Use of Server Mesh in the Context of Games

*Miguel Ferreira (1230199)*

*Instituto Superior de Engenharia do Porto, Portugal*

[[1]](#footnote-1)

***Abstract*—.The current architecture of online multiplayer games faces significant limitations regarding seamless data transfer, often forcing players to endure loading screens between transitions and server lag. Additionally, issues such as limited server capacity and poor scalability in distributed systems hinder the overall gaming experience. Addressing these challenges has become a focal point of technological innovation in the gaming industry, with several companies competing to develop solutions. One of the most promising approaches is server meshing, which enables dynamic load distribution and horizontal scalability across multiple servers. Cloud Imperium Games (CIG) is at the forefront of this innovation, demonstrating the potential of server meshing to revolutionize gaming. By enabling near-limitless scalability and reducing server load bottlenecks, this technology represents a significant leap toward creating vast, persistent game worlds with seamless player experiences. This paper explores the technological foundations of server meshing, its potential impact on game design, and its implications for the future of distributed systems in the gaming industry.**

***Index Terms*—Distributed systems, Dynamic server loading, Server mesh, Static server mesh, Dynamic Server Mesh, Static Server mesh, Start Citizen, Ashes of Creation, Usefulness.**

# I. INTRODUCTION

As the demand for lage-scale online multiplayer games grows, like mmos, so do the technical challenges associated with supporting vast game worlds and thousands of concurrent players. Traditional server architectures often impose limitations in terms of scalability and performance, leading to lag, the necessity for loading screens, needing to slow down the system, and overall poor user experiences. Addressing these issues requires innovation in distributed systems architectures in games.

One of the leading innovations in this landscape is server meshing, which is being pioneered by companies like Cloud imperium Games (CIG) and Interpid Studios and is expected to change the industry by enabling dynamic load balancing and near-limitless scalability. Server meshing allows multiple servers to work together, distributing player loads dynamically and thereby creating vast, and seamless game worlds without the performance bottlenecks associated with traditional server models.

## A. Abbreviations and Acronyms

CIG: Cloud Imperium Games

MMO: Massively Multiplayer Online

TiDi: Time Dilation

NPC: Non-Player Character

# II. Current Distributed software architecture

The current scenery of distributed systems offer solutions that are highly scalable and reliable, allowing for mass exchange in data, tools like Kubernetes allow for easy horizontal scalability and as distributed architectures become more prevalent it is a must use tool[1], there are already corroborated solutions for making systems fault tolerant like using redundancy between servers[2].

## Differences

Even if modern software architecture has been improving, even being able to work with millions of data, our games server architecture still has problems in proving these capabilities. The main problem is that games are full of state, an state management is still one of the biggest hurdles to solve, and this problem is accentuated on game servers due to its innate need for high state management [3]. As such, we have seen these kinds of limitations in display, for example, with eve online when the biggest online battle happened [4], it was needed to reinforce the servers and use a special technology created by the formers, called TiDi[5] which slows down game time so server do not crash.

# III. Server Meshing: A Technological Overview

Server meshing is the process of dynamically splitting the load of a game world across multiple servers, effectively treating them as a unified entity. This contrasts with the more traditional sharded servers’ model, like the mega servers of Guild Wars 2[6], where game worlds are divided into static instances. In server meshing, a game world can be hosted across multiple servers that coordinate dynamically based on real-time load and player activity.

Two primary types of server meshing have been proposed:

1. **Static Server Mesh**: In this configuration, the game world is partitioned into predefined areas, with each area assigned to a specific server. Players seamlessly transition between these areas without needing to experience loading screens. However, the static mesh does not allow for dynamic reallocation of server resources [7].
2. **Dynamic Server Mesh**: This more advanced configuration allows for real-time load distribution. The game world is not divided into static zones but rather allocated based on player density and server capacity. Servers can dynamically adjust to handle more players in high-traffic areas or offload areas with fewer players, optimizing performance and scalability [7].

