Profil Energi Pada Processor Perangkat *Edge Computing* Berbasis DVFS dalam Menjalankan Program Paralel

Lompo Ramos Emakarim[1], Andani Achmad[2], Adnan[3],

[1] Program Studi Teknik Elektro, Universitas Cokroaminoto Makassar, [2,3]Universitas Hasanuddin

**Correspondent Author: lompo_ramos@ucm-si.ac.id*

***Abstract* — This study aims to estimate the processor energy profile of an edge computing device during parallel program execution across various operating frequencies, utilizing Dynamic Voltage and Frequency Scaling (DVFS).**

The SparseLU and Strassen algorithms from the Barcelona OpenMP Tasks Suite (BOTS) were executed on an Intel NUC DN2820FYK PC using the Intel P-State driver. Both programs were run at different Intel NUC operating frequencies determined by DVFS principles. Processor energy consumption was estimated to establish the relationship between energy usage and operating frequency.

This research successfully generated processor energy usage profiles for SparseLU and Strassen across the various operating frequencies defined by DVFS. Furthermore, energy consumption efficiency was achieved by increasing the CPU frequency—resulting in faster execution times—even though the power consumed was higher.

***Keyword* — *Edge Computing*, DVFS, Processor, Energy, BOTS**

Abstrak — Penelitian ini bertujuan untuk mengestimasi profil energi processor pada suatu perangkat edge computing yang dibutuhkan pada suatu pemrograman paralel pada beberapa frekuensi kerja processor berbasis Dynamic Voltage and Frequency Scaling.

SparseLU dan Strassen dari Barcelona OpenMP Tasks Suite (BOTS) dijalankan pada PC Intel NUC DN2820FYK dengan menggunakan driver Intel P-State. Kedua program tersebut dijalankan pada setiap frekuensi kerja processor Intel NUC yang berbeda-beda berdasarkan prinsip Dynamic Voltage and Frequency Scaling. Energi yang digunakan oleh processor disetimasikan untuk mendapatkan profil penggunaan energi processor terhadap frekuensi kerja processor.

Pada penelitian ini profil penggunaan energi processor pada setiap frekuensi kerja berdasarkan Dynamic Voltage and Frequency Scaling dalam menjalankan SparseLU dan Strassen telah dibuat. Selain itu efisiensi konsumsi energi dapat dicapai yaitu dengan meningkatkan frekuensi CPU dan waktu eksekusi yang lebih cepat pada prosesor meskipun daya yang digunakan semakin besar..

Kata kunci — Edge Computing , DVFS, Processor, Energi, BOTS.

I. PENDAHULUAN

Edge computing dapat digunakan di jaringan telekomunikasi 5G [1], perumahan [2] bahkan di jaringan listrik [3]. Penggunaan sumber daya yang efisien dapat mengurangi jumlah energi atau daya listrik yang dibutuhkan untuk melakukan proses komputasi pada *edge computing*. Efisiensi komputasi dapat dilakukan dengan mengurangi beban kerja processor. Salah satu teknik yang dapat digunakan untuk mengurangi beban kerja processor adalah DVFS (*Dynamic Voltage Frequency Scaling*) [7]. Pengaturan kerja processor tersebut berdasarkan pengaturan frekuensi kerja processor. Semakin tinggi frekuensi kerja processor maka semakin tinggi pula tegangan listrik, daya, dan energi yang digunakan oleh processor. Untuk mengontrol kerja inti prosesor, driver Intel P-State dapat digunakan Selain itu, driver ini dapat dimanfaatkan untuk mengestimasi konsumsi energi inti prosesor oleh suatu program, khususnya pada prosesor kelas bawah yang mendukung P-State ACPI namun tidak memiliki fitur RPAL untuk pengukuran energi.[25]

Penerapan pemrograman paralel pada edge computing juga semakin berkembang, salah satunya adalah model DroMPI yang dikembangkan untuk mengatur komunikasi antara node serta menyediakan interfae API.[26]

Penelitian ini bertujuan untuk mengestimasi profil energi processor pada suatu perangkat edge computing yang dibutuhkan pada suatu pemrograman paralel pada beberapa frekuensi kerja processor yang dikendalikan oleh driver Intel P-State.

