J. Electrical Systems 19-4 (2023): 84-101

[1*] Yu Dai

3D Interior Design System Model Based on Computer Virtual Reality Technology

Abstract: - Globally, data volume is increases exponentially with increase in the proliferation with Cloud Computing. MapReduce is emerged as the prominent solution for the unprecedented growth in the efficient manner as it process both structured and unstructured data. The dynamic landscape of Virtual Reality has seen a significant shift towards technology-driven approaches, with data analytics and personalized learning becoming increasingly important. This paper introduces an innovative framework that leverages the power of Hadoop and MapReduce to elevate 3D virtual reality experiences within diverse VR Cloud settings. This paper presents the development of an efficient Cache-Based MapReduce framework (CMF) where Cache algorithms are effectively used to process queries on large-scale cloud-based data. The Hadoop System processes data in single-node Hadoop Clusters (Pseudo-distributed) as well as heterogeneous Hadoop Clusters (fully distributed nodes) within Amazon Web Services (AWS). The Hadoop System process the data in the single node Hadoop Cluster (Pseudo-distributed) heterogeneous Hadoop Cluster (fully distributed node) in the Amazon Web Services (AWS). The experimental analysis is evaluated for the SmallGutenberg and LargeGutenberg database. The developed model achieves the average reduction in job of 48.01% with reduction in execution time of 51.99%. The CMF of 7-node, 9-node, 15-node and 20-node reduction in execution time is measured as 49.91%, 51.38%, 54.71% and 45.29% respectively.

Keywords: Hadoop, MapReduce, 3D virtual reality, Big Data, Data Analytics, Personalized Learning, VR Cloud Technology, Text Processing, 3D VR Analytics, Content Recommendation.

I. INTRODUCTION

Interior design is a multifaceted discipline that focuses on enhancing the aesthetics, functionality, and overall atmosphere of interior spaces. It encompasses the art of arranging and decorating interiors to create environments that cater to the needs and preferences of the occupants while also adhering to practical considerations. Interior designers blend creativity with technical expertise to select color schemes, furniture, lighting, and decor that harmonize with the architectural elements of a space [1]. They pay close attention to layout, ergonomics, and the use of available space to optimize both visual appeal and functionality. Whether in residential, commercial, or institutional settings, interior design plays a crucial role in transforming empty spaces into inviting and functional areas that reflect the personality and purpose of the space. Interior design has been significantly transformed by the integration of virtual reality (VR) technology [2]. VR allows designers to create immersive, three-dimensional environments, providing clients and designers with an unparalleled tool for visualization and collaboration. Through VR, designers and clients can walk through spaces before they are physically constructed, enabling a more in-depth understanding of the design concept and an opportunity to make informed decisions about layout, materials, and furnishings. This technology enhances the design process by allowing real-time adjustments and fine-tuning of elements such as lighting, color schemes, and furniture arrangements, resulting in more precise and efficient design outcomes [3]. Moreover, it facilitates global collaboration, as clients and designers can interact and review designs from anywhere in the world, making interior design more accessible and dynamic. In essence, the incorporation of virtual reality has revolutionized interior design, offering an innovative and interactive approach that elevates the design experience to a new level of creativity and functionality [4]. The role of the Internet of Things (IoT) in interior design, particularly when combined with virtual reality (VR), is a game-changer in creating intelligent and immersive living and working spaces. IoT devices, such as smart lighting, thermostats, and security systems, can be seamlessly integrated into VR simulations of interior spaces [5]. This integration allows designers and clients to experience not just the visual aspects but also the functionality and automation of the designed space. For example, in a VR simulation, one can adjust the lighting and temperature, see how smart appliances and devices interact, and even anticipate energy consumption [6]. This not only enhances the realism of the virtual experience but also aids in making informed decisions about IoT device placement, programming, and their overall impact on the design. Furthermore, IoT sensors can collect data from physical spaces and feed it into the VR environment, offering valuable insights for design optimization. For instance, sensors can monitor occupancy patterns, air quality, and energy usage, allowing designers to create

[1]School of Design, Chongqing Industry Polytechnic College, Chongqing, 401120, China

*Corresponding author e-mail: TDMS1997@163.com

environments that are not only aesthetically pleasing but also energy-efficient and responsive to the needs of the occupants [7]. Additionally, IoT can assist in the personalization of interior spaces, as data from wearables and smart devices can be incorporated into VR simulations to tailor designs to individual preferences and needs.

In the current era of the digital age, global data has been rising at a very high speed due to the increased use of social networking sites, online shopping, the Internet of Things, the Cloud, smartphones, and other handheld devices [8]. A large volume of digital content is generated continuously through photos, videos, tweets, emails, text, and documents. It includes both structured and unstructured data. Database Management Systems like Oracle, DB2, MS SQL Server were previously used for the storage and processing of data [9]. But in today's scenario, these approaches get failed in handling a large volume of data. Traditional Relational Database Management System (RDBMS) cannot handle large volumes of data because they are designed for steady data retention and have inflexible schemas [10]. The studies on MapReduce framework show that it generates lots of intermediate values which get discarded in the environment after processing is finished [11]. MapReduce is unable to utilize this intermediate data for future use. Secondly, MapReduce does not have the techniques to identify duplicate computations and accelerate job execution. MapReduce is widely used in the analysis and computation of a large amount of data but the processing of the overall execution of the MapReduce task is slow [12]. This is due to intermediate loss of data and duplicate computation processing. The research that is being done in the area of MapReduce query processing largely focuses on the handling of intermediate data produced by Map and Reduce node [13]. MapReduce framework produces a large amount of intermediate data or values during job processing which gets discarded after the job is finished and again this data is processed when required. So, there is a need to hold this data for future re-use and save execution time for duplicate data processing. Caching is the technique that can be used to hold this data for future retrieval. It can be used to enhance the performance of MapReduce by making available the intermediate results of Map and Reduce tasks for the future [14]. It is one of the techniques used to enhance the performance of MapReduce by reducing the cost of I/O operations, query response time, CPU usage, and a load of the server [15]. It is a buffering technique that stores frequently-queried data in temporary memory. It makes data easier to be accessed and reduces workloads for databases. Therefore, these studies have acted as a motivation to develop an efficient technique of query processing using MapReduce through caching technique [16].

Several researchers have improved the performance of Hadoop MapReduce through implementing join operation in MapReduce. The join operation is an important operation for the processing of massive datasets in MapReduce, however, it is known that the join is expensive in terms of costs which makes the algorithms inefficient over the Hadoop framework [17]. Since several researchers have ported their attention to improve the performance of Hadoop through joining, therefore some techniques have been proposed to optimize the costs of the join algorithms, for example, by using the bloom filter technique to filter the inputs and decrease the data transferring in the network between the two sides Map and Reduce. In the cloud environment, different users make requests to the database simultaneously, which causes a heavy load on the database servers. It increases the query processing time of the queries. Caching is one of the solutions that can boost MapReduce performance by reducing the cost of I/O operations, the load of the server, query response time, and CPU usage [18]. In [19] proposed a general query processing scheme that can handle a single query along with sets of queries in an incremental way. They build the indexes in an ordered style, the sorted inverted indexes so that one can perform a quick pruning strategy that rejects unrelated objects. In [20] developed CAVA-Cache-Affinity and Virtualization Aware. It is a resource manager. It is used to compute the cache affinity at the runtime of MapReduce applications. It assigns a top priority to those applications which have high cache affinity. In this way, it manages limited-size distributed in memory caches efficiently. The experimental study exhibits that CAVA shows better performance for big data analytics applications. In [21] proposed a distributed cache to improve the join operation performance in Hadoop. The smaller dataset is saved in the mapper node cache memory. All the lookups are performed by the mapper to generate the final results. Hence, by using the distributed cache data load operations are managed efficiently at the mapper side and there is no need to use the reducer side. In the future, the fully distributed cluster mode can be used to perform such operations. In [22] proposed an In-Memory Cache approach for saving the data in cache memory before being used for processing. The cache size is 256 MB. It can be increased up to 512 MB or 1024 MB. When any Map task arrives, refreshing is done. An extra thread saves the input data block frequently from disk to In-Memory Cache and also evicts the data block based on some suitable cache eviction policy. Instead of transferring the large data blocks from disk to memory, it first checks the cache so that disk I/O operations can be reduced. This paper constructed a framework CMF-based data processing model for the E-infrastructure through MapReduce framework model.

