F-059

Optimizing Cloud Gaming Colocation with Virtual Machine

Soe Myat Min †　　Hiroaki Fukuda †　　Takumi Miyoshi †

Abstract—Cloud gaming is beneficial for multiple players to access games via remote virtual machines, offering scalability and convenience. However, prior research has largely focused on colocating and testing local environments without addressing real-world performance under network saturation.

This paper proposes a resource-efficient game colocation strategy in virtualized cloud gaming environments. We leverage machine learning to predict resource usage and optimize the number of concurrently hosted games while maintaining consistent performance (e.g., 60FPS). Our evaluation demonstrates over 90% accuracy in resource prediction, ensuring stable, high performance gaming experiences suitable for practical deployment.

Index Terms—Cloud gaming, game colocation, machine learning, GPU virtualization, cost optimization, Parsec

1. INTRODUCTION

Cloud gaming provides gaming resources over the Internet, eliminating the need for high-end hardware. [1] It allows players to enjoy high-quality games without downloading or updating the games. Instead, in cloud gaming, data centers handle computing power, reducing players' device requirements and enabling players to log in and play immediately without installations. Cloud gaming operates on a "pay-as-you-go" model, making it ideal for occasional players. [2] The basic setup involves a client device connected to a cloud gaming service, where inputs are processed in the cloud, and the game is streamed back as video, creating a seamless gaming experience as shown in Fig. 1 [3]. In traditional gaming, Fig. 1-A, the client device handles rendering, with game state information exchanged directly between the client and game server. In cloud gaming, Fig. 1-B, however, the rendering is done on server instances in the cloud, which stream video back to the client.

There are several key issues to focus on in a cloud gaming environment. First, server positioning must be reliable to ensure players can connect without interruptions. Second, cloud gaming services must prioritize minimizing latency, as delays can significantly degrade the Quality of Experience (QoE), making gameplay frustrating for users [4]. Finally, the cost of service is crucial, as players generally prefer high-performance services at an affordable price.

† 芝浦工業大学　Shibaura Institute of Technology

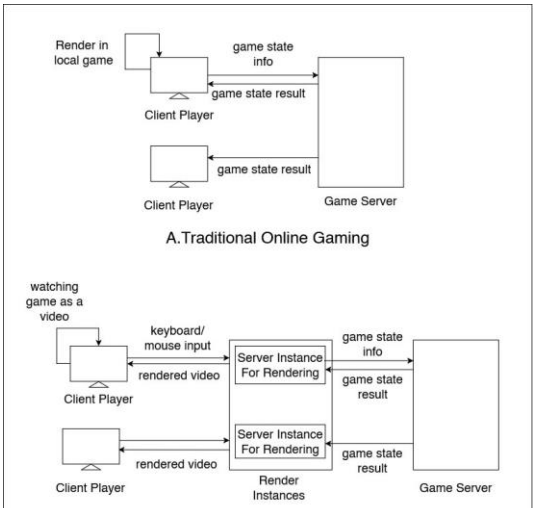

Fig. 1. Traditional and Cloud Gaming

When it comes to server positioning, cloud gaming platforms typically use either dedicated instances (e.g., physical or virtual machines) for each player or shared resources among multiple players. Colocating multiple games on a single server enhances resource utilization by sharing resources among games with varied demands. However, this introduces a tradeoff: increased competition for limited resources can potentially degrade game performance, affecting player satisfaction. The primary challenge here is to balance optimal resource usage with consistent game performance.

In game colocation, two main approaches exist: using virtual machines (VMs) and not using them. Using VMs for colocation is more realistic and closely resembles real-world applications, as it enables finer control over resource allocation. Regardless of the method, resource sharing focuses on CPU, memory, and GPU allocation, with each game requiring different resource levels based on its performance needs. Some games consume resources steadily, while others fluctuate depending on game state; sometimes using minimal resources and other times requiring high performance. Although these usage patterns are complex and unpredictable, every game has an identifiable range of high and low resource usage. Previous work [5] explored colocating multiple games on the same server without using VMs. While effective in a theoretical context, this approach does not fully align with practical, real-world solutions.

