LEVERAGING THE CURRENT TRENDS AND RESEARCH IN CROSS-PLATFORM MULTIVERSE FOR ENHANCING THE EFFICACY OF DATABASE MANAGEMENT SYSTEM

Shreya Bhardwaj

Panjab University, Chandigarh, India

ABSTRACT

The extended reality (XR) community is growing quickly and helping to develop the metaverse industry. A group of internet giants and individual developers have been drawn to this to create their metaverse products. A sub-metaverse could be considered for each XR community even though no such system can cross all these metaverses, which goes against the original idea of the metaverse: connecting each virtual world to an entire universe.

This paper proposes a cross-platform metaverse data management system (CMDMS) to address the issue and enable users to use their space and profile on multiple metaverse platforms.

INTRODUCTION

One of the hopes for the future is the creation of virtual worlds or the metaverse using extended reality technology. An extended reality platform for user-generated content and a development interface that conceals specific details of underlying technologies are required for extended reality technology to be widely used. In today's extended reality (XR) applications, developers must be familiar with user requirements and develop sophisticated software systems with features like memory management, multiple processes, information transfer and synchronization, and various process synchronization techniques. Developers must address low-level issues: Drivers for specific XR input or output devices, such as those that generate multiple stereo views, etc.

Users can call object models from an inventory to construct a vast virtual reality space.

Currently, various platforms are utilized for XR development; however, almost all are solely utilized by service programmers from a technology development perspective. Visual editing, a detailed attribute editor, and a dynamic game preview are all features of the current mainstream XR development platform Unity3D, for example. The Unity community provides talented developers with various scenarios and components. However, developers who use these components must still be proficient in one or two programming languages, such as C#, JavaScript, and others. This might be more user-friendly for people who don't know much about programming.

A platform for building virtual worlds that is simple to use and learn for people who don't know much about programming is needed.

A cross-platform metaverse data management system (CMDMS) is proposed in this paper. CMDMS creates plugins for various game development platforms, including Unity and Unreal, to enable dynamic metadata transfer between games developed on different platforms. It should be possible to use the platform across platforms and store user data. This enables users to achieve the effect of cross-platform metaverse data transfer and provides solutions for future metaverse world communication.

The idea for a metaverse data management system that works across platforms has already been mentioned. The article's second section will introduce the industry's current efforts and accomplishments and its current issues, and its third section will propose solutions.

RELATED WORK

The phrase "a universe beyond the physical world" refers to the metaverse. However, this concept has evolved over time to mean a vast network of multiple individual virtual worlds created by companies like Meta and Google, college students or researchers, or brilliant individual developers [2]. They have sufficient funds to support their growth thanks to the vision of well-known businesses. They dream of being at the forefront of the next fashion trend with their product because they love it so much. Horizon Worlds, the metaverse defined by Facebook's umbrella company meta, which just changed its name a year ago1, is an excellent example. It encourages users to construct their worlds with great talent and imagination.

Regarding Roblox, the metaverse concept inspired the initial company stock. In 20212, it became officially listed on the New York Stock Exchange. Bytedance invested significantly in August of that same year to acquire the virtual reality startup, Pico. Also, some people want to work in this field. The powerful rivals of meta are the enterprise metaverse from Microsoft and the Omniverse from Nvidia [3].

The metaverse concept has also been introduced into universities, including the Chinese The University of Hong Kong, in addition to these well-known businesses. CUHKSZ demonstrated a fantastic metaverse prototype on their campus and connected the students' actions to the metaverse [4]. The metaverse is being made accessible to the general public alongside numerous other educational institutions and personal developers [5].

In addition, a lot of research has been done on this topic, and several survey papers have been written about the metaverse. Xu and others 6] suggested conducting a comprehensive investigation of the edge-enabled metaverse from a computing, blockchain, networking, and communication standpoint. The goal was to help the metaverse networks, like 5G, cloud computing, and edge computing, run smoothly.