This paper presented effective 3D virtual reality based interior design model with the implementation MapReduce cloud computing environment. The specific contribution of the paper are stated as follows:

1. The paper introduces a Hadoop-based MapReduce framework tailored for 3D virtual reality, addressing the increasing demand for efficient data processing and analysis in the VR Cloud domain. It contributes to the streamlined handling of vast volumes of 3D VR data and VR Cloud content, which is crucial for data-driven VR Cloud approaches.
2. The framework's ability to perform text analysis aids in 3D VR assessment, offering 3D VR valuable insights into interior design. This contribution allows for more accurate and data-driven evaluation of 3D interior design, ultimately improving the quality of virtual reality.
3. The framework empowers 3D VR to conduct in-depth 3D VR performance analytics. It contributes to the identification of learning patterns, strengths, and weaknesses, enabling more targeted and effective 3D interior design strategies. This, in turn, enhances the quality of 3D virtual reality and promotes personalized learning.
4. With leveraging big data and analytics, the paper introduces content recommendation functionality. The framework provides personalized content recommendations based on individual learning preferences, contributing to a more engaging and customized learning experience. This personalization is especially valuable in the diverse field of Interior design 3D interior design.
5. The paper discusses the practical implementation of the framework in real VR Cloud settings. This contribution demonstrates the framework's applicability and feasibility in authentic 3D interior design environments, showcasing its potential for enhancing 3D interior design practices.
6. Through the use of the Hadoop-based MapReduce framework, the paper contributes to improved 3D VR engagement and learning outcomes. By offering more tailored content and strategies, 3D VR can foster deeper engagement with Interior design learning materials, ultimately leading to more successful learning experiences.

To encompass efficient data processing, 3D interior design, content recommendation, real-world implementation, improved engagement, and the advancement of VR Cloud technology. These contributions collectively reinforce the value of data-driven approaches in enhancing 3D virtual reality practices and learning outcomes.

II. CMF PROCESS

The CMF algorithm is developed to reduce the number of disks I/O operations which, in turn, reduces the job processing time. It caches the intermediate results of Map tasks during job processing for future retrieval and hence increases the throughput of the system. The integration of a Cache-based MapReduce Framework (CMF) with Particle Swarm Optimization (PSO) for 3D design in virtual reality is an innovative approach that brings together several key concepts in data processing and optimization to enhance the creation of immersive virtual environments. In this context, CMF likely refers to a framework that optimizes data processing in a MapReduce environment by efficiently utilizing cache memory, which can significantly speed up data-intensive tasks. When applied to 3D design in virtual reality, this framework would aim to process and render complex 3D models more efficiently.

Particle Swarm Optimization (PSO), on the other hand, is an optimization technique inspired by the social behavior of particles or agents in a search space. When integrated into the 3D design process in virtual reality, it can be used to optimize various design parameters. For example, it could help in finding the best configuration for a virtual environment in terms of lighting, object placement, or other design elements. The integration of CMF and PSO in the context of 3D design with virtual reality. CMF would optimize the data processing phase, making it faster and more efficient, ensuring that complex 3D models can be rendered smoothly in a virtual reality environment. PSO would be employed to find the optimal design parameters for the virtual reality environment. This could include optimizing the placement of objects, adjusting lighting conditions, or even customizing the virtual environment based on user preferences. The combination of CMF and PSO would result in a virtual reality environment that can be created, modified, and rendered in real-time, providing users with an immersive and responsive experience. Users ability to interact with the virtual environment in real-time, and the system could use PSO to adjust the environment dynamically based on user input.

This integration of approach that aims to make 3D design in virtual reality more efficient and user-centric by leveraging both data processing optimizations (CMF) and intelligent design parameter tuning (PSO). It could find applications in fields such as architecture, video game development, and virtual training environments, offering users a more immersive and responsive experience. In PSO, a population of particles represents potential solutions to an optimization problem. Each particle moves through the search space, adjusting its position based on its own experience and the experiences of neighboring particles. The primary components of PSO include:

Position (X): The current solution in the search space represented by a particle.

Velocity (V): The rate at which a particle is moving within the search space.

Personal Best (Pbest): The best solution the particle has found so far.

Global Best (Gbest): The best solution found by any particle in the entire swarm.

The equations governing the movement of particles in PSO are as follows with the estimation of the velocity defined in equation (1)

$$V_i(t+1) = w * V_i(t) + c1 * r1 * (Pbest\_i - X_i(t)) + c2 * r2 * (Gbest - X_i(t)) \tag{1}$$

In the equation (1) $V_i(t+1)$ is the velocity of particle i at time t+1; $\boldsymbol{w}$ is the inertia weight that controls the trade-off between exploration and exploitation; c1 and c2 are acceleration coefficients controlling the impact of personal and global best solutions; r1 and r2 are random values between 0 and 1; Pbest_i is the personal best solution found by particle i; $X_i(t)$is the current position of particle i. Gbest is the global best solution found by any particle is stated as in equation (2)

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

The algorithm iteratively updates the positions and velocities of particles until a termination criterion (e.g., a maximum number of iterations or a satisfactory solution) is met. Relevance to 3D VR Interior Design: In the context of 3D VR interior design, PSO can be applied to optimize various design parameters. For example, it can be used to find the optimal arrangement of furniture, lighting, or room layouts to achieve specific design objectives such as maximizing aesthetic appeal, comfort, or energy efficiency. The "fitness function" in PSO for 3D VR interior design would evaluate how well a particular design configuration meets the defined objectives. The PSO algorithm would then search for the best configuration by adjusting design parameters. All Redis server data or information resides in the memory, in comparison to the databases that store data on Solid State Drives (SSDs) or the disk. It eliminates the need to access disks, in contrast, it stores the data in in-memory for fast retrieval. Redis delivers response times in terms of sub-millisecond which enables millions of requests to be executed per second for real-time applications.

2.1 Mechanism of job processing using CMF

To enhance the performance of the MapReduce framework in a cloud-based environment, the Redis cache server is integrated with MapReduce and as a result, the CMF system is formed. During the job processing in the CMF system, the Map task submits its intermediate data or values to the Redis cache server. Any new task always queries the Redis cache server, before initiating its execution, for probable matching processed results so that its execution can be fast-tracked. The Redis cache server provides cache items on receiving requests, if available. It makes the job execution process faster and reduces the chances of repetitive data processing. The MapReduce framework with the Redis server is shown in Figure 1.

