Cloud Gaming and Containers

Essay for DD2482: Automated Software Testing and DevOps

Stefano Scolari
sscolari@kth.se

April 2023

I certify that generative AI, incl. ChatGPT, has not been used to write this essay. Using generative AI without permission is considered academic misconduct.

1 Cloud Gaming Background

Cloud Gaming has experienced rising popularity in the last 5 years. The market cap for Cloud Gaming was valued to be at around $244M in 2020, and, as indicated by the Global Cloud Gaming Report, its market cap in 2022 was reported to be at around $1.3B. And its growth is expected to reach $13B by 2028. [1].
Cloud Gaming was developed on the basis of *gaming on demand*, making it possible for gamers and users to play at any time and anywhere. This lifts end-users from the hassle of worrying about computational power and download time. As games are becoming increasingly complex, the direct demand for GPU power has parallelly increased.
The base mechanism for Cloud Gaming, if looked at from a high level is quite simple. Users connect to a gaming server that runs the game instance, and input is captured in real-time from the end-user and sent to the server. The input, when received, is used as input for the game instance. At this point the next frame is computed, encoded, and sent back to the end-user's thin client. It's basically an *interactive video streaming service.*

2 Virtualization

Having understood that in Cloud Gaming, the game itself is rendered on a distant cloud server, it may be intuitive to understand that such a service requires the use of very expensive and high-end hardware. Graphical Processing Units (GPUs) need to be shared and orchestrated amongst a number of simultaneous users, some of them, thanks to their computational power, can handle up to 30 simultaneous users [3].
This makes it possible to isolate and separate user's access and utilization of the underlying hardware through virtualization techniques.

2.1 Relevance

I chose to analyze and study a more in-depth and performance-related evaluation between Containers and Virtual Machines, because, although Containers are known to have great advantages compared to VMs, increasing awareness around it could help less innovation-prone and habitual sectors like the *Game Development* one, to more extensively use and implement this kind of technology [7]. As explained by Whitehead and Jantunen [2], Containers hold great potential to enhance, simplify and optimize the Game Development process. Furthermore, they offer a more scalable and optimizable solution for handling *game servers* and *backend microservices* [4].

2.2 Containers and VMs

Traditionally, cloud gaming instances were booted through a hypervisor used to generate hardware-level VMs. Making it possible to have isolation while also having resource sharing between users.
As previously mentioned, the most important hardware component for Cloud Gaming is represented by the GPU. Meaning that its performance level, although accessed through a VM, must resemble the native one.
Extensive benchmarking has been performed on both Virtual Machines (Vms) and Containers, the results show that the use of VMs to virtualize game instances can result in notable performance overhead [21, 12].

3 Containers

Before going into detail and explaining the *performance-related* advantages Containers were shown to have over traditional VMs, it is also important to remember the more well know general advantages, and explain why they are of vital importance for Cloud Gaming.
Containers have a smaller footprint when compared to VMs. This means that they require fewer resources to run, making them more efficient in terms of CPU and memory usage. This improved efficiency allows handling more game instances simultaneously on the same physical *Cloud Gaming server.*
This is of extreme importance for Cloud Gaming, in fact, it makes it possible for more players to be served with the same amount of resources, resulting in *increased scalability* and *lower costs.*
The greater flexibility provided by Containers also makes them a preferred solution over VMs. Containers can be easily moved between servers and platforms, which allows for more efficient resource allocation and better load balancing [18]. This is especially important for *Cloud Gaming*, where game instances need to be provisioned and de-provisioned on demand, and where traffic can fluctuate rapidly [6]. Containers can be deployed quickly and easily, allowing for fast scaling up and scaling down of resources as needed.
The existence of systems such as `Kubernetes` results in an improved and increased ease of management of containers. The deployment, management, and scaling of game instances can result in this way easier [24].
It's not a mystery that Google's own Cloud Gaming Service, Stadia, while active, had the best possible performance and lowest delay when compared to competitors [8]. As we all know, Kubernetes was developed by Google. There is no certain proof of a connection between the two things as Stadia's architecture was kept fairly secret, but assuming that Kubernetes played a crucial role in establishing a solid back-end infrastructure is a totally reasonable theory.

3.1 Containers performance advantage

After explaining the general advantages of Containers over traditional Virtual Machines, it is time to analyze in more detail the performance comparison between these two techniques. Most of the information and data presented from now onwards is mainly taken from the work of Kämäräinen et. al.[12]. This is a consequence of the fact that not much academic research has gone into evaluating and extensively benchmarking a performance comparison between Containers and Virtual Machine for Cloud Gaming. Research has dealt with benchmarking this comparison when training Machine Learning and Deep Learning models [22], or `MySQL` [10], but *gaming* and *computer graphics* applications have mostly flown under the radar. It may be a logical assumption to think that, proven Container's performance to be superior in many fields of application, better results will be expected also for GPU intensive applications. On the other hand, it is anyway important to have a basis for comparison and evaluation. In the following sections, I will be going over some areas of comparison between Containers and Virtual Machines, explaining why and how these are important for *Cloud Gaming*

3.1.1 CPU performance

Even though GPUs are usually more well suited for video encoding, as shown in the work by Safin et. al [19], the use of some hybrid CPU-GPU models has been researched with promising results [9].
Via *ffmpeg* and and *x264* video was encoded to `H.264/MPEG-4 AVC` video coding format. The results [12] reveal that Virtual Machine introduce an overhead of up to 16%.