Yang and others 7] looked into how could use AI and blockchain technologies to build a metaverse in the future. The metaverse ecology is created by combining several digital technologies into blockchain technology, which is the core technology of the metaverse. In August 2021, Nvidia announced plans to develop Omniverse, the first platform for virtual collaboration and simulation. Omniverse Nucleus, a database engine that enables multiple users to connect and jointly create a scene [8], is one of the three key components. In addition, this paper uses MongoDB as the database's backbone, and Radha et al. MongoDB, a proposed database, has an intuitive user interface that makes it simple for users to retrieve data.

However, as was previously mentioned, this paper aims to achieve cross-platform data management and connect each separate metaverse. This paper proposes a management system that enables users to use their profile and space across multiple metaverse platforms to accomplish this objective.

SOLUTION

The CMDMS is proposed in this paper to connect disparate metaverses. Through the collection of essential information about users' virtual world construction (such as components, coordinates, etc.) and the extraction of common characteristics of metaverse data storage, The CMDMS can gather useful data on accessible virtual spaces constructed by multiple users from various metaverse platforms from the corporations whose metaverse production is successful and who are willing to share metaverse data. The management system creates the data framework necessary to link various metaverses in this manner.

The figure depicts the three components of the CMDMS.

1. The available front-end plugin handles the model upload and download, and a server system handles the data storage, transformation, and arrangement. The construction diagram depicts the user level where the plugin is situated. Can use several metaverse development software call interfaces at the user level to upload and download data. The business and data modules are combined into the server system, as depicted in the construction figure. Model data from various platforms are accepted at the business level, where they are unpacked, analyzed, and stored at the data level. MongoDB, an open-source, cross-platform document-oriented database program that uses JSON-like documents and optional schemas, keeps models up to date at the data level.

System workflow

Upload

The developed universal plugin supports the upload procedure (this prototype currently supports the Unity and Unreal engine). Uploading is primarily broken down into three steps. First, upload the local FilmBoX (FBX) file describing the model, which should include the model's material and map files. Most modelling software and game engines are compatible

with the FBX format, a general 3D model file that includes animation, material property, map, bone animation, light, and camera. The plugin will then serialize the model-related files into byte streams. Second, upload the model's relevant attributes data (the relevant attributes are listed in the table below). Enter the description, model name, type, and uploader names. The back-end ought to determine the file size and upload time. The Assimp open-source library (Open Asset Import Library that supports reading various model resources3) extracts additional attributes from FBX model files. The plugin will serialize these attributes and combine them with the serialized byte stream of the model-related files, just like the first step did. Thirdly, the combined byte stream is compressed into resource-compressed packets using minizip4 as the underlying compression method. These packets are used as a unified resource processing standard between platforms and reduce the file size.

Attributes	Storage type & units
User-defined attributes	
Type	String
Model name	String
Uploader name	String
Description	String
Fixed attributes	
Upload time	Date
File size	MB
Model file attributes	
Volume	V^3
Number of model faces	Binary
Vertex coordinates	integer
Mesh	Binary
Skin	Binary
Texture	Binary
Material	Binary
Root	Binary
Animation layer	Binary
Animation Curve	Binary
Animation Curve Node	Binary
Cluster	Binary

Tips: Table 1 contains a list of existing attributes.

Thanks to the plugin, the goal models can be downloaded and loaded into the system.

2. Download

The plugin and server support the download procedure. The plugin side provides a display function similar to the resource store. Developers can browse the models stored in the CMDMS on the client side and filter specific models by model name or model type thanks to this visual display of the stored model preview and attached model information. The service level supports searching with model types and fuzzy and precise queries for model names when selecting the model and confirming the download.

The plugin downloads the model from our server and loads it into the development environment after users select the goal model they wish to load. The resource-compressed

package is then stored locally after being returned from the back end. Developers can alter the compressed resource package's local storage path.

3. Dynamic loading

The following are the primary steps of dynamic loading:

1) It converts the downloaded resource compressed package file into a byte stream by decompressing it with the assistance of mini zip. Its goal is to extract segments from the model, material, map, and model-related data files. As an entry in a zip file, each segment will be sealed;

2) To store the information contained in each zip file entry, convert each entry into a memory stream and write it into an asset loader context;

3) Utilizing the open-source library Assimp, examine the Asset Loader Context. Locate the root of the model set in the modelling software beforehand. This requires reading the model portion of the memory stream. The reader then separates the model's substream to locate the children's components from the root. A 1:1 mapping, including the size, can thus establish the relative position of each model component.

