http://ejournal.seaninstitute.or.id/index.php/InfoSains
Jurnal Info Sains : Informatika dan Sains, Volume 13, No 03 2023
E-ISSN.2797-7889, P-ISSN.2089-3329

Efficiency Analysis of Compression Software (WINRAR and 7-Zip) Across Diverse Data Types on Windows 11 and Ubuntu 23.10

[1]Rakha Maulana, [2]Bahteramon Bintang Sanjaya Manurung, [3]Nathanael Berliano Novanka Putra, [4]Aqwam Rosadi Kardian
[1,2,3]Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara, [4]Sistem Informasi,STMIK Jakarta STI&K
Email: rakha.maulana@student.poltekssn.ac.id[1], bahteramon.bintang@student.poltekssn.ac.id[2], nathanael.berliano@student.poltekssn.ac.id[3], aqwam@staff.jak-stik.ac[4]

Keywords

Compression Software, WINRAR, 7-Zip, Data Types, Operating Systems

Abstract. This paper presents a comprehensive analysis of the performance and efficiency of two widely used compression software, WINRAR and 7-Zip, across various data types. The study focuses on evaluating their effectiveness on different operating systems, specifically Windows 11 and Ubuntu 23.10. The analysis encompasses considerations such as compression ratios, resource utilization, and processing times. WINRAR and 7-Zip are examined in diverse scenarios, including the compression of text files (.txt), image files (.png), audio files (.flac), and video files (.mp4). The study reveals notable variations in compression outcomes influenced by intrinsic complexities of each file format. Moreover, the investigation extends beyond the initially studied operating systems, suggesting potential applications on other platforms like Kali Linux. The findings contribute insights into the nuanced performance of compression software across varied data types and operating environments, facilitating informed decision-making for users seeking optimal compression solutions.

1. INTRODUCTION

In the ever-evolving landscape of digital interactions, two compression software giants, WINRAR and 7-Zip, have ascended to prominence due to their unparalleled reliability, speed, and feature-rich capabilities. Their widespread adoption underscores their pivotal role in shaping user experiences within the digital realm.

This study embarks on a meticulous analysis, seeking to unveil the nuanced efficiency and efficacy of WINRAR and 7-Zip, with a specific lens on their performance across varied data types and two of the most prevalent operating systems Windows 11 and Ubuntu 23.10. In the realm of everyday digital data, file formats such as PNG for images, TXT for text, FLAC for high-quality audio, and MP4 for videos represent the common types of data handled by users. The decision to focus on Windows 11 and Ubuntu 23.10 is grounded in their status as the preeminent operating systems, collectively catering to a vast majority of the global user base.

The task at hand involves the file formats FLAC, TXT, PNG, and MP4. The references provide valuable insights into various file formats, including their applications and characteristics. Specifically, the references shed light on the use of file formats in different fields such as medicine, computer science, and data analysis. The references cover topics ranging from cataract surgery to genomics data mining and spectrum synthesis. By synthesizing these references, a comprehensive understanding of the file formats and their applications can be achieved.

FLAC is a lossless audio compression format known for its high-quality audio reproduction. It is widely used in the music industry and by audiophiles due to its ability to preserve the original audio quality [1]. The format is particularly valuable for applications where audio fidelity is crucial, such as music production and archiving.

TXT, as a plain text format, is commonly used for storing and exchanging textual data. It is a simple and universal format that can be easily read and edited by both humans and computer programs. The reference by highlights the prevalence of TXT files in research data, indicating its widespread use in scientific papers [2].

PNG is a popular image format known for its lossless compression and support for transparent backgrounds. It is widely used for web graphics and digital photography due to its ability to maintain

high image quality while reducing file size. The format is particularly suitable for images with text, line art, and graphics with sharp transitions [3].

MP4 is a versatile multimedia container format widely used for storing video, audio, and subtitles due to its flexibility, compression capabilities, and wide support across platforms and devices [4]. Its suitability for various applications, including video production, online streaming, and mobile devices, has made it a popular choice in the 'multimedia' industry. The format's ability to store different types of media in a single file makes it convenient for sharing and distributing multimedia content. Additionally, MP4's compression capabilities contribute to efficient storage and streaming of high-quality multimedia content.

