A Case Study of Energy Efficiency on a Heterogeneous Multi-Processor

Hitoshi Oi
The University of Aizu, Aizu-Wakamatsu, JAPAN
oi@oslab.biz

ABSTRACT

In this extended abstract, we present a case study of power-efficiency on a heterogeneous multi-core processor, Exynos 5422 based on the ARM big.LITTLE architecture. We show the effect of thermal management on the big (faster) cores and the comparisons between big and LITTLE (slower) cores using the EEMBC CoreMark-Pro benchmark. As expected, LITTLE cores are considered to be more energy efficient than the big cores at the maximum performances of both cores. However, the opposite is true for 4 out of 9 workloads when the performance of both cores are matched by lowering the clock frequency of big cores. Delay-insertion for matching the performance is only effective for one workload, but it may be useful in a multi-programmed environment when the frequency of each core cannot be set individually (which is the case for Exynos).

Keywords

Workload Analysis, Performance Evaluation, Energy Efficiency, Heterogeneous Multi-Core Processor

1. INTRODUCTION

Heterogeneous multi-core processing (HMP) is very popular these days, but in commercial products, they are mostly combinations of general purpose CPUs and compute-intensive cores (e.g. GPGPUs). Another type of heterogeneous multi-core architecture is that the cores are implemented in different microarchitectures but still have an identical instruction set. The big.LITTLE architecture from ARM is one of commercial products of this type. Samsung, among others, has implemented it as their Exynos series [1]. Task scheduling among different types of cores has been one of the most active topics in this type of HMPs [2]. However, it can also be a suitable platform to study the effect of microarchitecture on the performance and energy efficiency because all components other than cores are identical on a system with such an HMP processor. In this extended abstract, we present an energy efficiency study on an Odroid-XU3, a single-board computer with an Exynos 5422 processor [3]. In the next section, the measurement platform and methodologies are described. The measurement results and analysis are reported in Sections 3 to 6. Section 7 summarizes the results obtained so far and also mentions the status of the current work.

GreenMetrics 2017 Urbana-Champaign, IL, USA

2. PLATFORM AND METHODOLOGIES

Table 1 shows the specifications of the measurement platform Odroid-XU3 [3]. It is a single board computer from Hardkernel with an Exynos 5422 processor (Figure 1). Both A15 (big) and A7 (LITTLE) cores have core-private L1 instruction and data caches of 32KB each. The sizes of the shared L2 caches are 2MB (A15) and 512KB (A7). Both are DVFS capable in 0.1GHz steps with frequency ranges of 0.2 to 2GHz (big) and 0.2 to 1.4GHz (LITTLE). A15 is 3-way out-of-order superscalar and A7 is 2-way in-order.

Table 1: Odroid-XU3 Specifications

CPU	Samsung Exynos 5422 (ARM big.LITTLE: 4×A7 + 4×A15)
Memory	2GB (LPDDR3)
OS	Arch Linux ARM (Kernel 3.10.104-8)
Sensors	Power (big and LITTLE cores, GPU & Memory), Temperature

Odroid-XU3 has four built-in power sensors, measuring big and LITTLE cores, memory and GPU. It also has a temperature sensor measuring the die temperature. To measure the total power consumption, a power meter, Odroid Smart Power [4] is attached to the DC-input of the Odroid-XU3. The sensors and power meters are sampled every second. The idle power consumption is 0.1969 Watt by the sum of sensors and 3.2091 Watt by the power meter.

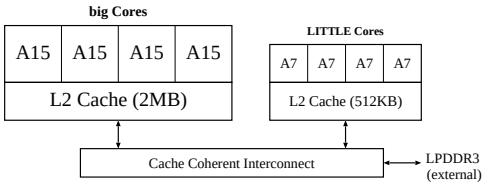

Figure 1: Exynos 5422 Diagram

We first obtain the normalized per-task energy $E_n = \sum_{ET}(P_d - P_i)/NT$, where P_d, the power consumption (either by sensors or power meter), P_i, the idle power, NT, the number of tasks executed during execution and ET, the execution time. As the energy efficiency metrics, we use the relative energy consumption which is the ratio of E_n of big cores with 2GHz against with the clock frequency in MPC in Table 3 (Section 3) and LITTLE cores against big cores (Sections 4 to 6).