In this paper, we investigate the possibilities of VMs on a single physical computer, allowing multiple players to share resources dynamically. To optimize colocating multiple

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

games with VMs, we create various game combinations and apply machine learning techniques and measure each game's frame per second (FPS) to confirm if a VM satisfies the requirement for the allocated game. Specifically, this paper makes the following contributions:

- We run games through various VMs in real-time online via the Internet, demonstrating the feasibility of using VMs in practical cloud gaming scenarios.
- We evaluate the performance results of using machine learning in handling complex tasks. This approach enables dynamic cloud server creation and resource allocation based on real-time game demands, enhancing scalability and efficiency in cloud gaming environments.

These contributions might help to provide comprehensive frameworks for optimizing cloud gaming performance and resource usage.

This paper is structured as follows. Section II provides the background, outline key concepts, and the context of cloud gaming and virtualization. Section III discusses resource sharing in the cloud for gaming, focusing on game colocation and VM configurations. Section IV details the experiment, including the setup, tools, and performance testing of the system. Section V covers Related Work, reviewing previous research relevant to cloud gaming and resource optimization. Finally, Section VI presents the conclusions and future work, summarizing the findings and proposing directions for future research.

2. BACKGROUND

Although cloud gaming was used before the 1990s, it has become popular in recent years. Players prefer playing games online rather than purchasing high-performance computers. High performance games require high performance devices. For an efficient gaming experience, choosing cloud gaming setup is the simpler, faster and cheaper way to start playing. For running and playing a game on the local computer, players are sometimes required to download games the size of which become over 100GB (e.g., a game called Call of Duty), taking time according to the bandwidth. Moreover, players might be required to upgrade their PC over time because of the graphic demand of a game.

For running a game on the cloud and playing it on the local computer, on the other hand, the server runs the game, renders graphics and sends them to players in addition to receiving player's actions such as clicking a button. Then, client software receives the graphics and renders them on the local computer, not requiring a GPU on the local computer.

The cloud gaming service for players has multiple ways for its setup. One of the popular ways is hosting dedicated servers, sharing resources just for one player according to the requirement of the game. Cloud gaming using dedicated server is more expensive than creating a single instance and sharing through multiple players. OnLive [6], a pioneer in cloud gaming, went bankrupt due to this reason. Other methods, such as Vector Bin Packing (VBP) which allowed playing multiple games on a single instance as long as the capacity does not exceed. This still has over-provisioning problem [5].

There are multiple cloud gaming services on the market, as well as several published research papers. Prior work [5], [7], [8] has already discussed game colocation on a single instance. Game colocation allows two or more games to run on a single instance. However, using a brute-force approach for colocating games is neither practical nor cost-effective. In [5] uses ASTER [9] that enables game colocation on Windows platform. Using ASTER, players connect multiple keyboards, mice and displays to the single computer and share resources among games, meaning that using ASTER cannot accomplish cloud gaming environments.

To realize cloud game environments in practice, using VMs can be considered an adequate option because cloud services such as Amazon EC2 use VMs. Using VMs provides a more realistic and modern cloud gaming architecture, allowing players to connect directly over the internet. VMs enable finetuned resource management for each game. It is important because they allow for efficient resource management and scalability, especially in cloud gaming environments. VMs enable the optimization of computing resources, such as CPU, memory, and bandwidth, by allocating and reallocating them dynamically based on user demand. This flexibility reduces hardware dependency, minimizes energy consumption, and improves overall performance. Additionally, isolation and security are enhanced, as each player gets a separate, dedicated environment, ensuring that resource consumption in one instance won't affect others [2].

As a gaming environment, most players are satisfied with 60FPS [10], and some games do not require even 60FPS such as turn based games. Therefore, in this paper, we consider 60FPS with a high graphic setting is enough for providing cloud gaming environments.

3. TOWARD RESOURCE SHARING ON CLOUD FOR GAMING

In this paper, we investigate the possibilities of colocation for cloud gaming with VMs. For this purpose, we use five games shown in Table I, and one physical computer on which Window 11 is running.

We chose these five games for testing based on their varying resource requirements and game types. All these games are highly popular on Steam and free to play. All these games allow the player to set their own in game graphic setting, resolution and FPS. These games were tested across different virtual machine levels to analyze resource allocation and ensure a balance between performance and efficiency.

