VPP-based Hybrid Connectivity Technology to Support Large-scale Metaverse Services

Jian Jeong
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
jian0219@khu.ac.kr

Gyulim Oh
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
ohrbfla1004@gmail.com

Sumin Son
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
itoodo12@gmail.com

Eui-Nam Huh
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
johnhuh@khu.ac.kr

***Abstract*—There are some major difficulties to provide large-scale metaverse services in current metaverse environments. The server structure using the TCP protocol and the sequential network processing approach are those. While servers using the TCP protocol offer reliability, their packet processing speed is slow and not well-suited for large-scale metaverse environments. They also face limitations in scaling the server infrastructure. Furthermore, even if server architecture is improved, packets are processed sequentially in traditional networking system, leading to bottlenecks in large-scale metaverse environments where hundreds of transmissions of packets occur simultaneously. In this paper, we aim to address these issues by proposing a hybrid connection technology that combines TCP and UDP supported by parallel packet processing using VPP(Vector Packet Processing), suitable for large-scale metaverse server architecture. Our study results showed that the maximum acceptable number of clients reached was 225, which is more than double the capacity of the existing system, allowing metaverse services to be provided to a larger number of users.**

Keywords: Large-scale Metaverse, Virtual Router, VPP, DPDK

I. Introduction

A metaverse is a virtual world platform that provides immersive experiences and real-time interactions in an environment that blends the real and the virtual. Due to the impact of the COVID-19 pandemic, offline meetings have been avoided to prevent the spread of the pandemic, and remote activities such as teleclasses, teleconferencing, and remote commuting have developed, leading to the growth of the metaverse industry. While various studies have been conducted on the metaverse industry, a number of issues have emerged that are currently facing the metaverse [1].

The biggest problem is scaling, which is indispensable to accommodate large numbers of users [2]. The number of users is limited due to the large amount of data generated within the metaverse space and the constraints on various aspects such as server resources and communication bandwidth. Past research has focused mainly on bandwidth to solve this problem [3], but it is still insufficient to develop into a real service. Therefore, we propose a new perspective and method to overcome the scaling problem of metaverse. The metaverse environments need to be more reliable and effective for large-scale users by being optimized their server architecture and improved their efficiency.

The metaverse platform servers we used is composed of four servers: a connection management server, a movement synchronization server, a video data processing server, and a voice data processing server. Each server ensures reliability through the use of TCP(Transmission Control Protocol). However, due to the inherent characteristics of TCP, its packet processing speed is slow and not suitable for large-scale metaverse environments. Therefore, a protocol addressing the efficiency issues associated with a single TCP central server structure needs to be considered.

Although server architecture improves, it is still insufficient to accommodate a large number of users because traditional network packet filtering and processing technologies occur sequentially, making it impossible to process multiple packets simultaneously. To enable metaverse servers to handle a large number of users effectively, it should be able to handle multiple packets concurrently.

This study proposes VHCT(VPP-based Hybrid Connectivity Technology), which is networking platform for supporting large-scale metaverse services in a network environment based on VPP and DPDK(Data Plane Development Kit), offering a hybrid protocol that combines TCP and UDP to address the scaling issues.

II. Related Works

A. Metaverse

To commercialize the metaverse platform, various technological challenges along with practical implementation issues must be actively addressed [4]. The seamless provision of a high-level immersive interaction necessitates the essential integration of AI technology and 3D rendering techniques [5]. These technologies enhance user experience and impart a sense of reality, enriching interactions within the metaverse environment.