The strategic choice of scrutinizing performance on both Windows and Ubuntu operating systems is crucial for providing insights relevant to a diverse spectrum of users. Windows 11 is a widely used operating system by users. The adoption of Windows 11 has been significant, with many users utilizing this operating system for various purposes. Windows 11, known for its user-friendly interface, enhanced security features, and improved performance, has gained significant adoption [5]

The widespread use of the Ubuntu operating system can be attributed to its representation of self in virtual interactions, its dynamic and changing approaches to upholding indigenous values, and its application in various fields such as digital entrepreneurship, education, and parenting [6] [7] [8]. Additionally, the Ubuntu operating system is designed to be easy to use, minimizing software piracy among students [9]. Furthermore, the incorporation of Ubuntu in operational policies can lead to meaningful social licenses from host communities in Africa [10]. The Ubuntu operating system is also widely used in virtualization technologies, contributing to its widespread adoption [11].

Additionally, the examination of both operating systems seeks to uncover the consistency in performance of WINRAR and 7-Zip across different environments. This approach aims to reveal whether these compression tools exhibit consistent behavior or if variations emerge based on the underlying operating system. As users navigate the intricacies of compression tools in their daily digital engagements, understanding how WINRAR and 7-Zip perform on these operating systems becomes imperative. This exploration not only contributes to the existing body of knowledge on compression software, but also assists users in making informed decisions based on the specificities of their chosen operating environment.

2. METHOD

In the pursuit of a comprehensive evaluation of WINRAR and 7-Zip, the following methodology outlines the systematic approach applied to configure, test, and analyze the compression software under controlled conditions:

Experimental Setup

a. Virtual Environment: Experimentation is conducted within a VirtualBox environment to ensure a controlled and reproducible testing ground.
b. Hardware Specifications:
   RAM: 4GB DDR4 2400 MHz allocated within the virtual machine.
   CPU: 4 cores assigned from a Ryzen 5 3550H processor.

Data Types and Operating Systems

a. Selection of Data Types: Various data types, including text files (.txt), image files (.png), audio files (.flac), and video files (.mp4), are chosen to represent common user scenarios.
b. Operating Systems: The analysis is conducted on Windows 11 and Ubuntu 23.10 to capture performance variations across different platforms.

Compression Process

a. Enhanced Compression Settings: The compression process is executed at the highest compression level available, prioritizing a smaller compressed output size.
b. The -m5 option in WinRAR and the -mx9 option in 7-Zip are effective choices for maximizing compression efficiency and minimizing file sizes. These options utilize advanced compression

algorithms such as LZ77 and LZSS, combined with Huffman coding, to achieve high compression ratios [12] [13]. Additionally, the use of these options can result in significantly reduced file sizes without sacrificing data integrity [14].

Performance Metrics

a. Key Performance Indicators (KPIs): Compression Ratio: Measured to assess the efficiency of the compression algorithms.
b. Resource Utilization: Monitoring RAM and CPU usage during compression and decompression processes.
c. Compression Time: Recording the time taken for each compression operation.

Data Collection

a. Rigorous Averaging: The test was carried out three times to achieve consistency, ensuring the reliability and repeatability of the results, which is crucial in experimental design and analysis [15].
b. Cross-Verification: Results are cross-verified to ensure consistency and reliability.

Cross-Platform Validation

a. Consistency Checks: To ascertain the consistency of performance, compression tools, WINRAR and 7-Zip, undergo validation across both Windows 11 and Ubuntu 23.10 operating systems.
b. Sequential Testing: Each compression scenario is sequentially executed on both Windows 11 and Ubuntu 23.10 to observe any divergences or consistencies in performance.
c. Performance Trends: The goal is to identify whether WINRAR and 7-Zip exhibit consistent behavior or if variations emerge based on the underlying operating system.
d. Cross-Verification Metrics: Compression ratios, resource utilization, and compression times are cross-verified, ensuring a comprehensive understanding of the tools' behavior in diverse operating environments.

