A Decision Framework for Intra Task Fixed Priority INTEL PXA270 Distributed Architecture for Soft RT- Applications Based on Deep Learning

Nasir Ayub
Department of Computer Science, Faculty of Computer Science & IT, Superior University Lahore, Pakistan
nasir.ayyub@hotmail.com

Muhammad Atif Imtiaz
School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Australia
matif@uow.edu.au

Esraa Ali
Faculty of Engineering Communications and Computer Engineering Department, Al-Ahliyya Amman University, Jordan
e.ali@ammanu.edu.jo

Abdullah M. Alqahtani
College of Engineering & Computer Science, Department of Electrical & Electronic Engineering, Jazan University, Saudi Arabia
amqahtani@jazanu.edu.sa

Arshad Ali
Faculty of Computer and Information Systems, Islamic University of Madinah, Al Madinah Al Munawarah, Saudi Arabia
a.ali@iu.edu.sa

Mirjalol Ashurov
Department of Information Technology in Mathematics and Education, Tashkent State Pedagogical University, Uzbekistan
author.uzb@mail.ru

Sami Albouq
Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia
salbouq1@iu.edu.sa

Foong Li Law
Department of Computer Science and Software Engineering, School of Computing, Asia Pacific University of Technology & Innovation (APU), Wilayah Persekutuan, Kuala Lumpur, Malaysia
foongli.law@apu.edu.my (corresponding author)

Received: 22 December 2024 | Revised: 30 January 2025 and 23 February 2025 | Accepted: 24 February 2025

 DOI: https://doi.org/10.48084/etasr.10006

ABSTRACT

Distributed architectures with fixed-priority scheduling using Dynamic Power Management (DPM) for CPU optimization are one of the serious concerns in INTEL PXA270. Increasing the number of transistors and task mapping on chips causes greater energy dissipation and power consumption. This study addresses the issue of system-level higher CPU energy dissipation during the execution of parallel workloads with common deadlines by introducing a framework that includes task migration based on DPM and an Adaptive Deadline First scheduling (A-DF) scheme to properly schedule migratable tasks. The DPM policy and efficient task allocation and scheduling using A-DF enhance overall throughput and optimize energy consumption to avoid delays and performance degradation in multiprocessor systems. The proposed model assigns processors to the ready task set to meet deadline requirements. A full task migration policy is also integrated to ensure proper task mapping and interprocess linkage among tasks with the same deadlines. The execution of a task can pause on one CPU and reschedule execution on another to avoid delay and ensure that the deadline is met. The proposed method shifts the context of the task from running to sleep and from idle to sleep using an adaptive DPM approach. The proposed scheme showed a promising reduction in energy dissipation compared to other conventional energy-aware task migration techniques. Simulations were conducted using a super pipelined microarchitecture Intel XScale PXA270 using instruction and data cache per core of 32 Kbyte I-cache and 32 Kbyte D-cache on various utilization factors (u_i) of 18% and 20%. The proposed approach consumed 6.3% less energy and achieved 2.1% and 2.4% improvements in terms of accuracy and precision when almost half of the CPU is running, and on a lower workload consumed 1.04% less energy. The proposed design provided significant improvements in clock rates of 100, 104, and 116 MHz.

Keywords-task migration; optimization methods; AI; ML; distributed computing; multiprocessor systems-on-chip; edge computing; data analysis; model evaluation; IoT; latency reduction

I. INTRODUCTION

Multicore processors are rapidly becoming the norm for embedded real-time systems due to the demand for multimedia and gaming systems. These processors offer greater adaptability, providing a substantially more productive environment for task assignments [1, 2]. This study presents a novel technique to reduce CPU energy consumption in multicore frameworks, reducing Worst-Case Execution Time (WCET) and the absolute energy utilization of the framework [3, 4]. In [5], dispatching domains were established along with their representations to maximize processing speed on multiprocessor platforms. Integration of components within MPSoCs results in lower energy consumption and power requirements [6, 7]. The energy consumption of MPSoCs occurs in many abstractions, from the logic level to the circuit. An MPSoC is an integrated circuit designed for applications with specific software and hardware [8]. A scheduling technique was proposed to assess CPU energy and memory utilization of a central framework for multimedia applications based on a Constant Bandwidth Server (CBS) [9, 10]. The energy-efficient CPU scheduler maps data using the Earliest Deadline First-Based Window constraint Migration (EDF-WM) for multiprocessors, in which memory allocation uses paging calculations called Shared Anonymous Private Pages (PSAP) [11-12]. In addition, an energy-aware technique is required for smooth task allocation to increase efficiency and result in increased CPU clock speed [13, 14].

