Received: 18 November 2023; Revised: 3 January 2024; Accepted: 1 April 2024; Published: 15 April 2024

Layered Approach in Energy-Efficient Information Technology (IT) Classroom Design

Okta Nurika[1*], Hafizul Fahri Hanafi[2], Low Tang Jung[3], Ahmed Abba Haruna[4]

[1]*Centre of Embedded Education Green Technology, Faculty of Computing and Meta-Technology, Sultan Idris Education University, Tanjung Malim, 35900 Perak, Malaysia; oktanurika@meta.upsi.edu.my*
[2]*Faculty of Computing and Meta-Technology, Sultan Idris Education University, Tanjung Malim, 35900 Perak, Malaysia; hafizul@meta.upsi.edu.my*
[3]*Computer and Information Sciences Department, Universiti Teknologi PETRONAS, Tronoh, Malaysia; lowtanjung@utp.edu.my*
[4]*College of Computer Science and Engineering, University of Hafr Al Batin, Al Jamiah, Saudi Arabia; aaharuna@uhb.edu.sa*

**corresponding author*

Abstract

Designing an energy-efficient IT classroom has not been straightforward due to the scattered recommended practices that involve different aspects. Some related aspects have not even been explored and need to be included - to increase the level of power saving and flexibility in implementation. Thus, a comprehensive layered approach or method is designed by this paper. It involves four (4) layers that represent different aspects of an IT classroom that can be made efficient in terms of energy consumption, namely hardware, platform, middleware, and application layers. Each layer is autonomous and hence can be implemented flexibly, subject to the bespoke feasibility of the use case. The proposed method is also expected to be future-proof and expandable due to its catering to both hardware and software-based optimizations. Such guidance is also crucial for the ongoing Industrial Revolution (IR) 4.0 and soon-to-be-established IR 5.0, where smart cities worldwide would be required to be energy-efficient.

Keywords: Energy-efficient classroom, layered method, IT classroom.

INTRODUCTION

Education institutions, especially universities are mostly comprised of numerous classrooms equipped with electronic devices, which consume power in huge amounts as a consequence of teaching and learning activities. Common power-saving practices, such as air conditioning and lighting scheduling are currently in practice, however, many other dimensions of power-saving have not been explored, specifically involving Information Technology (IT) classrooms. Research in the area of energy-efficient classrooms has been scarce and the term “green classroom” is often tangled with another meaning, which is a class delivered out in nature (Mason, Manzione, et al., 2022; Mason, Ronconi, et al., 2022; Mason, Zagni, et al., 2022; Vella-Brodrick & Gilowska, 2022, Zangori & Cole, 2019). This is even though the importance of green campus has been justified for over two (2) decades (Simpson, 2002), where it is claimed to be able to decrease power usage by about 30% and beyond. Lessening power utilization would also sequentially minimize pollution levels (Tiyarattanachai & Hollmann, 2016), for example, by lowering heat generation by electronic components.

The need for energy-efficient classrooms is an urgent matter in the current world of Internet of Things (IoT) driven facilities as proven by Patil et al. (2018), who proposed an IoT sensor-based solution to save power usage. Their method was human-oriented as one of the sensed triggers for the actuators (lights and fans) would be the presence of humans in the classroom – besides the levels of light intensity, temperature, humidity, and noise. The practicability of this solution is plausible since it utilizes affordable off-the-shelf Arduino devices as sensors. However, this solution is quite limited in the sense that it only involves lights and fans as actuators – along with SMS alerts in response to above-limit noise levels. In other words, it is a one-dimensional solution (hardware-only approach). In addition, it also lacks monitoring applications – either web-based or mobile-based.

An improved solution to Patil et al. (2018) was designed by researchers Diddeniya et al. (2020), who complemented the Arduino-based sensors with a sophisticated Microsoft Kinect sensor that would track human presence and movement in the classroom. Similar to the previous solution by Patil et al. (2018), this solution would also sense light intensity, temperature, and humidity. On the upside, it is also more manageable due to the integration of a web-based monitoring system. Likewise, the former is also a one-dimensional solution.