3. RESULTS AND DISCUSSION

In this section, we present a comprehensive analysis of the performance of two prominent compression software, WINRAR and 7-Zip, across different data types and operating systems. The tables below showcase the results of the experiments, focusing on compression time, RAM utilization, CPU usage, and compression ratio for various file formats. The software's efficiency is evaluated on both Ubuntu and Windows operating systems to provide insights relevant to a diverse user base.

Table 1. Trial Results for TXT File Compression

Software	Operation System	File	Compression Time (s)	Ram Usage (mb)	CPU Usage	Compression Ratio
Winrar	Ubuntu	Average 10MB.txt + 50MB.txt + 100MB.txt	2,45	12,5	45,02%	1222268,54%
		Average of all three txt files txt	1,79	12,5	45,80%	1655571,37%
	Windows	Average 10MB.txt + 50MB.txt + 100MB.txt	10,11	12,5	42,79%	516469,32%
		Average of all three txt files txt	9,33	12,5	31,97%	670804,42%
7ZIP	Ubuntu	Average 10MB.txt + 50MB.txt + 100MB.txt	2,79	21,20555667	58,18%	1225829,99%
		Average of all three txt files txt	1,8	21,0433	48,33%	1657493,38%
	Windows	Average 10MB.txt + 50MB.txt + 100MB.txt	1,06	21,112	54,06%	516832,03%
		Average of all three txt files txt	0,96	21,17667	53,50%	671277,51%

The analysis reveals that WINRAR excels in compression time, RAM utilization, and CPU usage. While the compression ratios are comparable, WINRAR is deemed superior in terms of efficiency and speed when handling text files.

Table 2. Trial Results for PNG File Compression

Software	Operation System	File	Compression Time (s)	Ram Usage (mb)	CPU Usage	Compression Ratio
Winrar	Ubuntu	Average 1MB.png + 10MB.png + 30MB.png	7,81	12,54	56,90%	100,01%
		Average of all three txt files png	7,46	12,5	61,10%	100,03%
	Windows	Average 1MB.png + 10MB.png + 30MB.png	6,40	20,99331	51,20%	100,01%
		Average of all three txt files png	5,36	21,1233	49,17%	100,03%
7ZIP	Ubuntu	Average 1MB.png + 10MB.png + 30MB.png	11,23	12,51333333	56,03%	100,02%
		Average of all three txt files png	15,07	12,54	48,00%	100,05%
	Windows	Average 1MB.png + 10MB.png + 30MB.png	5,03	20,8833	48,67%	100,02%
		Average of all three txt files png	6,96	20,9833	44,07%	100,04%

From the analysis results, WINRAR exhibits superiority in compression time, while there is no significant difference in RAM usage, CPU utilization, and compression ratio on PNG image files.

Table 3. Trial Results for FLAC File Compression

Software	Operation System	File	Compression Time (s)	Ram Usage (mb)	CPU Usage	Compression Ratio
Winrar	Ubuntu	Average 10MB.flac + 50MB.flac + 100MB.flac	37,37	12,48666667	62,23%	100,00%
		Average of all three txt files flac	37,76	12,58	68,76%	100,00%
	Windows	Average 10MB.flac + 50MB.flac + 100MB.flac	26,94	21,11766667	50,17%	100,00%
		Average of all three txt files flac	26,18	21,14	55,83%	100,00%
7ZIP	Ubuntu	Average 10MB.flac + 50MB.flac + 100MB.flac	92,40	12,51333333	48,28%	100,04%
		Average of all three txt files flac	118,21	12,46	48,43%	100,00%
	Windows	Average 10MB.flac + 50MB.flac + 100MB.flac	45,69	21,12753333	45,90%	100,04%
		Average of all three txt files flac	61,63	21,28	38,07%	100,00%

WINRAR exhibits superiority in compression time and CPU utilization, while there is minimal difference in RAM usage and compression ratio on flac video files. The choice between the two tools may depend on the specific priorities and requirements of the user.