A. Reference Architecture

Previous advances in CPU technology have offered better responses and improved execution times [15, 16]. An MPSA-based technique can optimize scheduling to increase throughput and meet the processor's mapping requirements [17]. The dissipation of energy refers to the discharge of energy from an MPSoC-based embedded system and occurs when the embedded system utilizes energy for its operation. Shrinking and increasing electronic transistors and frequency increases power density, causing many problems such as increased power consumption and thermal issues as well as higher energy dissipation [18]. A dvfs scheme can monitor parallel irregular divide-and-conquer algorithms to improve both performance and energy efficiency in frequency and voltage regulation [19, 20].

This study introduces a mechanism where the energy consumption (E_{cc}) of executing a task on an MPSoC mainly by switching resources, as discussed in [21, 22], is given by:

$$E_{cc} = P_{switching}.t = C_{eff}.V^2.f\frac{C_b}{f} \quad (1)$$

$$E_{cc} = C_{eff}.V^2.C_b(V_{dd} - V_{ths})/.{V_{dd}}^2 \quad (2)$$

The consumption of dynamic power due to the transistor's switching can be calculated using [23]:

$$P_{dy} = C_l * N_c * V_d * f \quad (3)$$

$$f = M * (V_d - V_t)/{V_d}^2 \quad (4)$$

B. Distributed Multiprocessor Architecture

C_l , V_d , N_c , and the number of state transitions with frequency f for fixed priority are set, while V_t represents the threshold voltage. L_g denotes the logic gates used in the CMOS chip [24]. The reduction constant is denoted as M as shown in (4). The accumulative power of the CMOS is:

$$P_t = (C * {V_d}^2 f \; + \; L_g * (V_d * M_3 \;) \quad (5)$$

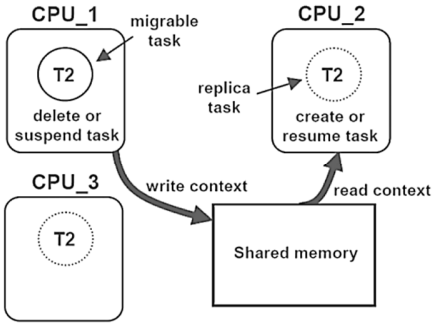

Fig. 1. Generalized task migration architecture.

The following are common assumptions used in the convergence analysis of the FL optimization algorithm [25, 26]. $f(x)$ is β-Lipschitz continuous if there exists $\beta \geq 0$ such that for all $x1, x2 \in Rd$:

$$|f(x1) - f(x2)| \leq \beta \left\| x1 - x2 \right\| \quad (6)$$

$f(x)$ is L-smooth if it has an L-Lipschitz continuous gradient, i.e., for all $x1, x2 \in Rd$:

$$f\left\| \nabla f\ (x1) - \nabla f\ (x2) \right\| \leq L \left\| x1\ - x2 \right\| \quad (7)$$

$$\left\| \nabla f\ (x1) - \nabla f\ (x2) \right\| \leq L \left\| x1\ - x2 \right\| \quad (8)$$

- Strongly Convex Objective Function (SCOF): $f(x)$ is μ-strongly convex if there exists an $\mu \geq 0$ such that for all $x1, x2\ \in\ Rd, f(x1) \geq f\ (x2) + (x1 - x2)T\nabla f(x2) + \mu$.
- Coercive Function (CF): $f(x)$ is coercive if $\lim_{||x||\to\infty} f(x) \to \infty$.

