International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 12– Issue 02, June 2023

Benchmarking Performances of Raspberry pi Microcomputer as Video Wall Devices

N. B. Paul
Department of Computer Engineering,
Kaduna Polytechnic, Kaduna.
Kaduna State Nigeria
Email: paulnaanman[AT] kadunapolytechnic.edu.ng

O. U. Okereke
Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University, Bauchi.
Bauchi State Nigeria
Email: ouokereke [AT] atbu.edu.ng

E. E. Omizegba
Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University, Bauchi.
Bauchi State Nigeria
Email: eeomizegba [AT] atbu.edu.ng

E. C. Anene
Department of Electrical and Electronics Engineering,
Abubakar Tafawa Balewa University, Bauchi.
Bauchi State Nigeria
Email: eanene [AT] atbu.edu.ng

Abstract— **Video wall development is affected by cost, power consumption, processing capabilities, algorithm, and video used. Literature has shown that using microcomputers reduces power consumption and costs, but performance remains a bottleneck. Benchmarking the performances of Raspberry pi (R-pi) devices with real-world loads will help understand R-pi video wall development, suitability, and utilization. The approach used is based on parallel video streaming using user datagram protocols (UDP) and broadcast addressing, while image splitting is done on clients. Nigel's performance monitoring (NMON) tool was used with videos of varying frames (15fps, 20fps, 24fps, 25fps, 30fps, 50fps, and 60fps) and resolutions of 144p, 240p, 360p, 480p, 720p, and 1080p to benchmark performances. Results revealed a maximum of 9.78%, 17.16%, 58.45 kB/s, and 1.13 kB/s for central processing unit (CPU), memory, network, and disk usage, respectively. Results also reveal that the R-pi as a video wall device, with the proposed approach, has the processing capability to enhance video wall development. These results reveal that for best performances, R-pi video walls are more suitable with videos of higher resolutions, such as 480p, 720p, and 1080p, and at lower frame rates, such as 24fps, 25fps, and 30fps.**

Keywords- Video wall, Raspberry pi, Microcomputer, Broadcast, Benchmark, Server.

I. INTRODUCTION

The first video wall, known as planar-tiled projected displays was built in the 1990s by Princeton University at Argonne National Laboratory in the United States [1]. Since then video walls are designed as distributed systems with group of monitors configured and synchronised to display content in a single larger screen fashion [2–4]. However, the major disadvantage of distributed computing in video wall is that communication to the display node must be from the sever node and it is necessary that such server must not be a bottle neck [5].

Today, video wall designs do not depend on specific hardware specifications and can be extended to more diverse environments with aim of cost-effectiveness, physical space, and network speed management as well as reduction in power consumption. To this end, microcomputers are common for video wall development [2-4, 6-10]. However, the use of microcomputers to perform high-end operations such as server is faced with the challenges of computational performance and high-speed connections required [11], [12].

In a fully Raspberry pi (R-pi) based video wall system (Figure 1), the display node (Clients) and the controller node (Server) are made up of multiple R-pi microcomputers connected to Liquid Crystal Display (LCD) monitors, with the server responsible for processing and streaming videos via a network switch to the clients. The server creates an image object designated with video frame synchronization identification (ID) and timestamp converted into packets and transmit to intend internet protocol (IP) broadcast addresses. All clients with fixed IP address receives the video frame packet from the cast address and display the contents as instructed by the server [10]. The multiple functions performed by the server requires an optimum utilization of computational resource to ensure system stability and durability.

Video wall image rending are based on typical server-client multimedia models, include the unicast in Figure 2(a), and broadcast in Figure 2(b) and multicast in Figure 2(c) addressing [13]. It has been established that when displaying one image across several monitors, the use of one server with IP multicast addressing will reduce network load and improve video frame synchronization [10]. However, the use of broadcast addressing increases speed and reduces bandwidth usage as compare to multicast and the unicast addressing [14]. In addition, the task on the server is reduced as it sends only one copy of each media content to the broadcast network, thereby keeping bandwidth requirements constant irrespective of the number of clients.