Table 2: CoreMark-Pro Benchmark Programs

Program	Description
CJPEG	JPEG Image Compression
Core	Derivation of CoreMark (list, matrix and state machine)
Linear	Linear algebra solver (from Linpack)
Loops	Modified Livermore loop
Neural Network	Pattern evaluation using neural net
XML Parser	XML parsing and ezxml creation
SHA	256-bit Secure Hash Algorithm
Radix2	FFT Radix 2
ZIP	zlib compression benchmark

For the workloads of this study, we use the CoreMark-Pro benchmark programs from EEMBC [5]. It is a collection of nine benchmark programs (five integer and four floating-point) as shown in Table 2. Each benchmark is run for minimum 1000 iterations or 10 seconds (whichever more/longer), and the averages of ten runs are reported.

3. THERMAL MANAGEMENT EFFECT

To achieve the maximum performance, we first ran each benchmark with four threads on the big cores by fixing their clock to 2GHz. However, running all big cores at the highest clock frequency activated the thermal management (triggered at 95°C) which resulted in lowering the clock frequency to 900MHz. We then lowered the clock frequency and sought the maximum performance Odroid-XU3 could achieve.

Table 3: Thermal Management Effect. TM: % of time thermal management was enabled. MPC: Clock frequency for the maximum performance.

Benchmark	CJPEG	Core	Linear	Loops	Nnet
TM (%)	39.3	45.7	22.5	48.6	33.6
MPC (GHz)	1.7	1.8	1.7	1.7	1.6

Benchmark	Parser	Radix2	SHA	ZIP
TM (%)	22.2	14.1	44.1	23.1
MPC (GHz)	1.9	1.9	1.8	1.9

Table 3 shows the the percentage of the execution time during which the thermal management is activated (TM) and the clock frequency at which the maximum performance was achieved (MPC). Figure 2 shows the relative performance (left Y-axis) and the energy consumption (right Y-axis) of the the 2GHz execution against the execution with the MPC clock frequency in Table 3. Linear (linear algebra solver) was suffered from the thermal management most and running it with the 2GHz clock actually slowed it down to 64% of the 1.7GHz execution. Please note that the higher percentage of the thermal management activation did not necessarily result in the lower performance. We can see this example in Linear vs Loops: the latter has a higher thermal management percentage (22.5 vs 48.6) but the former has a lower relative performance (64.4 vs 82.4).

The center and right bars in Figure 2 show the relative energy consumptions of the 2GHz cases versus the maximum performance cases. In general, the total (DC) energy shows higher values than the CPU+Memory (P+M). While further investigation is required, the total includes the power consumption of the cooling fan, which consumes 0.84 Watt of power, and this could be the reason of total's being higher than the CPU+Memory. CJPEG benchmarks had the highest energy overhead. If it was run at 2GHz, it consumed 22 and 38 % more energy than 1.7GHz in CPU+Memory and total, respectively.

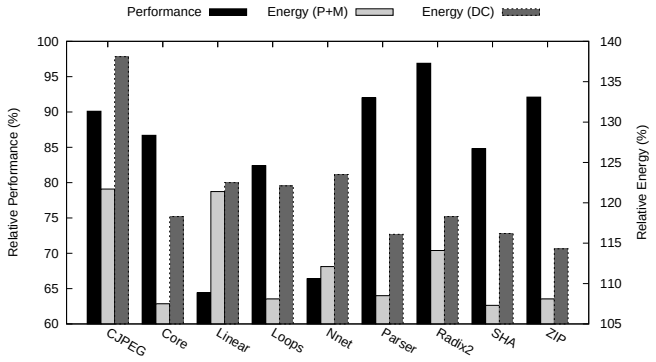

Figure 2: Effect of Thermal Management on Performance (left y-axis) and Normalized Energy (right y-axis).