Table 4. Trial Results for MP4 File Compression

Software	Operation System	File	Compression Time (s)	Ram Usage (mb)	CPU Usage	Compression Ratio
Winrar	Ubuntu	Average 10MB.mp4 + 50MB.mp4 + 100MB.mp4	407,72	12,51333333	65,78%	100,91%
		Average of all three txt files mp4	441,98	12,5	68,55%	100,28%
	Windows	Average 10MB.mp4 + 50MB.mp4 + 100MB.mp4	298,91	21,34111333	55,53%	100,91%
		Average of all three txt files mp4	313,34	23,18667	54,84%	100,20%
7ZIP	Ubuntu	Average 10MB.mp4 + 50MB.mp4 + 100MB.mp4	808,77	12,5	56,24%	100,81%
		Average of all three txt files mp4	629	12,5	67,14%	100,20%
	Windows	Average 10MB.mp4 + 50MB.mp4 + 100MB.mp4	424,89	19,68334333	52,37%	100,87%
		Average of all three txt files mp4	426,55	14,83667	67,06%	100,28%

WINRAR exhibits superiority in compression time and CPU utilization, while there is minimal difference in RAM usage and compression ratio on MP4 video files. The choice between the two tools may depend on the specific priorities and requirements of the user. Based on the proprietary RAR

compression algorithm developed by Eugene Roshal, WinRAR may be considered superior to 7-Zip due to its exceptional compression capabilities. The RAR algorithm employs a combination of Huffman coding, run-length encoding, and adaptive coding to exploit data structures and patterns effectively, resulting in consistently high compression rates [16]. The emphasis on Huffman coding, a key component of the RAR algorithm, signifies a meticulous approach to compression [17]. Moreover, the use of advanced compression techniques aligns with established principles in the field, as demonstrated by Ziv and Lempel [18]. The specific optimization over time of the RAR algorithm may contribute to WinRAR's potential superiority, particularly in scenarios where maximum compression efficiency is a priority [19]. However, it is important to note that among the various lossless compression algorithms, canonical Huffman coding is particularly favorable for systems with limited computational and energy resources, due to its low complexity and high compression efficiency [20].

WINRAR demonstrates superior performance in compression time and CPU utilization, attributed to its proprietary RAR compression algorithm. However, 7-Zip, being open source and free, presents an attractive alternative, especially for users prioritizing cost savings and open-source principles. The choice between the two ultimately hinges on user preferences, with WINRAR offering advanced features for a cost, and 7-Zip providing a free and open-source solution.

4. CONCLUSION

In conclusion, the comparison between WINRAR and 7-Zip reveals distinct trade-offs. While WINRAR demonstrates superior performance in compression time and CPU utilization, leveraging its proprietary RAR algorithm with advanced compression techniques, 7-Zip stands out for being open-source and freely accessible. Users must weigh their priorities, considering factors such as cost, advanced features, and the principles of open-source software. WINRAR may be preferred for those valuing advanced capabilities and are willing to pay, whereas 7-Zip is an attractive choice for those prioritizing cost-free access and open-source values. Ultimately, the decision between the two depends on individual preferences, specific needs, and the importance attached to factors such as cost and software philosophy..

REFERENCES

[1] A. Khodadadi and M. Teimouri, "Dataset for file fragment classification of audio file formats", BMC Research Notes, vol. 12, no. 1, 2019. https://doi.org/10.1186/s13104-019-4856-1

[2] R. Cerda-Cosme and E. Méndez, "Analysis of shared research data in spanish scientific papers about covid-19: a first approach", Journal of the Association for Information Science and Technology, vol. 74, no. 4, p. 402-414, 2022. https://doi.org/10.1002/asi.24716

[3] R. Fakouri and M. Teimouri, "Dataset for file fragment classification of image file formats", BMC Research Notes, vol. 12, no. 1, 2019. https://doi.org/10.1186/s13104-019-4812-0

[4] M. Allendorf and C. Melius, "Bac-mp4 predictions of thermochemistry for gas-phase tin compounds in the sn−h−c−cl system", The Journal of Physical Chemistry A, vol. 109, no. 22, p. 4939-4949, 2005. https://doi.org/10.1021/jp040625+

[5] R. Beucher, "Configuring and installing petsc for underworld",, 2022. https://doi.org/10.59350/v12vh-c2r96