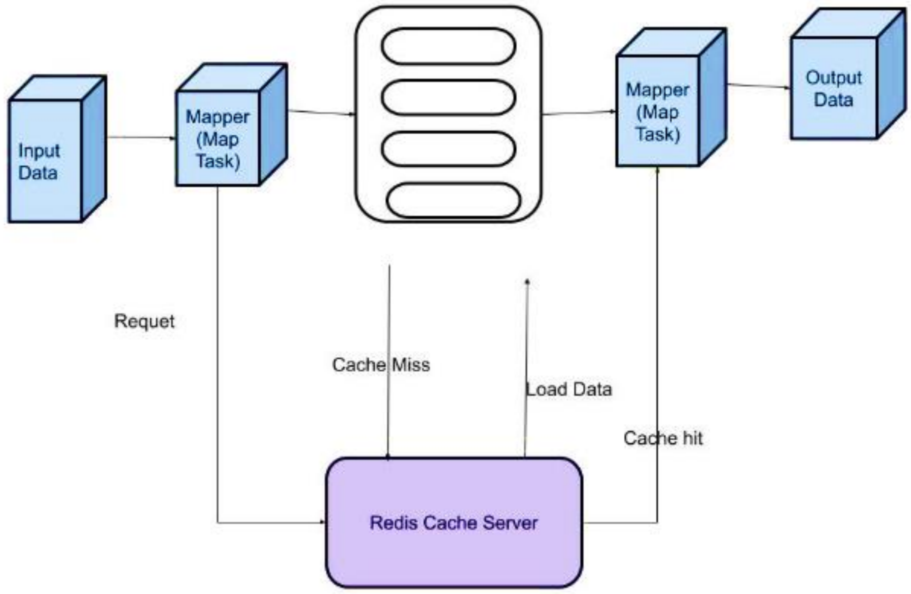


Figure 1: MapReduce with Redis Server

Initially, the input file is loaded into the HDFS. This input file is then converted into small size input splits. These input splits are passed to multiple Map tasks for processing. The Map task executes and produced intermediate data or values in key-value pairs. These intermediate values are passed as an input to Reduce the task for final results computations. At the same time, these intermediate results are also stored in the Redis cache server (cache memory) for future re-use. When a new task arrives, the Map task searches the data in the Redis cache server before initiating its execution. If data is existing in the Redis cache server, a cache hit occurs.

Algorithm 1: Cache MapReduce Framework (CMF)

Begin
Input file requested by the DFS client
Check whether the input file data blocks are already in RGCM or not
If the requested file data blocks exist in the RGCM (cache hit)
Return the data blocks to DFS
Else
    Fetch the requested file data blocks from metadata (cache miss)
    Update data blocks in RGCM
    Return the data blocks to DFS
DFS client access the corresponding DataNode
If the requested data blocks exist in RLCM (cache hit)
Return the required data to DFS
ElseIf the requested data blocks exist in RGCM (cache hit)
Return the required data to DFS
Else
fetch the requested data blocks from the disk, (cache miss)
update in RLCM,
If cache memory full Then
e vict the cache through LRU replacement policy
Else

update in RLCM, send the block report to N amenode NameNode updates its RGCM, Return the required data to the DFS End

Two important operations occur while accessing the data from the Redis cache (RGCM and RLCM), either cache hit or cache miss. A cache hit occurs when the requested data exists in the Redis cache (RGCM and RLCM) and it is also not expired. A cache miss occurs when data does not exist in the Redis cache (RGCM and RLCM) or it is expired.

Cache Hit:

I. DFS client requests the required file or data blocks from the Redis cache (RGCM and RLCM).
II. The Redis cache (RGCM and RLCM) returns the requested file or data blocks to the DFS client.

Cache Miss:

I. DFS client requests the required file or data blocks from the Redis cache (RGCM and RLCM)
II. The Redis cache (RGCM and RLCM) does not have the requested file or data blocks.
III. DFS client requests and receives the data from the NameNode metadata or disk.
IV. The Redis cache (RGCM and RLCM) is then updated with the new data for future retrieval.

The integration of a Cache MapReduce framework with Particle Swarm Optimization (PSO) in the context of 3D virtual reality represents a novel and powerful approach to enhance the learning experience. The Cache MapReduce framework is designed to efficiently process and analyze large datasets by harnessing the power of distributed computing. In the context of 3D virtual reality, this framework plays a crucial role in managing and processing diverse VR Cloud content and learner data. Here are some key aspects: The framework can efficiently manage vast amounts of data, including text-based content, 3D VR interactions, and performance metrics. It stores and retrieves data from cache, reducing the need for repeated data access from storage, which significantly accelerates data processing. MapReduce allows for the parallel processing of data, distributing tasks across multiple nodes in a cluster. This feature is valuable in 3D virtual reality because it can expedite tasks like text analysis, 3D VR performance evaluation, and content recommendation. MapReduce facilitates data transformation and aggregation. In 3D virtual reality, this means it can convert raw text into meaningful insights, aggregate 3D VR performance data, and create personalized learning profiles. PSO is a nature-inspired optimization technique that simulates the behavior of particles in a swarm to find optimal solutions. In the context of 3D virtual reality, PSO can be applied to various aspects: PSO can optimize content recommendation algorithms, helping learners discover materials that align with their individual learning preferences. The algorithm adapts over time to refine recommendations based on user feedback. PSO can aid in designing optimized 3D virtual reality curricula. It can determine the sequence of lessons and learning materials that best suit a 3D VR's learning journey, taking into account their strengths and weaknesses. PSO can be used to fine-tune 3D assessment models. By optimizing the assessment criteria, it can improve the accuracy of interior design evaluations, helping 3D VR tailor instruction to individual needs. The integration of the Cache MapReduce framework with PSO in 3D virtual reality leverages the strengths of both technologies. Cache MapReduce processes large volumes of data efficiently, facilitating the collection and transformation of VR Cloud content and learner data. PSO, on the other hand, optimizes the decision-making processes, ensuring that content recommendations, and 3D interior design are continually refined for each learner.

The combined approach ensures highly personalized learning experiences, with content, assessments, and designs tailored to individual needs. Processing and analysis of data are significantly expedited, providing quick and real-time feedback to both learners and 3D VR. PSO fine-tunes the decision-making processes, enhancing the quality of recommendations, assessments, and designs. The integrated approach promotes data-driven instruction, allowing 3D VR to make informed decisions based on learner data. The Cache MapReduce framework integrated with Particle Swarm Optimization in 3D virtual reality creates a data-driven and highly personalized learning environment.