Another hardware approach to reduce power consumption in the classroom was developed using an active green wall (Taemthong & Plitsiri, 2023) that consisted of live plants. Compared to electronic air conditioners, it managed to reduce power usage by up to over 50%. This came with an additional benefit in that it also decreased the amount of CO2 in the classroom. The proposed solution replaced the role of air conditioners and fans, therefore it was limited to only the context of air conditioning. Another drawback of it was that the embedded plants required maintenance (watering and nutrition) and the idea of having live plants in a classroom may not be aesthetically attractive to all people. In terms of suitability, this solution may not be appropriate for a classroom with a considerable number of computers that demand a high cooling system, which is usually provided by electronic air conditioners.

IoT-driven passive methods to control power usage have also been proposed (Santos & Ferreira, 2019; Sasane et al., 2017). They are mostly based on monitoring and therefore are incomprehensive and lack proactive methods, which will be proposed in this paper.

As discussed above, most of the existing solutions or methods to save power utilization in a classroom are limited to only one dimension of many possible aspects of a classroom. Thus, this paper will present a layered approach that covers multiple dimensions of power saving in a classroom. Section 2 would describe the proposed methodology, Section 3 would conclude the paper, and at last, Section 4 would conjecture about potential future improvement.

Received: 18 November 2023; Revised: 3 January 2024; Accepted: 1 April 2024; Published: 15 April 2024

METHODOLOGY

Our proposed methodology would cover multiple aspects of energy saving in a classroom. It is a layered bottom-up approach, where energy-saving proposed practices would start from where the data is in the form of an electronic signal (hardware layer). This signal will sequentially be converted to digital data format and be forwarded to the hardware controller (platform layer), which is the computer platform a.k.a. Operating System (OS). After the data is inside the OS, it is now ready to be used by a middleware (middleware layer) or directly used by an application (application layer). Each one of these layers can be optimized by applying certain configurations that would reduce power consumption. This layered approach is taking inspiration from how data travels through the computer network as empirically explained in the Internet Protocol (IP) stack (Nath et al., 2015). Below Figure 1 visualizes the bottom-up approach of how sequentially energy-saving practices can be implemented, although conceptually it is modular, where users may optimize any number of layers in any order, considering that power consumption does not regard the ownership of data subject to which application owns it. In other words, the proposed power-saving practices would work obliviously.

Application
Middleware
Platform
Hardware

Figure 1: Layered approach in energy-efficient classroom

Hardware Layer

The most fundamental layer is the hardware layer, where hardware components with lower wattage would be preferred. With regards to CPU clock on Linux, it was discovered that the power consumption is not linear to the increase in CPU clock in the case of a CPU with 4GHz of maximum clock speed (Herzog et al., 2021). The optimal clock setting that generated the least energy usage was at 3.5 GHz. Table 1 below presents possible power-saving practices at the hardware layer that can be considered for implementation. They are related to optimal hardware configurations and choices of technology.

Table 1: Efficient hardware requirements

Hardware Type	Efficiency Requirement
Power supply	At least 90% efficiency in converting Alternate Current (AC) to Direct Current (DC).
Processor/CPU	Low power. Pre-assembled onto the motherboard. Clock speed setting at 3.5 GHz. Use of Wattch software to measure and optimize the processor's power allocation at the architecture level.
Random Access Memory	Free choice due to insignificant power requirement.

(RAM)	
Hard drive storage	Solid State Drive (SSD) is preferred over magnetic drive.
Motherboard	Automatic disablement of unused components.
Graphic card	Free choice is subject to purpose.
WiFi card	WiFi 7 standard supporting card.
Cabled network card	Fiber optic based (90% less energy than copper).
Air conditioning	Sensor-driven temperature adjustment.
Lighting	Sensor-driven intensity adjustment.
IoT sensors	Opting for Zigbee or NB-IoT as the wireless connection protocol.