II. TINJAUAN PUSTAKA

A. Benchmarking Pemrograman Paralel

Barcelona OpenMP Tasks Suite (BOTS) merupakan salah satu benchmark yang digunakan dalam pemrograman paralel[27]. Barcelona Open MP memiliki 8 benchmark yang dapat dijalankan pada sistem untuk menganalisis karakteristiknya yang terkait dengan pemrograman paralel. Dua benchmark tersebut berkaitan dengan tugas matriks, yaitu SparseLU dan Strassen. Strassen adalah perkalian matriks padat menggunakan dekomposisi matriks hierarkis, sedangkan SparseLU adalah pemanfaatan matriks jarang untuk memfaktorkan matriks LU. SparseLU sendiri memiliki 5 versi, yaitu 1 versi serial dan 4 versi paralel, sedangkan Strassen memiliki 6 versi paralel.[29]

B. DVFS dan Pengaturan Daya

Efisiensi komputasi dapat dilakukan dengan mengurangi beban kerja processor. Salah satu teknik yang dapat digunakan untuk mengurangi beban kerja processor adalah DVFS (Dynamic Voltage Frequency Scaling) [7]. DVFS diwujudkan salah satunya dalam bentuk ACPI _PSS (Advanced Configuration and Power Interface Processor

Performance States). Dengan DVFS, berbagai algoritma dapat dikembangkan untuk mengatur kerja processor berdasarkan tugas tertentu [8], atau mencari kombinasi frekuensi-tegangan yang sesuai [9]. DVFS juga dapat diterapkan di perangkat mobile [10], SoC (Sistem on Chip) [11], sistem komputasi awan [12] bahkan untuk skala waktu yang sangat kecil [13]. Pengaturan kerja processor tersebut berdasarkan pengaturan frekuensi kerja processor. Semakin tinggi frekuensi kerja processor maka semakin tinggi pula tegangan listrik, daya, dan energi yang digunakan oleh processor.

C. NUC dan Edge Computing

Evaluasi Komputer mini seperti Next Unit of Computing (NUC) menggunakan energi listrik yang lebih kecil dibandingkan server dengan mendapatkan penghematan energi sebanyak 7,7 kali [29]. Hal ini menjadikannya sebagai salah satu pilihan pada edge computing. Seperti pada penggunaan Edge Keeper, sebuah platform yang dikembangkan sebagai mobile edge computing yang menjalankan jaringan LTE [28]

Studi tentang *edge computing* telah banyak dilakukan [1], [2], [3], [4], [14], [15], [16], [17],[18], [19]. Studi tentang Dynamic Frequency and Voltage Scaling (DFVS) yang mengatur tegangan dan frekuensi processor juga telah dilakukan [5], [6], [7], [8], [9], [10], [11],[12], [13],[20], [21], [22],[23],[24]. Beberapa menerapkan DVFS pada cloud data center [6],[7], [12][23]. Namun, dari semua penelitian tersebut, DFVS tidak diterapkan di edge computing. Padahal dengan menerapkan DVFS maka pengontrolan terhadap frekuensi dan tegangan prosessor dimungkinkan untuk dilakukan pada edge computing. Namun pengontrolan tersebut harus dimulai dari bagaimana mengetahui profil energi pada processor tersebut.

III. METODE PENELITIAN

A. Sistem Hardware

Sebuah sistem dibangun seperti yang dijelaskan pada [25] dengan menggunakan mini PC Intel NUC DN2820FYK yang dilengkapi prosesor *dual-core*. Prosesor tersebut tidak memiliki fitur *hyper-threading*, sehingga jumlah *thread* terbatas pada dua. frekuensi maksimum ditetapkan sebesar 2159 MHz dan minimum sebesar 498 MHz, serta fitur *boost* dinonaktifkan. Sistem operasi yang digunakan adalah Linux Ubuntu 15.04 *generic* x86_64 dengan *kernel* 3.19.0-42. Driver Intel P-State diatur sebagai *default* untuk mengontrol P-State inti prosesor, dan *power governor* diatur ke mode *power save*.