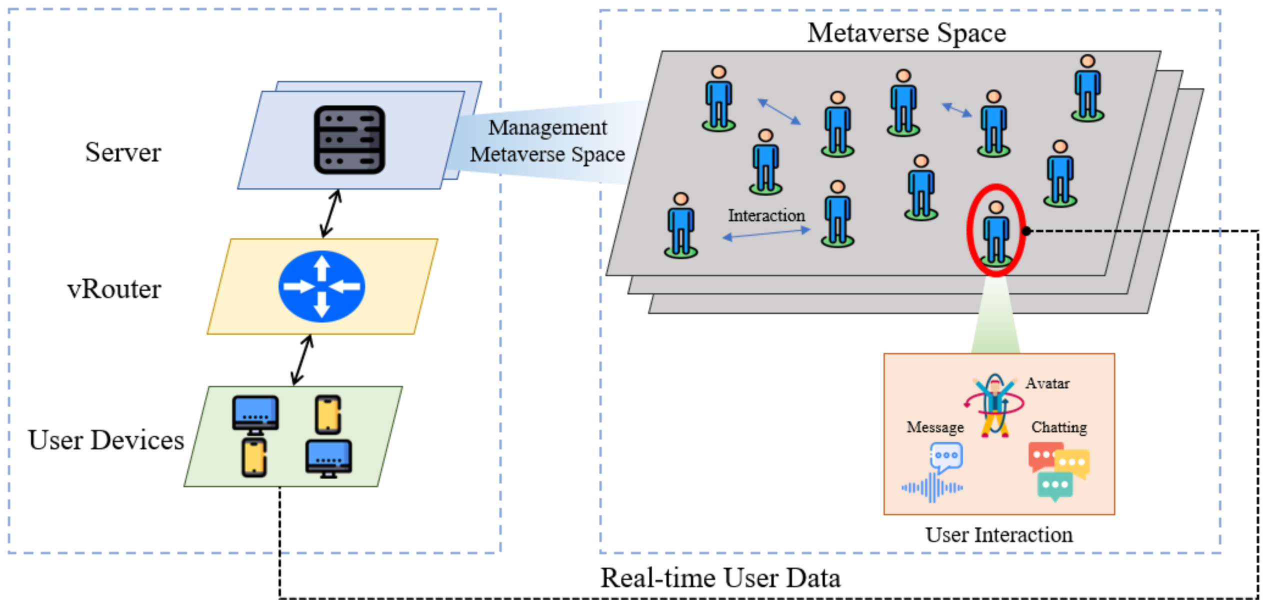

Fig. 1. VHCT Architecture

Moreover, accommodating huge number of users in the metaverse rises a fundamental challenge in handling massive amounts of data. The current limitations of single-server systems in effectively managing such large-scale data necessitate the indispensable adoption of a distributed server architecture as a solution [6]. However, operating large-scale distributed servers comes with substantial resource costs, emphasizing the crucial need for efficient cost management and resource optimization [7].

In particular, given this scenario, optimizing servers and ensuring swift packet processing are imperative to effectively increase the client capacity per server. Performance optimization and efficient network communication play an important role in supporting smooth interactions between large groups of users, which will play an important role in the successful commercialization of the metaverse platform.

By overcoming these challenges, the metaverse will provide users an innovative experience where the boundaries between reality and the virtual world converge.

B. *VPP with DPDK*

VPP and DPDK enhances performance in the field of network packet processing. VPP (Vector Packet Processor) is a high-performance packet processing stack that offers L2-L4 functionality and is designed as a scalable framework for vector packet processing technology [8]. It optimizes packet processing using the computer processor's instruction cache. DPDK (Data Plane Development Kit) bypasses the kernel, eliminating context switches, and employs Poll Mode Driver to reside on CPU cores, continuously monitoring incoming packets from external sources. When packets arrive at the Network Interface Card (NIC), DPDK processes them immediately, reducing interrupt overhead. This allows for high-performance network applications and accelerated packet processing in user space [9].

C. *Shared Memory*

Shared memory refers to memory that can be accessed by two or more processes and facilitates data sharing and communication between server processes. Typically, to share data in distributed servers, data is stored in a database and synchronized. However, accessing a database involves communication with the database server over the network, which introduces a significant amount of overhead. Therefore, in this paper, shared memory is used instead of a database to reduce server overhead and enhance response times [10, 11].

