European Journal of Information Technologies and Computer Science
Vol 4 | Issue 4 | October 2024
ISSN 2736-5492

RESEARCH ARTICLE

A Comparative Study of CPU and GPU Power Consumption while using Open-Source and Proprietary Media Players

Afzal Ahmed*, Mohammad Tariq Iqbal, and Mohsin Jamil

ABSTRACT

This study presents a comparative analysis of power consumption between open-source and proprietary media players when playing open-media format videos (.webm). As media consumption grows, energy-efficient software is critical for both environmental sustainability and device performance. Using tools like HWiNFO, key metrics such as GPU and CPU power consumption, memory usage, and efficiency were evaluated for popular open-source (e.g., VLC, Kodi) and proprietary (e.g., GOM Player, KMPlayer) players. The results reveal that open-source players generally consume less GPU power but more CPU resources, while proprietary players balance CPU and GPU usage with higher memory demands. The findings suggest that careful selection of media players can lead to significant energy savings over time, offering insights for developers and users focused on energy-efficient computing.

Keywords: Media players, open source, power consumption, proprietary.

Submitted: September 10, 2024
Published: October 26, 2024

10.24018/ejcompute.2024.4.4.135

Department of Electrical and Computer Engineering, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, Canada.

****Corresponding Author:***
Email: afzala@mun.ca

1. Introduction

The continuous advancement in media technology has brought about a diverse range of software platforms that cater to the consumption of digital media, such as videos and audio. These platforms can broadly be categorized into two types: open-source and proprietary media players. Open-source media players are developed and distributed freely, with their source code available to the public, allowing for transparency, customizability, and collaborative improvements. In contrast, proprietary media players are typically commercial products with closed-source code, developed by corporations or private entities, and come with restricted access to their inner workings. Both types of media players have their unique advantages and limitations, particularly in terms of performance, cost, security, and energy consumption, especially when handling various media formats like open standards such as .webm.

The growing adoption of open-source media players in the technology landscape has been driven by the desire for more transparency, flexibility, and user control [1]. On the other hand, proprietary software continues to dominate certain market segments, owing to the perceived superiority in performance, customer support, and proprietary features [2]. With the global increase in video consumption across various devices, an often-overlooked factor is the power consumption of these media players, particularly as it pertains to their use of CPU, GPU, and memory resources.

Power consumption in software applications is increasingly relevant, given the growing awareness of environmental sustainability and energy conservation [3]. In this context, a comparison of open-source and proprietary software from the perspective of energy efficiency is both timely and necessary. These studies revealed that the choice between open-source and proprietary software significantly affects the usage perspective [4], but perspective of power usage is still unexplored, which can have broader implications when considering large-scale deployments or extended usage scenarios.

This paper seeks to explore the energy consumption differences between open-source and proprietary media players when playing open media formats, specifically. webm videos. We aim to investigate whether open-source media players are more energy-efficient in terms of CPU and GPU power consumption, memory usage, and overall system resource utilization, compared to their proprietary counterparts. Our experiments will focus on real-world usage scenarios, and the results will be evaluated in the context of daily use and longer-term energy conservation strategies.

2. Literature Review

The debate between open-source and proprietary software isn't just about accessibility and features; it also hinges on performance optimization and energy usage. Media players, in particular, reveal stark differences in how they utilize hardware. In the following sections, we will delve into the detailed comparison between open-source and proprietary software, particularly in terms of media players

2.1. *Open Source vs. Proprietary Software*

The comparison between open-source and proprietary software has been a central theme in the field of software development and digital technology. Open-source software (OSS) is defined by its availability to the public, allowing users to view, modify, and distribute the source code [5]. In contrast, proprietary software is typically closed-source and is sold as a product, with restrictive licenses preventing unauthorized access to the code [6].