B. Pengumpulan Data Daya dan Estimasi Energi

Program PERF dijalankan untuk mengumpulkan peristiwa *power:p_state_sample* dari program BOTS. Dengan mengumpulkan peristiwa tersebut, nilai dinamis frekuensi *core* dapat dikumpulnkan selama waktu eksekusi program. Frekuensi tersebut kemudian dikonversi menjadi nilai daya berdasarkan data ACPI _PSS [30] menggunakan persamaan perbandingan [28]. Untuk mengestimasi energi program itu sendiri, kami menggunakan metode yang dijelaskan dalam [28], di mana grafik daya terhadap waktu eksekusi diintegralkan. RAPL, sebuah fitur untuk mengukur konsumsi energi prosesor, tidak tersedia pada sistem ini. Data untuk setiap program diambil sebanyak lima kali untuk mendapatkan nilai rata-rata.

C. Versi BOTS

Semua versi SparseLU dan Strassen dari BOTS dievaluas dan dikompilasi menggunakan GCC. SparseLU mencakup 4 versi paralel yaitu SparseLU.gcc.for-omp-task, SparseLU.gcc.for-omp-task-tied, SparseLU.gcc.single-omp-task, dan SparseLU.gcc.single-omp-task-tied—serta 1 versi serial, yaitu SparseLU.gcc.serial. Sementara itu Strassen mencakup 6 versi paralel—yaitu Strassen.gcc.omp-task, Strassen.gcc.omp-task-if_clause, Strassen.gcc.omp-task-if_clause-tied, Strassen.gcc.omp-task-manual, Strassen.gcc.omp-task-manual-tied, dan Strassen.gcc.omp-task-tied—serta 1 versi serial, yaitu Strassen.gcc.serial.

IV. HASIL DAN PEMBAHASAN

A. Daya, Waktu Eksekusi, dan Energi

Melalui estimasi energi pada semua varian SparseLU dan Strassen, diperoleh Tabel 1 dan Tabel 2 yang memuat rata-rata waktu eksekusi, rata-rata energi, dan rata-rata daya pada kedua *core* setelah lima kali percobaan. Nilai rata-rata daya dan energi pada Tabel 1 dan Tabel 2 merupakan hasil penjumlahan nilai rata-rata daya dan energi dari *core* 1 dan *core* 2. Berdasarkan Tabel 1 dan Tabel 2, terlihat bahwa versi paralel SparseLU dan Strassen memiliki nilai yang saling berdekatan; waktu eksekusi berkisar antara 24,12 detik hingga 24,16 detik untuk SparseLU dan 34,76 detik hingga 35,20 detik untuk Strassen, sedangkan energi berkisar antara 86,65 J hingga 86,79 J untuk SparseLU dan 124,80 J hingga 126,37 J untuk Strassen. Perbedaan hanya terlihat pada versi serial. Versi serial SparseLU memiliki waktu eksekusi sekitar 47,975806 detik dan estimasi energi sekitar 172,3329 J, sementara versi serial Strassen memiliki waktu eksekusi sekitar 55,80951 detik dan estimasi energi sekitar 200,40021 J. Hasil penelitian menunjukkan bahwa pada semua versi, rata-rata daya untuk seluruh *core* berada di angka sekitar 3,59 J/detik. Hal ini berarti bahwa dalam kondisi tersebut, *P-state*, frekuensi, dan daya *core* cenderung tetap pada nilai tertentu, sehingga energi lebih banyak dipengaruhi oleh durasi waktu eksekusi.

B. Variasi P-State

Dengan menetapkan nilai minimum dan maksimum frekuensi ke angka yang sama, frekuensi *core* akan cenderung beroperasi pada nilai tertentu, yaitu frekuensi dari

suatu *state* dalam P-States. Kami menetapkan nilai satu per satu sebagaimana tercantum dalam informasi ACPI _PSS. Kami kemudian memperoleh informasi frekuensi yang berbeda dari yang disebutkan pada [28], seperti yang tercantum dalam Tabel 3.

