

A New Core Level Utilization Algorithm for Energy-Efficient Multicore Systems

Samar Nour
Computer and Systems Engineering
Faculty of Engineering
Badr University
Cairo, Egypt

Sameh A.Salem
Computer and Systems Engineering
Faculty of Engineering
Helwan University
Cairo, Egypt

Shahira M.Habashy
Computer and Systems Engineering
Faculty of Engineering
Helwan University
Cairo, Egypt

Abstract—The energy consumption is becoming a constraint on all computer devices, from smartphones to supercomputers. Consequently, the focus has moved from performance to energy and power consumption. Design metrics are not only based solely on performance, as the energy performance of application executions is becoming the main aspect of architecture. Also, Design metrics depend on, the manufacturers of semiconductor chips which, have implemented multicore processors to boost the level of energy efficiency by using verified techniques for voltage and frequency scaling. To utilize the maximum potential of such architectures, we need to make the right decisions because parameters such as core type, frequency, and utilization typically affect power dissipation and performance. This paper proposes a new algorithm to achieve energy-efficient by monitoring core energy and level utilization control such as: Increasing the number of cores to execute the task, scaling voltage, and frequency. Based on the built model, we analyze the energy efficiency variations for different platform configurations providing the same level of performance. We show that trading the number and type of core with frequency and voltage level and core utilization rate can lead to substantial energy efficiency gains.

Index Terms—Energy efficiency, DVFS, power consumption, utilization, multicore

I. INTRODUCTION

According to Gartner, Inc., a consultancy group, the handheld industry for service is one of the fastest-growing industries of the Industry in computing [1]. Battery operated devices are continuously facing a different type of workloads with different performance requirements. All these modifications in software should be executed on hardware by means of sometimes a limited energy budget. For a particular task, savings often come from the right option of execution unit to use therefore the narrow amount of energy is one reason why many such computing devices are taking advantage of multiple processors system on chip (MPSoC) [1] [2]. The next increasing problem is how to map the execution on multicore platforms of parallel applications. This was an area under investigation with a wide range of goals. Some of the main goals are efficiency, fairness, predictability, reliability, etc ...

One of the strategic targets considered in the use of energy is MPSoC. The intrinsic metric of energy consumption that

depends on various considerations. Apart from the ratio of dynamic and static dissipation of strength, the time of execution of the application and the physical architectural elements must be considered in order to establish a winning minimization approach to the consumption of power. The optimal choice for mapping software to hardware is a challenging problem, as the form of workload is diverse [3].

To achieve improved efficiency and energy levels, one of the new strategies studied by the research community includes using multicores. Since the consideration of parameters such as core form, frequency, and core utilization rate affects the dissipation of power, we will explore the effect of these arguments on multicore energy efficiency in this article.

DVFS (Dynamic Voltage Frequency Scaling) and DPM (Dynamic Power Management) are two commonly used methods for energy-aware multicore scheduling. Processor power dissipation is divided into static power and dynamic power. DVFS works to increase system efficiency by reducing the supply voltage, reducing dynamic power, and improving overall energy dissipation [4]. Taking into account that the reduction in frequency contributes to decreasing utilization of the task, and the efficiency constraints of the task must be taken into account [5]. To minimize the leakage current, DPM is used to establish a particular point at which the core is shifted to sleep mode [6]. That leads to a reduction in its static strength. To decrease overall energy consumption, the selection of power management points should be carefully selected. A new hybrid technique has emerged between the DVFS and DPM methods, to achieve benefits from both techniques [7] [8]. The DVFS was mentioned in this paper.

Adjusting the number of cores to perform the task, we use this idea trying to get minimum energy and multicore efficiency, the use of tasks can be scalable [9]. We investigate how, dynamic voltage and frequency scaling (DVFS), number and type of core and core utilization influence the overall energy efficiency of multicores based computing systems.

The rest of the paper is organized as follows. Section II presents related work. In section. III The proposed algorithm, section. IV Experimental Setting. In sections. V and VI, Experimental Results of the quantitative evaluation review, and finally conclusions are introduced.

Identify applicable funding agency here. If none, delete this.

II. RELATED WORK

Several types of research have suggested DVFS-based solutions for real-time embedded systems operating on traditional multicore platforms as part of recent energy management research.