III. Proposed Solution

The VHCT architecture we propose is described in Fig. 1. To address the scaling issues of large-scale metaverse services, we use VPP and DPDK. They ensure that all network traffic generated between clients and servers is forwarded through the virtual router. It optimizes network packet processing by bypassing the kernel, and improves data transmission speeds. Not only VPP and DPDK, but also the hybrid protocol ensures high reliability and speed in the large-scale metaverse environment.

A. *Virtual router environment*

The network structure we propose is described in Fig. 2. Within the virtual router (vRouter), there are two VPP instances - VPP1 and VPP2 - communicating via Linux sockets. When a client sends a packet to the server, the vRouter's DPDK intercepts the packet, and VPP1 processes up to 256 packets in parallel. VPP1 then sends the packets to VPP2 through a socket, and VPP2 forwards them to the server. When the server needs to send packets to the client, the process operates in the reverse order. Both vpp instances act as virtual routers for client and server, respectively, and operate independently, so they enable faster and safer packet processing.

B. *Limitations of a Single TCP Server*

In the large metaverse platform using only TCP servers, latency is usually too high, so the metaverse platform cannot be serviced when responding to a client's request. The main reason for high-latency is the 3-way handshake method to ensure stable data transmission in TCP communication. This is particularly suitable for login and entry/exit packets where reliability is important, but these safety rules slow packet processing speed, and the server experiences high-latency in

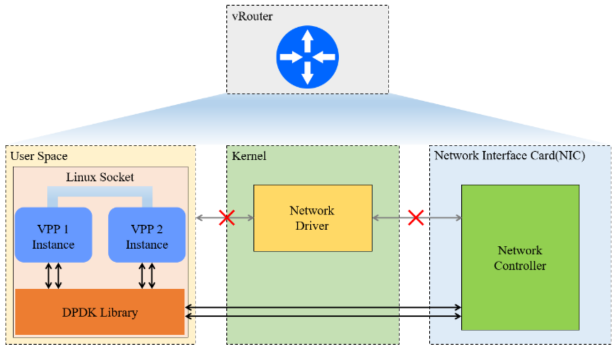

Fig. 2. Network Architecture Diagram using VPP with DPDK

processing packets received from clients. Therefore, it is better not to use TCP-only servers except for data communication where reliability is important.

C. Limitations of a Single UDP Server

Fast packet processing is essential for large-scale metaverse platform, and TCP-only servers are not appropriate for large-scale metaverce platform due to high latency. Instead, UDP is more suitable as a server protocol because UDP is much faster than TCP. However, UDP communication does not use a 3-way handshake method, so it does not guarantee data reliability. This is appropriate for voice calls and real-time video transmissions that require real-time communication, but can cause data loss or damage for important data transmission.

A 3-way handshake algorithm should be applied when communicating important data, such as client entry/exit, to operate all protocols of the metaverse platform server fixed to UDP. However, UDP servers also have some problem. Several distributed servers perform 3-way handshake logic to avoid losing packets such as entry/exit. As a result, each server determines and manages the client's state information, which can lead to a synchronization problem in which the client's state information between servers does not match. The large-scale metaverse platform also needs Algorithms for data consistency between another servers. Therefore, it is preferable that data shared between servers is managed by one server.

So, the connection management server that manages login and entry/exit uses TCP, and the motion synchronization server, voice data server, and video data server use UDP. When logged in, all servers transmit data received from clients from TCP-based servers to UDP-based servers through shared memory to maintain the same data and state.