We run Windows Hyper-V for generating and controlling VMs. Besides, to avoid both over-provisioning (wasting

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

resources) and under-provisioning (causing performance issues), we create five types of VMs, categorized into levels, where each level contains different resources, as shown in Table II. Then, we check which types of VMs each game can run on based on its requirements (e.g., FPS) in preparation for applying machine learning. Due to the limitations of CPU, memory, and GPU, we can only create five levels.

TABLE I
CHOSEN GAMES LIST AND INFORMATION.

Game Name	Game Type	Description	Require VM Level
DotA Underlords	Auto Battler	Small and light weight game	Level 2
DotA 2	Multiplayer Online Battle Arena	Advanced server stability and data exchange	Level 3
Counter-Strike 2	First-person Shooter	Advanced physics and particle systems	Level 4
The Sims 4	Social Simulation Game	Authentication via EA	Level 3
Team Fortress 2	First-person Shooter	Older source engine game	Level 3

TABLE II
VIRTUAL CPU, VIRTUAL MEMORY, AND ALLOWED GPU PERCENTAGES PER LEVEL OF VIRTUAL MACHINE.

	V-CPU (Core)	V-Memory (GB)	Allowed GPU %
Level 1	1	2	20
Level 2	2	4	40
Level 3	4	8	60
Level 4	6	12	80
Level 5	8	16	100

A. Profiling games and types of VMs

We created five types of VMs shown in Table II, from Level 1 to Level 5 in which Level 1 is the lowest and Level 5 is the highest and runs each game with each level to measure the resource usage such as physical memory load, core usage and CPU/thread usage respectively. In this measurement, we capture this resource usage data every two seconds that emulates practical gaming scenarios. We then check whether each type of VM satisfies the 60FPS requirement for each game.

Fig. 2 illustrates the average resource usages across different types of VMs. We have found the relation between the resource usages and the games, for example Level 1 cannot satisfy requirements of the games because core usage and CPU/thread usage show 100%. When it comes to FPS, Level 3 can satisfy most of the games' requirements (i.e., 60FPS). Consequently, three out of five games can run on Level 3 VMs, and the rest of games must be assigned to Level 2 and Level 4 respectively shown in Table I. This information is useful to run multiple VMs on a physical computer at a time explained in Section III.

B. Game Combination

The games are colocated using combinations determined by a reputation-based formula in equation 1, *C* represents the combination, *n* represents the total number of games available for colocation, and *r* represents the number of games chosen to be colocated together at a time. Due to the resource limitations of a physical computer, we can only run three VMs at a time. Therefore, in this case, *n* is 5 as well as *r* is 3, resulting in 35 unique colocation combinations. In addition, due to the vCPU limitations detailed in Section III-C, some combinations out of 35 are unavailable (e.g., three Level 4 games are unavailable due to the restrictions of vCPU and three games at a time), thereby we remove the unavailable combination from 35 combinations, resulting in 24 available combinations. This approach enables us to evaluate each possible game colocation configuration.

$$ {}^{n}C_{r} = \frac{(n+r-1)!}{r! \cdot (n-1)!} \qquad (1) $$

C. Virtual Machine Setting

The physical computer used in this paper contains 16 CPU cores. When more than 12 virtual CPU cores are assigned in total to the VMs, the computer crashes due to the limitation of its 16 physical CPU cores. Therefore, based on this result and the observations in Section III-B, the possible combinations of VMs are 6 as shown in Table III. There are 10 combinations in total, however 6 of the combination is feasible.