Tabel 1. Waktu Eksekusi, Energi Core, dan Daya pada semua versi SparseLU

SparseLU	Time (S)	All Cores Energy(J)	Average Power of All Cores (J/S)
Serial	47.9758	172.3329	3.5920797
omp single task	24.1576	86.764188	3.5915850
omp single task tied	24.1656	86.792995	3.5915848
for omp-task	24.1264	86.651195	3.5915483
for omp task tied	24.1400	86.698005	3.5914631

Tabel 2. Waktu Eksekusi, Energi Core, dan Daya pada semua versi Strassen

Strassen	Time (S)	All Cores Energy(J)	Average Power of All Cores (J/S)
Serial	55.80951	200.40021	3.5907896
Strassen.omp-tasks	35.2084	126.37475	3.5893329
Strassen.omp-tasks-tied	35.1419	126.20441	3.5912738
Strassen.omp-tasks-if_clause	34.7696	124.80998	3.5896239
Strassen.omp-tasks-if_clause-tied	34.8937	125.31305	3.5912738
Strassen.omp-tasks-manual	34.8401	125.05987	3.5895366
Strassen.omp-tasks-manual-tied	34.8537	125.14709	3.5906382

Selisih antara nilai input adalah 166 MHz, sedangkan selisih antara frekuensi output pada Tabel 3 memiliki dua nilai, yaitu 83,3 MHz dan 166,6 MHz. Dengan membaca MSR_FSB_FREQ berdasarkan arsitektur Silvermont [31], kami memperoleh kecepatan *bus* yang dapat disesuaikan (*scalable bus speed*), yaitu 83,3 MHz. Kecepatan *bus* ini bersesuaian dengan selisih frekuensi output sebesar 83,3 MHz. Nilai ini juga merupakan setengah dari 166,6 MHz, yang berarti mungkin terdapat satu *state* tambahan di antara selisih 166,6 MHz tersebut sehingga dapat diperoleh selisih yang konsisten sebesar 83,3 MHz di antara semua *state*. Oleh karena itu, dengan menetapkan 2415,7 MHz sebagai P0, dapat dieroleh 23 P-States sebagaimana tercantum dalam Tabel 4, yang mensimulasikan frekuensi P-state pada *driver* Intel P-State. Jumlah P-States yang benar-benar tersedia dapat bergantung pada konfigurasi *driver* P-State itu sendiri.

Tabel 3. Penyimpangan antara *State* Input dan Output

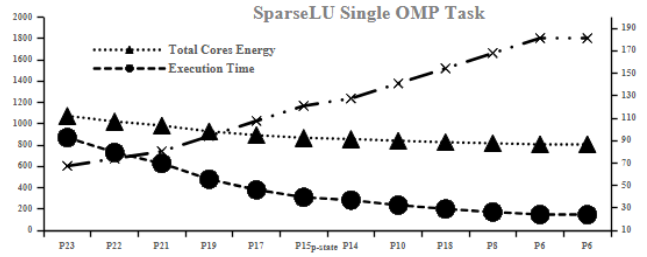
Input (MHz)	Deviasi (MHz)	Frekunsi Output (MHz)	Deviasi (MHz)
498		499.8	
664	166	583.1	83.300
830	166	666.4	83.300
996	166	833	166.600
1162	166	999.6	166.600
1328	166	1166.2	166.600
1494	166	1249.5	83.300
1660	166	1416.1	166.600
1826	166	1582.7	166.600
1992	166	1749.3	166.600
2158	166	1915.9	166.600
2159	1	1915.9	0

Tabel 4. Simulasi Frekuensi P-State

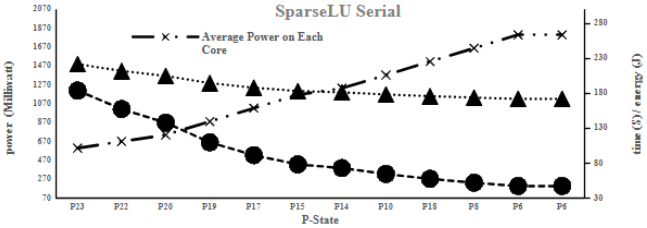
P-*state* #	Freq (MHz)	P-*state* #	Freq (MHz)
P11	1499.4	P23	499.8
P10	1582.7	P22	583.1
P9	1666	P21	666.4
P8	1749.3	P20	749.7
P7	1832.6	P19	833
P6	1915.9	P18	916.3
P5	1999.2	P17	999.6
P4	2082.5	P16	1082.9
P3	2165.8	P15	1166.2
P2	2249.1	P14	1249.5
P1	2332.4	P13	1332.8
P0	2415.7	P12	1416.1

