight-thread case that runs on XU3 with big cores at 1200 MHz and LITTLE cores at 1000 MHz.

Operating Systems Design: *Schedule tasks to the right core at the right time.* The scheduling algorithm directly affects the energy consumption and performance of systems. On the application level, we found that different applications have different optimal configurations, for example, LU.A prefers big core at 800 MHz while UA.A works better on LITTLE core at 1200 MHz. If the workload is scheduled to a wrong type of core, the energy difference can be as high as near 30%. On the system level, there are usually multiple applications running concurrently. The system needs to detect non-CPU intensive phrases and schedules them to LITTLE cores with the consideration of migration cost, so that the total system energy is saved. Moreover, the energy consumption is not the only aspect that we care about during scheduling, user experience should also be considered.

V. Related Work

Heterogeneous platforms used to appear in clusters and high performance multi-core computers to improve the performance. Srinivasan *et al.* [3] studied the existing operating system and hardware interaction in three kinds of heterogeneous core architectures. Aside from performance improvement, heterogeneity is also helpful for energy saving. Kumar *et al.* [1] proposed single-ISA heterogeneous multi-core architecture and discussed the potential power saving. In the mobile field, Lin *et al.* [2] proposed a combination of runtime and OS support on asymmetric processors to achieve workload energy proportionality. Most of the previous work use simulators to evaluate their solutions, in this paper, we run experiments on real hardware to explore the benefits of heterogeneous CPU architecture.

With the emerging of the ARM big.LITTLE technology and Samsung Exynos SoC, more and more researchers are attracted and begin to working in this field. Carroll and Heiser [8] built a Linux frequency governor Medusa which leverages DVFS and offlining to save energy of modern smartphones. Hähnel and Härtig [11] investigated the detailed energy characteristics of the hardware components and applications on the big.LITTLE platform. We analyzed and compared the two big.LITTLE platforms and proposed several implications for developers to improve the energy efficiency.

VI. Conclusions and Future Work

Heterogeneous platforms lead the trend of future devices, especially in the mobile market. We found that heterogeneous platforms indeed have great potential for energy saving which mostly comes from idle and low workload situations, however, there are several steps that should be taken seriously by the community, including hardware vendors, application developers, and operating systems designers, to maximize the potential of heterogeneous platforms. In the future, we will build an energy prediction tool that can help systems to choose the optimal configuration for heterogeneous platforms.

Acknowledgment

This work is supported by Natural Science Foundation of China (No.61472109, 61003077), and Natural Science Fund of Zhejiang Province (LY14F020045).

References

[1] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, "Single-isa heterogeneous multi-core architectures: The potential for processor power reduction," in *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 36. Washington, DC, USA: IEEE Computer Society, 2003, pp. 81–92.

[2] F. X. Lin, Z. Wang, and L. Zhong, "Supporting distributed execution of smartphone workloads on loosely coupled heterogeneous processors," in *Proceedings of the 2012 USENIX Conference on Power-Aware Computing and Systems*, ser. HotPower'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2.

[3] S. Srinivasan, L. Zhao, R. Illikkal, and R. Iyer, "Efficient interaction between os and architecture in heterogeneous platforms," *SIGOPS Operating Systems Review*, vol. 45, no. 1, pp. 62–72, Jan 2011.

[4] ARM, "Arm big.little technology," http://www.arm.com/products/processors/technologies/biglittleprocessing.php.

[5] Samsung, "Heterogeneous multi-processing solution of exynos 5 octa with arm big.little technology," http://www.arm.com/files/pdf/Heterogeneous_Multi_Processing_Solution_of_Exynos_5_Octa_with_ARM_bigLITTLE_Technology.pdf, white paper.

[6] Hardkernel, "Odroid xu3, odroid xu+e," http://www.hardkernel.com/main/main.php.

[7] B. P. Corp., "BP Precision," http://www.bkprecision.com/, model 1785B.

[8] A. Carroll and G. Heiser, "Mobile multicores: Use them or waste them," *SIGOPS Operating Systems Review*, vol. 48, no. 1, pp. 44–48, May 2014.

[9] A. Gutierrez, R. Dreslinski, T. Wenisch, T. Mudge, A. Saidi, C. Emmons, and N. Paver, "Full-System Analysis and Characterization of Interactive Smartphone Applications," in *the proceedings of the 2011 IEEE International Symposium on Workload Characterization (IISWC)*, Austin, TX, USA, 2011, pp. 81–90.

[10] N. A. S. Division, "Nas parallel benchmarks," http://www.nas.nasa.gov/publications/npb.html.

[11] M. Hähnel and H. Härtig, "Heterogeneity by the numbers: A study of the odroid xu+e big.little platform," in *6th Workshop on Power-Aware Computing and Systems (HotPower 14)*. Broomfield, CO: USENIX Association, 2014.