Platform Layer

The next layer to be made efficient is the platform layer. It refers to a computing system platform, which directly communicates with the hardware. The choice and configuration of the platform would significantly affect the power consumption. The first part to optimize in a computing system platform is the Operating System (OS). An OS bridges or mediates between hardware and software (Mwiinga, 2023; Sunil et al., 2018). Additionally, an OS also provides an interface – either via graphical or command line (text mode) to the user (Rathee, 2019; Ahamed, 2013; Bresnahan & Blum, 2020). A comparative study has been done on different OS' regarding their Graphical User Interface (GUI) comprehensiveness, hardware compatibility, portability, process management features, memory management, text mode, file system features, architecture, and security features (Odun-Ayo et al., 2020; Tan & Hakala, 2023). However, their comparisons are missing the power consumption aspect although the importance of power efficiency in OS has been highlighted (Khan, 2013; Sinha, 2014; Vahdat et al., 2000; Youssef, 2022).

To fill the above-identified gap in power-saving practice at the platform layer, this paper proposes the following purpose-aimed recommendations listed in Table 2. The previous recommendations pertain to the purposes of the platform. Text mode OS may be sufficient for services that do not require GUI navigation or GUI-driven monitoring capability. While certain OSes that are better at memory management are preferable for services with frequent memory access.

Table 2: Power saving recommended practices in the platform layer

Purpose	Recommended Configuration
Application server	Linux Text Mode
Authentication server	Linux Text Mode
Firewall server	Linux Text Mode
Proxy server	Linux Text Mode
Security monitoring server	Linux GUI Mode
Network routing	Linux Text Mode
Memory-intensive application	Macintosh GUI
Distributed computing server	Linux Text Mode

Received: 18 November 2023; Revised: 3 January 2024; Accepted: 1 April 2024; Published: 15 April 2024

Middleware Layer

Middleware is an intermediary software that connects two or more applications by delivering messages or data between them. It converts or bridges the gaps between heterogeneous systems with purposes for automation, networking, digitizing, optimization, and simplifying intra-communication. An example of conversion done by middleware is when two (2) systems implement different data formats; one uses JSON and the other one uses XML. Middleware is also capable of duplicating a request message to query multiple back-end servers. Furthermore, it may also behave as a proxy that conceals the back-end server.

The previously mentioned functions of middleware put a highly intense workload on it and hence raise the need for it to be efficient. In the context of mobile applications, transforming memory data layout from Transform Array of Structure (AOS) to Structure of Array (SOA) has managed to decrease energy consumption by middleware (Almusalli et al., 2017). A wider context of middleware role is in the area of distributed computing, such as a Campus Grid or Wireless Sensor Network (also known as the Internet of Things). The components inside this distributed computing are connected using a middleware called runtime infrastructure (RTI). A modified RTI named Green Runtime Infrastructure (G-RTI) was developed to reduce its power consumption by aggregating messages (Biswas et al., 2018), where the degree of energy saving is linear to the degree of message aggregation. Both push and pull IoT subscription models were supported.

The above-mentioned middleware optimizations have niche scopes, each for mobile phone ecosystems and IoT. Thus, further improvisations aimed at power saving on middleware would be recommended by this paper via the following Table 3.

Table 3: Power saving recommended practices in the middleware layer

Purpose	Recommended Configuration
General message queue	RabbitMQ and/or Apache Kafka
IoT message queue	G-RTI
Mobile phone queue	Transform memory data layout from AOS to SOA

Application Layer

Power saving at the application layer is the most practical and accessible to execute due to its portability and flexibility since applications are pluggable and can be moved between on-premise and on-cloud deployments. Efficient practices ranging from coding practices to configuration settings may be applied to reduce power consumption. Refer to the upcoming Table 4 for the respective recommended practices.