Frekuensi spesifik ditetaip sebagaimana yang tercantum pada Tabel 3—yang sesuai dengan nomor P-State pada Tabel 4—dan semua versi SparseLU serta Strassen dijalankan pada setiap frekuensi tersebut. Data mengenai waktu eksekusi, daya rata-rata, dan estimasi energi diperoleh untuk setiap frekuensi. Dihasilkan grafik yang mencakup seluruh versi SparseLU dan Strassen. Karena semua versi paralel memiliki

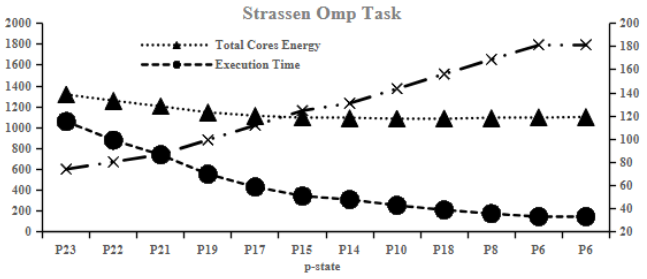
waktu eksekusi dan konsumsi energi *core* yang serupa, hanya ditampilkan grafik untuk Strassen.omp.task (Gambar. 1) dan SparseLU single-omp-task (Gambar. 3) sebagai representasi versi paralel, serta grafik untuk versi sekuensial (Gambar. 2 dan Gambar. 4). Keempat grafik tersebut menunjukkan bahwa seiring dengan peningkatan frekuensi, waktu eksekusi dan energi mengalami penurunan, sedangkan daya justru meningkat. Hal ini berarti bahwa dalam eksperimen kami yang melibatkan eksekusi program SparseLU dan Strassen, waktu eksekusi memberikan pengaruh yang lebih besar terhadap estimasi energi dibandingkan daya. Meskipun daya menunjukkan kecenderungan meningkat seiring dengan naiknya frekuensi *core*, estimasi energi justru menunjukkan penurunan yang selaras dengan berkurangnya waktu eksekusi.



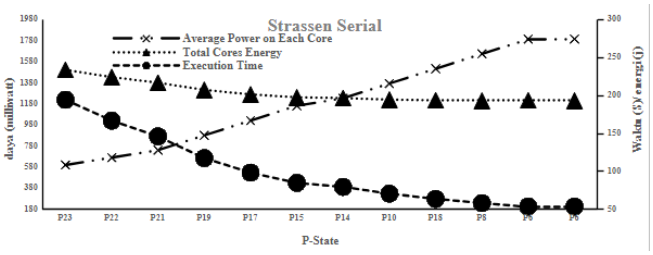
Gambar 1 Waktu Eksekusi , Power, Energy VS P-State dari SparseLU single omp pada Intel P-State



Gambar 2 Waktu Eksekusi , Power, and Energy VS P-State dari SparseLU Serial pada Intel P-State



Gambar 3 Waktu Eksekusi , Power, Energy VS P-State dari Strassen Omp pada Intel P-State



Gambar 4 Waktu Eksekusi , Power, and Energy VS P-State dari Strassen Serial pada Intel P-State

Dalam kondisi di mana *core* cenderung bertahan pada satu status P-State tertentu tanpa beralih ke P-State lain yang lebih tinggi atau lebih rendah, konsumsi energi hanya bergantung pada waktu eksekusi. Dalam eksperimen ini dimana digunakan dua *thread,* didapatkan bahwa energi dan waktu eksekusi untuk semua versi hampir sama. Hasil serupa juga terlihat dalam [29], di mana peningkatan kecepatan (*speedup*) waktu eksekusi hampir identik saat menggunakan dua *thread*—baik pada versi *if-clause-cut-off*, *manual cut-off*, maupun tanpa *cut-off*, serta antara versi *tied* dan *untied*. Studi [29] tersebut juga menyebutkan bahwa peningkatan jumlah *thread* menghasilkan tren peningkatan kecepatan yang berbeda, di mana versi *manual cut-off* dan *untied* menunjukkan laju peningkatan kecepatan yang lebih tinggi.