D. Hybrid Protocol using IPC

When a client accesses the metaverse space as shown in Fig. 3, the client's data is transmitted to the TCP-based connection management server. When the login is successful, the connection management server delivers the received data to all UDP servers through shared memory and sends a response for the login request to the client. After the client receives a login success response, the client communicates in real time with other clients in the metaverse space. In this case, real-time communication between clients is performed with UDP-based motion synchronization, voice data, and video data servers. If a client terminates the connection, the TCP-based connection management server detects a disconnection, processes the client, and transmits the exit information to all

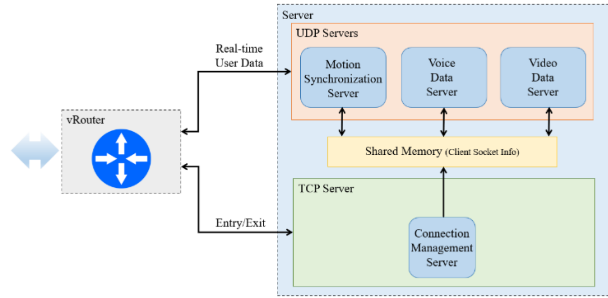

Fig. 3. Hybrid Protocol Server Architecture

UDP servers through shared memory. The UDP server leaves the client based on the received data.

By ensuring client entry/exit packets through TCP-based servers, all clients in the metaverse space can fully synchronize each client's state, and communication using shared memory between servers can maintain the same data and improve speed over using databases. Speed can be improved by using UDP to process real-time generated data rather than login/logout and entry/exit packets.

IV. Design Of Experiment

To evaluate the server structure of the large-scale metaverse service, we developed a test client program and measured its performance. The clients of the actual metaverse platform consist of two users - user 1 and user 2 -. User 1 generates 60 frames of video data per second and transmits it to the server, which averages 1,800 bytes/sec in size. User 2 generates location data and voice data by 32 bytes/sec and 100 bytes/sec, respectively, and transmits it to the server. That is, the size of the data received by the server is about 2,000 bytes/sec, and the test client transmits 2,000 bytes/sec of data per second to the server.

A. Network Latency

To measure the delay time, the time at which data is transmitted to the packet was recorded. We use the following heuristic to estimate latency.

$$Latency = t_c - t_p \quad (1)$$

$t_c$is the current time of the test client, and t_p is the time stored in the packets. Assuming each test client program as one user, the delay time was measured for 10 minutes after all clients accessed the server, and the average of all delay times was determined as the final delay time.

V. Performance Analysis

The experimental results regarding communication latency based on the number of clients in the TCP, hybrid protocol, and VPP-based hybrid protocol environment, using the architecture shown in Fig. 1, Fig. 2 and Fig. 3, are illustrated in Fig. 4.

In Fig. 3, the vertical rise of each graph for each experiment signifies a huge increase in latency. When the number of clients surpasses this point, it indicates the emergence of server bottlenecks, rendering service provision unfeasible.

Comparing TCP and the hybrid protocol, as the number of clients increases, the difference in latency between the two becomes more pronounced. When 75 clients are connected, the latencies are approximately 0.35 seconds and 0.2 seconds

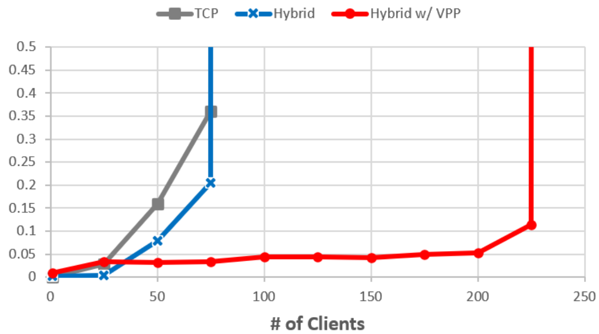

Fig. 4. Communication latency based on the number of clients in a TCP, ybrid protocol, and VPP-based hybrid protocol environment

for TCP and the hybrid protocol, respectively, making the hybrid protocol about 1.5 times faster. However, both methods experience a sharp increase in latency when the number of clients exceeds 100. This is attributed to the bottleneck caused by the sequential processing of network packets on the server, which suggests that the server architecture alone cannot address this issue.

On the other hand, in the case of the VPP-based hybrid protocol, even with 225 clients, the latency remains around 0.1 seconds, allowing it to accommodate a significantly larger number of clients compared to the previous two methods.