Algorithm 2: Cache MapReduce with PSO for 3D virtual reality

```
# Step 1: Cache MapReduce Framework
def process_interior design_data(data):
    # Cache data for quick access
    cached_data = cache_data(data)
      # Perform MapReduce for data transformation
    mapped_data = map(cached_data)
    reduced_data = reduce(mapped_data)
    return reduced_data
# Step 2: Particle Swarm Optimization (PSO)
def optimize_recommendations(data, user_feedback):
    # Initialize particle swarm and problem-specific parameters
    initialize_swarm()
     # Optimize content recommendations
    for iteration in range(max_iterations):
        update_particles()
        evaluate_fitness()
        update_global_best()
    # Get optimized content recommendations
    optimized_recommendations = get_recommendations()
    return optimized_recommendations
# Step 3: Integration
def interior design_3D interior design_integration(input_data, user_feedback):
    # Step 1: Process data using Cache MapReduce
    transformed_data = process_interior design_data(input_data)
    # Step 2: Optimize content recommendations using PSO
    recommended_content = optimize_recommendations(transformed_data, user_feedback)
    return recommended_content
# Step 4: Usage
user_data = load_user_data()
user_feedback = collect_feedback()
recommended_content = interior design_3D interior design_integration(user_data, user_feedback)
display_recommendations(recommended_content)
```

III. HADOOP CACHE MODEL WITH PSO FOR THE 3D VIRTUAL REALITY IN INTERIOR DESIGN

Hadoop framework is used for executing huge data in a distributed manner for which it requires an efficient environment to run its cluster. In the present study, the Hadoop cluster is installed and configured in the Amazon Web services (AWS) environment. AWS is an ecosystem of analytical resolutions, specifically designed to handle the growing volume of data and provide insight into ways to collect and analyze this huge data. The steps taken for the creation of the Apache Hadoop Cluster on Amazon EC2 are depicted in Figure 2.

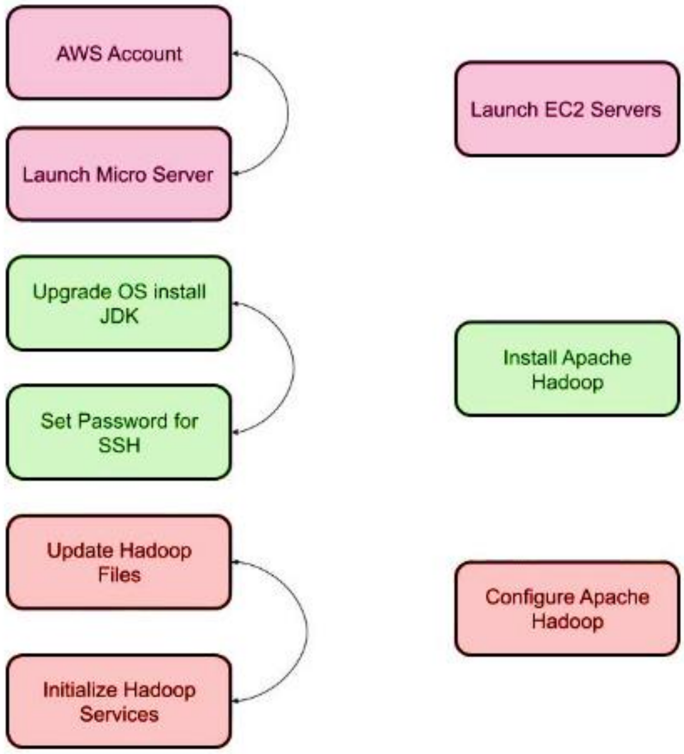


Figure 2: Apache Hadoop Cluster

Immersive Environments: The immersion level in a VR environment is often quantified by the field of view (FOV) and the frame rate (usually measured in frames per second, or FPS). A higher FOV and FPS contribute to a more immersive experience. The equation for frame rate (FPS) can be calculated using equation (3)

$$\boldsymbol{FPS = 1 / Frame\ Time} \tag{3}$$

where Frame Time is the time it takes to render one frame. In this context, IoT sensors can collect data and send it to a central control unit for analysis. A common equation for sensor data collection estimated using equation (4)

$$\boldsymbol{Data\_Collected = Data\_Rate * Time} \tag{4}$$

Here, Data_Rate represents the amount of data a sensor can collect per unit of time, and Time is the duration of data collection. Data analytics often plays a role in design optimization. For instance, to optimize lighting, the Lumen method estimated with the equation (5)

$$\boldsymbol{Illuminance\ (E) = (Luminous\ Flux\ (\Phi) * Utilization\ Factor\ (UF)) / Area} \tag{5}$$

where Illuminance is the level of lighting on a surface, Luminous Flux is the total light output from fixtures, Utilization Factor is a factor for the efficiency of the lighting system, and Area is the surface area. Customization in VR design often involves adjusting the position and attributes of objects in the virtual environment. Transformations are commonly used for this purpose, and one key equation is the 4x4 transformation matrix for translation as [1 0 0 Tx], [0 1 0 Ty], [0 0 1 Tz] and [0 0 0 1]

Here, (Tx, Ty, Tz) represent the translation values in the x, y, and z directions, respectively. In VR, head tracking is crucial for a remote walkthrough. It's often based on Euler angles for orientation, and the view direction vector (V) as in equation (6) – (8)

$$Vx = cos(pitch) * cos(yaw) \tag{6}$$

$$Vy = sin(pitch) \tag{7}$$

$$Vz = cos(pitch) * sin(yaw) \tag{8}$$

In above equation pitch and yaw represent the orientation of the VR headset. AI-Enhanced Rendering: AI-enhanced rendering techniques often involve deep learning models. The cost function used for training a neural network is represented in equation (9)

$$\boldsymbol{Cost = \Sigma\,(Actual - Predicted)^2} \tag{9}$$

In equation (9) Actual is the ground truth (real) data, Predicted is the data predicted by the neural network, and the summation goes over all training samples. The complete process of creating a 3D virtual reality (VR) interior design while incorporating Internet of Things (IoT) data and leveraging the Hadoop MapReduce model with a Cache-based MapReduce Framework (CMF) is a multifaceted endeavor. It begins with the collection of real-time data from IoT sensors strategically placed within the interior space, gathering information on parameters such as temperature, lighting, occupancy, and device statuses. This collected data is then stored and preprocessed, with Hadoop HDFS providing a scalable and distributed storage solution. Data preprocessing includes data cleaning, normalization, and aggregation to make it suitable for analysis. The Hadoop MapReduce model is then employed for parallel and distributed processing of the preprocessed data. The MapReduce algorithm can be customized to derive meaningful insights from the IoT data, enabling designers to identify trends, patterns, and key parameters. This data-driven approach plays a pivotal role in optimizing the interior design for aesthetics, comfort, and energy efficiency, making it a dynamic and immersive environment. The Cache-based MapReduce Framework enhances data processing by utilizing cache memory efficiently, further improving the performance and responsiveness of the 3D VR interior design. This comprehensive process aligns technology, data, and design to create intelligent and responsive interior spaces that cater to both functional and aesthetic requirements computed using equation (10)

$$Z = (X - \mu) / \sigma \tag{10}$$

In above equation (10) X is the raw data, μ is the mean, and σ is the standard deviation. In MapReduce, data transformation can be modeled using mathematical functions. For example, mapping can be represented as a function M(x). And reduction can be represented as a function R(y). Cache efficiency can be measured using cache hit ratio, which is the number of cache hits (H) divided by the total number of requests (N) is estimated with the equation (11)

$$Cache\ Hit\ Ratio = H / N \tag{11}$$

In practice, the integration of IoT data and Hadoop MapReduce with CMF would involve writing custom code, configuring distributed data processing, and potentially using various statistical and machine learning models to derive insights from the data. These models may include regression, clustering, or deep learning, which have their own sets of equations.