Several studies have focused on the economic implications of choosing between OSS and proprietary software. Chesbrough highlighted that open-source software often reduces costs for organizations, as there are no licensing fees, and the collaborative nature of open-source development leads to faster bug fixes and feature updates [7]. However, proprietary software is often seen as more stable and better supported by vendors, offering dedicated customer support and advanced functionalities [8].

When it comes to performance, some studies suggest that proprietary software has an edge, particularly in specialized use cases [9]. However, the flexibility and adaptability of OSS make it a popular choice among developers, particularly for customization-heavy applications. In contrast, proprietary software may offer more seamless integration and user-friendly interfaces. Additionally, security is often a key concern, with OSS sometimes criticized for potential vulnerabilities due to its open nature, though this is often mitigated by the large community of developers contributing to the code [10].

2.2. *Media Players: Open Source and Proprietary Solutions*

Media players are an essential category of software, with various open-source and proprietary options available for consumers. Some well-known open-source media players include VLC, Kodi, and MPV, while proprietary options include Windows Media Player, RealPlayer, and GOM Player. These platforms cater to a variety of media formats, including open formats like .webm, which are commonly used due to their efficiency and lack of licensing restrictions [11].

Previous studies on media player performance have shown that open-source solutions like VLC and Kodi are highly regarded for their versatility and support for a wide range of formats [12]. VLC, for example, is renowned for its ability to play virtually any file format without needing additional codecs [13]. However, proprietary media players often boast more polished user interfaces and optimized performance, especially in hardware-accelerated tasks such as 4K video playback [5].

The impact of open-source software (OSS) on proprietary software has been substantial, particularly in competitive settings. Zhou *et al*. found that competition from OSS could push proprietary providers to enhance both the quality and price of their software, contrary to earlier assumptions that OSS would lower quality. This dynamic is evident in the media player market, where OSS forces proprietary players to innovate, often at the cost of higher energy consumption due to feature expansion [14].

Costa *et al*. examined proprietary software ecosystems (SECOs), highlighting the importance of intellectual property protection while fostering innovation. In the case of media players, proprietary platforms often consume more resources due to their advanced features and background processes. These governance mechanisms, while enhancing platform stability, often come with a higher energy cost compared to their open-source counterparts [15].

2.3. *Power Consumption of Daily-Use Software*

The energy consumption of daily-use software, including media players, has been the subject of several studies in recent years. Energy efficiency is increasingly becoming a priority for software developers and users alike, particularly in the context of climate change and rising energy costs. Katal *et al*. have analyzed the power consumption of various software applications with a focus on minimizing resource usage and improving energy efficiency [4].

Media players are particularly significant in this regard, as they are used for extended periods in many daily routines. This was attributed to the lightweight nature of the codebase in open-source projects, as well as the community-driven focus on efficiency and performance optimizations [3].

Zhang *et al*. discussed energy consumption differences between open-source and proprietary systems, noting that OSS, with its modular architecture, is generally more energy-efficient. Proprietary media players, on the other hand, tend to be more feature-rich but resource-intensive, resulting in higher power consumption [16].

Henkel found that many commercial firms adopt hybrid models, where open-source projects complement proprietary software. This approach is particularly useful in media players, where open-source components can reduce development costs and improve efficiency while maintaining core proprietary functionality [6].

2.4. *Open Media Formats and Resource Efficiency*

Open media formats such as .webm, developed by Google, are increasingly popular due to their royalty-free status and efficient compression algorithms [11]. Studies has shown that media players optimized for these open formats tend to be more resource-efficient in decoding and playback [13]. This is in contrast to proprietary formats like .mp4 or .mov, which often require specialized hardware or software to decode, leading to increased power consumption [17].

VLC, an open-source media player, has been found to be particularly efficient when playing open media formats like .webm, utilizing less CPU and memory compared to proprietary players. This aligns with the broader trend observed in open-source software, where community-driven development often leads to more lightweight and efficient codebases [5].