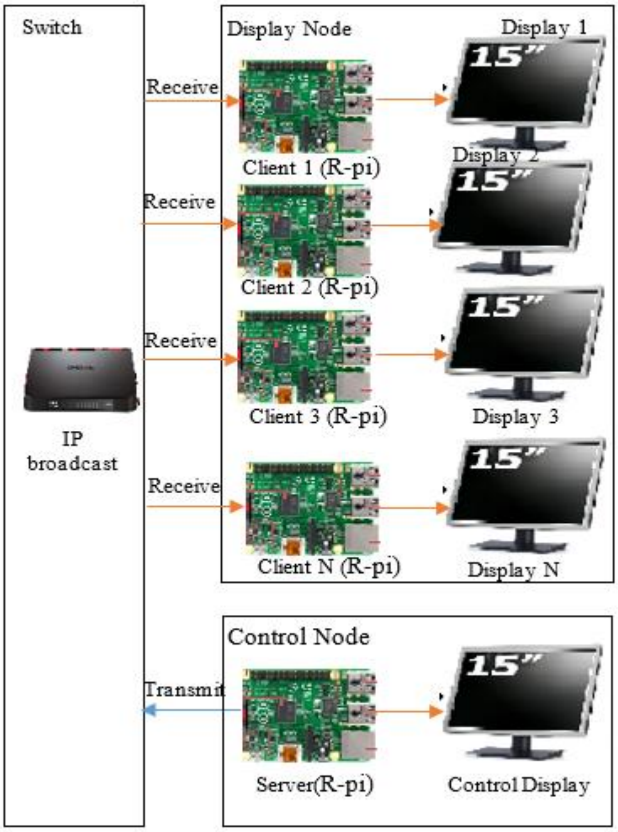
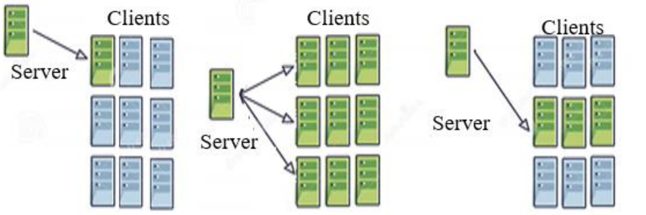


Figure 1: R-pi Microcomputer Based Video Wall



(a) Unicast (b) Broadcast (c) Multicast

Figure 2. Server-Client Multimedia Streaming Models

Evaluating computational performance, networking protocols, and devices under reference conditions is referred to as benchmarking. Application throughput and node power consumption are identified as the major benchmarking metrics [15] and are referred to as primary performance metrics which can be collected directly from the system under test (SUT). Benchmarking helps in setting limits, conditions, and type of input/output for a SUT. It sets reference parameters for performance improvement and enhancing innovation; Identifies best technological approach and components with superior performance; It exposes the gap between the expected performance and the actual state of a system, hence setting room for improvement; provides the direction for technological or process changes; and helps in measuring the efficiency of their operational metrics [16], [17]. A fundamental practice to determine performances, is to use a set of benchmarks [18] with a benchmarking model [19].

In video walls, algorithms such as offset and overlay described in [3], [4] are identified as major image processing algorithms used in dealing with visualisation challenges as associated with image splitting. Unfortunately, human perceptions of video quality associated with these algorithms have yielded inconsistency and inconclusiveness [20–24]. Therefore, there is a need to investigate the computational impact associated with these algorithms, particularly when R-pi microcomputer is used as video wall server.

In this work, we propose taking advantage of the powerful central processor unit (CPU) in R-Pi 3B+ microcomputers to provide a good platform for a cheaper infrastructure with significantly less CPU usage, less amount of streaming bandwidth and memory usage. Low computational algorithms with relatively good-speed connection protocol are used to provide a new approach to drive high-resolution tiled displays directly with a microcomputers as server.

The work uses, less computational streaming protocols, user datagram protocols (UDP) with broadcast addressing for video streaming. The impact of real world workloads on the computational performance of the R-pi as the server has been verified practically with different visualisation scenarios, at interactive frame rates and resolutions. Evaluation was achieved through benchmarking with Nigel's performance monitoring (NMON) tool. Parameters considered for benchmarking include; the CPU, memory, network and disk usages of a R-Pi 3B+ as 3 by 3 tiled LCD based video wall server.

II. RASPBERRY PI MICROCOMPUTER AS SERVERS

The R-pi microcomputer is a multi- interface, low-cost, low power microcomputer, designed to perform the functions of desktop computers [25]. The R-pi model 3B+ uses a 64 bit, 1.4GHz processor with static random access memory (SRAM) of 1 GB, Ethernet speed of up to 300 Mbps and consumes about 15 Watts, these makes it suitable for use as a server.

Literature such as [26] - [29], used a large collection of R-pi to build clusters without study on the performance of any node or the whole cluster under realistic workload. Using R-pi server with realistic loads in a parking lot design [30], demonstrated that R-pi can be used as a server for internet of things (IoT) network. The author of [31], used a single node R- pi to evaluate the virtualization impact on CPU, memory I/O, disk I/O, and network I/O and concluded that the overhead is negligible, relative to native execution. Unfortunately, the experiments conducted predominantly centered on the system benchmarking and did not reflect real world workload. In [32], the author, studied the feasibility of using R-pi based cluster for big data applications with more realistic workloads but used Apache Hadoop framework, while, TeraSort was used to evaluate the cluster performance and energy consumption.

The use of R-Pi as client-server; in [33] for webserver showed processor performance averaged of 23% and maximum of 27% ,with highest speed of 91 Mbit/s when served with 10000 concurrent HTTP-requests. Similarly [34] used R-Pi for light intensity monitoring but could not evaluate.