Recently, [10] submitted a survey of energy management techniques for embedded systems, while [11] submitted a survey focused on hard real-time systems. In this article [12], where the aim is to reduce energy consumption, they have discussed the problem of partition allocation in mixed-criticality systems. Instead of relying on new frequency adjustment scheduling algorithms to save resources, they recommend a partition for CPU distribution that takes into account, not just the different frequencies at which the CPU operates, but also the utilization of CPU.

In [13] The authors propose a runtime system in their work that tracks the performance of the applications through the heartbeat framework to reduce power under a performance requirement. They just consider the type of core to use and the frequency level in the configuration space. Unlike here, we also perceive the degree of utilization as being an extra parameter in the space for the configuration. Besides, they proposed an analytical model to measure the dissipation of energy, while we are constructing power models based on experimental data in our work. To achieve better energy efficiency, [14] writers explore parallelism within the task and how it can be utilized. During execution, parallel parts of the application are not known, as in some instances, even the minimum level of output expected is uncertain. By voicing these two parameters in the application directly, source code, a run-time power manager in different phases for resource allocation, they be achieved to make optimal decisions. In contrast with this work, in our paper besides DVFS and the level of parallelism, we exploit to scale utilization, which gives the possibility to choose between different numbers of cores, to evaluate the possibility to achieve better energy efficiency. The use of DVFS, and thread planning was fine. Studied for various works. Authors in [15] attempt to enhance the Efficiency of power by scaling and scheduling frequencies. They recommend a method for estimating the efficiency of power metric in various application stages, with different frequencies core forms, and using hardware output, this method counters (HPC) like the number of instructions fetched and retired, Cache hits/misses, expected branch numbers and IPC. Instead of concentrating on energy efficiency output counters, as a more useful measure, they focus on energy effectiveness. (For battery operating devices in particular). The economic model of price the theory was introduced in [16] to make the correct decision to reach a degree of achievement under the requirement of minimum power. In this proposed Framework It distributes and coordinates DVFS, migration of tasks, and load balancing to achieve the defined quality under a certain (TPD) thermal architecture.

Our work emphasizes, in comparison to the previous works, on energy management when delivering various applications and

comparing Performance on average. We consider the effect of the utilization of changing energy performance according to the application type.

III. THE PROPOSED ALGORITHM

With the improvements made in semiconductor manufacturing, Industry, technology, can provide an impressive Integration level at the transistor level. This mechanism gave birth to MPSoC. Nevertheless, with post-Dennard scaling [17] the power with every processor generation, density has increased. Recent studies indicate that the power for the same chip area after 2005 for each process technology, dissipation increased by a factor of 2 [18]. That's why the industry has changed in recent years. For architectures on a single chip with multiple core types, these architectures can offer programmable levels at different levels. Logic beside conventional cores, which altogether can result in comparison with symmetric architectures, a more convenient choice is [19] [20]. It is possible for the platforms based on this technology to offer execution to a wide range of workloads, varying from one workload to another. Databases of memory (requiring a small computational power) to Multimedia applications that may be hungry for computing. Workload mapping followed on many-core architectures aimed to achieve specific goals such as efficiency, latency, throughput, and reliability [21] [22]. Mapping for energy efficiency is one of the latest concerns. In addition to unused cores for power gating, the conventional cores DVFS and techniques for achieving energy conservation used varying core types in modern architectures.

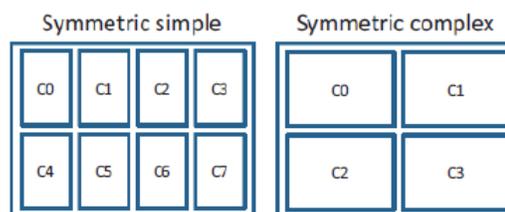


Fig. 1: Symmetric Multicore.

To the best knowledge, the architecture of a multicore processor enables the communication between all available cores to ensure that the processing tasks are divided and assigned accurately. One of the most popular homogeneous design of the cores is a processor with symmetrical cores Figure 1 [21]. The cores included in this form of the processors are identical to each other and are intended to be used for all tasks and types of purposes. The benefit of these processor types is that because only one type of core design exists, Compared to processors with asymmetrical cores, developing applications for them is easier. Moreover, because the cores are generalized, it is easier to apply the unused processing power of one core to accomplish the task of another. The only obvious drawback of this model is, of course, that it cannot be optimized to perform a specific type

of task because the cores are designed for general use [24]. In our work, we use Symmetric MultiCore to get minimum power and run one task on more than one core to get the best utilization and minimum energy.