Le Feuvre *et al*. demonstrated the effectiveness of the GPAC multimedia framework, an open-source solution that handles media playback with minimal resource consumption. This study underscored the benefits of open-source frameworks in managing complex tasks like video encoding and playback with lower energy demands, making them ideal for open media formats like .webm [18].

2.5. HWiNFO Studies on Power Consumption

HWiNFO is a widely used tool for monitoring system resources such as CPU, GPU, and memory usage, making it an ideal choice for measuring the power consumption of media players [12]. Study by Singh *et al*. has utilized HWiNFO to gather detailed performance metrics during predictive analysis, providing valuable insights into how different analysis affect overall system power consumption [8].

Although previous research has explored software efficiency and market competition between OSS and proprietary systems, there is still a gap in understanding the energy consumption specific to media players, especially for open media formats like .webm. This study seeks to address this gap by analyzing the power consumption of open-source and proprietary media players, providing valuable insights for optimizing energy efficiency and understanding trade-offs between feature-rich proprietary software and OSS.

3. Experiment

The primary objective of this experiment was to compare the power consumption of open-source and proprietary media player software. The focus was on evaluating CPU power package, GT core power, percentage of CPU usage, and physical memory consumption in megabytes (MBs).

3.1. Media Players Evaluated

In this study, five open-source and seven proprietary media players were evaluated to compare their power consumption and system resource usage during video playback. The open-source media players tested were VLC, Kodi, MPV, SMP, and MPC. The proprietary media players included Windows Media Player, KMPlayer, GOM Player, RealPlayer, ALLPlayer, LAPlayer, and POT Player. These players were selected based on their popularity, diverse functionalities, and compatibility with open media formats such as .webm, ensuring that the results represent common real-world usage.

3.2. Hardware Configuration

The experiments were conducted on the following hardware configuration:

- *Processor:* 12th Gen Intel(R) Core(TM) i7-12700H
- *Base Clock Speed:* 2300 MHz
- *Cores: 14* cores
- *Logical Processors:* 20 logical processors
- *Physical Memory Available:* 16 GB DDR4
- *Operating System:* Microsoft Windows 11 Home
- *OS Version*: 10.0.22631 Build 22631
- *Display:* Intel(R) Iris(R) Xe Graphics
- *Adapter Type:* Intel(R) Iris(R) Xe Graphics Family, Intel Corporation compatible
- *Driver Version:* 31.0.101.4575

This hardware was chosen to ensure consistency across all tests, minimizing any performance variability caused by hardware differences.

3.3. Tools Used

To measure and record power consumption and system resource usage, the following tools were utilized:

- HWiNFO: Used to monitor real-time system performance, including CPU power consumption, GPU power consumption, memory usage, and CPU utilization. HWiNFO was critical for capturing high-resolution power consumption data at 1-second intervals (1000 ms).
- Microsoft Excel: Used for processing and visualizing the collected data, and performing statistical analyses.

3.4. Media Playback Testing Environment

All media players were tested using a standard video file to ensure consistency:

- *Video Resolution:* 4K (3840 × 2160)
- *Video Length:* 3 minutes and 14 seconds
- *File size:* 101 MBs
- *Codec*: VP9
- *Format:* .webm

This video was chosen to represent typical high-definition media consumption. During testing, no other background tasks or applications were running on the system to ensure that resource consumption could be attributed solely to the media players.

In this experiment, we used WebM because it stands out as a superior choice for the following reasons:

- *Royalty-Free:* Unlike H.264, H.265, and other proprietary formats, WebM (VP8/VP9) is completely royalty-free, eliminating licensing fees and legal complexities [19], [20].
- *Open-Source:* WebM and its codecs are fully open-source, aligning perfectly with the goals of your study focused on open formats [21]. This allows transparency and flexibility in development and distribution, as opposed to proprietary standards like H.264 and H.265 [22], [23].
- *Web Support:* WebM is optimized for the web and is natively supported by HTML5 [24], making it a common choice for web streaming and online platforms. Its broad compatibility with browsers like Chrome and Firefox makes it ideal for open and accessible media use, unlike the limited web support of H.265 [25].

