uCPE Edge Computing for Real-Time Interactions in the Metaverse

Lichuan Ren[1], Jinyi Huang[1], Jamie Wang[1], Grace Zhu[1], Daniel Wang[1],
Jordan Lambert[2], Yanyan Zheng[2], Zheng Li[2], Sandra Liu[2], Vivian Ye[2], Alan Yen[2],
Larry Zhou[2], David Mahar[2], John Gibbons[2], and Michael Satterlee[2]

[1]AT&T Labs Internship Program, Middletown, NJ
[2]AT&T Labs, Middletown, NJ

***Abstract*—The metaverse is an integration and intertwining of the physical and virtual worlds. To create an immersive metaverse, it is important to meet the network requirements needed to effectively simulate real-time interactions. This paper discusses some of the issues involved in implementing the metaverse, and how many of them can be solved with uCPE edge computing.**

***Index Terms*—Metaverse, uCPE, Edge Computing, Private 5G, Virtual and Augmented Reality, Artificial Intelligence, Avatar, Digital Twin**

I. Introduction

The metaverse is a connected physical-virtual world, facilitated by the convergence between the Internet and Web technologies, and Extended Reality (XR) [1]. Many technical challenges remain to be resolved in building a truly interactive and immersive metaverse. First, the digitization of the virtual world is not fully automatic, and avatar representation is not realistic. Second, controller-based interaction approaches cannot provide an immersive user experience in the metaverse. Most notably, existing technologies are unable to meet the computing requirement of large-scale multi-users in the metaverse.

Current head-mounted displays (HMDs) are offered in either standalone or wired models. Standalone models contain limited computing power comparable to a smartphone, while wired versions require a direct tether to a powerful and often expensive gaming machine in order to render graphics to the device. Due to the limited computing power of current HMDs, the seamless and smooth rendering for the metaverse remains to be one of the key challenges. Moreover, the complexity of both rendering and network transportation scales to $\mathcal{O}(N)$ for N users in the metaverse.

Cloud computing [1] [2] is a possible solution for combining the power of high-end graphics hardware and the ease of use of wireless standalone HMDs, however this creates the problems of high latency and low throughput. The slightest bit of latency is enough to detach a user from an immersive virtual reality experience, due to its reliance on realtime head and hand tracking.

Edge computing is a promising technology for the metaverse because of its low latency, low jitter, and high bandwidth features. In this paper, we propose a framework to apply edge computing in real-time interaction in the metaverse.

II. Design

In our design, we utilized many components to implement a functional and immersive metaverse. Firstly, a virtual world has been created with computer-aided design (CAD) software. Then, multiple RGB-D sensors have been set up for 3D data acquisition to create realistic avatars. Meanwhile, the user's hand gesture is recognized using spatial and temporal feature and applied to interact with the virtual world. Lastly, edge computing is employed for real-time rendering.

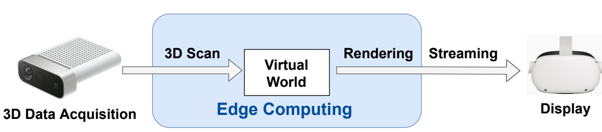

Fig. 1. 3D Data-Flow

A. Static Model

In order to construct a baseline static model, we used CAD software to model real life buildings to establish a backdrop for contextualisation of our project. By referencing building blueprints and utilizing 3D modeling software such as Rhinoceros 7 and Blender, we were able to construct detailed to-scale models. These models allow users to explore and freely navigate the buildings in the virtual world at their own pace, according to their interests. In addition, we utilized photogrammetry as a quick and effective way to produce high quality models of complex objects, such as statues. We used Unreal Engine's Twinmotion as a means to render and contextualize the built environment with ray tracing technology.

B. Dynamic Model

1) Dynamic Avatar Creation: To create realistic avatar representations in the metaverse, real-time photogrammetry using multiview RGB-D images has been utilized.

In our system, multiple Azure Kinect sensors are set up for 3D data acquisition. In order to reconstruct the 3D human body using RGB-D images, camera calibration and pose estimation data is required. Since the camera poses are fixed, the camera pose estimation, i.e. extrinsic calibration, has been implemented offline. Extrinsic calibration consists of two steps. First, each camera pose is roughly estimated by finding a 3D rigid transformation [3] from the reference coordinate system and the camera coordinate system. Then, the transformation from the subordinate camera to the master device is refined through the colored iterative closest point (ICP) [4].

We utilize Microsoft Azure Kinect Body Tracking SDK for human instance segmentation. 3D Point clouds are registered using the segmented RGB-D images.

2) Vision-Based Hand Gesture Recognition: Current methods of interaction in the metaverse are based on controllers, which are not very intuitive or user-friendly. Hand gestures are one of the most widely used forms of communication in the physical world. Hence, we detect and recognize hand gesture to interact within the metaverse. The hand segmentation can be acquired from Azure Kinect Body Tracking SDK. In order to classify hand gesture, heuristic-based methods [5] and NN-based methods [6] can be applied.

C. Real-Time Rendering on Edge Computing

Real-time 3D graphic rendering requires huge computational power, which is carried out utilizing edge computing. Our 3D model is created by merging the static model and real-time scanned dynamic model. The edge computing hardware described in the next section can perform the computation required to render our model, and deliver it to users.