[25] A learning-directed DVFS algorithm for single-core and multicore embedding was proposed in this paper. Although this paper proposed an effective DVFS learning-directed method for single-core and multicore embedded platforms, it is possible to further develop and extend the proposed technique to high-end high-performance computing (HPC) systems. Many studies [26] [27] are working on similar issues for HPC systems.

In our work we focus on architectures that are composed of Symmetric MultiCore that can be enabled simultaneously in a more lightweight process.

[28] They suggested the proper scenario features for efficient control of the power of new mobile devices and a new scenario-aware DVFS policy has been proposed that adjusts CPU clusters' operating frequencies. The Reform Plan The level of parallelism is considered to provide sufficiently processing speed for optimum energy performance. We consider a system with full HMP instead of a multicore scheduling considered from the authors. This gives us more flexibility in scheduling decisions.

This paper considers the ability to regulate the degree of the load to further improve the energy efficiency of each core, varying energy efficiency constraints as shown in Algorithm 1. As a limitation of results, we consider the necessary throughput levels. Recently, the new gem5 simulator offers the possibility of determining the extent of the utilization of tasks. The gem5 Simulator [29] as a new element in our study we will explore the task utilization parameter to obtain a near-optimal configuration. We consider a Symmetric Multicore in this paper it consists of four separate cores, each one has performance-optimized and energy-optimized, where each core can have a single voltage and a corresponding frequency level as shown in Figure 1.

We address the question in this paper about which platform the configuration offers the most energy-efficient implementation of, under various constraints. Using the following headings assumptions:

- the application has a configurable level of parallelism
- the platform provides as actuators different DVFS levels and core utilization rates

The core utilization is also considered in this work as an indeed, using the newly added parameter, The Simulator Gem5. indeed, previous work has already been completed, It has been shown that the core usage rate can have a direct effect on multicore architectures' power efficiency. We describe the configurations of the platform as:

- number of parallel instances from the application to be executed
- number and type of cores to utilize

- DVFS level of the core
- load level (utilization rate) of each core

Application analysis would be focused on basic performance analysis, and Levels that the application includes. Taking into consideration the number of DVFS levels, the number of cores for each task, each core can provide, and utilization levels there, There is a vast range of possible configurations to consider. For example, if we have N cores , each core type has Fh(high frequency) and FL (low frequency). Frequency levels and the levels of utilization of the tasks are L, then the number of configurations will be as equation(1):

$$C = N * Fh * FL * Lh * LL + N * Fh * Lh + N * FL * LL \quad (1)$$

In this analysis, we assume that DVFS can be applied to the Core level and that the task has all activities mapped to a core, which means the same level of utilization (load level). A diagram of the steps involved in the Flow chart for The Proposed Algorithm can be seen in Figure 2.

Algorithm 1: The Proposed Algorithm.

```

1 Li : System level at certain frequency and certain
  voltage
2 L1 : low frequency
3 Lh : high frequency
4 Cn : The number of cores ∈ N
5 Ui : Core utilization at V/F level Li
6 Ei : Energy at at V/F level Li
7 Eb : best Energy
8 Ub : best utilization
9 Set the frequency to minimum V/F Level L1
10 Calculate the new Ui, Ei and power
11 for Li from L1+1 to Lh do
12     for each Ci i ∈ N do
13         Running each task on Ci
14         Calculate the new Ui, Ei and power
15         if no such Ei exists then
16             if no such Ui exists then
17                 Eb=min{Ebold , Ei}
18                 Ub=argUi min{Eb}
19                 Save V/F leavel at Eb and Ub
20             end
21         end
22     end
23 Find V/F level L that corresponds to minimum Ei
  and add it to the best Ui.
24 end

```

IV. EXPERIMENTAL SETTINGS

A. Platforms

This research was conducted on the state-of-the-art of Microservers ARMv8: Applied Micro's (now Ampere's) Computing) X-Gene 3, which consists of 4 cores compatible with 8 and 32 64-bit ARMv8. Both microprocessors provide high-end