Hasil serupa juga ditemukan pada grafik energi versus frekuensi dalam [32]. Grafik tersebut membedakan antara beban kerja yang intensif terhadap memori—yaitu ketika lalu lintas memori melebihi 90% kapasitas bus memori akibat *Last Level Cache Misses*—dan beban kerja yang intensif terhadap CPU—yaitu ketika lalu lintas memori berada di bawah 10% kapasitas bus memori. Studi tersebut juga menunjukkan bahwa peningkatan frekuensi CPU pada kedua jenis beban kerja menyebabkan kenaikan konsumsi daya dan penurunan waktu eksekusi. Namun, pada beban kerja yang intensif terhadap memori, rasio penurunan waktu eksekusi lebih besar daripada rasio kenaikan daya; sebaliknya, pada beban kerja yang intensif terhadap CPU, rasio penurunan waktu eksekusi lebih kecil daripada rasio kenaikan daya. Oleh karena itu, peningkatan frekuensi cenderung meningkatkan konsumsi energi pada beban kerja yang intensif terhadap memori, sedangkan pada beban kerja yang intensif terhadap CPU, konsumsi energi cenderung menurun. Dalam eksperimen kami, seluruh program BOTS terkait matriks yang dijalankan pada platform tersebut menunjukkan hasil yang serupa dengan beban kerja intensif CPU dalam [32], yang mendukung pernyataan umum dalam [33] bahwa semakin cepat program berjalan, semakin sedikit energi yang dikonsumsi, kecuali jika terjadi hambatan *bottleneck* pada lalu lintas memori [32] atau masalah terkait peristiwa OpenMP [34].

V. KESIMPULAN

Profil energi processor pada perangkat *edge computing* berbasis Intel NUC dan *driver Intel P-State* dengan menjalankan program paralel *Sparse-LU* dan *Strassen* dari Barcelona OpenMP Task Suite telah dihasilkan. Profil energi menunjukkan bahwa waktu eksekusi dan energi yang dibutuhkan memiliki kesamaan untuk versi program paralel.

Dari profil energi yang dihasilkan, juga dapat ditunjukkan bahwa efisiensi energi dapat dicapai dalam menjalankan suatu program paralel dengan meningkatkan frekuensi CPU dimana hal tersebut ditentukan oleh waktu eksekusi yang lebih cepat pada prosesor *dual-core* meskipun nilai daya processor meningkat.

DAFTAR ACUAN

[1] A. Flizikowski, M. A. Rahman, M. M. Mowla and F. Kooshki, "Workload Prediction of Virtualized RAN in the Edge Micro Data Center: An Experimental Progress," 2023 IEEE Conference on Standards for Communications and Networking (CSCN), Munich, Germany, 2023, pp. 334-338, doi: 10.1109/CSCN60443.2023.10453198.

[2] V. Gudepu et al., "Edge Computing in Micro Data Centers for Firefighting in Residential Areas of Future Smart Cities," 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives, 2022, pp. 1-6, doi: 10.1109/ICECCME55909.2022.9988101.

[3] D. Jiang, R. Xu, J. Guo, J. Li, X. Li and L. Peng, "Research on Integrated Value-Added Operation Model of Grid Assets Combined with Micro Data Center," 2024 Boao New Power System International Forum - Power System and New Energy Technology Innovation Forum (NPSIF), Qionghai, China, 2024, pp. 577-584, doi: 10.1109/NPSIF64134.2024.10883244.

[4] D. Scano et al., "Enabling P4 Network Telemetry in Edge Micro Data Centers With Kubernetes Orchestration," in IEEE Access, vol. 11, pp. 22637-22653, 2023, doi: 10.1109/ACCESS.2023.3249105.

[5] S. Liu, "A High-Performance Self-Adjustable Power Management System with Dynamic Voltage Frequency Scaling for Low-Power Computing," 2025 Asia-Europe Conference on Cybersecurity, Internet of Things and Soft Computing (CITSC), Rimini, Italy, 2025, pp. 133-137, doi: 10.1109/CITSC64390.2025.00032.

[6] J. Masoudi, B. Barzegar and H. Motameni, "Energy-Aware Virtual Machine Allocation in DVFS-Enabled Cloud Data Centers," in IEEE Access, vol. 10, pp. 3617-3630, 2022, doi: 10.1109/ACCESS.2021.3136827

[7] J. Mao, T. Bhattacharya, X. Peng, T. Cao and X. Qin, "Modeling Energy Consumption of Virtual Machines in DVFS-Enabled Cloud Data Centers," 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 2020, pp. 1-6, doi: 10.1109/IPCCC50635.2020.9391552.