$$f(x1) \geq f(x2) + (x1 - x2)T\nabla f(x2) \quad (9)$$

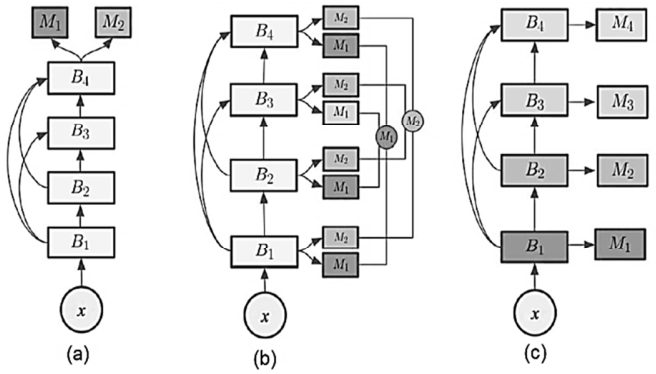

Fig. 2. Migration stages: (a) Fixed priority, (b) Deadline-aware, (c) Hybrid hierarchy.

C. Migration in Adaptive Runtime Manager

In [27], a task migration mechanism was proposed to move an executing task from one host CPU in a distributed architecture to another. The selection of a task and movement to the host CPU for a new task and the creation of the task on that host increase performance, reliability, and processing speed. However, the lack of on-time task migration can affect the overall system performance. Most current techniques improve energy and power management-related problems. The mapping of ready task allocation to the CPU is another approach to scheduling. Three main classes of task scheduling exist: multiprocessor partitioned scheduling, restricted-migration scheduling, and full-migration scheduling [28, 29].

II. PROPOSED DECISION FRAMEWORK

The scheduling algorithm applies migration on a task $t_i \in \tau$ when a multiprocessor system allows it. This task starts execution on one core of the processor and later switches, due to high utilization, and migrates it to another core.

A. Adaptive-Deadline First (A-DF) for Multiple Cores Based on Deep Learning

In the proposed deep learning-based A-DF strategy for Intel distributed architecture, all tasks $t_i \in \tau$ are allowed to migrate at any point as per the condition of u_i during their execution. However the $t_i \in \tau$ cannot execute in parallel on multiple cores but can execute on any single core selected from the configuration at a time. This strategy is more flexible and increases performance by allowing migration. Figure 3 shows the full migration of tasks using A-DF. The task set contains a ready task set that is allocated to the scheduler. The proposed scheduler monitors the utilization factor and migrates the t_i according to the configurations mentioned in (10), in which c_i refers to the continuous integration of a critical deadline of the current task and τ_i is a CPU that includes a ready task set.

$$\sum_{i=\{1.,.,.n\}/k}^{n} \frac{c_i}{\tau_i} + \frac{c_k+\alpha_k}{\tau_k} \quad (10)$$

Equations (11) to (15) show that with an increase in c_i the deadline of a current task parameter, such as sleep state and deep sleep state of $c_k + \alpha_k$, gradualy increases to unity while the τ_i of the specific CPU architecture that contains a ready task also enhances with each task execution up to n tasks. c_j and α_j are the variable running tasks on the CPU that depend on CPU utilization as shown in (15).

$$\sum_{i=\{1.,.,.n\}/(k,j)}^{n} \frac{c_i}{\tau_i} + \frac{c_k+\alpha_k}{\tau_k} + \frac{c_j+\alpha_j}{\tau_j} = 1 \quad (11)$$

$$\sum_{i=\{1.,.,.n\}/(k,j)}^{n} \frac{c_0}{\tau_0} + \frac{c_0+\alpha_0}{\tau_0} + \frac{c_0+\alpha_0}{\tau_0} \quad (12)$$

$$\sum_{i=\{1.,.,.n\}/(k,j)}^{n} \frac{c_1}{\tau_1} + \frac{c_1+\alpha_1}{\tau_1} + \frac{c_1+\alpha_1}{\tau_1} \quad (13)$$

$$\sum_{i=\{1.,.,.n\}/(k,j)}^{n} \frac{c_3}{\tau_3} + \frac{c_3+\alpha_3}{\tau_3} + \frac{c_3+\alpha_3}{\tau_3} \quad (14)$$

$$\sum_{i=\{1.,.,.n\}/(k,j)}^{n} \frac{c_n}{\tau_n} + \frac{c_n+\alpha_n}{\tau_n} + \frac{c_n+\alpha_n}{\tau_n} \quad (15)$$