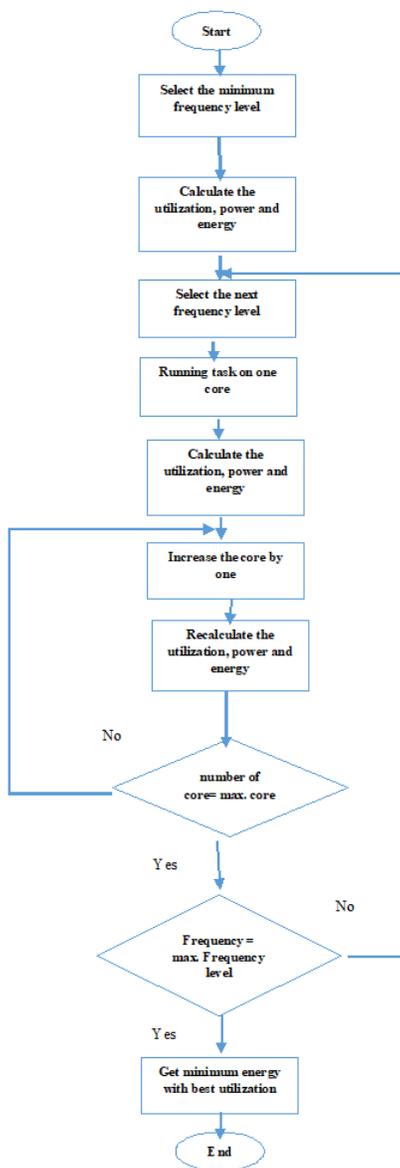


Fig. 2: Flow chart for The Proposed Algorithm.

efficiency in processing and come with a scalable subsystem. Cpu for Lightweight Intelligent Control (SLIMpro) in time to enable power management flexibility, resiliency, and end-to-end security for a wide range of applications.

The dedicated processor SLIMpro tracks system sensors, configures system sensors, Device attributes (e.g. regulation of supply voltage, etc.) which can be accessed by a Linux kernel running on the system. The key power domain of the X-Gene 3 microprocessor includes the CPU cores, L1, L2, and L3 cache memories, as shown in Figure 3, the memory controller, which is called the PCP (Processor Complex) power domain, is the one that consumes the largest part of the overall power consumption. Figure 3 presents the architecture of X-Gene3.

The operating voltage of the primary domain of control In X-

Gene 3, can shift from 870mV downwards. Although all the cores for the CPU In both chips, each pair of cores (PMD-Processor MoDule), can work at the same voltage at different frequencies. The frequency in XGene3 will range from 375MHz to 3GHz (at 1/8 of the maximum clock frequency steps of both microprocessors).

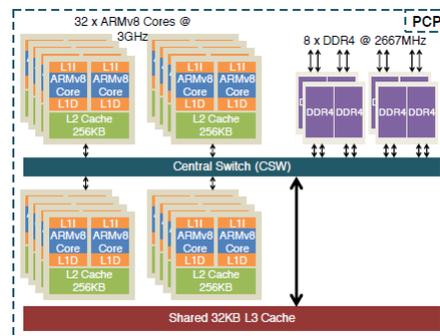


Fig. 3: X-Gene 3 block diagram.

B. Experimental Configuration

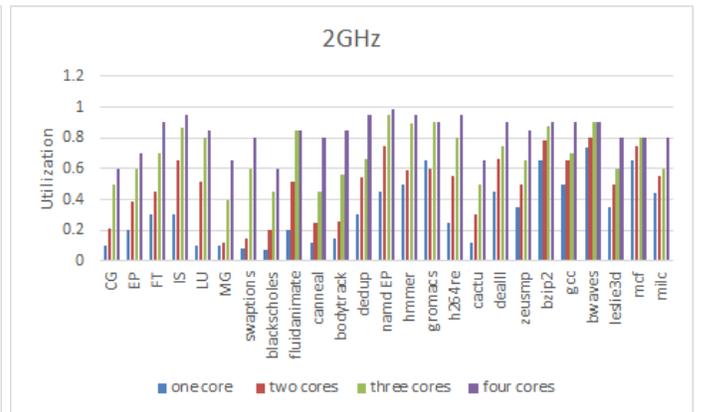
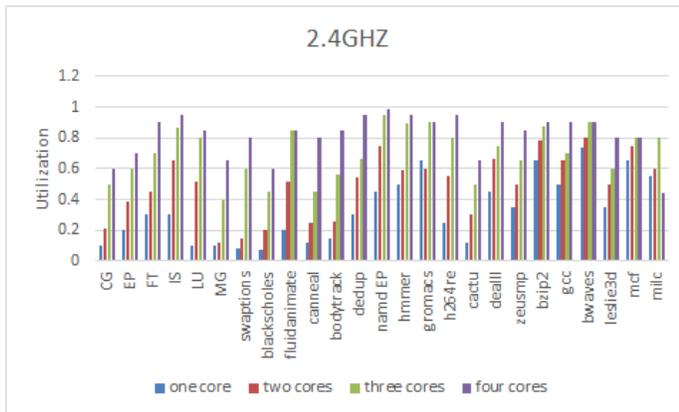
We use 25 benchmarks out of 3 separate benchmarks in our study. Parallel Benchmark Suites: The Parallel Benchmark Suite NAS V3.3.1 (NPB) [30], the suite Of SPEC CPU2006 [31], and PARSEC suite v3.0 [32]. NPB are systems developed for the purpose of Assess the efficiency of parallel supercomputers and they have been used in many performance and energy efficiency studies [33] [34].