In [35] R-pi was used for big data with cluster of 12 nodes, httperf benchmarking tool, an apache spark frame work and Hadoop Distributed File System (HDFS). CPU usage, network throughput and average response time of the R-pi were measured. Results revealed response time varies from 2809 req/s to 98 req/s while the CPU usage also varied from 67.2%, to 22.3% for 1 kB and 100 kB workload respectively.

In [36 - 38] R-pi has been used to develop video walls, intending to reduce cost, energy and space. These designs used R-pi as clients rather than servers as such could not benchmark or evaluate computational performances of the R-pi microcomputer as a video wall server.

In [36] a subjective test of R-pi as server and clients for video wall using videos with .3GP, .MP4, and .MKV formats revealed, lower video file format produce worse video quality (resolution). Furthermore the results recommended .MP4 or .MKV formats as better in image quality and when video has moving objects.

Literature has shown that in Linux based systems such as in R-pi, the kernel provides data in the/proc/stat file and can be displayed using utilities such as Atop, Htop, top. Iostat, nmon and Dstat commands. Benchmarking tools such as 'top command', 'SPECviewperf 11' and Wireshark network analyser have been used by [39], [40] to evaluate the performances of Linux based PC system as servers. Despite using a PC, [39] showed that streaming a video of 720p results in CPU usage of 24 % and 27 % as the number of client's increases from one to two. Other benchmarking carried out as in [40], [41] did not mention the tool used, however, [42] see benchmarking tools are programs that run directly on the CPU hardware which are better than processor simulators. In addition, the use of kernel-based benchmarking tools such as NMON, for benchmarking has remained unexplored.

III. METHODOLOGY

The 3×3 video wall testbed developed in our previous study [4] with R-pi microcomputers (model 3B+) was setup in an air-conditioned Laboratory with thermometer, uninterrupted power supply and Voltmeter using the model in Figure 3.

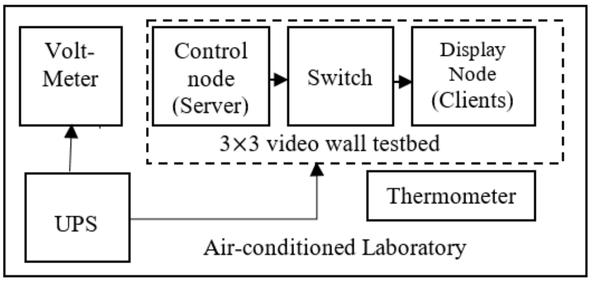


Figure 3: Experimental Testbed Model

In this work, the video stored in the server are encoded and transmitted, frame by frame to all the clients within the IP address range. Each client receives the image frame, spilts and displays only the segment that matches its IP address. Our proposed benchmarking model as in Figure 4 is a closed-loop controlled system with feedback at three different stages.

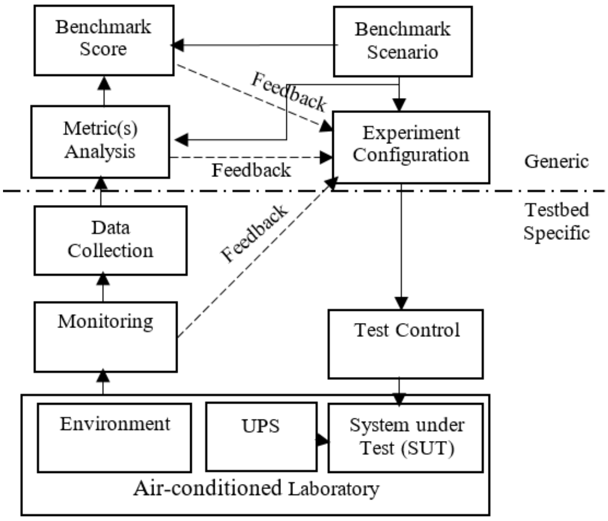


Figure 4: Performance Evaluation Model

The lower portion of the chart shows testbed specific operations mainly conducted using the experimental test bed setup in Figure 5.

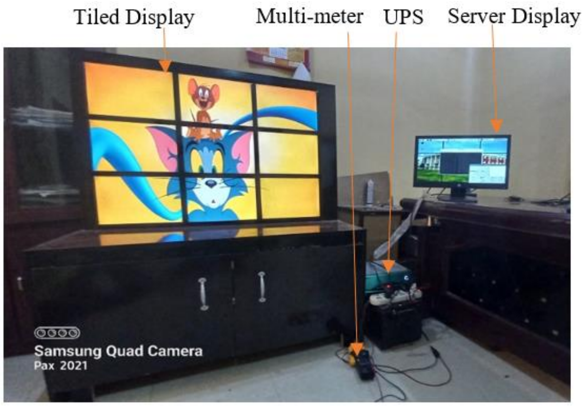


Figure 5: Experimental Testbed Setup

The three feedback conditions include; 1. Monitoring the environmental (temperature) streaming and electrical conditions. 2. Metric analysis and 3. Benchmarking scores. In comparison with benchmarking scenario, negative or out of range parameters from any of these units in our model will result in reconfiguration or restarting of the experiment. In this way, all results will be free from possible external influence. Details of the methodology are as follows.