## Benefits of Server Meshing

1. **Horizontal Scalability**: Server meshing enables scaling horizontally by adding more servers to the network. This approach contrasts with traditional vertical scaling, where upgrading server hardware becomes cost prohibitive as player numbers grow.[8]
2. **Seamless Player Experience**: By reducing or eliminating loading screens between server transitions, server meshing allows for a more immersive experience. Players can traverse vast game worlds without interruptions, making the game feel more persistent and alive [9].
3. **Reduced Bottlenecks**: Dynamic server meshing addresses the issue of overcrowding in high-traffic areas by allocating more servers to those areas, reducing lag and enhancing performance [8][9].

# V. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the article, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

# Appendix

Appendixes, if needed, appear before the acknowledgment.

References

1. *Kubernetes Components | Kubernetes*. (n.d.). Retrieved September 28, 2024, from https://kubernetes.io/docs/concepts/overview/components/
2. *Fault Tolerance in Distributed Systems: Strategies and Case Studies - DEV Community*. (n.d.). Retrieved September 27, 2024, from <https://dev.to/nekto0n/fault-tolerance-in-distributed-systems-strategies-and-case-studies-29d2>
3. Distributed State — Challenges and Options | by Koshy | Medium. (n.d.). Retrieved October 1, 2024, from https://nittikkin.medium.com/distributed-state-management-80c8100bb563
4. *Fury at FWST-8 Battle Report | EVE Online*. (n.d.). Retrieved September 27, 2024, from <https://www.eveonline.com/news/view/fury-at-fwst-8-battle-report>
5. Introducing Time Dilation (TiDi) | EVE Online. (n.d.). Retrieved October 1, 2024, from <https://www.eveonline.com/news/view/introducing-time-dilation-tidi>
6. Introducing the Megaserver System – GuildWars2.com. (n.d.). Retrieved October 1, 2024, from <https://www.guildwars2.com/en/news/introducing-the-megaserver-system/>
7. *The Nosy Gamer: The Server Meshing Wars Have Begun*. (n.d.). Retrieved September 25, 2024, from <https://nosygamer.blogspot.com/2024/07/the-server-meshing-wars-have-begun.html>
8. Horizontal scaling vs vertical scaling: Choosing your strategy | DigitalOcean. (n.d.). Retrieved October 1, 2024, from <https://www.digitalocean.com/resources/articles/horizontal-scaling-vs-vertical-scaling>
9. *(42) Ashes of Creation Alpha Two Server Meshing Technology Preview - YouTube*. (n.d.). Retrieved September 27, 2024, from https://www.youtube.com/watch?v=pdav0as54mU&t=360s
10. *Server meshing - Ashes of Creation Wiki*. (n.d.). Retrieved September 27, 2024, from https://pt.ashesofcreation.wiki/Server\_meshing
11. Duong, T. N. B., & Zhou, S. (2003). A dynamic load sharing algorithm for massively multiplayer online games. *IEEE International Conference on Networks, ICON*, 131–136. https://doi.org/10.1109/ICON.2003.1266179
12. Aote, S. S., & Kharat, M. U. (2009). A game-theoretic model for dynamic load balancing in distributed systems. *Proceedings of the International Conference on Advances in Computing, Communication and Control, ICAC3’09*, 235–238. https://doi.org/10.1145/1523103.1523153
13. *Server Meshing and Persistent Streaming Q&A - Roberts Space Industries | Follow the development of Star Citizen and Squadron 42*. (n.d.). Retrieved September 25, 2024, from https://robertsspaceindustries.com/comm-link/transmission/18397-Server-Meshing-And-Persistent-Streaming-Q-A
14. *What is Star Citizens Serving Meshing and Persistent Entity Streaming | by Marvas McCladdie | Medium*. (n.d.). Retrieved September 25, 2024, from https://medium.com/@MarvasMcCladdie/star-citizen-is-one-of-the-most-ambitious-and-highly-anticipated-video-games-of-all-time-486abb0a02a9
15. *(25) CitizenCon 2951: Server Meshing & The State Of Persistence - YouTube*. (n.d.). Retrieved September 25, 2024, from https://www.youtube.com/watch?v=TSzUWl4r2rU

1.  [↑](#footnote-ref-1)