For each experiment we run the workload with utilization levels from 10% to 90%, avoiding the run at 100%. when executing a task on more than one core, the utilization is increasing every time. First, with one instance running on only one core, we explore the entire range of available frequencies and after that increase by another one core with a varying utilization from 10% to 90%. For each experiment, we log the power data energy and total utilization and collect also the Performance data from the application in terms of operations per second. Each data point in the graphs is obtained by running experiments 10 times. For each frequency and each utilization level, we take the average power dissipation from the logs.

In the second stage of the experiments, we measure the power dissipated by running multiple cores in each round. The frequency governor cpufreq in Linux gives the possibility to define 5 frequency levels on the cores, from 900MHz to 2.4GHz. Those intervals correspond to two discrete voltage levels for driving the cores. The voltage and frequency levels for both cores are shown in Table 1.

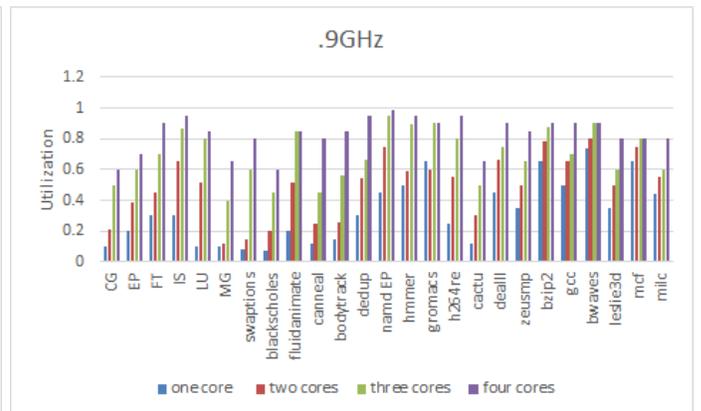
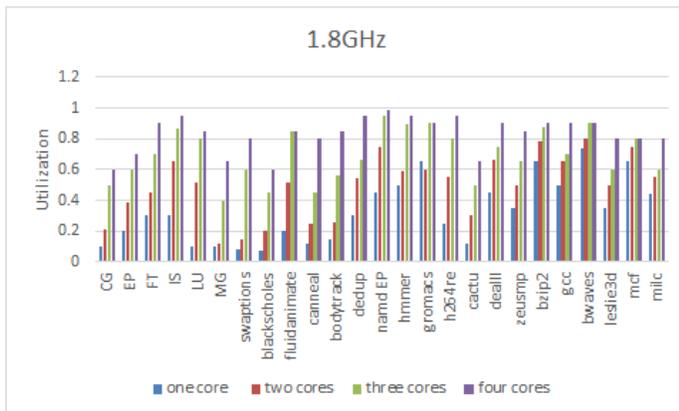
TABLE I: Frequency and voltage relation for X-Gene 3.

Core Frequency (GHz)	.9	1.2	1.8	2	2.4
Core Voltage (mV)	870			888	



(a) Utilization at 2.4 GHZ

(b) Utilization at 2 GHz



(c) Utilization at 1.8 GHZ

(d) Utilization at 0.9 GHz

Fig. 4: Experimental Results on different benchmarks at various frequency levels.