A. Experimental Configuration

The primary performance of this server was evaluated under 25°C temperature (monitored with a thermometer), with stable power supply of 220 Vac provided from a UPS and monitored with a digital multi-meter. A display monitor was also connected to the server to visualise the Graphical user interface (GUI) control panel. The primary performance parameters data are saved on the server memory as .nmon file. The experimental test bed setup is as in Figure 5.

B. Test Control

To capture vast range of video resolution and frame rate as in [4]. 4 minutes 59 seconds, Mp 4 video [43] was downloaded at varying frame rates of 15fps, 20fps, 24fps,25fps, 30fps, 50fps and 60fps and resolutions of 144p, 240p, 360p, 480p, 720p and 1080p using source site [44] and stored on the SUT as test video.

C. Laboratory Condition

Uninterrupted Power Supply (UPS) and air-condition set at 25°C were provided in the room with voltmeter and thermometer monitoring possible variations.

D. Monitoring

During experiments, monitoring functions are checked at intervals of 1 minute to ensure stability. The maximum operating temperature of R-Pi’s GPU or CPU is 70ºC and 85ºC respectively. Therefore, to reduce the effect of overheating on the system performance, the temperature mathematically expressed as in Equation 2 was monitored using **“`vcgencmd measure_temp`”** to maintain half the average temperature range of 35-40°C. Operating voltage of the CPU and GPU were also monitored as 1.2 V using **“`vcgencmd measure_volts`”** and where changes are observed, experiment is restated after re-configuration and stabilization of environmental and electrical conditions see Figure. 6.

$$CPU_T = A_T + L_T \tag{1}$$

Where; A_T = Ambient temperature, L_T = load induced temp. rise and CPU_T = CPU operating temperature.

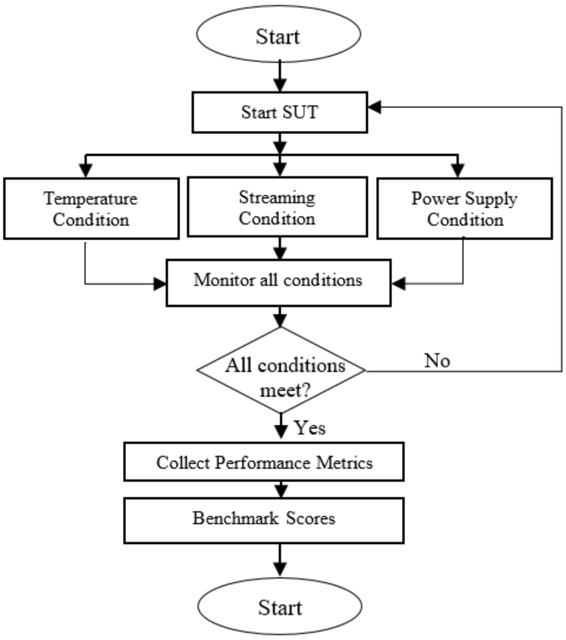


Figure 6: Performance Evaluation Model

E. Data Collection

During the experiment, raw data was gathered using NMON and benchmarking performance information such as; CPU usage, memory usage, network throughput, disk usage, and file system resources were obtained. Data were acquired in a .nmon file format with the following NMON command and processed using the NMON analyzer [45];

“nmon –F **filename**.nmon –s f_s –c D_s"

Where; nmon is the instruction to carry out NMON analysis,
–F means write file,
filename.nmon is the name to be assigned to the file,
–s instruction to carry out sampling at certain interval for the performance metrics,
f_s is the Sampling intervals (seconds),
–c instruction to collect and record certain number of samples of performance metrics,
D_s Means number of sample to collect and record

To ensure uniformity of evaluation, the settings for this command Sampling Interval (f_s) was 5 seconds while the number of runs was set at 60.

F. Performance Metrics

Performance Metrics considered in this work include CPU usage, memory usage, network usage and disk usage. These metrics are used to score the performance of the R-pi microcomputer as a server.

G. Benchmark Score

The benchmark score for each of the performances metrics were drawn from the percentage differences in performances.

The percent difference was used to establish a comparison in different groups, different categories or levels and is the ratio of the absolute difference between two values to their average multiplied by 100. [46], mathematically expressed as in Equation 2;

$$\text{Percentage difference (PD\%)} = \frac{|\,f_1 - f_2\,|}{\left(\frac{f_1 + f_2}{2}\right)} \times 100\ \% \qquad (2)$$

Where, $PD(\%)$ is Percentage differences

f_1 Is the measured value from experiment 1

f_2 Is the measured value from experiment 2

To benchmark, the various metrics are converted into integer values between 0 and 100% for comparison using percentage difference; the smaller the differences, the better the performance score.