Table 4: Power saving recommended practices in the application layer

Purpose	**Recommended Configuration**
Anti-virus installation	Installation in the gateway firewall instead of individual computers.
Intrusion Prevention System (IPS) installation	Installation in the gateway firewall instead of individual computers.
Compute-intensive application	Distributed execution in multiple computers.
Code interpretation	Online code interpretation on the cloud.
Code compilation	Online code compilation on the cloud.
Frequent productivity application	Online productivity applications on the cloud.
Application design	Usage of dark theme. Minimization of the number of animations. Minimization of unnecessary scripts. Minimization of unnecessary dynamic web pages. Optimization of caching to reduce CPU cycles. Opting for push-based notifications instead of pull-based ones.
Application programming	Opting for compiled languages over-interpreted ones. Multi-threading. Efficient loop (avoiding recursion). Use of vectorization and instruction sets. Buffering and batching data requests into one operation. Minimization of cache size. Minimization of the number of data transfers between processor and memory.
Programming tools to save energy	Use of PowerEscape Insight, PowerEscape Analyzer, and PowerEscape Architect for C language programming. Use of VTune Analyzer for C, C++, Fortran, NET, Java, C#, or Visual Basic programming. Use of Intel Performance Tuning Utility (Intel PTU) to analyze code performance. Use of Koala software to predict power usage of the application.
Application usage	Use of one application only at one time. Use of Intel Power Checker to measure energy efficiency of applications running on Intel processors. Use of PowerTOP to detect power consumption and management issues on Linux computers. Use of powercfg to identify if an application has triggered to spike the platform timer resolution that may cause higher power consumption. Use of POWERAPI to calculate power usage of applications in real-time.

The above-suggested practices revolve around software/application management and efficient coding practices. The implementations are expected to be the quickest compared to the rest of the layers since the application layer is the closest to users.

CONCLUSION

Our proposed multi-layer method to save power consumption in a classroom covers all essential technology-related controllable parts. It consolidates different aspects or dimensions of energy saving, which is expected to significantly minimize power consumption compared to the unoptimized one, subject to the percentage of compliance. Considering its comprehensiveness and autonomy among every layer, the public may selectively comply according to their bespoke feasibility. This method could be a one-stop reference for hardware and software engineers to assist them in providing efficient classroom solutions for their clients. Affordable green classrooms would accelerate its construction,

Received: 18 November 2023; Revised: 3 January 2024; Accepted: 1 April 2024; Published: 15 April 2024

especially in areas where power generation is limited, therefore promoting equal IT education in remote districts.

FUTURE WORK

Future expansion of our proposed method would be affected by the quickly growing IoT-driven smart cities all over the world, which may evolve new hardware and software technologies that require new optimization techniques – depending on parameters. The spawning of smart cities would encourage hardware and software vendors to produce new devices and applications, which may eventually be adopted in IT classrooms. These classrooms may be able to develop sophisticated software products that could subsequently improve the economy of global nations.

REFERENCES

Ahamed, F. (2013). *The Fdos operating system*. Computer Society of India.

Almusalli, F., Zaman, N., & Rasool, R. (2017). Energy efficient middleware: Design and development for mobile applications. *19th International Conference on Advanced Communication Technology (ICACT),* 541-549. https://doi.org/10.23919/ICACT.2017.7890149.

Biswas, A., Fujimoto, R., & Hunter, M. (2018). Energy efficient middleware for dynamic data driven application systems. *2018 Winter Simulation Conference (WSC),* 628-639. https://doi.org/10.1109/WSC.2018.8632433.

Bresnahan, C. and Blum, R. (2020). *LPI Linux essentials study guide*. Wiley. https://doi.org/10.1002/9781119657712.ch1.

Diddeniya, S. A. I. P., Gunawardana, H. D. C. N., Maduwantha, K., Koswattage, K. R., Randima, M. V., & Vasanthapriyan, S. (2020). IoT based energy efficient smart classroom. *Journal of Multidisciplinary Engineering Science Studies*, *6*(12), 3581-3586.

Herzog, B., Hugel, F., Reif, S., Honig, T., & Schröder-Preikschat, W. (2021). Automated selection of energy-efficient operating system configurations. *e-Energy '21: Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, 309–315. https://doi.org/10.1145/3447555.3465327.

Khan, W. (2013). Energy efficient mobile operating systems. *International Journal of Advanced Networking and Applications*, *5*(1), 1812-1817.

Mason, L., Manzione, L., Ronconi, A., & Pazzaglia, F. (2022). Lessons in a green school environment and in the classroom: Effects on students' cognitive functioning and affect. *International Journal of Environmental Research and Public Health*, *19*(24), 1-16. https://doi.org/10.3390/ijerph192416823.