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

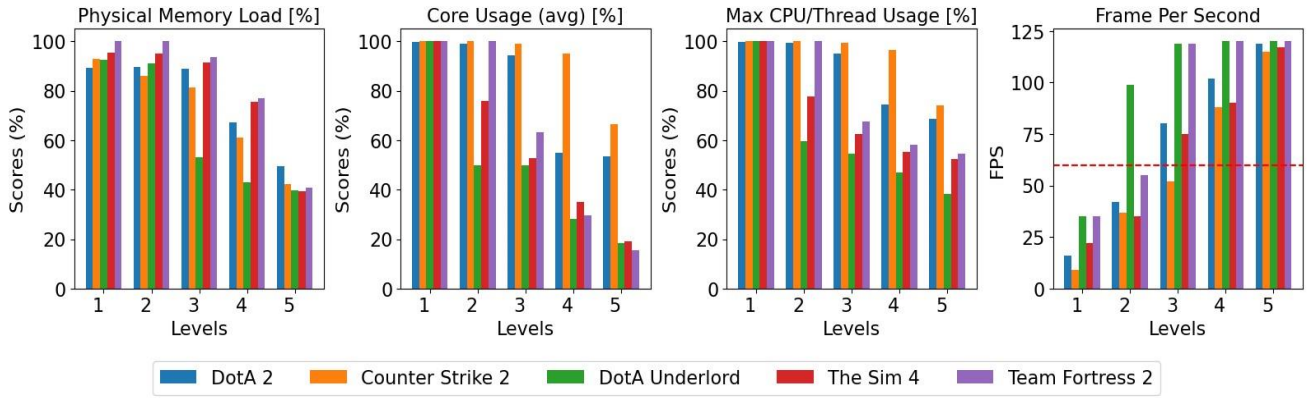

Fig. 2. Resource usage for different types of VMs

4. APPLY MACHINE LEARNING AND EXPERIMENT

We apply machine learning techniques to run and/or choose an adequate VM for players who will join and start a game running in a cloud environment. Using restrictions and/or conditions, we collect data to be used by machine learning algorithms and measure their performances.

A. Experimental setup

We show the specification of the hardware used in this experiment in Table IV. We use Windows 11 as the operating system for the physical computer with Hyper-V Manager, which controls the VMs running on it. Easy-GPU-PV [11] is a GPU paravirtualization tool for Windows Hyper-V, enabling VMs to access GPU resources on Windows. Parsec [12] is a high-frame-rate remote desktop software, allowing users to connect to VMs for gaming. This computer connects to 100 Mbps Ethernet and players connect VMs with Wi-Fi via the Internet, which is the same condition players play a game running in a cloud environment.

B. Data Collection

We collect logs for each colocation every 2 seconds over a 10-minute duration. These logs contain resource usage data from the physical computer and VMs. Although we can collect over three hundred kinds of data, we only select the essential data for machine learning applying feature importance algorithm [13]. As a result, we focus on Memory, CPU, and GPU in addition FPS as shown in Table V. We collected 9,392 samples being applied to machine learning.

TABLE III
ALL COMBINATIONS OF VM LEVEL

No.	VM 1	VM 2	VM 3	feasibly
1	Level 2	Level 2	Level 2	feasible
2	Level 2	Level 2	Level 3	feasible
3	Level 2	Level 2	Level 4	feasible
4	Level 2	Level 3	Level 3	feasible
5	Level 2	Level 3	Level 4	feasible
6	Level 3	Level 3	Level 3	feasible
7	Level 4	Level 3	Level 3	infeasible
8	Level 4	Level 4	Level 2	infeasible
9	Level 4	Level 4	Level 3	infeasible
10	Level 4	Level 4	Level 4	infeasible

TABLE IV
SPECIFICATION OF PHYSICAL PC

CPU	16 Cores i9-12900KF
Memory	32GB RAM
Storage	2 TB
GPU	NVIDIA GeForce RTX 3080
Network Speed	100 Mbps
Required Software	Hyper-V Manager Easy-GPU-PV Parsec

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

C. Machine Learning

We apply machine learning techniques to optimize resource allocation while maintaining game performance stability. Resource partitioning, contention, and meeting performance requirements are inherently complex due to the varying demands of different games. Instead of manually predicting resource usage, we leverage machine learning to enable automated and adaptive optimization.

To make computations efficient, we extract key features from CPU, memory, GPU, and other system metrics, reducing the complexity while retaining essential performance indicators. This allows the machine learning model to accurately predict resource needs and ensure smooth gameplay while maximizing resource utilization.

By analyzing resource usage across different types of VMs in Fig2, we observe that resource consumption patterns do not follow a simple linear trend with respect to allocated resources. Some games exhibit nonlinear scaling, where increased CPU, memory, or GPU allocation does not directly correlate with proportional performance improvements. This variation makes manual optimization challenging, as resource demands fluctuate dynamically based on gameplay conditions.