III. Implementation

The AT&T Universal Customer Premises Equipment (uCPE) [7] with Vyatta network operating system (VNOS) is the hardware of choice for our edge computing solution because of the following reasons: 1) plug-and-play capabilities, 2) cloud-native management and control, 3) ease of use, 4) affordability, 5) built-in 10G switch, and 6) flexible add-on 3rd party modules.

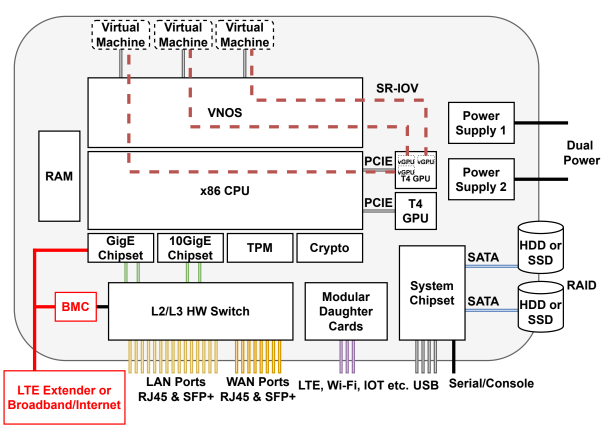

Fig. 2. Edge Computing Enhancement for AT&T uCPE

Fig. 2 illustrates the hardware design of edge computing enhanced uCPE. Combined with the base OS, VNOS, uCPE can support routing, firewall, segment routing, and other VNFs. The edge computing system makes use of an enhanced version of AT&T's large uCPE Whitebox specification, which adds two NVIDIA Tesla T4 GPUs to support GPU virtualization. Utilizing SR-IOV PCI passthrough, we provide virtual machines with virtual GPUs (vGPUs) to render virtual reality content, as well as support other metaverse computation.

A possible use case of edge computing enhanced uCPE is to apply it on private 5G networks. As shown in Fig. 3, edge computing enhanced uCPE can be inserted into private 5G network by slicing data path between DU and CU to achieve better QoS for end users. It can also be stacked to dynamically expand edge computing capacity.

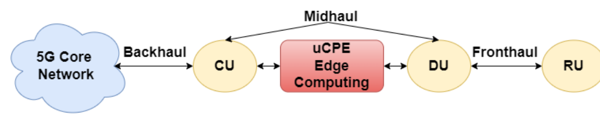

Fig. 3. 5G with Edge Computing

Fig. 4 illustrates the process of real-time 3D rendering on our edge computing device. Each user connecting with an HMD can be assigned an individual virtual machine with a corresponding vGPU and our virtual reality streaming software stack. The content rendered within the virtual machine is streamed to the user's HMD to provide high-end rendering quality.

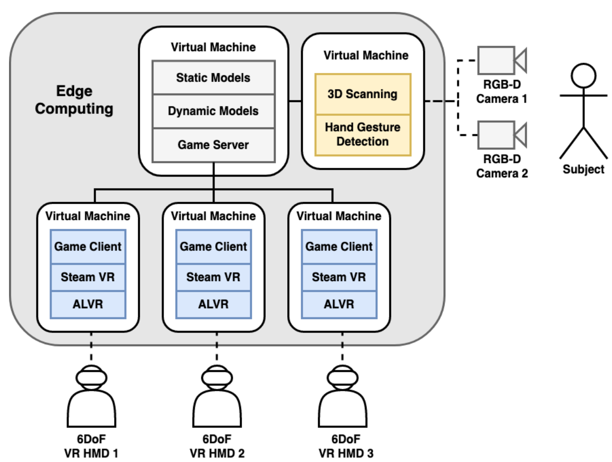

Fig. 4. Real-Time 3D Rendering on Edge Computing

With the design of edge computing enhancement for AT&T uCPE, we are able to achieve real-time rendering of 3D models within 200ms.

IV. Conclusion

There are many aspects to consider when building an immersive metaverse, but the main concern is the necessity to render the virtual world and avatars in real time to give users a more compelling experience. We propose a feasible approach to achieve a low latency rating is to use an edge computing enhancement for AT&T uCPE.

References

[1] L.-H. Lee, T. Braud, P. Zhou, L. Wang, D. Xu, Z. Lin, A. Kumar, C. Bermejo, and P. Hui, "All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda," 2021. [Online]. Available: https://arxiv.org/abs/2110.05352

[2] H. Ning, H. Wang, Y. Lin, W. Wang, S. Dhelim, F. Farha, J. Ding, and M. Daneshmand, "A survey on metaverse: the state-of-the-art, technologies, applications, and challenges," *CoRR*, vol. abs/2111.09673, 2021. [Online]. Available: https://arxiv.org/abs/2111.09673

[3] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine Vision and Applications*, vol. 9, pp. 272–290, 1997.

[4] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 143–152.

[5] M. Perimal, S. Basah, M. Safar, and H. Yazid, "Hand-gesture recognition-algorithm based on finger counting," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-13, pp. 19–24, 2018.

[6] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, "Construct dynamic graphs for hand gesture recognition via spatial-temporal attention," 2019. [Online]. Available: https://arxiv.org/abs/1907.08871

[7] "AT&T uCPE Specification at Open Compute Project," https://www.opencompute.org/files/ATT-OCP-Specs-April-2018-002.pdf, accessed: 2022-07-01.