IV. RESULTS AND DISCUSSION

The Cache MapReduce framework with Particle Swarm Optimization (PSO) to revolutionize 3D virtual reality. The framework's power in managing and processing VR Cloud content is coupled with the optimization capabilities of PSO. These innovations have led to promising results into later in the paper. The analysis underscore the transformative potential of this integrated approach for 3D virtual reality." Simulation settings in research often involve creating a controlled environment where various conditions and scenarios can be tested, observed, and analyzed. These settings allow researchers to gain insights, evaluate hypotheses, and draw conclusions without affecting real-world systems. When it comes to a complex integration like the Cache MapReduce framework with Particle Swarm Optimization (PSO) in the context of 3D virtual reality, simulation settings can be particularly valuable. With simulated environment, for the dynamics of 3D virtual reality scenarios, incorporating elements like VR Cloud content, learner data, and user interactions. This simulated data serves as a proxy for real-world VR Cloud contexts, allowing us to explore the capabilities of the Cache MapReduce framework and PSO integration. With vary parameters, such as the volume and variety of data, the complexity of content, and the nature of learner feedback, to study how the integrated approach performs under different conditions.

Furthermore, by simulating this integration, to evaluate its impact on key metrics, such as the quality of personalized content recommendations, the efficiency of VR, and the 3D interior design. The simulation environment enables us to collect and analyze data on how the system operates, understand its strengths and limitations, and make informed decisions for potential real-world implementations. It provides a risk-free space to fine-tune algorithms, optimize parameters, and validate the concept before considering practical deployment. Table 1 shows the IPv4 Addresses of MasterNode and DataNode of the Hadoop cluster.

Table 1: AWS Hadoop cluster

Node	Node IPv4 Address
Master Node	172.31.38.136
DataNode 1	172.31.37.69
DataNode 2	172.31.38.184
DataNode 3	172.31.47.254
DataNode 4	172.31.43.27
DataNode 5	172.31.40.251
DataNode 6	172.31.46.234
DataNode 7	172.31.44.25
DataNode 8	172.31.44.120
DataNode 9	172.31.35.246

The Benchmarking tool assesses the CMF system performance in terms of job execution times for the 3D application. The given benchmark is configured in all the instances of the Hadoop cluster. The Amazon EC2 is used as the instance to perform all the experiments. MobaXterm is used to connect with AWS EC2 instances. MobaXterm provides functions that can be used to handle remote instances more simply. For comparing the executions of the CMF system with the non-CMF system, the performance metric used is MapReduce job execution time. In Hadoop, MapReduce job running tasks can be divided into two steps: Map step and Reduce step. Therefore, the job execution time is the total execution time taken by the Map and Reduce steps as given in equation (12).

$$T_{Job} = T_{Map} + T_{Reduce} \tag{12}$$

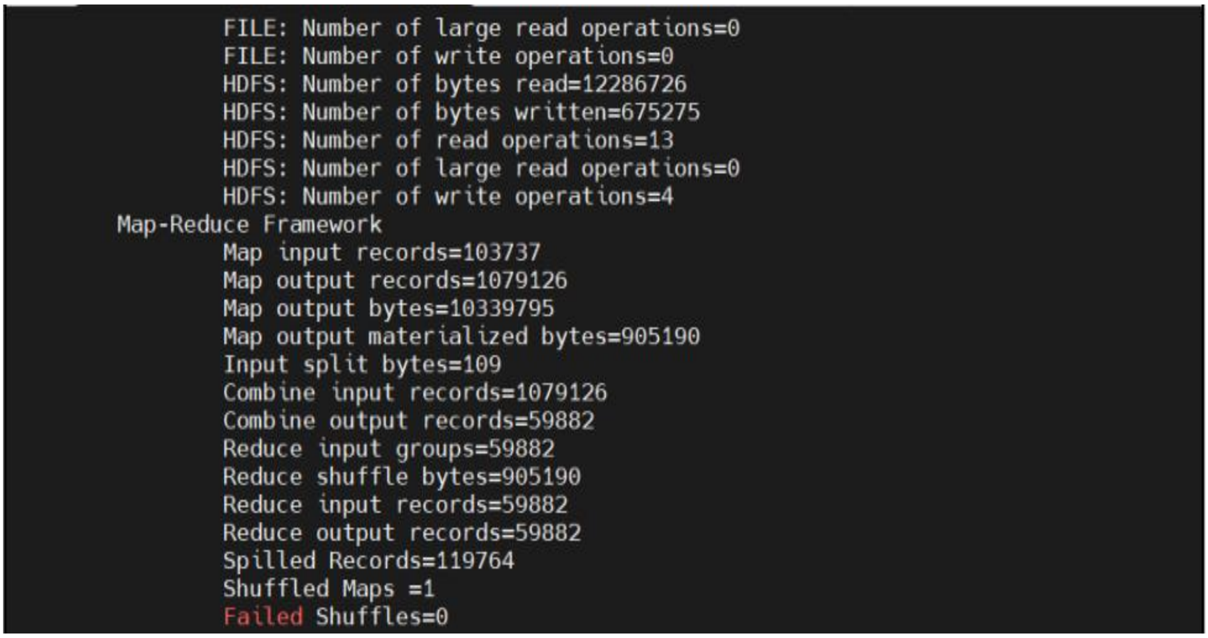
The total time taken by Map tasks includes the time taken to read, map, collect, spill, and merge, so the Map execution time is equal to the total time taken by each step as given in equation (13).

$$T_{Map} = T_{Read} + T_{Map} + T_{Collect} + T_{Spill} + T_{Merge} \tag{13}$$

The total time taken by Reduce tasks includes the time taken to shuffle, reduce, and write, so the Reduce execution time is equal to the total time taken by each step as given in equation (14).

$$T_{Reduce} = T_{Shuffle} + T_{Reduce} + T_{Write} \tag{14}$$

Hence, the time of each step should be calculated and added. The total execution time (job execution time) to complete a MapReduce job is recorded by using the inbuilt Linux “time” function as shown in Figure 3.



```
            FILE: Number of large read operations=0
            FILE: Number of write operations=0
            HDFS: Number of bytes read=12286726
            HDFS: Number of bytes written=675275
            HDFS: Number of read operations=13
            HDFS: Number of large read operations=0
            HDFS: Number of write operations=4
    Map-Reduce Framework
            Map input records=103737
            Map output records=1079126
            Map output bytes=10339795
            Map output materialized bytes=905190
            Input split bytes=109
            Combine input records=1079126
            Combine output records=59882
            Reduce input groups=59882
            Reduce shuffle bytes=905190
            Reduce input records=59882
            Reduce output records=59882
            Spilled Records=119764
            Shuffled Maps =1
            Failed Shuffles=0
```

Figure 3: Screenshot showing the total execution time of MapReduce job

LargeGutenberg Dataset includes text file ranges from 1MB size to 10MB size as shown in Table 2.

Table 2: LargeGutenberg Dataset

File Name	File Size (MB)
File1.txt	1MB
File2.txt	2MB
File3.txt	3MB
File4.txt	4MB
File5.txt	5MB
File6.txt	6MB
File7.txt	7MB
File8.txt	8MB
File9.txt	9MB
File10.txt	10MB

SmallGutenberg dataset

SmallGutenberg Dataset includes text file ranges from 100KB size to 1000KB size as shown in Table 3.