III. RESULTS AND DISCUSSIONS

In the experimental evaluation, a uniform multiprocessor was considered using STORM for the multithreaded application that executes an XML file containing the task parameters for $t_i \in \tau$: $n = 4, n = 10$, and $n = 17$. The period ranges between [4-25] ms. The task set in Table I on $u_i = 12$ and $u_i = 20$ were evaluated for $s_d = 1000$ ms simulation duration for the embedded architecture supporting symmetric as well asymmetric CPU arrangement with m CPUs. Task t_i is allocated and mapped on the CPU cores ($\lambda_2, \lambda_4, \lambda_6 \in \lambda_n$).

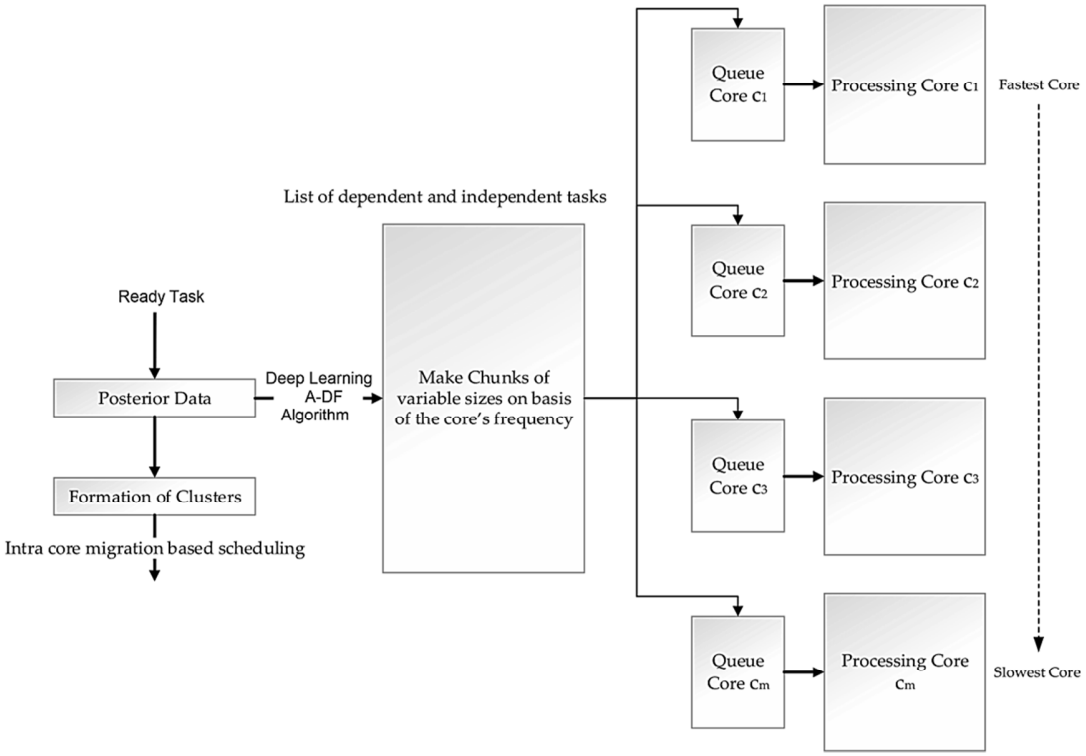

Fig. 3. Proposed task-migration model based on A-DF.