IV. RESULT AND DISCUSION

Average metric values (Avg) from performance data collected during experiments for videos with the two algorithms were recorded and plotted as shown in Figures 7 to 14. Videos with varying resolutions as well as varying frame rate were used. Also analyzing processed data percentage changes within the range of measurement reveal performances of the algorithms with varying resolution and frame rate changes in Tables 1 and 2.

A. Effect of Video Resolution

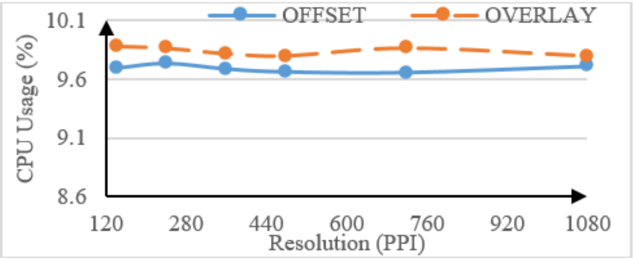


Figure 7: CPU Usage with Resolution Changes

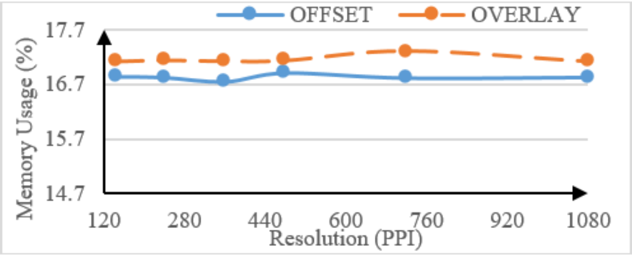


Figure 8: Memory Usage with Resolution Changes

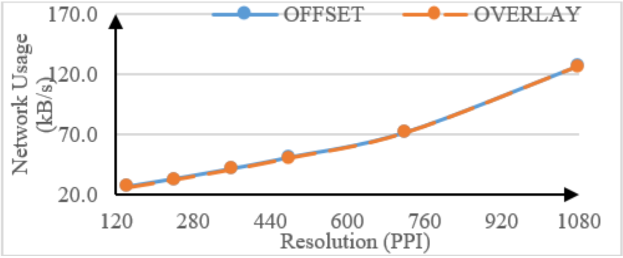


Figure 9: Network Usage with Resolution Change

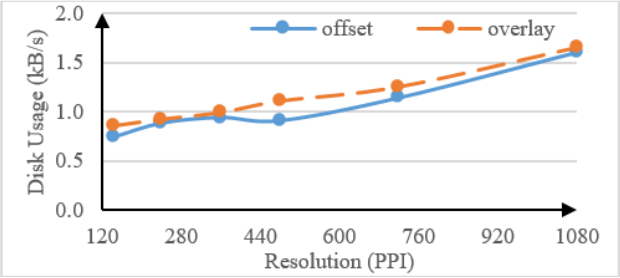


Figure 10: Disk Usage with Resolution Changes

Table 11: PD% with Resolutions

Resolution	**CPU %**	**Memory %**	**Network %**	**Disk %**
144	1.84	1.67	0.79	14.03
240	0.96	2.47	0.17	9.18
360	1.33	2.25	0.35	5.30
480	1.42	1.37	0.28	19.55
720	2.12	2.88	0.13	9.44
1080	**0.91**	1.75	0.46	3.39