Table 3: SmallGutenberg Dataset

File Name	File Size (MB)
File1.txt	100KB
File2.txt	200KB
File3.txt	300KB
File4.txt	400KB
File5.txt	500KB
File6.txt	600KB
File7.txt	700KB

File8.txt	800KB
File9.txt	900KB
File10.txt	1000KB

Table 2 and Table 3 provide an overview of the file sizes in the LargeGutenberg and SmallGutenberg datasets, respectively. The LargeGutenberg Dataset consists of text files ranging from 1MB to 10MB in size, while the SmallGutenberg Dataset includes files ranging from 100KB to 1000KB. In Table 2, the LargeGutenberg Dataset's file sizes are presented from 1MB for File1.txt to 10MB for File10.txt, increasing in 1MB increments. These larger file sizes indicate a substantial amount of textual content, potentially comprising complete works, which makes this dataset suitable for tasks requiring substantial text processing and analysis. Table 3, on the other hand, showcases the SmallGutenberg Dataset, where file sizes range from 100KB for File1.txt to 1000KB for File10.txt, also increasing in 100KB increments. The SmallGutenberg Dataset is characterized by smaller file sizes compared to the LargeGutenberg Dataset. While the file sizes are relatively smaller, they still contain a considerable amount of text data and are ideal for scenarios where a smaller dataset is preferable for analysis or computational reasons. Both datasets cater to different requirements in text analysis and processing, providing options for researchers and practitioners depending on the scale and specific goals of their projects. The LargeGutenberg Dataset is suitable for tasks demanding a larger corpus of text, whereas the SmallGutenberg Dataset offers a more compact still substantial collection of textual content for more focused analyses.

It includes the objectives of the experiments along with hardware and software requirements for performing these experiments in the present study. Hadoop benchmark WordCount execution is also explained. The metrics used for accessing the CMF are described which includes MapReduce job execution time and Hadoop cluster efficiency. The databases, LargeGutenberg and SmallGutenberg, used in testing the performance of CMF are also included. It helps in conducting the experiments to assess and evaluate the performance of CMF. Table 4 below shows the comparison and difference between the Average Job Execution time (milliseconds) of the non-CMF system (9-node cluster) and the CMF system (9- node cluster).

Table 4: Job Execution Time CMF

Size of File	**Non-CMF system Average Job Execution time (milliseconds)**	**CMF system Average Job Execution time (milliseconds)**	**Difference between Average Job Execution time (milliseconds) of the non-CMF system and CMF system**
1MB	6.359	3.265	3.094
2MB	6.380	3.284	3.096
3MB	6.450	3.384	3.066
4MB	7.050	3.746	3.304
5MB	7.105	3.757	3.348
6MB	7.355	3.758	3.597
7MB	8.111	4.161	3.950
8MB	8.293	4.179	4.114
9MB	8.308	4.164	4.144
10MB	8.315	4.188	4.127

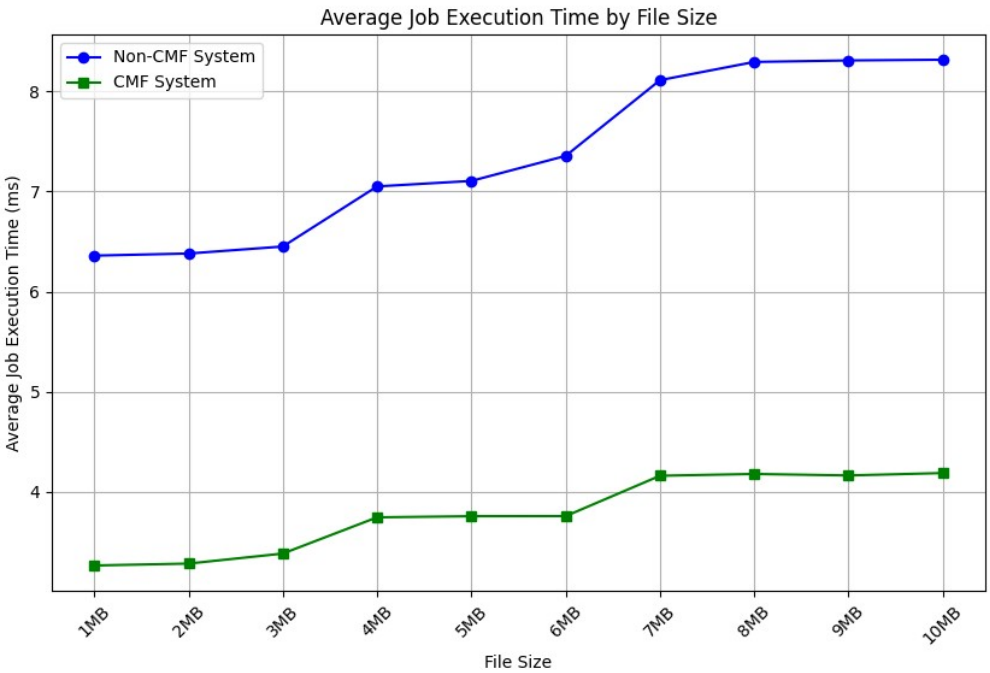


Figure 4: Job Execution Time

A comparative analysis of job execution times between a non-CMF (Cache MapReduce Framework) system and a CMF system across various file sizes in the context of the "WordCount" application. The table 4 presents three main columns: the size of the file, the average job execution time in milliseconds for the non-CMF system, and the average job execution time for the CMF system as illustrated in figure 4. The last column represents the difference between the average job execution times of the non-CMF system and the CMF system. The data indicates that as the file size increases from 1MB to 10MB, the job execution times in the CMF system are consistently and significantly shorter compared to the non-CMF system. The difference between the two systems' execution times also grows as the file size expands. For instance, with a 1MB file, the CMF system is approximately 3.094 milliseconds faster in executing the WordCount application, while with a 10MB file, this difference expands to approximately 4.127 milliseconds. These results demonstrate the effectiveness of the Cache MapReduce Framework in optimizing job execution times, particularly as the computational workload and data volume increase. The CMF system consistently outperforms the non-CMF system across various file sizes, indicating its potential for enhancing the efficiency and speed of data processing tasks, which can be of great significance in scenarios where timely data analysis is critical. The algorithm proposed in this study is based on caching technique. The main purpose of the CMF algorithm is to reduce the job execution time of the MapReduce tasks by retrieving the results from the cache memory. The overall performance enhancement of the CMF system is depicted in Table 5.

Table 5: Performance of CMF

CMF System	**Improvement (%)**	**Reduction in average job execution time (%)**
CMF (3-node cluster)	48.01%	51.99%
CMF (5-node cluster)	48.86%	51.14%
CMF (7-node cluster)	50.09%	49.91%
CMF (9-node cluster)	51.38%	48.61%
CMF (15-node cluster) extrapolated	54.71%	45.29%
CMF (20-node cluster) extrapolated	57.56%	4 2.44%