[8] X. Liu, "Low Energy Consumption and Time Deterministic Energy-saving Workflow Task Scheduling Algorithm based on DVFS," 2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom), Xi'an, China, 2022, pp. 56-61, doi: 10.1109/CSCloud-EdgeCom54986.2022.00019.

[9] A. Yeganeh-Khaksar, M. Ansari, S. Safari, S. Yari-Karin and A. Ejlali, "Ring-DVFS: Reliability-Aware Reinforcement Learning-Based DVFS for Real-Time Embedded Systems," in IEEE Embedded Systems Letters, vol. 13, no. 3, pp. 146-149, Sept. 2021, doi: 10.1109/LES.2020.3033187.

[10] S. Hu, X. Xu, Y. Chen, M. Qin, Y. Lian and Y. Zhao, "A ResNet-Based DVFS Regulator for Heterogeneous Multi-Core Mobile Processors," TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON), Singapore, Singapore, 2024, pp. 1274-1277, doi: 10.1109/TENCON61640.2024.10902794.

[11] Q. Zhou et al., "A Lightweight DRDPG-Based RL DVFS for Video Rendering on CPU-GPU Integrated SoC," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 71, no. 5, pp. 2119-2131, May 2024, doi: 10.1109/TCSI.2024.3351791.

[12] H. Noguchi and M. Kaneko, "Energy-Efficient CPU Core Allocation in DVFS and CPU Pinning Enabled Cloud," 2024 IEEE Cloud Summit, Washington, DC, USA, 2024, pp. 67-72, doi: 10.1109/Cloud-Summit61220.2024.00018.

[13] Y. Shen and A. Zou, "Learning-Based Performance and Power Model for Processor Microsecond DVFS," 2023 China Semiconductor Technology International Conference (CSTIC), Shanghai, China, 2023, pp. 1-3, doi: 10.1109/CSTIC58779.2023.10219252

[14] M. D. M. da Silva, A. Gamatié, G. Sassatelli, M. Poss and M. Robert, "Optimization of Data and Energy Migrations in Mini Data Centers for Carbon-Neutral Computing," in IEEE Transactions on Sustainable Computing, vol. 8, no. 1, pp. 68-81, 1 Jan.-March 2023, doi: 10.1109/TSUSC.2022.3197090

[15] M. H. Samsuri et al., "Comparative Performance Analysis of Edge-AI Devices in Deep Learning Applications," 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 2024, pp. 1-6, doi: 10.1109/ICIEA61579.2024.10665079

[16] G. L. Santos et al., "Analyzing the Impact of Micro Data Centers Failures in Cellular Networks: a Road Race Study," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 2020, pp. 3096-3102, doi: 10.1109/SMC42975.2020.9283450

[17] E. Minazzo et al., "A New Edge Micro Data Center and its Passive Thermosyphon Cooling System at TUe: Cooling System Thermal Performance Tests," 2024 40th Semiconductor Thermal Measurement, Modeling & Management Symposium (SEMI-THERM), San Jose, CA, USA, 2024, pp. 1-5

[18] P. J. Kuehn and M. Mashaly, "DVFS-Power Management and Performance Engineering of Data Center Server Clusters," 2019 15th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Wengen, Switzerland, 2019, pp. 91-98, doi: 10.23919/WONS.2019.8795470

[19] H. Noguchi and M. Kaneko, "Application-Independent DVFS Method Based on OS Metrics Monitoring for Cloud Servers," 2024 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Beijing, China, 2024, pp. 1-6, doi: 10.1109/CCCI61916.2024.10736461.

[20] C. Park et al., "A new DVFS algorithm to minimize energy consumption on system-on-chip architecture and electrical characteristics," 2022 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2022, pp. 1-4, doi: 10.1109/ICCE53296.2022.9730423.