- *Compression Efficiency*: WebM (VP9) offers competitive compression efficiency, similar to H.265/HEVC, but without the licensing costs [26]. This makes it efficient for streaming high-quality videos while saving bandwidth, which is key for the performance and power efficiency aspects of this study [27].
- *Media Players compatibility*: Another reason for choosing WebM (VP9) is its vast compatibility and it was supported by all the media players under consideration. Windows media players and real media players do not natively support WebM or VP8/VP9 playback without installing third-party codecs, and for this study, K-lite codec pack was installed, which helped run all the applications smoothly [28].

Additionally, during the study, an MP4 file was successfully converted to WebM using a third-party app called BeeConverter, demonstrating the ease with which files in other formats can be transformed into the open WebM format for better compatibility with your research goals [29].

Considering these factors, WebM was the most fitting open media format for this research, as it aligns with the principles of openness, efficiency, and accessibility.

3.5. Data Collection Methodology

For each media player, the following metrics were recorded every 1000 milliseconds (1 second) during video playback:

- GPU Power Consumption (Watts)
- CPU Power Consumption (Watts)
- Memory Usage (MB)
- CPU Utilization (%)

Each media player was tested over three playback sessions. The data from these sessions were averaged to provide the final figures used in the analysis.

3.6. Repetition and Averaging

To ensure accuracy and account for variations in system performance, each test was repeated three times per media player. The results from each session were averaged to reduce any anomalies or irregular spikes in power consumption, ensuring the reliability of the data.

3.7. Data Analysis

Once data collection was completed, Microsoft Excel and Matplotlib were used to analyze the results. The average GPU power consumption, CPU power consumption, memory usage, and CPU utilization for each media player were calculated. Independent t-tests were conducted to determine if the differences in power consumption and resource usage between open-source and proprietary media players were statistically significant.

3.8. Additional Considerations

To ensure the accuracy and reliability of the results, the following considerations were taken into account:

- *Cooling:* The system's cooling fans were set to a constant speed to ensure that varying fan speeds did not interfere with power consumption readings.
- *Performance Mode:* The system was set to "High Performance" mode in Windows to prevent power-saving features from influencing the results.
- *Background Applications:* All non-essential background tasks and services were disabled to ensure that the recorded data reflected only the resource usage of the media players.

4. Results

The following sections results of key metrics such as GPU and CPU power consumption, memory usage, and CPU utilization were analyzed and compared.

4.1. GPU Power Consumption

Open-source media players exhibited lower GPU power consumption on average, with VLC consuming the least at 0.0856 W and MPV the highest at 0.5070 W.

Proprietary media players demonstrated more consistent, but generally higher, GPU power consumption, with KMPlayer averaging 0.2291 W and POT Player at 0.2781 W. Graphs in Figs. 1 and 2 show the GPU power consumption for each open-source media player and Proprietary media player, respectively, highlighting the efficiency of open-source players like VLC in comparison to proprietary options such as KMPlayer.

4.2. CPU Power Consumption

1. *Open-source media players:*

VLC and Kodi showed the lowest CPU power consumption at 3.68 W and 4.03 W, respectively. MPV had the highest CPU power consumption among open-source players at 5.96 W.

2. *Proprietary media players:*

KMPlayer consumed 6.02 W, and Windows Media Player consumed 5.56 W, reflecting slightly higher CPU power usage compared to their open-source counterparts.

Figs. 3 and 4 illustrate the CPU power consumption for each media player, with open-source media players generally exhibiting better efficiency.