4. BIG.LITTLE COMPARISON

Figure 3 compares the performance and energy consumption of the LITTLE cores against the big cores. The clock frequency of LITTLE cores is fixed at 1.4GHz (the highest frequency). The big cores were executed at the clock frequency listed in the MPC columns in Table 3 because the highest frequency (2GHz) resulted in sub-optimal performance and energy efficiency.

The relative performance of the LITTLE cores (left bars) varies widely among benchmarks, from 26% of Radix2 to 51% of Core. The relative energy consumption seems to be in a reciprocal relation to the performance: the higher the performance, the lower the energy consumption (vise versa). For example, Radix2 and Core consumed the most and the least energy, which is the opposite to the relation of these two programs in terms of performance. It should also be noticed that the total energy was smaller than the CPU+Memory energy. The possible reason would be because the cooling fan might be active only in the big core executions.

5. LOWERING CLOCK FREQUENCY

We lowered the clock frequency of the big cores and sought the frequency at which the big cores achieved the nearest throughput as the LITTLE cores (Figure 4). The right bars represent the clock frequency (GHz) at which the big cores performed the nearest throughput as the LITTLE cores.

In terms of energy efficiency, the LITTLE cores may not always be advantageous against the big cores as in Figure 3. In Radix2 and Parser (Linear, Loop, Nnet and Zip) the LITTLE cores consumed more (similar) energy per task than (as) the big cores. In other words, it might be better to run these workloads on the big cores with a lower clock frequency than on the LITTLE cores if the energy efficiency is the primary concern. Another characteristic we noticed in Figure 3

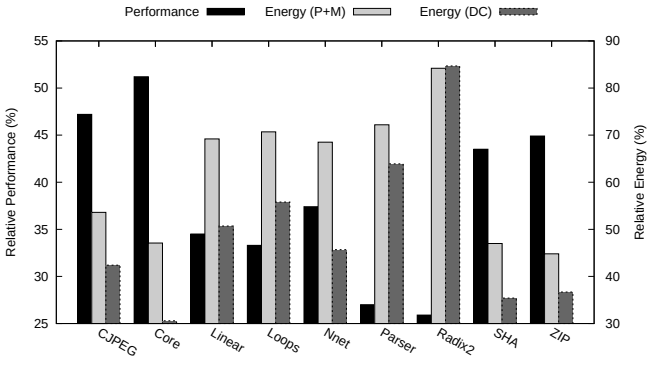

Figure 3: Relative Performance and Energy Consumption of LITTLE Cores against big Cores (maximum performance).

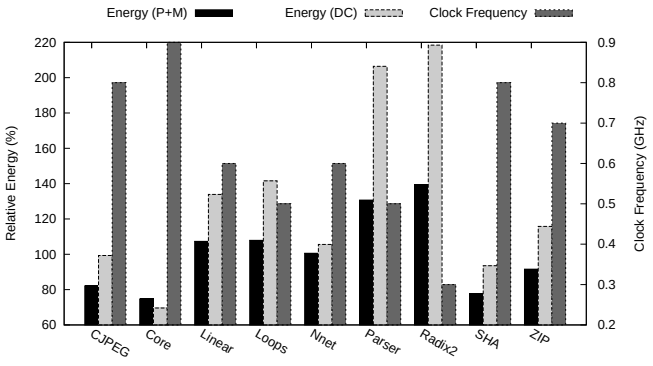

Figure 4: Relative Performance and Energy Consumption of LITTLE Cores against big Cores (lowered clock frequency).

was that the total was higher than the CPU+Memory. This may be because when the clock of big cores was lowered, the cooling fan was turned off and the energy overhead of the cooling fan was eliminated. Accordingly, the relative energy consumption of the LITTLE cores became higher.