The model's mesh information is gradually loaded from the root node using the LoadMesh method following the mapping node correspondence, and then the model's shape is restored.

4) Utilizing the mapping function found within the "Asset Loader Context," read the memory stream to locate the data for the materials and textures. This method can map the material and texture to the model.

We will restore a complete model to the game engine at this point.

The model-related data in the zip file entry will be deserialized into specific attributes and encapsulated in the plugin's data result. Developers can choose whether or not to use the data result's output for game development.

System features

(1) Support for model search The CMDMS's database of choice is MongoDB.

Analytic functions do not require complex joins, and it supports dynamic queries on documents, facilitating deep queries, in contrast to the conventional RDB (Relational Database) like Oracle or MySQL.

Most importantly, it can store many different kinds of data in the simplest way possible, and lets embedded documents make nested structures. Each of these features supports CMDMS in a way that is both efficient and effective.

A fast data search is supported when creating an index for the model's creator's id, its type, and, finally, the full-text index for the model's name. This is quick and can provide users with sufficiently detailed search results [10, 11].

(2) Data security Every CMDMS user must log in with the account they use on multiple platforms. Users will be denied access if they attempt to upload or download data using an unregistered account. In addition, CMDMS routinely copies processes and prepares a distributed database system for data backup. Additionally, it has an open-source private mechanism that enables users to share or safeguard their work. The system's compatible services can effectively assist users in anticipating the workload and potential risks associated with the database migration.

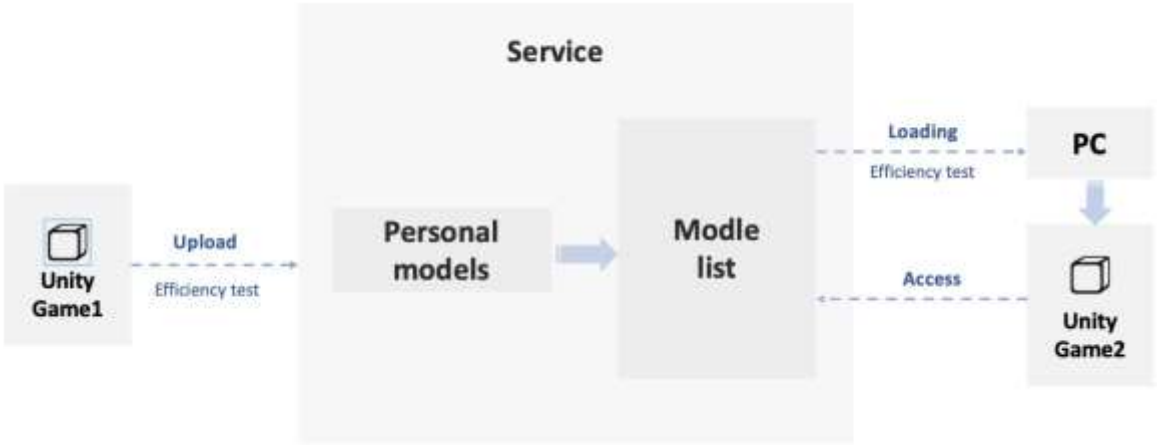

Fig. 1. Detailed CMDMS procedure between two terminals

A firewall is an essential barrier between the front and back end. A firewall should serve as middleware for the front-end system's input and output, filtering the model and information. The filter first ensures that the user has permission to upload or download models and whether the model's destination or download request is legal. Flow control is yet another crucial function of the firewall.

The firewall would limit the upload and download requests at the same time, rejecting requests from particular IPs or MAC addresses if they exceed the limit on the number of requests that can process in a given time frame.

(3) Problem-locating functions and data change prediction systems are designed due to the labour-intensive and time-consuming nature of locating and resolving system issues.

Analyses of family ties: beginning with an entity and tracing its processing back to the interface for the data source. The relationship between the data links is shown graphically to show how an entity is processed. This makes data problem analysis easier, saves money on labour, and locates the issue in a single step. It can provide data at the field level, working its way up to

locate more complex data for analysis on particular fields in particular tables and then solving the issue.