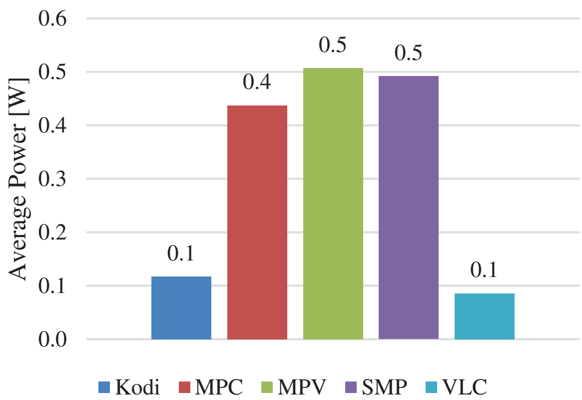

Fig. 1. Average GPU power consumption of open-source media players.

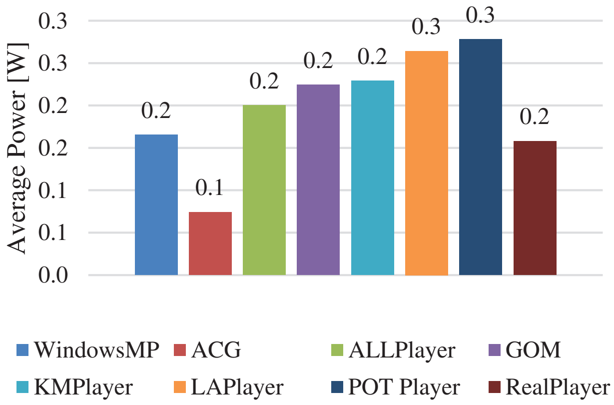

Fig. 2. Average GPU power consumption of proprietary media players.

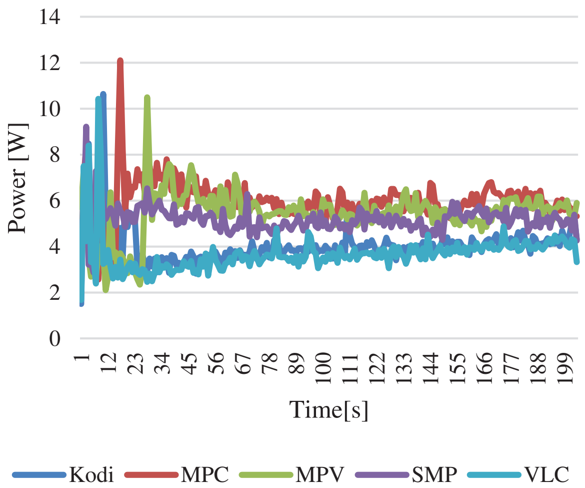

Fig. 3. Power consumption of open-source media players.

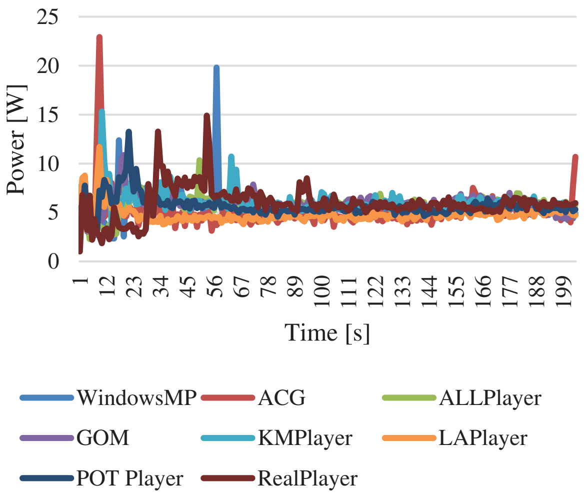

Fig. 4. Power consumption of proprietary media players.

4.3. CPU Utilization

1. *Open-source media players:*

VLC showed the lowest CPU utilization at 6.41%, while MPC reached 10.75%.

2. *Proprietary media players:*

GOM Player had the highest CPU utilization at 17.5%, followed by KMPlayer with 12.69%.