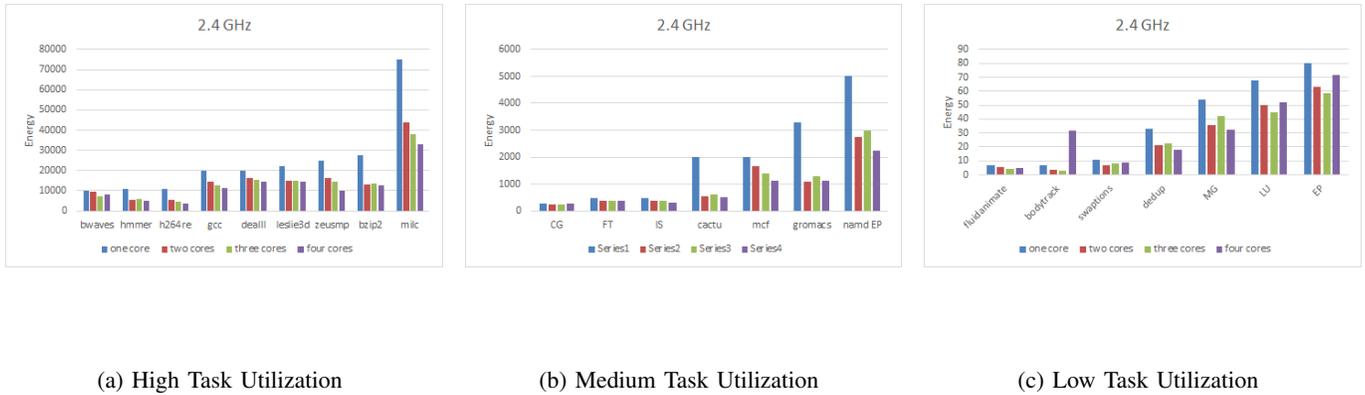


Fig. 5: Energy efficiency characterization results for SPEC benchmarks of on X Gene 3 at 2.4 GHZ with varying number of cores.

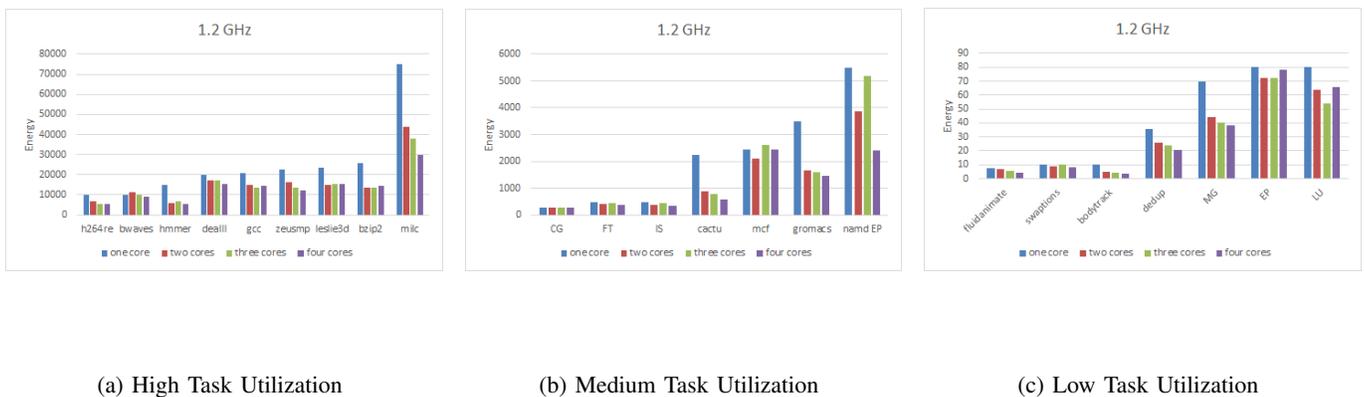


Fig. 6: Energy efficiency characterization results for SPEC benchmarks of on X Gene 3 at 1.2 GHZ with varying number of cores.

V. EXPERIMENTAL RESULTS

For each core form, we developed an energy model using experimental settings mentioned in the section before. Based on the energy model obtained, at the split application of several cores, we derive the energy efficiency of the different cores and measure total utilization at different frequencies. For all possible platform configurations, we list in a table the energy efficiency and the corresponding performance rating. We can therefore use this table to select the most energy-efficient configurations for a level of performance. We can therefore use this table to select the most energy-efficient configurations for optimal performance.

A. Core Utilization

CPU utilization refers to the sum of a CPUs work handled. Depending on the quantity and type of computational activities, actual CPU utilization varies. Some tasks need a significant amount of Processor power, while others need less because Of specifications for non-CPU resources. In other words, the Utilization of the CPU is the proportion of the total available processor cycles that each operation utilizes. Figure 4, tracks the utilization of the processor (average for varying frequencies in all work sets). When running the task on one, two, three, or four cores, as shown in the figure, as the number of cores increases, the use of the system also increases, but when the frequency increases, the utilization of the system decreases, so that when the frequency decreases, we watch saving in the system energy, and we can compensate for the number of cores and get more use of the system.