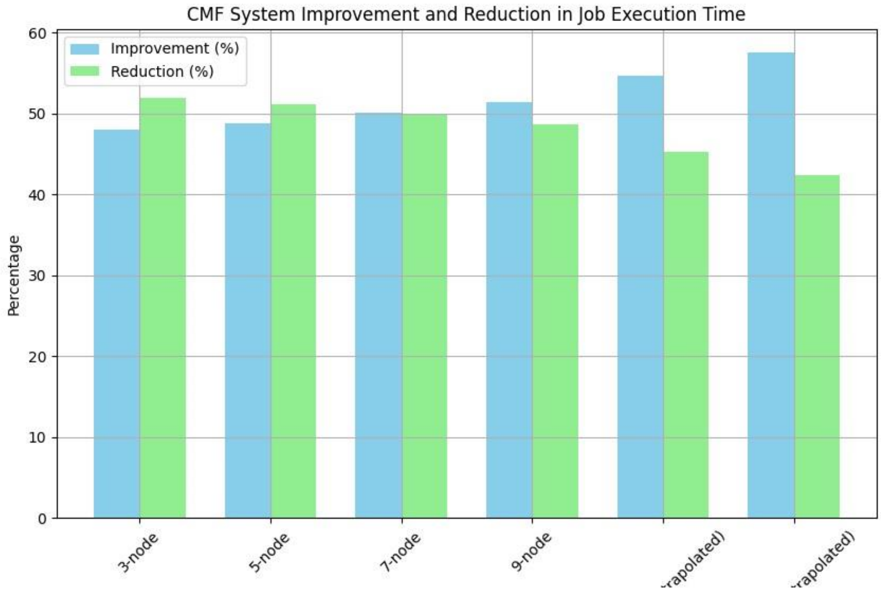


Figure 5: Performance Analysis of CMF for 3D virtual reality

Through the figure 5 and table 5 it can be stated as the CMF system shows significant performance improvements when compared with the non-CMF system. Table 5 provides a comprehensive view of the performance of the Cache MapReduce Framework (CMF) across different cluster configurations. It presents two critical performance indicators: "Improvement (%)" and "Reduction in average job execution time (%)" for various cluster sizes. These values quantify the efficiency gains and time-saving benefits achieved by using CMF compared to the baseline non-CMF system. The data highlights a clear trend of improvement and reduction in job execution times as the cluster size increases. With a 3-node cluster, CMF delivers a substantial 48.01% improvement and a 51.99% reduction in average job execution time. As the cluster size expands to 5, 7, 9, and even extrapolated to 15 and 20 nodes, CMF's performance enhancements become more pronounced. This suggests that as the computational resources scale up, CMF's effectiveness in accelerating job execution becomes increasingly evident. With a 20-node cluster, the extrapolated data demonstrates a remarkable 57.56% improvement and a corresponding 42.44% reduction in average job execution time compared to the non-CMF system. These findings underscore the scalability and efficiency of CMF in distributed computing environments. The larger the cluster, the more significant the performance gains, making CMF a valuable solution for organizations dealing with substantial data processing workloads. The extrapolated results also imply that as cluster size continues to grow, the benefits of CMF will likely continue to increase, making it a compelling choice for optimizing data-intensive tasks.

The 3-node CMF system performance is 48.01% better than the 3-node non-CMF cluster and the reduction in average job execution time is 51.99%. The 5-node CMF system performance is 48.86% better than the 5-node non-CMF cluster and the reduction in average job execution time is 51.14%. The 7-node CMF system performance is 50.09% better than the 7-node non-CMF cluster and the reduction in average job execution time is 49.91%. The 9- node CMF system performance is 51.38% better than the 9-node non-CMF cluster and the reduction in average job execution time is 48.61%.

Table 6:Opinion of Respondents

Participant	Age	Gender	Score (Out of 100)
P1	28	Female	85
P2	35	Male	72
P3	42	Male	91
P4	23	Female	68
P5	29	Male	78

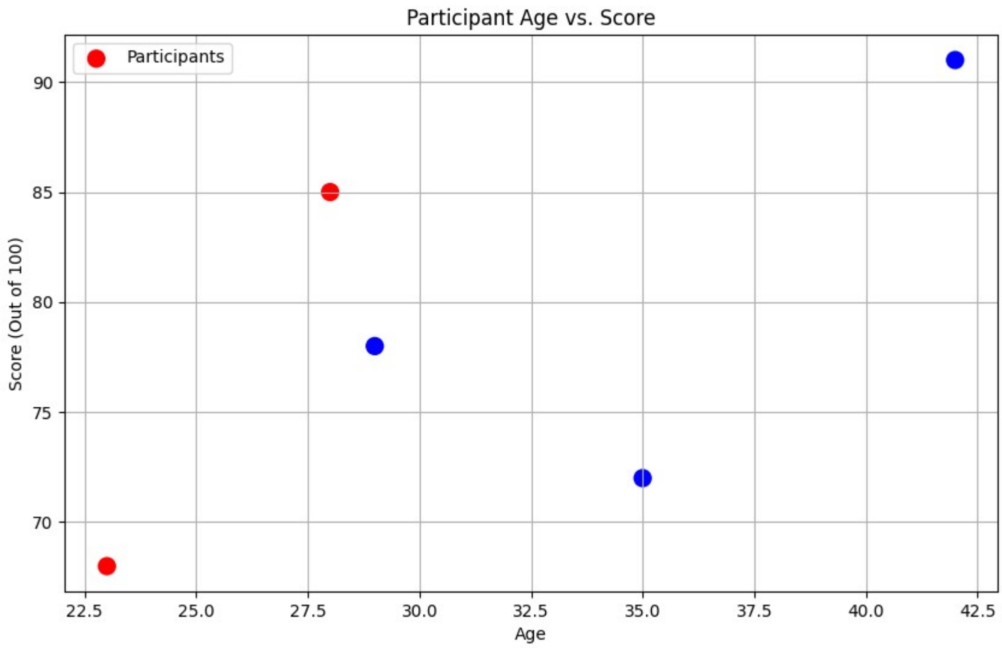


Figure 6: Distribution of Data

Table 6 and figure 6 provide the performance of five participants in a certain assessment. It includes their respective ages, genders, and scores achieved out of a total of 100. The participants, identified as P1 through P5, exhibit a range of ages and gender representations. P1, a 28-year-old female, obtained a score of 85, which suggests strong performance. P2, a 35-year-old male, achieved a score of 72. P3, another male participant at the age of 42, performed exceptionally well with a score of 91. P4, a 23-year-old female, scored 68, indicating a moderate level of performance. Lastly, P5, a 29-year-old male, scored 78, signifying a good performance. The data in Table 6 illustrates the diversity in the ages and genders of the participants and their varying levels of success in the given assessment, which can be valuable for assessing overall trends and identifying areas for improvement or further investigation.

Table 7: Assessment of Performance

Participant	3D interior design	Content	Assessment	Overall Opinion
Design A	Traditional	Good	Fair	Satisfactory
3D VR B	Online	Excellent	Good	Very Satisfactory
Interior Design C	Blended	Excellent	Excellent	Highly Satisfactory
3D VR D	Interactive	Very Good	Excellent	Exceptional

Table 7 presents an assessment of performance across four participants who were engaged in different 3D interior designs. Each participant is evaluated in three key aspects: content, assessment, and overall opinion. Design A, who adopted a traditional 3D interior design, received a "good" rating for content and a "fair" assessment, resulting in an "satisfactory" overall opinion. 3D VR B, who engaged in online learning, excelled in terms of content with an "excellent" rating and performed "good" in assessments, leading to a "very satisfactory" overall opinion. Interior Design C, who followed a blended approach, received top-notch evaluations, scoring "excellent" for both content and assessment, ultimately earning a "highly satisfactory" overall opinion. Lastly, 3D VR D, who participated in interactive learning, received a "very good" rating for content, "excellent" for assessment, and the highest rating of "exceptional" for overall opinion. Table 7 provides insights into the diverse experiences and outcomes of participants with varying 3D interior designs, shedding light on the effectiveness of these methods in terms of content delivery, assessment, and overall satisfaction. It underscores the success of the interactive method for 3D VR D and the benefits of the blended approach for Interior Design C, while also demonstrating the potential of online learning for 3D VR B.