TABLE I. PARAMETERS FOR SYSTEM TIMING REQUIREMENTS (MS) AT $u_i = 20\%$ FOR TASK SET τ $N = 8$ AND $m = 4$

Tasks τ	$p_i(ms)$	$r_i(ms)$	d_i	p_r
t_1	4	4	4	7
t_2	15	10	15	8
t_3	16	11	16	5
t_4	15	8	15	6
t_5	4	4	4	7
t_6	11	16	5	11
t_7	16	11	16	5
t_8	15	8	15	6

In (16) to (20), $\delta^{(n)}$ shows that ∂L current task parameters with deep sleep state of $\partial z^{(n+1)}$ increases to unity while $\partial a^{(n)}$ is the normal ratio of task architecture that contains a ready task and also enhances with each task execution up to n task. $\partial a^{(n)}$ and $\partial z^{(n)}$ are variables with stable tasks on the multiprocessor whose utilization is variable according to the task.

$$\delta^{(l)} = \frac{\partial L}{\partial z^{(l)}} = \delta^{(l+1)} \cdot \frac{\partial z^{(l+1)}}{\partial a^{(l)}} \cdot \frac{\partial a^{(l)}}{\partial z^{(l)}} \quad (16)$$

$$\delta^{(0)} = \frac{\partial L}{\partial z^{(0)}} = \delta^{(0+1)} \cdot \frac{\partial z^{(0+1)}}{\partial a^{(0)}} \cdot \frac{\partial a^{(0)}}{\partial z^{(0)}} \quad (17)$$

$$\delta^{(1)} = \frac{\partial L}{\partial z^{(1)}} = \delta^{(1+1)} \cdot \frac{\partial z^{(1+1)}}{\partial a^{(1)}} \cdot \frac{\partial a^{(1)}}{\partial z^{(1)}} \quad (18)$$

$$\delta^{(2)} = \frac{\partial L}{\partial z^{(2)}} = \delta^{(2+1)} \cdot \frac{\partial z^{(2+1)}}{\partial a^{(2)}} \cdot \frac{\partial a^{(2)}}{\partial z^{(2)}} \quad (19)$$

$$\delta^{(3)} = \frac{\partial L}{\partial z^{(3)}} = \delta^{(3+1)} \cdot \frac{\partial z^{(3+1)}}{\partial a^{(3)}} \cdot \frac{\partial a^{(3)}}{\partial z^{(3)}} \quad (20)$$

$$\delta^{(n)} = \frac{\partial L}{\partial z^{(n)}} = \delta^{(n+1)} \cdot \frac{\partial z^{(n+1)}}{\partial a^{(n)}} \cdot \frac{\partial a^{(n)}}{\partial z^{(n)}} \quad (21)$$

Tables II and III show training and testing comparisons on various clock frequencies with state-of-the-art algorithms, where A-DF achieved more efficient results at a lower u_i. Figures 5 and 6 illustrate the various core configuration states of the proposed m-core distribution model using a full task migration policy for $t_i \in \tau$ on a MARVEL INTEL PXA-270 embedded MPSoC based on utilization factor (u_i).

TABLE II. TRAINING PERFORMANCE ANALYSIS

	SVM	CNN	NBB	RNN	ANN	RF	A-DF (Proposed)
Accuracy	90%	85.2%	86.3%	84%	81%	81%	92%
Precision	85.2%	86.3%	82%	90%	85.2%	85.2%	88.2%
Recall	86.2%	87.4%	83.3%	85.2%	86.3%	86.3%	89.3%
F-Measure	88.5%	89.4%	84.5%	88.5%	89.4%	89.4%	87.4%
Specificity	90.5%	90.3%	87.4%	90.5%	90.3%	90.3%	88.3%
Sensitivity	95.7%	91.5%	89.4%	81%	78%	78%	92%

TABLE III. PERFORMANCE ANALYSIS WITH DIFFERENT ALGORITHMS

Classifier	Training Accuracy %					Testing Accuracy %						
SVM	75.22	72.34	89.65	79.89	87.78	77.92	91.34	80.62	73.23	82.19	89.89	89.89
CNN	91.34	80.62	89.65	79.89	87.78	77.92	81.30	80.18	76.71	87.34	84.32	74.32
NBB	81.30	80.18	76.71	73.37	70.18	74.32	70.77	89.65	79.89	87.78	77.92	67.92
RF	70.77	89.65	79.89	65.79	61.87	73.37	72.49	72.08	84.32	81.28	70.21	60.21
RNN	72.49	70.77	89.65	79.89	87.78	75.79	72.41	72.01	87.92	86.84	83.68	73.68
DT	75.79	72.49	72.08	84.32	81.28	61.87	73.37	72.49	72.08	84.32	81.28	65.79
ANN	60.64	72.41	72.01	87.92	86.84	73.37	72.49	72.01	87.92	86.84	83.68	72.01
A-DF	92.16	74.32	85.79	84.32	81.07	84.72	82.19	71.98	80.21	88.92	97.97	87.97