However, for client numbers less than 25, it can be observed that the VPP-based method exhibits longer latency than the other two methods. This seems to be due to internal VPP operations for packet parallel processing. Therefore, when the number of clients is small, using only the hybrid protocol without VPP appears to be a better choice.

VI. Conclusion

We proposed a hybrid protocol and VPP-based routing technology to support large-scale metaverse environments in this paper. The hybrid protocol introduced in this paper allowed us to reduce communication latency compared to traditional TCP servers, and it laid the foundation for distributed servers through shared memory. Furthermore, the VPP-based routing technology improved server communication bottlenecks, significantly increasing the number of clients that can connect simultaneously. Our study sufficiently demonstrated the feasibility of implementing large-scale metaverse platform that can accommodate more than twice the number of concurrent users compared to existing metaverse services.

In the future, to further enhance server performance, applying distributed servers using the hybrid protocol to a cloud environment for effective load balancing is considered advantageous for server performance improvement. Additionally, based on the experimental results, it is better to add an automatic switching function that applies VPP depending on the number of clients. To achieve this, further experiments will be conducted to determine the optimal client number for applying VPP and set it as a threshold. When the threshold is exceeded, VPP will be applied automatically to optimize performance. Therefore, we propose a cloud-based metaverse server platform with VPP switching technology.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2023-RS-2023-00258649) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

References

[1] S. -M. Park and Y. -G. Kim, "A Metaverse: Taxonomy, Components, Applications, and Open Challenges," IEEE Access, vol. 10, pp. 4209-4251, 2022

[2] M. Xu et al., "A Full Dive Into Realizing the Edge-Enabled Metaverse: Visions, Enabling Technologies, and Challenges," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 656-700, Firstquarter 2023

[3] R. Cheng, N. Wu, S. Chen and B. Han, "Will Metaverse Be NextG Internet? Vision, Hype, and Reality," IEEE Network, vol. 36, no. 5, pp. 197-204, September/October 2022

[4] Feifei Shi, Huansheng Ning, Xiaohong Zhang, Rongyang Li, Qiaohui Tian, Shiming Zhang, Yuanyuan Zheng, Yudong Guo, Mahmoud Daneshmand, "A new technology perspective of the Metaverse: Its essence, framework and challenges", Digital Communications and Networks, 2023.

[5] L. Chang et al., "6G-Enabled Edge AI for Metaverse: Challenges, Methods, and Future Research Directions," in Journal of Communications and Information Networks, vol. 7, no. 2, pp. 107-121, June 2022

[6] Sun Teh-Jen, Jeong Seung-Woo, Jeong Hong-Ju, Hong Choong Seon, Park Seong-Bae, Huh Eui-nam "MetaOps: Metaverse Operations for Large-Scale Metaverse Services on Distributed Cloud", 2023

[7] Y. Cai, J. Llorca, A. M. Tulino and A. F. Molisch, "Compute- and Data-Intensive Networks: The Key to the Metaverse," 2022 1st International Conference on 6G Networking (6GNet), Paris, France, pp. 1-8, 2022

[8] D. Barach et al., "High-Speed Software Data Plane via Vectorized Packet Processing," in IEEE Communications Magazine, vol. 56, no. 12, pp. 97–103, 2018

[9] Jeong Gab-Joong, Choi Kang-Il. Study on the Implementation of a Virtual Switch using Intel DPDK. The Journal of the Korea Institute of Electronic Communication Sciences, 2015

[10] Venkataraman, Aditya and Kishore Kumar Jagadeesha. "Evaluation of Inter-Process Communication Mechanisms." 2015

[11] Patricia K. Immich, Ravi S. Bhagavatula, Dr. Ravi Pendse, "Performance analysis of five interprocess communication mechanisms across UNIX operating systems", Journal of Systems and Software, Volume 68, Issue 1, 27-43, 2003