3.1.2 Memory and power usage

Virtual Machines usually are assigned with a static and constant amount of the host's RAM, leading to sub-optimal memory allocation and usage. On the other hand, Containers are not affected by this problem.
Moreover, power-consumption was found to be lower when using Containers over Virtual Machines [15], which is an important aspect for a Cloud Gaming service.

3.1.3 GPU performance

Coming now to the GPU comparison, VMs were found to have up to an 8% loss compared to native performance. Containers, on the other hand, achieve no significant loss in performance. Furthermore, as Containers are basically instances of processes running in the OS, sharing the GPU resources is automatic. Quite different if we compare it with VMs, which require specific GPU passthrough technology such as Nvidia's vGPU [5] to achieve similar results.
On top of that, VMs showed a 22% loss in performance in full *HD* and a staggering 33% loss in 4K resolution.

3.1.4 Game performance

After analyzing factors influencing scalability and cost on the infrastructure's side, we are now evaluating the different Quality of Services (QoS) and Quality of Experiences (QoE).
Also in this case, VMs presented significant overheads and worse performance when benchmarked by testing various games. Depending on the GPU used, overhead could reach values of up to 27%.

3.1.5 Startup time performance

Although overseen, startup time is of vital importance to assure a high *QoE*. Users on the thin-client, often a simple Browser, are promised instant-into-gaming experiences and introducing waiting times results in a decreased *QoE*. Containers can offer almost instantaneous startup times, compared to almost half a minute of waiting [13] for traditional VMs. Moreover, recent research has discovered new ad-hoc dependency scheduling methods to further improve containers' startup time [11]

4 Limitations

Up until this point, we have only considered the advantages of containerization above traditional Virtual Machines.
Despite the positives, a number of limitations need to be listed and presented.

4.1 Security

Containers are known to offer minimal separation between the hosting OS and other containers on the same machine. This, as a consequence, leads to security boundaries being less robust compared to Virtual Machines [23].
This means that performance and scalability come with a trade-off when compared to security. Despite the former's importance, security must be a central concern.
Considering the case of Docker, the work from Lee et. al [14] demonstrate that the network connection amongst Docker containers can lead to weakened independence container characteristics. Because of this, `DDoS` and *Man-In-The-Middle* (MITM) attacks are potentially possible. Considering that both accounts and profiles used to access various stores and specific game-related content are accessed from within the *Cloud Gaming* instances, means that in case of security failures or MITM attacks, a data breach is to be considered as a potential risk.
Hybrid solutions have been proposed, where containerization is performed from within Virtual Machines to solve the security issues and concerns [17]. Although an interesting solution, it's difficult to see this as a feasible option for

Cloud Gaming, as performance limitations would still be an issue due to VMs' hardware-level virtualization.

Also, the more limited isolation in Containers means that if a security vulnerability exists in the kernel, it could affect all the containers running on that host, causing (not only) decreased *QoS* and *QoE*.

5 Containers in Game Development

As previously mentioned, an in-depth analysis of the advantages of Containers when dealing with virtualization requiring GPU-intensive computations is of great importance for *AAA* Game Development.

Several articles have shown how, despite there being great advantages, the progress and speed at which this new technology is being integrated and used could be greatly improved [7].

A good example of efficient integration of Containers in a AAA Game Development environment is given to us directly from a Riot blog post [20] by Stewart. Here, he explains how the creators of League Of Legends, transitioned to using containers for *Continuous Deployment*. Using Docker container files meant they could achieve their *core principle*: engineering ownership over their build environments. Developers found it faster to iterate and deploy improvements to their audience, which meant League of Legends players were always getting the best possible experience and the latest code.

As explained in [16, 4], containers together with Kubernetes can make it possible to better scale the game servers.

Improved and more easily manageable back-end architectures for game servers will help create new and amazing online gaming experiences, capable of handling a greater number of players.

6 Conclusions

As thoroughly presented, *Containers* have important advantages over traditional virtualization methods for *Cloud Gaming* solutions.
The lightweight, portable, and scalable nature of *Containers* makes them an ideal choice for *game developers* and Cloud Gaming providers.
The clear benefits of *Containers* extend beyond the realm of gaming, with the potential to revolutionize *game development* itself. Providing a consistent environment for developers, allows them to create and deploy games with ease.

As the popularity of *Cloud Gaming* continues to rise, *Containers* are sure to play an increasingly critical role in the industry, becoming the *go-to* solution for *game developers* and *Cloud Gaming* providers.

References

[1] Cloud Gaming Market: Global industry trends, share, size, growth, opportunity and forecast 2023-2028. https://www.imarcgroup.com/cloud-gaming-market. Accessed: 2023-04-06.