TABLE I: Average CPU Usage and Power Consumption

Media player	Average CPU power consumption	Average CPU usage percentage
Kodi	4.032	7.2%
MPC	5.959	10.8%
MPV	5.445	9.1%
SMP	5.177	9.1%
VLC	3.685	6.4%
WindowsMP	5.568	11.8%
ACG	5.041	8.4%
ALLPlayer	5.859	11.0%
GOM	5.983	17.5%
KMPlayer	6.017	12.7%
LAPlayer	4.745	8.6%
POT Player	5.697	10.3%
RealPlayer	5.927	13.2%

Table I depicts the average CPU utilization percentages and average power consumption, showing that proprietary media players generally consume more CPU resources and, hence, burn more power.

4.4. Memory Usage

Proprietary media players generally used more memory compared to open-source players, with KMPlayer consuming the most memory (7562 MB), while VLC consumed around 6705 MB.

If we calculate the average memory power consumption for the media players, it can be calculated based on their memory usage, with open-source players consuming less power on average. For example, VLC, being the most efficient, consumed approximately 1.43 W from memory as it uses the least memory, while KMPlayer, being the most resource occupant, used around 2.13 W.

Figs. 5 and 6 display the physical memory usage of open-source and proprietary media players, respectively, with open-source players consistently using less physical memory.

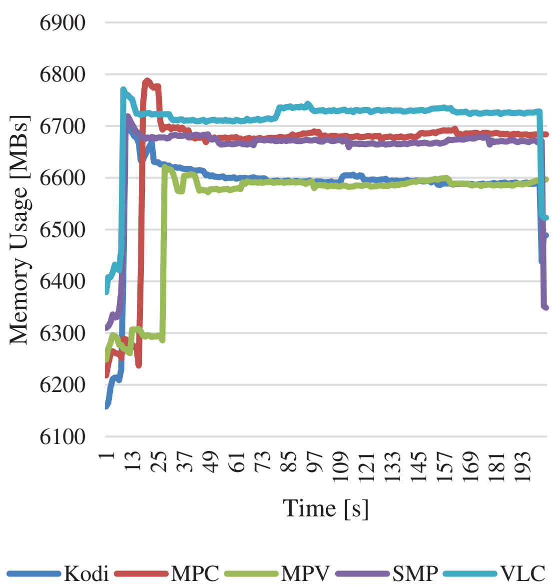

Fig. 5. Physical memory usage by open-source media players.

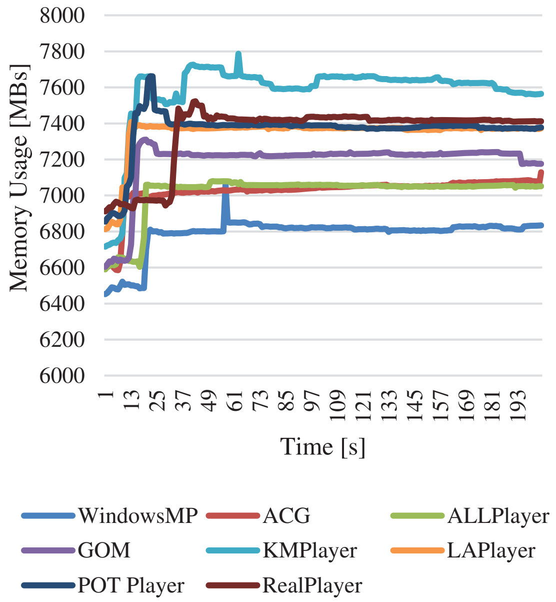

Fig. 6. Physical memory usage by proprietary media players.