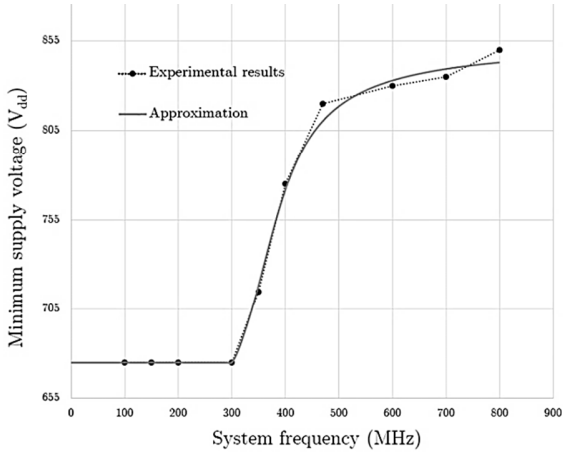

Fig. 4. V_{dd} for each system at 416 MHz.

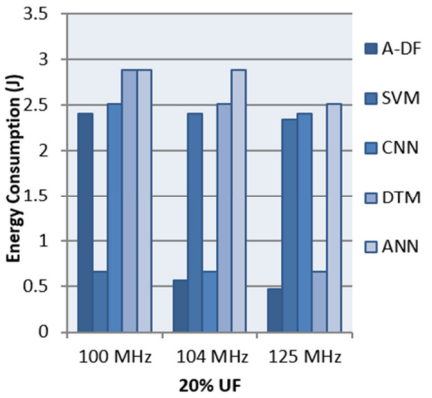

Fig. 5. Intra-task fixed mapping on 20% UF using A-DF.

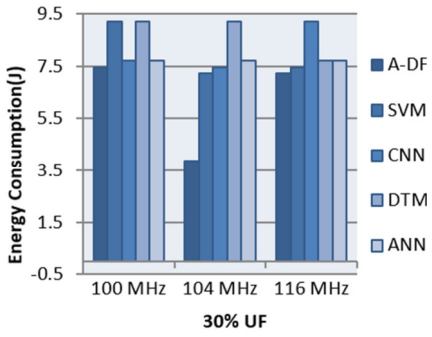

Fig. 6. Intra-task mapping on 30% UF using A-DF.

IV. CONCLUSION

A full task migration policy based on Dynamic Power Management (DPM) techniques for efficient task allocation and scheduling is the finest integration to decrease energy consumption. MARVEL INTEL PXA-270, and PXA-271, combined with PXA-250 use system-level DPM that allows CPU mode switches according to energy-power requirements. The proposed model achieved superior results compared to other models because efficient task scheduling and migration operate as key elements to control energy and power reduction, particularly in multiprocessor designs. There is an increasing market demand for applications that use multiple threads. On utilization factors (u_i) of 18% and 20%, the proposed approach consumed 6.3% less energy and provided 2.1% and 2.4% improvements in terms of accuracy and precision. The proposed design accumulatively provides significant improvements on three clock rates of 100, 104, and 116 MHz. The proposed system-level task migration policy offers solutions for improper core allocation and inefficient scheduling of real-time applications with deadline and priority constraints while minimizing energy dissipation to achieve better system performance.

REFERENCES

[1] M. J. Irwin, L. Benini, N. Vijaykrishnan, and M. Kandemir, "Chapter 2 - Techniques for Designing Energy-Aware MPSoCs," in *Multiprocessor Systems-on-Chips*, A. A. Jerraya and W. Wolf, Eds. Morgan Kaufmann, 2005, pp. 21–47.