B. Effect of Video Frame Rate

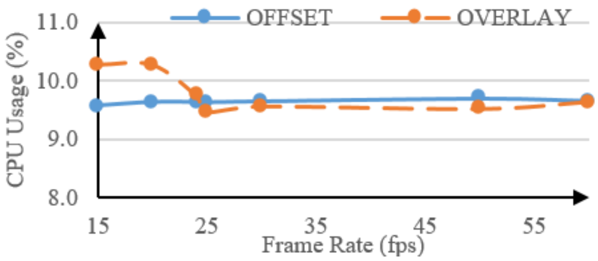


Figure 11: CPU Usage with Frame Rate Changes

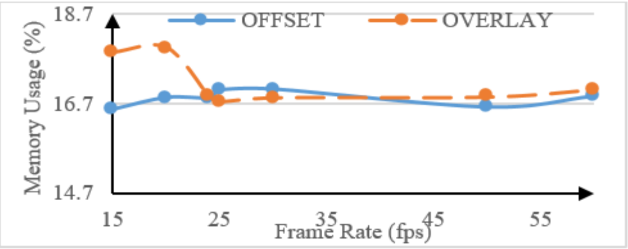


Figure 12: Memory Usage with Frame Rate Changes

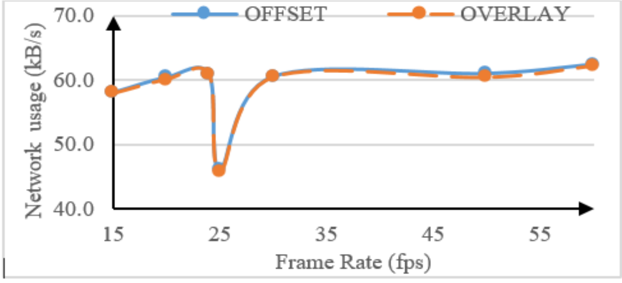


Figure 13: Network Usage with Frame Rate Changes

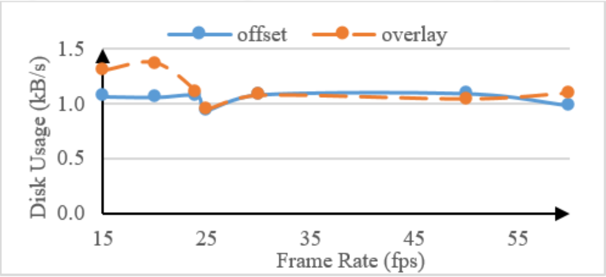


Figure 14: Disk Usage with Frame Rate Changes

Table 22: PD% with Frame Rates

Frame Rate	CPU %	Memory %	Network %	Disk %
15	7.02	7.33	0.16	20.26
20	6.14	7.11	0.37	30.89
24	0.98	0.19	0.09	2.96
25	1.71	1.48	0.88	1.41
30	0.85	1.08	0.08	0.52
50	1.79	1.31	0.89	4.59
60	0.13	0.84	0.28	11.20

Figure 11 shows that the overlay uses more CPU at lower fps with 10.3%, 10.3%, 9.8% at 15 fps, 20 fps, and 24 fps respectively, while at the offset remained at 9.6% and 9.5%. Interestedly, both algorithms become stable at 25 fps, 30 fps, 50 fps, and 60 fps with the 24fps and 25fps having more closed values of 9.5% and 9.6%. Similarly, in Figure 12 that the overlay uses more memory of 17.9% and 18% compare to 16.6% and 16.8% for offset at 15 fps, 20 fps respectively, while the offset remained relatively stable between 16.8 and 17%.

Interesting Figure 13 shows that additional bandwidth will be required for offset to effectively stream videos will be up to; 57.9 kB/sec for 15fps to 62.2 kB/sec for 60fps respectively. The steady increased in network required dropped sharply to 45.8 kB/sec at 25fps as a result of GPU/CPU capacity.

Figure 14 also shows that the overlay uses more disk than the offset but showed a drop in disk usage from 1.1 kB/sec at 24fps to 1.0 kB/sec at 25fps.

Table 2, shows 24fps, 25fps and 30fps video produces less difference of 0.08% to 2.96% in performances for both algorithms in terms of CPU, memory, network, disk usage, demonstrated good performance. This shows that moderate frame rate videos such as 24fps, 25fps and 30fps video are most suitable for application in R-pi video wall.

V. CONCLUSION

This work affirm that though there exist PD% of 0.36% to 10.15% between the two algorithms. Disk usage introduces differences of 10.15% and 8.81% while network differences are minimal of 0.36 and 0.39%. It also reveal R-pi microcomputers as server in video wall with image splitting algorithms implemented on clients reduces CPU usages to 9.78% memory usage to 17.16% , network usages to 58.45 kb/s and disk space to 1.13 kb/s. These work has reveal that R-pi video walls are more suitable with videos such as 480p, 720p and 1080p at 24fps, 25fps and 30fps.

REFERENCES

[1] M. Saleem, H.E. Valle, S, Brown, V. I. Winters and A. Mahmood, "The hiperwall tiled-display wall system for big-data research", *J Big Data*, vol. 5, no. 41, 2018, doi.org/10.1186/s40537-018-0150-7.

[2] D. Marfil, F. Boronat, J. Gonzalez, and A. Vidal, "Design and assessment of a scalable and customizable low-cost tiled display system", in *IEEE Latin America Transactions*. vol. 19, no. 5, 2021, pp 708 – 716, doi: 10.1109/TLA.2021.9448284.

[3] N. B. Paul, O. U Okereke, E. E. Omizegba, and E. C. Anene, "Comparative study of overlay and offset algorithms in 3-by-3 video wall using objective metrics", *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, vol. 8 issue 7, pp. 14389 -14396, July. 2021.

[4] N. B. Paul, E. E. Omizegba, O. U Okereke, and E. C. Anene, "Implementation of bezel compensation algorithms on Raspberry pi microcomputer and evaluation of video impairments using video quality metric", *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, vol. 8 issue 9, pp.14598-14608, Sep. 2021.

[5] M. Han, I Wald, W. Usher, N. Morrical, A. Knoll and C.R. Johnson, "A virtual frame buffer abstraction for parallel rendering of large tiled display walls". *2020 IEEE Visualization Conference (VIS).* DOI 10.1109/VIS47514.2020.00009 Pp 11-15. 2020.

[6] M. R. Jakobsen and H. Kasper, "Up close and personal: collaborative work on a high-resolution multitouch wall display", *in ACM Trans. Comput.-Hum. Interact. 21*, 2, Article 11, 2014, pp. 34, doi: 10.1145/2576099.