In [2], core heterogeneity was emulated by lowering the clock frequencies for slower cores. However, the effect of lowering clock frequency varies among workloads in both performance and energy-efficiency.

6. DELAY INSERTION

SPEC Server Efficiency Rating Tool (SERT) compares the energy consumption of servers at different load levels [6]. For example, for the CPU-intensive workloads, load levels are 100, 75, 50 and 25%. SERT achieves these target load levels by inserting delays between tasks. Using this methodology, the performance of big cores were matched to that of LITTLE cores and the results are shown in Figure 5. The left and center bars for each benchmark show the relative energy of LITTLE cores against big cores (CPU+Mem and DC-input). The right bar indicates the duty cycle (execution time divided by turnaround including delay) in percent. The delay-insertion is not as effective as lowering clock frequency in most workloads: LITTLE cores can perform the same task at 31 to 80% (P+M) or 20 to 57% (DC) of the energy of big cores. The only exception is Radix2, for which LITTLE cores consume 59 and 14% more CPU+Memory and total energy than big cores, respectively. Radix2 seems to be able to take the performance advantage of the big cores and we can reduce the duty cycle to as low as 32% which should be one of the reasons of its energy-efficiency in the delay-insertion based execution.

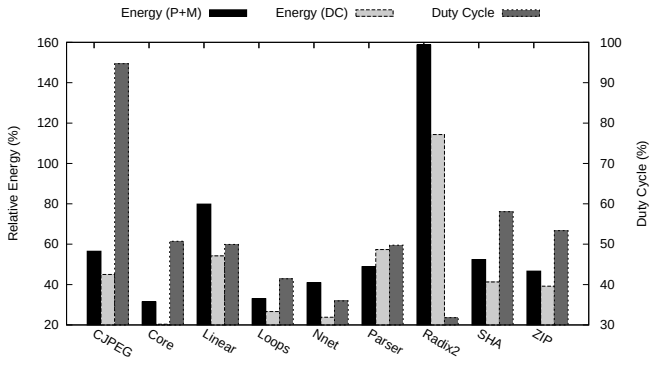

Figure 5: Relative Energy of LITTLE cores against big cores with delay (left) and Duty Cycle (right).

7. SUMMARY

In this work, we presented a case study of energy-efficiency on a platform with a heterogeneous multi-core processor. It was found that the thermal management could lower the performance and energy efficiency of big cores when run at the highest clock frequency. We compared the big and LITTLE cores, at their maximum performance and also at the matched performance by means of lowering clock frequency and delay-insertion.

Until recently, it was considered that only LITTLE cores could be profiled with the performance counters [7]. Actually, it was not true and a tool to profile both types of cores became available [8]. It is planned to confirm the results in this work and further investigate the behavior of the HMP using such tools.

8. REFERENCES

[1] Exynos Processor - Samsung , http://www.samsung.com/semiconductor/minisite/Exynos/w/

[2] Juan Carlos Saez et. al, "Leveraging workload diversity through OS scheduling to maximize performance on single-ISA heterogeneous multicore systems", Journal of Parallel and Distributed Computing, Vol. 71, Issue 1, Jan. 2011, pp114–131,

[3] Odroid-XU3, http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127

[4] Odroid Smart Power, http://www.hardkernel.com/main/products/prdt_info.php?g_code=G137361754360

[5] CoreMark®-Pro http://www.eembc.org/coremark

[6] Server Efficiency Rating Tool (SERT), https://www.spec.org/sert/ .

[7] "Access the performance-counter on Ubuntu/Linux, " http://odroid.com/dokuwiki/doku.php?id=en:odroid_linux_performance_counter

[8] "PMCTrack on Odroid-XU3/4," https://pmctrack.dacya.ucm.es/install/pmctrack-on-odroid-xu3-4/