[2] P. M. Dhulavvagol and S. G. Totad, "Performance Enhancement of Distributed Processing Systems Using Novel Hybrid Shard Selection Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13720–13725, Apr. 2024, https://doi.org/10.48084/etasr.7128.

[3] M. Alkhawatrah, "Energy-Harvesting Cooperative NOMA in IOT Networks," *Modelling and Simulation in Engineering*, vol. 2024, no. 1, Jan. 2024, Art. no. 1043973, https://doi.org/10.1155/2024/1043973.

[4] M. Z. Iskandarani, "Investigation of Energy Consumption in WSNs Within Enclosed Spaces Using Beamforming and LMS (BF-LMS)," *IEEE Access*, vol. 12, pp. 63932–63941, 2024, https://doi.org/10.1109/ACCESS.2024.3395932.

[5] A. Burns and A. Wellings, "Dispatching Domains for Multiprocessor Platforms and Their Representation in Ada," in *Reliable Software Technologiey – Ada-Europe 2010*, 2010, pp. 41–53, https://doi.org/10.1007/978-3-642-13550-7_3.

[6] G. Gonzalez-Martinez *et al.*, "A Survey of MPSoC Management toward Self-Awareness," *Micromachines*, vol. 15, no. 5, May 2024, Art. no. 577, https://doi.org/10.3390/mi15050577.

[7] J. Liu, M. Mao, J. Gao, J. Bai, and D. Sun, "Hardware-Accelerated YOLOv5 Based on MPSoC," *Journal of Physics: Conference Series*, vol. 2732, no. 1, Nov. 2024, Art. no. 012013, https://doi.org/10.1088/1742-6596/2732/1/012013.

[8] P. Verma, A. K. Maurya, and R. S. Yadav, "A survey on energy-efficient workflow scheduling algorithms in cloud computing," *Software: Practice and Experience*, vol. 54, no. 5, pp. 637–682, 2024, https://doi.org/10.1002/spe.3292.

[9] T. Yu *et al.*, "Collaborative Heterogeneity-Aware OS Scheduler for Asymmetric Multicore Processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1224–1237, Feb. 2021, https://doi.org/10.1109/TPDS.2020.3045279.

[10] W. Rao and H. Li, "Energy-aware Scheduling Algorithm for Microservices in Kubernetes Clouds," *Journal of Grid Computing*, vol. 23, no. 1, Dec. 2024, Art. no. 2, https://doi.org/10.1007/s10723-024-09788-w.

[11] K. Gaffour, M. K. Benhaoua, A. E. H. Benyamina, and A. K. Singh, "A new efficient multi-task applications mapping for three-dimensional Network-on-Chip based MPSoC," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 10, 2021, Art. no. e6194, https://doi.org/10.1002/cpe.6194.

[12] Y. Hu, Y. Liu, and Z. Liu, "A Survey on Convolutional Neural Network Accelerators: GPU, FPGA and ASIC," in *2022 14th International Conference on Computer Research and Development (ICCRD)*, Shenzhen, China, Jan. 2022, pp. 100–107, https://doi.org/10.1109/ICCRD54409.2022.9730377.

[13] R. Gonzalez and M. Horowitz, "Energy dissipation in general purpose microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1277–1284, Sep. 1996, https://doi.org/10.1109/4.535411.

[14] S. Choi, V. K. Prasanna, and J. Jang, "Minimizing energy dissipation of matrix multiplication kernel on Virtex-II," in *Reconfigurable Technology: FPGAs and Reconfigurable Processors for Computing and Communications IV*, Jul. 2002, vol. 4867, pp. 98–106, https://doi.org/10.1117/12.455487.

[15] H. Ali *et al.*, "A survey on system level energy optimisation for MPSoCs in IoT and consumer electronics," *Computer Science Review*, vol. 41, Aug. 2021, Art. no. 100416, https://doi.org/10.1016/j.cosrev.2021.100416.

[16] J. Feliu, J. Sahuquillo, S. Petit, and L. Eeckhout, "Thread Isolation to Improve Symbiotic Scheduling on SMT Multicore Processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 2, pp. 359–373, Oct. 2020, https://doi.org/10.1109/TPDS.2019.2934955.