[7] J. M. E. M. Van Der Werf, et al., "Facilitating collaborative decision making with the software architecture video wall", *in Proc-IEEE, ICSAW 2017: Side Track Proceedings*, 2017, pp.137–140, doi: 10.1109/ICSAW.2017.27.

[8] L. R. Rivera, A. M. Viveros and S. C. Vergara, "User interface features for tiled display environments", *in proc. 10th Computr Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, 2013*, pp. 296-301, doi: 10.1109/ICEEE.2013.6676036.

[9] S. Noda, et al., "Implementation of high presence video communication system for multiple users using tiled display environment", *in Proc-IEEE WAINA, 2015*, pp. 494–499, doi:10.1109/WAINA.2015.81.

[10] Y. Sun-Jin, P. Jae-Pyo, and Y. Seung-Min, "Design and implementation of IP video wall system for large-scale video monitoring in smart city environments. *Journal of the Korea Academia-Industrial cooperation Society Vol.20, No.9 pp.7-13, 2019.* doi.org/10.5762/KAIS.2019.20.9.7.

[11] I. S. B. Md Isa, C. Ja Yeong, N. L. Azyze and b. M. S.Azyze, "Real-time traffic sign detection and recognition using Raspberry Pi", *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 12, No. 1, February 2022, pp. 331~338 ISSN: 2088-8708, DOI: 10.11591/ijece.v12i1.pp331-338

[12] T. Biedert, P. Messmer, T. Fogal and C. Garth. “Hardware-accelerated multi-tile streaming for realtime remote risualization”, *Eurographics Symposium on Parallel Graphics and Visualization*, H. Childs, F. Cucchietti (Editors), 2018, pp 33-43, doi: 10.2312/pgv.20181093

[13] S. Kaur, K. Singh and Y. Singh, “A Comparative analysis of Unicast, Multicast, Broadcast and Anycast addressing schemes routing in MANETs”, *International Journal of Computer Applications (0975 – 8887)* Volume 133 – No.9, January 2016 pp. 16-22.

[14] C. Shen, , C. Liu, and R. Rouil, “A comprehensive analysis on Multicast and Unicast performance and selection”, IEEE Global Communications Conference 2021 (Globecom), Madrid, ES, [online], https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=931401 (Accessed March 17, 2022)

[15] S. Bouckaert, J. V. V. Gerwen, I. Moerman, S. C. Phillips, and J. Wilander, “Benchmarking computers and computer networks”. *EU FIRE White Paper*. *Avaliable online* http://www.sop.inria.fr/members/Thierry.Turletti/WP11.pdf

[16] C. Priya, “What is Benchmarking? Definition, Types, Process, Advantages, Disadvantages, Scope - The Investors Book”, 2018. The investors book.com/benchmarking.html

[17] Ryus et al., “A Methodology for performance measurement and peer comparison in the public transportation Industry”, TRB, TCRP REPORT 141. www.researchgate.net/publication/241809119_A_Methodology_for_Performance_Measurement_and_Peer_Comparison_in_the_Public_Transportation_Industry

[18] United Nation, Economic and Social Council. “A Benchmarking literature review – definitions concepts and methodologies”. 29 June 2020. https://unece.org/sites/default/files/2021-09/ECE-TRANS-WP5-2020-06e.pdf

[19] B. Robert, L. Y. Chen, and E. Smirni, “Data centers in the wild: A large performance study”, IBM, Zurich, Switzerland, Rep., Z1204–002, 2012.

[20] D. S. Tan, and M. Czerwinski, “Effects of visual separation and physical discontinuities when distributing information across multiple displays”, *Human-Computer Interaction -- INTERACT'03 M.* Rauterberg et al. (Eds.) Published by IOS Press, IFIP, 2003, pp. 252-255

[21] C. S. Campbell, and P. P. Maglio, “Segmentation of display space interferes with multitasking”. Conference: Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th Sep. 2003, Zurich, Switzerland Pp. 575–582.

[22] X.-D. Yang, E. Mak, D. McCallum, P. Irani, X. Cao, and Izadi, S. Lensmouse, “Augmenting the mouse with an interactive touch display”, *in proc. CHI 2010*, pp. 2431–2440.

[23] A. Jimenez and D. Rincon, “Videowall disforme sobre redes IP”, *UPCommons*, pp. 1–77. 2016.

[24] A. M. McNamara, F. Parke, and M. Sanford, “Evaluating performance in tiled displays: navigation and wayfinding”, *proc. VRCAI 2011*, pp. 483–490.

[25] S. J. Johnston and S. J. Cox, “The Raspberry pi: a technology disrupter, and the enabler of dreams,” *Electronics*, vol. 6, no. 3, 2017, doi: 10.3390/electronics6030051.