Mason, L., Ronconi, A., Scrimin, S., & Pazzaglia, F. (2022). Short-term exposure to nature and benefits for students' cognitive performance: A review. *Educational Psychology Review*, *34*(2), 1-39. https://doi.org/10.1007/s10648-021-09631-8.

Mason, L., Zagni, B., Bacchin, F., Frison, C., & Scrimin, S. (2022). Children's attentional processes in outdoor and indoor environments: The role of physiological self-regulation. *International Journal of Environmental Research and Public Health*, *19*(20), 1-14. https://doi.org/10.3390/ijerph192013141.

Mwiinga, P. (2023). *Operating system*. Real Deal.

Nath, P. B. & Uddin, M. M. (2015). TCP-IP model in data communication and networking. *American Journal of Engineering Research*, *4*(10), 102-107.

Odun-Ayo, I., Okokpujie, K., Oputa, K., Ogbu, H., Emmanuel, F., Shofadekan, A., & Okuazun, G. (2020). Comparative study of operating system quality attributes. *IOP Conf. Series: Materials Science and Engineering*. https://doi.org/10.1088/1757-899X/1107/1/012061.

Patil, B. K., Badgujar, R. D., Nil, N. L., Suryawanshi, S. N., Patel, D. C., & Bagul, P. P. (2018). Energy efficient smart classroom. *International Journal for Research in Applied Science & Engineering Technology, 6*(6), 85-89.

Rathee, P. (2019). *Basic principles of an operating system.* BPB Publications.

Santos, D. & Ferreira, J. C. (2019). IoT power monitoring system for smart environments. *Sustainability*, *11*(19), 5355-5378. https://doi.org/10.3390/su11195355.

Sasane, N. N., Sakat, S.N., Nemane, S. K., Pallav, P. K., & Kaushik, V. R. (2017). IoT based energy meter billing and monitoring system - A case study. *International Research Journal of Advanced Engineering and Science*, *2*(4), 64-68.

Simpson, W. (2002). Energy sustainability and the green campus. *Planning for Higher Education*, *31*(3), 150-158.

Sinha, A. (2014). *Energy efficient operating systems and software* [Doctoral dissertation, Massachusetts Institute of Technology].

Sunil, J., Vincy, A. J., & Thirumalai, K. (2018). *Fundamentals of operating systems concepts*. Lambert Academic Publications.

Taemthong, W. and Plitsiri, I. (2023). Using active green wall systems for both saving energy and improving indoor air quality in classrooms. *E3S Web of Conferences*. https://doi.org/10.1051/e3sconf/202337904002.

Tan, X. & Hakala, L. (2023). StateOS: A memory-efficient hybrid operating system for IoT devices. *IEEE Internet of Things Journal*, *10*(11), 9523-9533. https://doi.org/10.1109/JIOT.2023.3234106.

Tiyarattanachai, R. & Hollmann, N. M. (2016). Green campus initiative and its impacts on quality of life of stakeholders in green and non-green campus universities. *Springerplus*, *5*(1), 84-100. https://doi.org/10.1186/s40064-016-1697-4.

Vahdat. A., Lebeck, A., & Ellis, C. S. (2000). Every Joule is precious: The case for revisiting operating system design for energy efficiency. *EW 9: Proceedings of the 9th workshop on ACM SIGOPS European workshop: Beyond the PC: New challenges for the operating system*. 31-36, https://doi.org/10.1145/566726.566735.

Vella-Brodrick, D. & Gilowska, K. (2022). Effects of nature (greenspace) on cognitive functioning in school children and adolescents: A systematic review. *Educational Psychology Review*, *34*(3), 1-38. https://doi.org/10.1007/s10648-022-09658-5.

Youssef, J. (2022). *The influence of operating system on the energy consumption of software and algorithms* [Bachelor thesis, University of Stuttgart]. OPUS - Publication Server of the University of Stuttgart. http://dx.doi.org/10.18419/opus-12181.

Zangori, L. & Cole, L. (2019). Assessing the contributions of green building practices to ecological literacy in the elementary classroom: An exploratory study. *Environmental Education Research*, *25*(11), 1674-1696. https://doi.org/10.1080/13504622.2019.1662372.