In this paper, we apply two machine learning algorithms: Random Forest (RF) and Support Vector Machine (SVM). RF and SVM can effectively analyze non-linear relationships. RF, as an ensemble method, reduces over fitting by averaging multiple decision trees. [14] SVM is effective in high-dimensional spaces and prevents over fitting through margin maximization. [15]

We divide the obtained samples into 75-25 ratio for training and testing. We then add a parameter called High Performance (shortly H), which is a binary value: 0 or 1, as shown in equation 2. In this equation, fx (x=1,2,3) represents FPS for each VM (VM1, VM2 and VM3). If all FPS are above 60, H becomes 1, otherwise 0. Applying two machine learning techniques, we obtain five well-known values: Accuracy, Precision, Recall, Specificity and F1 Score respectively as shown in Fig. 3.

D. Evaluation

In terms of accuracy, both RF and SVM demonstrate 90% to 95%, indicating that both models can assign players to an appropriate VM. This result does not imply that 5% to 10% of players are unable to play their required games; rather, it means that players experience insufficient FPS for 5% to 10% of the total playing time, as shown in Fig. 4.

Precision represents the proportion of true positives among the colocations identified by the model. In our evaluation, RF achieved 85% precision, while SVM reached 77%. However, the actual percentage of players able to play smoothly may be higher, as minor FPS fluctuations do not always impact gameplay quality.
Furthermore, although the insufficient FPS does not meet the requirements, the degradation ratio is small, suggesting that our machine learning model could be viable for practical use.

TABLE V
ESSENTIAL DATA BEING APPLIED TO MACHINE LEARNING

Category	Machine Log Info
Memory	Virtual Memory Load [%] Physical Memory Load [%]
CPU	Core Usage (avg) [%] Total CPU Usage [%] Total CPU Utility [%]
GPU	GPU Video Engine Load [%] GPU Memory Usage [%] GPU D3D Usage [%]
Other	Framerate [FPS] Current Running Game

$$H = \begin{cases} 1 \text{ if } f_1 \geq 60 \text{ and } f_2 \geq 60 \text{ and } f_3 \geq 60 \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

5. RELATED WORK

There are multiple ways to find the optimal trade-off between performance and visual quality when playing games. Players tend to prioritize the best visual experience as long as the performance remains sufficient to meet their QoE standards [16].

Previous work has highlighted the sensitivity and intensity in measuring QoE degradation [5], [17]. Sensitivity refers to the degree of performance degradation a game experiences when there is competition for shared resources, while intensity describes the level of pressure a game places on those resources. “Pressure” here means intentionally reducing resource availability during gameplay to observe how well a game can perform under constrained conditions. Prior research utilized ASTER, which is designed specifically for game colocation in a local environment.

However, our approach differs by using VM levels to achieve similar insights into QoE degradation. This method allows us to balance performance and resource usage more effectively for cloud gaming providers, offering a simpler way to add new games by testing VM levels. Moreover, we used VM and tested on real world saturation and made it more realistic.