[26] P Abrahamsson, S. Helmer, N. Phaphoom, L Nicolodi, N. Preda, L. Miori, M. Angriman, J. Rikkila, X. Wang, K. Hamily, et al. “Affordable and energy-efficient cloud computing clusters: The Bolzano Raspberry Pi cloud cluster experiment”. *In Proceedings of the 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom),* Bristol, UK, 2–5 December 2013; Volume 2, pp. 170–175

[27] A. Anwar, K.R Krish and A.R Butt,. “On the use of Microservers in supporting eadoop applications”. *In Proceedings of the 2014 IEEE International Conference on Cluster Computing (CLUSTER),* Madrid, Spain, 22–26 September 2014; pp. 66–74.

[28] M.F Cloutier, C. Paradis, and V.M. Weaver, “Design and analysis of a 32-bit embedded high-performance cluster optimized for energy and performance.” *In Proceedings of the 2014 Hardware- Software Co-Design for High Performance Computing (Co-HPC),* New Orleans, LA, USA, 17 November 2014; pp. 1–8.

[29] S.J. Cox, J.T. Cox, R.P. Boardman, S.J Johnston, M. Scott, and N.S. O’brien, Iridis-pi: “A low-cost, compact demonstration cluster.” *Clust. Comput.* **2014**, 17, 349–358.

[30] S. M. Florence, M. Uma, C. Fancy, and G. Saranya, “A study of remotely booking slot for vehicle using Internet of Things”, *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 10, No. 5, October 2020, pp. 5392~5399 ISSN: 2088-8708, DOI: 10.11591/ijece.v10i5.pp5392-5399

[31] R. Morabito, A performance evaluation of container technologies on internet of things devices. 2016, arXiv:1603.02955. arXiv.org e-Print archive. Available online: http://arxiv.org/abs/1603.02955 (accessed on 20 April 2016).

[32] N. Schot, “Feasibility of Raspberry Pi 2 based micro data Centers in big data applications”. *In Proceedings of the 23th University of Twente Student Conference on IT, Enschede*, The Netherlands, 22 June 2015.

[33] B. Madoš, J. Hurtuk, E. Chovancová, P. Feciák, anf D.Bajkó, “Downsizing of web Server design using Raspberry Pi 3 single board computer platform”, *2017 IEEE 14th International Scientific Conference on Informatics*, pp. 238-242.

[34] X. Sun, N. Ansari, and R. Wang, “Optimizing resource utilization of a data center,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, Fourth Quarter 2016.doi: 10.1109/COMST.2016.2558203.

[35] W. Hajji and F. Po Tso, “Understanding the performance of low power Raspberry Pi cloud for big data”, *Electronics 2016*, 5, 29; pp. 1-14, doi:10.3390/electronics5020029.

[36] N. Chandra, and A. P. Sujana, “Design and implementation video wall based on Raspberry pi for e-madding”, *Komputika: Jurnal Sistem Komputer (Computing: Journal of Computer Systems)* vol. 7, no. 1, April 2018, hlm. 39 - 46 doi: 10.34010/komputika.v7i1.1507

[37] “PiWall.” http://www.piwall.co.uk (último acceso: abril 2020).

[38] A. A. Batool, F. N. Alsalman, Z.H. AlShakhs, F. A. Hafiz and M. Azher, “Dynamic video wall tile creation using Raspberry pi3”. *In Proceedings IEEE Computer Society, IEEE 13th International Symposium on Autonomous Decentralized Systems,* 2017, *pp. 268- 271,* doi: 10.1109/ISADS.2017.45.

[39] G.G. Ramón and C.V. Rodríguez, *“Implementation of a low cost video wall using R-pi devices”,* Master Thesis, M.Sc in Telecom. Eng’g and Mgt, Universitat Politechnica De Calalunya. 2014.

[40] C. Papadopoulos, K. Petkov, A. E. Kaufman, and K. Mueller, “The Reality deck--an immersive GigGUIxel display”. *IEEE computer graphics and applications,* vol. 35, no.1, pp 33-45. 2014.

[41] R. Bundulis and G. Arnicans, “Concept of virtual machine based high resolution display wall”, *Information, Electronic and Electrical Engineering*, pp. 1-6. 2014.

[42] D. Varad, M., Nishchay and K. K. Shrirang, “Techniques for benchmarking of CPU Micro-Architecture for performance evaluation”. https://www.google.com/search?client=firefox-b-d&q=Techniques+for+Benchmarking+of+CPU+Micro-Architecture+for+Performance+Evaluation

[43] YouTube.Tom and Jerry | No Way Out | Boomerang Africa - Sunday Morning Shake Up. https://www.youtube.com/watch?v=bShpwwJR3Tg. Downloaded 29th May 2021

[44] Tube offline https://www.tubeoffline.com/download-youtube-videos.php

[45] N. Griffiths, “NMON analyser — A free tool to produce AIX performance reports”. https://developer.ibm.com/articles/au-nmon_analyser/. Accessed Aug. 2021.

[46] T.J. Cole, and D. G. Altman, “Statistics notes: what is a percentage difference? *Research Methods & Reporting”.* https://www.bmj.com/content/bmj/358/bmj.j3663.full.pdf. Accessed 15 Sep. 2021.