B. Energy efficiency results

From the power measurement data, we can assess the energy efficiency of each core type under different levels of frequencies. We express the energy efficiency using the achieved number of operations per joule metric. For different utilization and frequency levels, we derive the corresponding core level energy efficiency. Figures 5 and 6 show the energy efficiency X-Gene 3 cores respectively, using one, two, three, and four cores to running each task under different Utilization. From the figures, we observe for all frequency a non-linearity of the energy efficiency. Therefore, we can expect different efficiency levels for different platform configurations if the utilization rate of the core can be controlled. From the figures we notice the efficiency variation range from around 10% to 60% depending on the frequencies and utilization of core. As explained when the utilization increases the energy decrease because the execution time of the task is decreasing that happens on some tasks like sequence task as shown in figures 5 and 6, for example, milc and mcf. Unlike parallel tasks the energy increase or does not affect by increasing the utilization of the system as shown the figures 5 and 6, for example, tasks like EP and LU. In another word, the idea of increasing system utilization by increasing the number of cores in different levels of frequency is efficiency for application and not efficiency for others.

VI. CONCLUSION

In this paper, a new algorithm has been introduced to improve energy efficiency while exploiting homogeneity, voltage and frequency scaling, and utilization rate control techniques at the same time. For various platform configurations that provide the same level of performance, we measured the variance in energy efficiency. We demonstrated that changing the frequency and voltage level and the core utilization rate of the number of cores and form of the core will lead to significant energy gain. Simultaneously, we reached a combination of energy efficiency and performance unlike the usual in the field of research later and as shown in the result that was explained before, where we get energy efficiency and the best utilization rate with different platform configurations.

In future work, the general model focused on a better instruction mix, providing the probability of reflecting a wider range of the modern world Applications. The probability of running time will be weighed and the system to split the execution of an application into Phases will be also studied. For each phase, the appropriate architecture configuration can be established and selected at runtime. This will have a runtime system that is energy efficient at the granularity of the Phases implementation.

REFERENCES

- [1] Nunez-Yanez, Jose. "Energy Proportional Heterogenous Computing with Reconfigurable MPSoC." International Conference on High Performance Computing & Simulation (HPCS), 2019.
- [2] Junior, Francisco Carlos Silva, Ivan Saraiva Silva, and Ricardo Pezzuol Jacobi. "Evaluation and Proposal of a Lightweight Reconfigurable Accelerator for Heterogeneous Multicore." IEEE Latin America Transactions, 2020.
- [3] Salami, Bagher, Hamid Noori, and Mahmoud Naghibzadeh. "Fairness-Aware Energy Efficient Scheduling on Heterogeneous Multi-Core Processors." IEEE Transactions on Computers 2020.
- [4] W. Knight, Two heads are better than one [dual-core processors], in IEEE Review, vol. 51, no. 9, pp. 3235, September 2005.
- [5] J. W. S. Liu, Real-Time Systems, Upper Saddle River, NJ, USA: Prentice-Hall, 2000.
- [6] E. Seo, J. Jeong, S. Park, and J. Lee, Energy efficient scheduling of realtime tasks on multicore processors, in IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 11, pp. 15401552, November 2008.
- [7] Hassan, Hadeer A., Sameh A. Salem, and EL-Sayed M. Saad. "Energy Aware Scheduling for Real-time Multi-Core Systems.", in International Journal of Computer Science Engineering (IJCE) 2018.
- [8] Hajiaminia, Shervin, and Behrooz A. Shirazib. "A study of DVFS methodologies for multicore systems with islanding feature." Advances in Computers 119, 2020.
- [9] Nour, Samar, and Shahira Mahmoud. "ARMSS: Adaptive Reconfigurable Multi-core Scheduling System.", in International Journal of Scientific & Engineering Research Volume 9, Issue 9, September-2018.
- [10] Mittal S. A Survey of Techniques For Improving Energy Efficiency in Embedded Computing Systems. CoRR, 2014.
- [11] Bambagini M, Marinoni M, Aydin H, Buttazzo G. Energy-Aware Scheduling for Real-Time Systems: A Survey. ACM Trans Embed Comput Syst, 2016.
- [12] Guasque, Ana, et al. "Energy efficient partition allocation in mixed-criticality systems." PloS one 14.3, 2019.
- [13] E. Del Sozzo, G. C. Durelli, E. M. G. Trainiti, A. Miele, M. D. Santambrogio, and C. Bolchini. Workload-aware power optimization strategy for asymmetric multiprocessors. In 2016 Design, Automation Test in Europe Conference Exhibition (DATE), pages 531534, March 2016.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US