[21] J. Guerreiro, A. Ilic, N. Roma and P. Tomás, "Modeling and Decoupling the GPU Power Consumption for Cross-Domain DVFS," in IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 11, pp. 2494-2506, 1 Nov. 2019, doi: 10.1109/TPDS.2019.2917181

[22] M. A. Uddin, M. Lin and L. T. Yang, "Dynamic Slack-Sharing Learning Technique With DVFS for Real-Time Systems," in

IEEE Transactions on Sustainable Computing, vol. 9, no. 3, pp. 261-270, May-June 2024, doi: 10.1109/TSUSC.2023.3283518

[23] W. Lin, X. Luo, C. Li, J. Liang, G. Wu and K. Li, "An Energy-Efficient Tuning Method for Cloud Servers Combining DVFS and Parameter Optimization," in IEEE Transactions on Cloud Computing, vol. 11, no. 4, pp. 3643-3655, Oct.-Dec. 2023, doi: 10.1109/TCC.2023.3308927

[24] F. Ziaee, M. Ghazvini and M. Kaedi, "A Learning-Based DVFS Power Management Scheme for Multicore Processors," 2024 19th Iranian Conference on Intelligent Systems (ICIS), Sirjan, Iran, Islamic Republic of, 2024, pp. 18-23, doi: 10.1109/ICIS64839.2024.10887414

[25] Adnan, Andani Achmad and Lompo Ramos Emakarim, "Leveraging ACPI _PSS Data to Estimate Energy of Processor Core between ACPI-Cpufreq and Intel P-State Driver on Low-End Intel Pentium Celeron N2830," in International Journal of Innovative Computing, Information and Control (IJICIC), Vol. 13, Number 6, pp. 1993-2008, 2017, DOI: 10.24507/ijicic.13.06.1993

[26] G. -M. Grosu, S. -E. Nistor, R. -I. Ciobanu, C. Dobre and F. Pop, "DroMPI: Parallel Computation Over Drop Computing," 2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW), Philadelphia, PA, USA, 2024, pp. 120-127, doi: 10.1109/CCGridW63211.2024.00021

[27] Jinghao Sun, Tao Jin, Yekai Xue, Liwei Zhang, Jinrong Liu, Nan Guan, Quan Zhou, ompTG: From OpenMP Programs to Task Graphs,Journal of Systems Architecture,Volume 126,2022,102470,ISSN 1383-7621,https://doi.org/10.1016/j.sysarc.2022.102470.

[28] N. Rahman, T. Atahary, C. Yakopcic, T. M. Taha and S. Douglass, "Q-Learning Based Cognitive Domain Ontology Representation and Solving on Low Power Computing Platforms," in IEEE Access, vol. 12, pp. 131-147, 2024, doi: 10.1109/ACCESS.2023.3346908

[29] Duran, A., Teruel, X., Ferrer, R., Martorell, X. and Ayguad´e, E. Barcelona OpenMP Task Suite: A Set of Benchmarks Targeting the Exploitation of Tied Parallelism in OpenMP. Proceedings of International Conference on Parallel Processing (ICPP). 2009. Page(s):124 – 131. DOI: 10.1109/ICPP.2009.64

[30] Hewlett-Packard Corporation, Intel Corporation Microsoft Corporation, Phoenix Technologies Ltd., and Toshiba Corporation. Advanced Configuration and Power Interface Specification Revision 5.0 Errata A. 2013

[31] Intel Corporation. 2015. Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 (3A, 3B & 3C): System Programming Guide.

[32] Kim, Shin-gyu, Choi, Chanho, Eom, Hyeonsang, Yeom, Heon Y., Byun, Huichung. Energy-centric DVFS Controling Method for Multi-core Platforms. Proceedings of SC Companion: High Performance Computing, Networking Storage and Analysis. 2012.Pages: 685 - 690, DOI: 10.1109/SC.Companion.2012.94

[33] Porterfield, A.K. Olivier, S.L., Bhalachandra S., and Prins, J.F. Power Measurement and Concurrency Throttling for Energy Reduction in OpenMP Programs. Proceedings of IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum. 2013. Pages: 884 - 891, DOI: 10.1109/IPDPSW.2013.15

[34] Leidel, J and Yong Chen. Communication Avoiding Power Scaling. Proceedings of 44th International Conference on Parallel Processing Workshops.2015. Pages: 166 - 174, DOI: 10.1109/ICPPW.2015.26.