[2] Containers for game development. https://pages.awscloud.com/Containers-for-Game-Development_2020_0322-GAM_OD.html?&trk=el_a131L0000084iEtQAI&trkCampaign=March_2020_0322-GAM&sc_channel=el&sc_campaign=pac_Q1-2020_exlinks_blog_OTT_03DGAB&sc_outcome=Product_Adoption_Campaigns&sc_geo=NAMER&sc_country=mult. Accessed: 2023-04-07.

[3] How many users can share a gpu? https://www.citrix.com/blogs/2013/02/22/how-many-users-can-share-a-gpu/. Accessed: 2023-04-06.

[4] Optimize game servers hosting with containers. https://aws.amazon.com/it/blogs/gametech/optimize-game-servers-hosting-with-containers/. Accessed: 2023-04-07.

[5] Quantifying the impact of virtual GPUs: See how nvidia benchmarks ux in vmware virtualized environments. https://images.nvidia.com/aem-dam/Solutions/design-visualization/solutions/resources/documents1/quantifying-impact-of-vgpu-whitepaper.pdf. Accessed: 2023-04-07.

[6] Wei Cai, Ryan Shea, Chun Ying Huang, Kuan Ta Chen, Jiangchuan Liu, Victor C.M. Leung, and Cheng Hsin Hsu. A survey on cloud gaming: Future of computer games. *IEEE Access*, 4:7605–7620, 2016.

[7] Game Design. Kubernetes for Game Development. 2021.

[8] Andrea Di Domenico, Gianluca Perna, Martino Trevisan, Luca Vassio, and Danilo Giordano. A network analysis on cloud gaming: Stadia, geforce now and psnow. *Network*, 1(3):247–260, 2021.

[9] Ling Yu Duan, Wei Sun, Xinfeng Zhang, Shiqi Wang, Jie Chen, Jianxiong Yin, Simon See, Tiejun Huang, Alex C. Kot, and Wen Gao. Fast MPEG-CDVS encoder with GPU-CPU hybrid computing. *IEEE Transactions on Image Processing*, 27(5):2201–2216, 2018.

[10] Wes Felter, Alexandre Ferreira, Ram Rajamony, and Juan Rubio. An updated performance comparison of virtual machines and Linux containers. *ISPASS 2015 - IEEE International Symposium on Performance Analysis of Systems and Software*, pages 171–172, 2015.

[11] Silvery Fu, Radhika Mittal, Lei Zhang, and Sylvia Ratnasamy. Fast and efficient container startup at the edge via dependency scheduling. *HotEdge 2020 - 3rd USENIX Workshop on Hot Topics in Edge Computing*, 2020.

[12] Teemu Kamarainen, Yuanqi Shan, Matti Siekkinen, and Antti Yla-Jaaski. Virtual machines vs. containers in cloud gaming systems. *Annual Workshop on Network and Systems Support for Games*, 2016-January, 2016.

[13] Kuljeet Kaur, Tanya Dhand, Neeraj Kumar, and Sherali Zeadally. Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers. *IEEE Wireless Communications*, 24(3):48–56, 2017.

[14] Haneul Lee, Soonhong Kwon, and Jong-Hyouk Lee. Experimental analysis of security attacks for docker container communications. *Electronics*, 12(4), 2023.

[15] Martin Lindström. Containers Virtual machines comparison. 2022.

[16] "Mark Mandel". Scaling dedicated game servers with kubernetes. `https://www.gamedeveloper.com/programming/scaling-dedicated-game-servers-with-kubernetes-part-1-containerising-and-deploying`. Accessed: 2023-04-08.

[17] Ilias Mavridis and Helen Karatza. Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing. *Future Generation Computer Systems*, 94:674–696, 2019.

[18] Manoj Kumar Patra, Sanjay Misra, Bibhudatta Sahoo, and Ashok Kumar Turuk. Gwo-based simulated annealing approach for load balancing in cloud for hosting container as a service. *Applied Sciences*, 12(21), 2022.

[19] Ramil Safin, Emilia Garipova, Roman Lavrenov, Hongbing Li, Mikhail Svinin, and Evgeni Magid. Hardware and software video encoding comparison. *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2020*, (September):924–929, 2020.

[20] "Maxfield F. Stewart". Thinking inside the container. `https://technology.riotgames.com/news/thinking-inside-container`. Accessed: 2023-04-08.

[21] Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, Timoteo Lange, and Cesar A.F. De Rose. Performance evaluation of container-based virtualization for high performance computing environments. *Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2013*, pages 233–240, 2013.

[22] Pengfei Xu, Shaohuai Shi, and Xiaowen Chu. Performance Evaluation of Deep Learning Tools in Docker Containers. *Proceedings - 2017 3rd International Conference on Big Data Computing and Communications, BigCom 2017*, pages 395–403, 2017.

[23] Anuj Kumar Yadav, M. L. Garg, and Ritika. *Docker containers versus virtual machine-based virtualization*, volume 814. Springer Singapore, 2019.

[24] Qi Zhang, Ling Liu, Calton Pu, Qiwei Dou, Liren Wu, and Wei Zhou. A Comparative Study of Containers and Virtual Machines in Big Data Environment. *IEEE International Conference on Cloud Computing, CLOUD*, 2018-July:178–185, 2018.