TABLE II: Energy Saving of VLC as Compared to All Other Media Players

Media player	Annual energy consumption (kWh)	Energy savings (kWh)
VLC	2.75	0
Kodi	3.03	0.28
MPV	4.72	1.97
SMP	4.14	1.39
MPC	4.29	1.54
Windows MP	4.19	1.43
KMPlayer	4.56	1.81
GOM Player	4.42	1.67
RealPlayer	4.53	1.78
ALLPlayer	4.35	1.6
LAPlayer	3.67	0.91
POT Player	4.44	1.69

4.5. *Long-term Energy Consumption*

As shown in Table II over a year of daily video playback (assuming two hours of use per day), using VLC could save approximately 1.80 kWh of energy compared to KMPlayer. While this may seem like a small amount, these savings can add up significantly when considering large-scale deployments or heavy users.

Fig. 6 visualizes the estimated annual energy savings for each media player based on daily usage scenarios.

4.6. *Statistical Analysis*

Statistical tests revealed significant differences in memory usage between open-source and proprietary media players ($p < 0.05$), with proprietary players generally consuming more memory. However, no significant differences were observed in GPU power consumption ($p > 0.05$), suggesting that GPU efficiency is relatively consistent across both types of players.

Table III presents the results of the t-tests for key performance metrics, highlighting where statistically significant differences exist.

TABLE III: Statistical Analysis of Significant Differences

Performance metric	p-value	Significance
GPU power consumption	0.1264	Not significant
CPU power consumption	0.0855	Marginally significant
Memory usage	0.00053	Significant
Memory power consumption	0.0015	Significant
CPU utilization	0.0012	Significant

5. Conclusions and Future Work

This study highlights the significant differences in power consumption and resource usage between open-source and proprietary media players during 4K video playback. Open-source media players, particularly VLC, demonstrated superior efficiency in both GPU and CPU power consumption, making them more suitable for energy-conscious users. Proprietary players, however, often consume more physical memory and CPU resources due to their feature-rich environments, advanced user interfaces, and additional background services. Despite their higher resource demands, proprietary media players provide a more comprehensive media experience. These findings emphasize the importance of selecting media players based on the specific needs of the user, whether prioritizing energy efficiency or enhanced functionality. The study also underscores the role that software choices can play in long-term energy savings, especially when media consumption is a regular part of users' daily activities.

Future research could expand on this study by examining a broader range of media formats and codecs, including those more commonly used in proprietary ecosystems, to explore whether similar trends in power consumption persist.

Conflict of Interest

The authors declare that they do not have any conflict of interest.

References

[1] Fortunato L, Galassi M. The case for free and open source software in research and scholarship. *Philos Trans R Soc A*. 2021;379:2179.

[2] Blancaflor EB, Samonte SA. An analysis and comparison of proprietary and open-source software for building e-commerce websites. *J Adv Inf Technol*. 2023;14(3):426–31.

[3] Blind K, Böhm M, Grzegorzewska P, Katz A, Muto S, Pätsch S, Schubert T. The impact of Open Source Software and Hardware on technological independence, competitiveness and innovation in the EU economy. *Final Study Report. Europ Commiss, Brussels*. 2021;10:430161.

[4] Katal A, Dahiya S, Choudhury T. Energy efficiency in cloud computing data centers: a survey on software technologies. *Cluster Comput*. 2023;26(3):1845–75.

[5] Pearce JM. Economic savings for scientific free and open source technology: a review. *HardwareX*. 2020;8:e00148.

[6] Henkel J. Open source software from commercial firms: tools, complements, and collective invention. *Zeitschrift für Betriebswirtschaft*. 2004;4:1–23.

[7] Chesbrough H. *Measuring the Economic Value of Open Source*. San Francisco: Linux Foundation; 2023.

[8] Kelty CM. *Two Bits: the Cultural Significance of Free Software*. Durham, NC: Duke University Press; 2020.

[9] Jin C, Bai X, Yang C, Mao W, Xu X. A review of power consumption models of servers in data centers. *Appl Energy*. 2020;265:114806.

[10] Khajuria R, Kumar A, Anand T, Jain S, Dayal P, Banerjee S. Efficient video streaming on raspberry Pi 4B: reducing CPU utilization through optimization techniques. *2024 International Conference on Expert Clouds and Applications (ICOECA)*, pp. 942–7, Bengaluru, Karnataka, India: IEEE; 2024, April.