[17] S. Dey, S. Isuwa, S. Saha, A. K. Singh, and K. McDonald-Maier, "CPU-GPU-Memory DVFS for Power-Efficient MPSoC in Mobile Cyber Physical Systems," *Future Internet*, vol. 14, no. 3, Mar. 2022, Art. no. 91, https://doi.org/10.3390/fi14030091.

[18] E. Jiang, L. Wang, and J. Wang, "Decomposition-based multi-objective optimization for energy-aware distributed hybrid flow shop scheduling with multiprocessor tasks," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 646–663, Jul. 2021, https://doi.org/10.26599/TST.2021.9010007.

[19] Y. L. Chou, S. Liu, E. Y. Chung, and J. L. Gaudiot, "An Energy and Performance Efficient DVFS Scheme for Irregular Parallel Divide-and-Conquer Algorithms on the Intel SCC," *IEEE Computer Architecture Letters*, vol. 13, no. 1, pp. 13–16, Jan. 2014, https://doi.org/10.1109/L-CA.2013.1.

[20] M. Alam, R. A. Haidri, and M. Shahid, "Resource-aware load balancing model for batch of tasks (BoT) with best fit migration policy on heterogeneous distributed computing systems," *International Journal of Pervasive Computing and Communications*, vol. 16, no. 2, pp. 113–141, Apr. 2020, https://doi.org/10.1108/IJPCC-10-2019-0081.

[21] K. Baital and A. Chakrabarti, "Dynamic Scheduling of Real-Time Tasks in Heterogeneous Multicore Systems," *IEEE Embedded Systems Letters*, vol. 11, no. 1, pp. 29–32, Mar. 2019, https://doi.org/10.1109/LES.2018.2846666.

[22] K. Huang *et al.*, "Expected Energy Optimization for Real-Time Multiprocessor SoCs Running Periodic Tasks with Uncertain Execution Time," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 3, pp. 398–411, Jul. 2021, https://doi.org/10.1109/TSUSC.2018.2853621.

[23] H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "System-level application-aware dynamic power management in adaptive pipelined MPSoCs for multimedia," in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2011, pp. 616–623, https://doi.org/10.1109/ICCAD.2011.6105394.

[24] J. R. B. Bantock, V. Tenentes, B. M. Al-Hashimi, and G. V. Merrett, "Online tuning of Dynamic Power Management for efficient execution of interactive workloads," in *2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Taipei, Taiwan, Jul. 2017, pp. 1–6, https://doi.org/10.1109/ISLPED.2017.8009195.

[25] P. Bogdan, R. Marculescu, and S. Jain, "Dynamic power management for multidomain system-on-chip platforms: An optimal control approach," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 18, no. 4, Jul. 2013, Art. no. 46, https://doi.org/10.1145/2504904.

[26] H. Khan, I. U. Din, A. Ali, and M. Husain, "An Optimal DPM Based Energy-Aware Task Scheduling for Performance Enhancement in Embedded MPSoC," *Computers, Materials and Continua*, vol. 74, no. 1, pp. 2097–2113, Aug. 2022, https://doi.org/10.32604/cmc.2023.032999.

[27] X. Zhang, W. Zhang, W. Sun, H. Wu, and A. Song, "A Real-time Cutting Model Based on Finite Element and Order Reduction.," *Computer Systems Science & Engineering*, vol. 43, no. 1, 2022.

[28] A. K. Coskun, T. S. Rosing, K. Mihic, G. De Micheli, and Y. Leblebici, "Analysis and Optimization of MPSoC Reliability," *Journal of Low Power Electronics*, vol. 2, no. 1, pp. 56–69, Apr. 2006, https://doi.org/10.1166/jolpe.2006.007.

[29] R. Urunuela, A. M. Deplanche, and Y. Trinquet, "STORM a simulation tool for real-time multiprocessor scheduling evaluation," in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, Bilbao, Sep. 2010, pp. 1–8, https://doi.org/10.1109/ETFA.2010.5641179.