Analyses of impact: CMDMS understands the relationships between metadata and provides the capability to analyze the data flow and demonstrate the relationships between the metadata, with the current analysis object serving as the starting point and the endpoint being the most affected subobject.

As a result, the graph's flow of data will examine which data linked to this source have changed due to changes in the current metadata.

USAGE SCENARIO

The CMDMS divides the usage conditions into two aspects from a macro perspective: users on an individual and institutional level.

The following are the conditions for individual users:

1. Zero foundations for the plugin's data migration implementation.

The CMDMS's data migration function allows individual users to use a VR plugin incompatible with their own VR devices. It does not require individual users to have any prior programming experience.

2. System security is ensured by data backup.

Users can back up their data in two different ways with this CMDMS.

Data can be stored on local disks or backed up to the local system's cloud server by users on various platforms. Users can use the CMDMS to back up their data to their devices if a device failure or interruption results in data loss. The CMDMS will alert individual users before data migration operations to the amount of data to be transferred and any potential risks. This will enable individual users to intuitively assist individual users in controlling cross-platform operation risks and safeguarding their devices.

The following are the conditions for institutional users:

1. Enterprise user costs are reduced through whole-process data migration.

Institutional users can use the CMDMS's data migration solution to migrate all relevant data to the required devices. As a result, institutional users' VR plugins can be used on VR devices that did not originally support it. The CMDMS includes Unity support, making it possible to transfer data between Unity editors. This plan saves institutional users more time and money than the traditional operation method, and it makes it easier for institutional users to increase the platform's capabilities quickly.

2. Provide a search engine to boost productivity at work.

Through the CMDMS's search engine, institutional users can quickly locate the necessary business index data, making retrieval more accurate, quick, and professional.

Institutional users' work efficiency is enhanced because they no longer have to perform complex search operations, as is the case with the traditional operation method.

3. improves data security through the provision of risk management functions.

The CMDMS's database classification function allows institutional users to classify data based on their security policies. The CMDMS's historical data search function allows the retracing of previously used historical data to better meet institutional users' stringent data security requirements and improve institutional users' risk resistance.

CONCLUSION

The metaverse will likely become increasingly commercialized and available to more people. One possibility for achieving the metaverse's decentralized structure is CMDMS. CMDMS has demonstrated the inter-transmission of Unity and Unreal-based game components.

CMDMS allows Unity game components to be uploaded and downloaded by users. The results of the efficiency test and the restoration test are currently favourable, and more types of tests will be carried out.

REFERENCES

[1] Bierbaum A, Just C, Hartling P, et al, (2001), VR Juggler: A virtual platform for virtual reality application development, in *2001 Proceedings IEEE Virtual Reality*, pp. 89-96.

[2] Dionisio J D N, III W G B, Gilbert R, (2013), 3D virtual worlds and the metaverse: Current status and future possibilities," *2013 ACM Computing Surveys (CSUR)*, pp. 1-38.

[3] Kim J, (2021), Advertising in the Metaverse: Research Agenda, *Journal of Interactive Advertising*, pp. 141-144.

[4] Duan H, Li J, Fan S, et al, (2021), Metaverse for social good: A university campus prototype" in *2021 Proceedings of the 29th ACM International Conference on Multimedia*, pp. 153-161.

[5] Radha, T., James, P., Simi, S., Davis, S.P., Nazeem, P.A., Shylaja, M.R. and Deepu Mathew, (2018) MangoDB: A database of mango varieties and landraces of the Indian subcontinent. *Current Science*, vol.114, no.3, pp.2022-2025.

[6] Chickerur, Satyadhyan, Anoop Goudar, and Ankita Kinnerkar. (2015), Comparison of relational database with document-oriented database (mongodb) for big data applications. *2015 8th International Conference on Advanced Software Engineering Its Applications (ASEA).* IEEE.

[7] Sharma, Monika, Vishal Deep Sharma, and Mahesh M. Bundele. (2018), Performance analysis of RDBMS and no SQL databases: PostgreSQL, MongoDB and Neo4j." 2018 3rd *International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE).* IEEE, 2018. 150