[11] Ashok Kumar L, Angalaeswari S, Mohana Sundaram K, Bansal RC, Patil A. Intelligent Solutions for Sustainable Power Grids. *IGI Global*. 2024. https://doi.org/10.4018/979-8-3693-3735-6.

[12] Hrauda W. Virtual binaural acoustics in VLC player: hRTFs and efficient rendering. 2015. Available from: https://iem.kug.ac.at/fileadmin/03_Microsites/01_Kuenstlerisch_wissenschaftliche_Einheiten/01_Institute/Institut_17_Elektronische_Musik_und_Akustik/Projekte/2014/hrauda.pdf.

[13] Timmerer C, Wien M, Yu L, Reibman A. Special issue on open media compression: overview, design criteria, and outlook on emerging standards. *Proc IEEE*. 2021;109(9):1423–34.

[14] Zhou ZZ, Vidyanand C. Impact of competition from open source software on proprietary software. *Prod Oper Manag*. 2022;31(2):731–42.

[15] Costa LA, Fontão A, Santos R. Investigating proprietary software ecosystem governance and health: an updated and refined perspective. *Proceedings of the XVII Brazilian Symposium on Information Systems*, pp. 1–8, Uberlândia, Brazil, June 2021.

[16] Zhang X, Wen Y, Fan R. Energy-aware cloud computing: simulation and experimentation of energy-aware resource allocation. *IEEE Trans. Cloud Comput*. 2022;10(2):123–35.

[17] Bamhdi A. Requirements capture and comparative analysis of open source versus proprietary service oriented architecture. *Comput Stand Inter*. 2021;74:103468.

[18] Le Feuvre J, Concolato C, Moissinac JC. GPAC: open source multimedia framework. *Proceedings of ACM International Conference on Multimedia*, pp. 1009–12, Augsburg, Germany, 2007.

[19] WebM Project. An open, royalty-free media file format designed for the web. WebM Project. 2024. Available from: https://www.webmproject.org. Accessed: Sept. 9, 2024.

[20] Free Software Foundation. Licenses for open source projects. FSF. 2024. Available from: https://www.fsf.org. Accessed: Sept. 9, 2024.

[21] WebM Project. The WebM Project: Open, royalty-free media file format. WebM Project. 2024. [Online]. Available from: https://www.webmproject.org/about/. Accessed: Sept. 9, 2024.

[22] VideoLAN VideoLAN. x264, the best H.264/AVC encoder. VideoLAN. 2024. Available from: https://www.videolan.org/developers/x264.html. Accessed: Sept. 9, 2024.

[23] HEVC Advance. Licensing information for H.265/HEVC. HEVC advance. 2024. Available from: https://hevcadvance.com. Accessed: Sept. 9, 2024.

[24] Google Developers. WebM and the VP8 codec: royalty-free video. Google Developers. 2024. Available from: https://developers.google.com/web/updates/2010/05/WebM-VP8-Codec. Accessed: Sept. 9, 2024.

[25] CanIUse. Browser support for webm video format. CanIUse. 2024. Available from: https://caniuse.com/webm. Accessed: Sept. 9, 2024.

[26] Streaming Media. *VP9 vs H.265: a Comparison of Streaming Performance*. Medford, NJ, USA: Streaming Media; 2024.

[27] FFmpeg. The comprehensive codec library: vP9 and WebM support. FFmpeg. 2024. Available from: https://ffmpeg.org. Accessed: Sept. 9, 2024.

[28] BeeConverter. MP4 to WebM conversion: a simple guide. BeeConverter. 2024. Available from: https://beeconverter.com. Accessed: Sept. 9, 2024.

[29] RealNetworks. RealPlayer—Media player. RealNetworks. Available from: https://www.real.com. Accessed: Sept. 9, 2024.