[6] O. Fernandes, N. Fernandes, F. Júnior, A. Leão, & M. Costa, "Symbolic consumption and representation of self: a study of interactions in a virtual community of ubuntu-br users", Cadernos Ebape Br, vol. 17, no. spe, p. 717-731, 2019. https://doi.org/10.1590/1679-395174446x

[7] M. Abubakre, I. Faik, & M. Mkansi, "Digital entrepreneurship and indigenous value systems: an ubuntu perspective", Information Systems Journal, vol. 31, no. 6, p. 838-862, 2021. https://doi.org/10.1111/isj.12343
[8] S. Nxumalo and D. Mncube, "Using indigenous games and knowledge to decolonise the school curriculum: ubuntu perspectives", Perspectives in Education, vol. 36, no. 2, p. 103-118, 2018. https://doi.org/10.18820/2519593x/pie.v36i2.9
[9] B. Achmad, I. Nur, & K. Siti, "Development of linux ubuntu open source distribution based open source distribution system to minimize students 'software study",, 2019. https://doi.org/10.2991/icesre-18.2019.63
[10] A. Ogunranti, "Localizing the ungps – an afrocentric approach to interpreting pillar ii", Business and Human Rights Journal, vol. 8, no. 1, p. 66-84, 2022. https://doi.org/10.1017/bhj.2022.35
[11] A. Younge, R. Henschel, J. Brown, G. Laszewski, J. Qiu, & G. Fox, "Analysis of virtualization technologies for high performance computing environments",, 2011. https://doi.org/10.1109/cloud.2011.29
[12] T. Bell and S. Zhang, "Multiwavelength depth encoding method for 3d range geometry compression", Applied Optics, vol. 54, no. 36, p. 10684, 2015. https://doi.org/10.1364/ao.54.010684
[13] J. Konstantinides and I. Andreadis, "Performance analysis for canonical huffman coding with fixed window size", Electronics Letters, vol. 52, no. 7, p. 525-527, 2016. https://doi.org/10.1049/el.2015.3299
[14] P. Ou and S. Zhang, "Natural method for three-dimensional range data compression", Applied Optics, vol. 52, no. 9, p. 1857, 2013. https://doi.org/10.1364/ao.52.001857Rationale for Enhanced Compression: This approach aims to push the boundaries of each compression tool's capabilities, providing insights into their performance when tasked with achieving the utmost compression ratios.
[15] M. Rahman, A. Tutul, S. Abdullah, & M. Bayzid, "Chapao: likelihood and hierarchical reference-based representation of biomolecular sequences and applications to compressing multiple sequence alignments", Plos One, vol. 17, no. 4, p. e0265360, 2022. https://doi.org/10.1371/journal.pone.0265360
[16] V. Nguyen, H. Nguyen, H. Duong, & V. Snasel, "n-gram-based text compression", Computational Intelligence and Neuroscience, vol. 2016, p. 1-11, 2016. https://doi.org/10.1155/2016/9483646
[17] B. Wijaya, S. Siboro, M. Brutu, & Y. Lase, "Application of huffman algorithm and unary codes for text file compression", Sinkron, vol. 7, no. 3, p. 1000-1007, 2022. https://doi.org/10.33395/sinkron.v7i3.11567
[18] A. Tanjung and S. Nasution, "Comparison analysis with huffman algorithm and goldbach codes algorithm in file compression text using the method exponential comparison", The Ijics (International Journal of Informatics and Computer Science), vol. 4, no. 1, p. 29, 2020. https://doi.org/10.30865/ijics.v4i1.1387
[19] A. Fariña, G. Navarro, & J. Paramá, "Boosting text compression with word-based statistical encoding", The Computer Journal, vol. 55, no. 1, p. 111-131, 2011. https://doi.org/10.1093/comjnl/bxr096
[20] G. Navarro and J. Tarhio, "Lzgrep: a boyer-moore string matching tool for ziv-lempel compressed text", Software Practice and Experience, vol. 35, no. 12, p. 1107-1130, 2005. https://doi.org/10.1002/spe.663