- [14] S. Holmbacka, E. Nogues, M. Pelcat, S. Lafond, and J. Lilius. Energy efficiency and performance management of parallel dataflow applications. In 2014 Conference on Design and Architectures for Signal and Image Processing (DASIP), pages 18, October 2014.
- [15] Arunachalam Annamalai, Rance Rodrigues, Israel Koren, and Sandip Kundu. An opportunistic prediction-based thread scheduling to maximize throughput/watt in AMPs. In Proceedings of the 22nd international conference on Parallel architectures and compilation techniques, pages 6372. IEEE Press, 2013.
- [16] Thannirmalai Somu Muthukaruppan, Anuj Pathania, and Tulika Mitra. Price Theory Based Power Management for Heterogeneous Multi-cores. In Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 14, pages 161176, New York, NY, USA, ACM, 2014.
- [17] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc. Design of ion-implanted MOSFETs with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256268, October 1974.
- [18] M. B. Taylor. A landscape of the new dark silicon design regime. In 2014 Design, Automation Test in Europe Conference Exhibition (DATE), pages 11, March 2014.
- [19] D. H. Woo and H. H. S. Lee. Extending Amdahls Law for Energy-Efficient Computing in the Many-Core Era. *Computer*, 41(12):2431, December 2008.
- [20] E. S. Chung, P. A. Milder, J. C. Hoe, and K. Mai. Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPUs In 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture, pages 225236, December 2010.
- [21] Sparsh Mittal and Jeffrey S. Vetter. A Survey of CPUGPU Heterogeneous Computing Techniques. *ACM Comput. Surv.*, 47(4):69:169:35, July 2015.
- [22] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel. Mapping on multi/many-core systems: Survey of current and emerging trends. In 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pages 110, May 2013.
- [23] Mihai Pricopi and Tulika Mitra. 2014. Task scheduling on adaptive multi-core. *IEEE transactions on Computers* 63, 10 2014.
- [24] Ke Ning, Gabby Yi, and Rick Gentile. 2005. Single-chip Dual-core Embedded Programming Models for Multimedia Applications. *ECN Magazine* 2005.
- [25] Chen, Yen-Lin, et al. "Learning-Directed Dynamic Voltage and Frequency Scaling Scheme with Adjustable Performance for Single-Core and Multi-Core Embedded and Mobile Systems." *Sensors* 18.9, 2018.
- [26] Etinski, M.; Corbalan, J.; Valero, M. Understanding the future of energy-performance trade-off via DVFS in HPC environments. *J. Parallel Distrib. Comput.* 2012.
- [27] Calore, E.; Gabbana, A.; Schifano, S.F.; Tripiccion, R. Evaluation of DVFS techniques on modern HPC processors and accelerators for energy-aware applications. *Concurr. Comput. Pract. Exp.* 2017.
- [28] Butko, Anastasiia, et al. "Exploration of performance and energy trade-offs for heterogeneous multicore architectures." *arXiv preprint arXiv:1902.02343*, 2019.
- [29] Lowe-Power, Jason, et al. "The gem5 Simulator: Version 20.0+." *arXiv preprint arXiv:2007.03152*, 2020.
- [30] NAS Parallel Benchmarks Suite, v3.3.1. <https://www.nas.nasa.gov/publications/npb.html>.
- [31] J. L. Henning, Spec cpu2006 benchmark descriptions, *SIGARCH Comput. Archit. News*, vol. 34, pp. 117, Sept. 2006.
- [32] C. Bienia, S. Kumar, J. P. Singh, and K. Li, The parsec benchmark suite: Characterization and architectural implications, in Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, PACT 08, (New York, NY, USA), pp. 7281, ACM, 2008.
- [33] B. Lepers, V. Quema, and A. Fedorova, Thread and memory placement on numa systems: Asymmetry matters, in Proceedings of the 2015 USENIX Conference on Usenix Annual Technical Conference, USENIX ATC 15, (Berkeley, CA, USA), pp. 277289, USENIX Association, 2015.
- [34] M. Curtis-Maury, Improving the Efficiency of Parallel Applications on Multithreaded and Multicore Systems. PhD thesis, Virginia Tech, 2008..