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

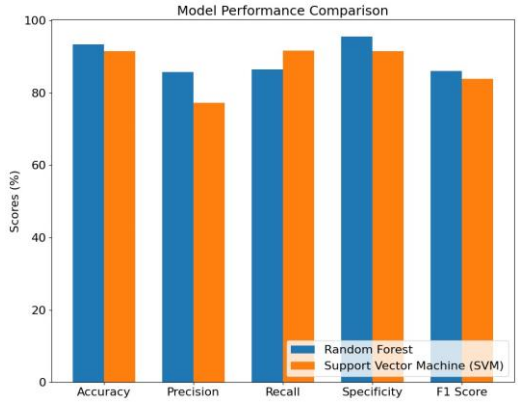

Fig. 3. Performance comparisons between RF and SVM

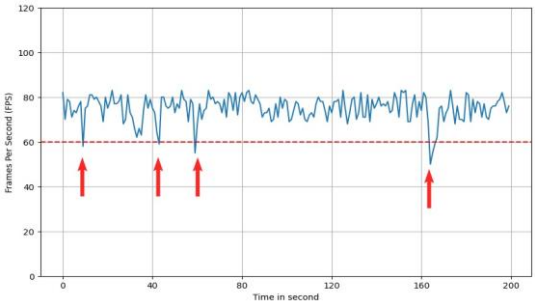

Fig. 4. An example of FPS transition over time

6. CONCLUSIONS AND FUTURE WORKS

This paper investigates the possibilities of using VMs for colocating multiple games running in a cloud environment. Different from prior works, we use VMs running on a physical computer that represents a practical scenario to realize cloud game environment. Considering several restrictions and complexities, we apply machine learning techniques to create models that assign players to adequate VMs while balancing resource usage and performance. Experimental results show that our models might work effectively in a practical cloud gaming environment.

As future work, we plan to evaluate more games with different genres to test the scalability and generalization of our approach. Additionally, we aim to integrate deep learning models to enhance performance prediction further. Finally, we will consider testing our model across multi-cloud environments, including AWS, Azure, or GCP, to assess its adaptability to different infrastructures.

REFERENCES

[1] Chun-Ying Huang, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. Gaminganywhere: The first open-source cloud gaming system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(1s):1–25, 2014.

[2] Sawsan Ali Hamid, Yassine Boujelben, and Faouzi Zarai. Enhancing cloud gaming experience through optimized virtual machine placement: A comprehensive review. *Journal of Network and Systems Management*, 32(4):80, 2024.

[3] Ryan Shea, Jiangchuan Liu, Edith C-H Ngai, and Yong Cui. Cloud gaming: architecture and performance. *IEEE network*, 27(4):16–21, 2013.

[4] Saeed Shafiee Sabet, Steven Schmidt, Saman Zadtootaghaj, Babak Naderi, Carsten Griwodz, and Sebastian Moller. A latency compensation¨ technique based on game characteristics to mitigate the influence of delay on cloud gaming quality of experience. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 15–25, 2020.

[5] Yusen Li, Changjian Zhao, Xueyan Tang, Wentong Cai, Xiaoguang Liu, Gang Wang, and Xiaoli Gong. Towards minimizing resource usage with qos guarantee in cloud gaming. *IEEE Transactions on Parallel and Distributed Systems*, 32(2):426–440, 2020.

[6] Onlive. http://onlive.com/, 2013.

[7] Iryanto Jaya, Wentong Cai, and Yusen Li. Rendering server allocation for mmorpg players in cloud gaming. In *Proceedings of the 49th International Conference on Parallel Processing*, pages 1–11, 2020.

[8] Yusen Li, Chuxu Shan, Ruobing Chen, Xueyan Tang, Wentong Cai, Shanjiang Tang, Xiaoguang Liu, Gang Wang, Xiaoli Gong, and Ying Zhang. Gaugur: Quantifying performance interference of colocated games for improving resource utilization in cloud gaming. In *Proceedings of the 28th international symposium on high-performance parallel and distributed computing, pages 231–242, 2019.2*

[9] https://ibiksoft.com/. Aster. https://ibiksoft.com/, 2013.

[10] Kajal T Claypool and Mark Claypool. On frame rate and player performance in first person shooter games. *Multimedia systems*, 13(1):3– 17, 2007.

[11] jamesstringerparsec. Easy-gpu-pv. https://github.com/jamesstringerparsec/Easy-GPU-PV, 2022.

[12] Parsec. https://parsec.app/, 2024.

[13] Geeksforgeeks. Feature importance with random forests. https://www. geeksforgeeks.org/feature-importance-with-random-forests/, 2024.

[14] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[15] Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273– 297, 1995.

[16] Steven Schmidt, Saman Zadtootaghaj, and Sebastian Moller. Towards¨ the delay sensitivity of games: There is more than genres. In *2017 Ninth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2017.

[17] Jason Mars, Lingjia Tang, Robert Hundt, Kevin Skadron, and Mary Lou Soffa. Bubble-up: Increasing utilization in modern warehouse scale computers via sensible co-locations. In *Proceedings of the 44th annual IEEE/ACM International Symposium on Microarchitecture*, pages 248– 259, 2011.

Copyright © 2025 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.