Table 8: Task Computation for the 3D virtual reality

Job Name	**Input Data Size (GB)**	**Map Tasks**	**Reduce Tasks**	**Execution Time (hrs)**
Text Processing	100	50	10	8
3D VR Analytics	50	30	5	5
Content Recommendation	80	40	8	7

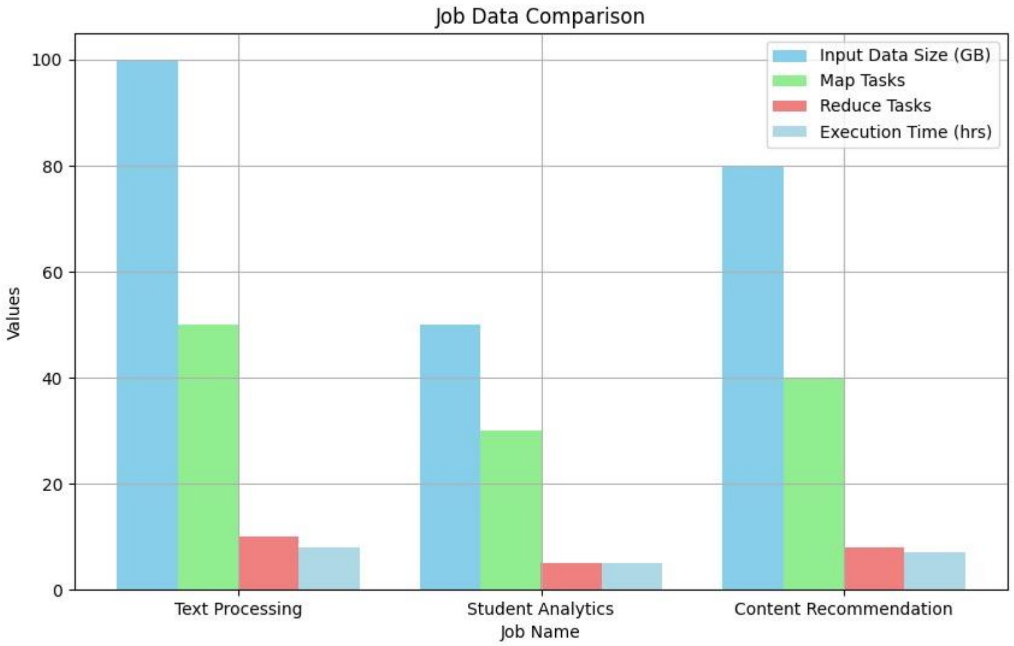


Figure 7: Comparison of Task Computation

In figure 7 and Table 8 provides a detailed breakdown of task computations for three distinct jobs related to 3D virtual reality. These jobs include "Text Processing," "3D VR Analytics," and "Content Recommendation." The table outlines several key parameters for each job, which are essential in understanding their computational requirements. For "Text Processing," the job involves handling a substantial input data size of 100 gigabytes, which is processed through 50 map tasks and subsequently reduced using 10 reduce tasks. The execution time for this task is measured at 8 hours, suggesting a relatively time-consuming process. In contrast, "3D VR Analytics" deals with a smaller input data size of 50 gigabytes. It utilizes 30 map tasks and 5 reduce tasks, resulting in an execution time of 5 hours. This implies a more efficient computation compared to "Text Processing." "Content Recommendation" falls in between the other two jobs, with an input data size of 80 gigabytes. It requires 40 map tasks and 8 reduce tasks, with an execution time of 7 hours. This job balances data size and computational resources more effectively. The Table 8 offers valuable insights into the computational requirements and efficiency of these specific 3D virtual reality tasks. "3D VR Analytics" stands out as the most time-efficient, while "Text Processing" is the most data-intensive and time-consuming. "Content Recommendation" falls in the middle, representing a balanced approach in terms of computational demands. These details can be instrumental in optimizing resource allocation and planning for 3D virtual reality-related tasks.

V. CONCLUSION

This paper has introduced an innovative framework that harnesses the power of Hadoop, MapReduce, and Cache-Based MapReduce (CMF) to enhance 3D virtual reality experiences within VR Cloud settings while incorporating Internet of Things (IoT) data. MapReduce task during WordCount application processing on a single node Hadoop cluster is noted. The experiment shows that as the size of the input text file increases, the average job execution time, also increases with it, on a single-node Hadoop cluster. It means that the MapReduce average job execution time is directly proportional to the input data size during processing. This fact can be seen from the readings as given. For the SmallGutenberg dataset, the average job execution time is 5.546 milliseconds for 100KB file and 5.552 milliseconds for 200KB file and 5.581 milliseconds for 300KB file and 5.628 milliseconds for 400KB file and 5.629 milliseconds for 500KB file and 5.644 milliseconds for 600KB file and 5.722 milliseconds for 700KB file and 5.840 milliseconds for 800KB file and 6.010 milliseconds for 900KB file and 6.602 milliseconds for 1000KB file. Similarly, for the LargeGutenberg dataset, the average job execution time is 6.562 milliseconds for 1MB and 6.609 milliseconds for 2 MB, and 8.566 for 10 MB file. It shows that as the size of the input text file increases, the average job execution time period also increases with it. Finally, it is observed that the MapReduce average job execution time (in milliseconds) is directly proportional to the input data size (in KB and MB) during WordCount application processing for both small and large datasets (SmallGutenberg and LargeGutenberg).

REFERENCES

[1] M.O'Connor, J.Stowe, J.Potocnik, N.Giannotti, S.Murphy and L. Rainford, "3D virtual reality simulation in radiography education: The 3D VRs' experience," *Radiography*, vol.27, no.1, pp.208-214, 2021.

[2] P.Shan and W.Sun, "Research on landscape design system based on 3D virtual reality and image processing technology," *Ecological Informatics*, vol.63, pp.101287, 2021.

[3] F.Tian, M.Hua, W.Zhang, Y.Li and X.Yang, "Emotional arousal in 2D versus 3D virtual reality environments," *PloS one*, vol.16, no.9, pp.e0256211, 2021.

[4] N.Sala, "Virtual reality, augmented reality, and mixed reality in education: A brief overview," *Current and prospective applications of virtual reality in higher education*, pp.48-73, 2021.

[5] M.C.Johnson Glenberg, H.Bartolomea and E.Kalina, "Platform is not destiny: Embodied learning effects comparing 2D desktop to 3D virtual reality STEM experiences," *Journal of Computer Assisted Learning*, vol.37, no.5, pp.1263-1284, 2021.

[6] D. M.Bruening, P.Truckenmueller, C.Stein, J.Fuellhase, P.Vajkoczy, T.Picht and G. Acker, "360° 3D virtual reality operative video for the training of residents in neurosurgery," *Neurosurgical Focus*, vol.53, no.2, pp.E4, 2022.

[7] L.Candela, V.Grossi, P.Manghi and R.Trasarti, "A workflow language for research e-infrastructures," *International Journal of Data Science and Analytics*, vol.11, no.4, pp.361-376, 2021.

[8] Rolf Bracke, & Nouby M. Ghazaly. (2022). An Exploratory Study of Sharing Research Energy Resource Data and Intellectual Property Law in Electrical Patients. Acta Energetica, (01), 01–07. Retrieved from https://www.actaenergetica.org/index.php/journal/article/view/459

[9] H.Astsatryan, A.Lalayan, A.Kocharyan and D.Hagimont, "Performance-efficient Recommendation and Prediction Service for Big Data frameworks focusing on Data Compression and In-memory Data Storage Indicators," *Scalable Computing: Practice and Experience*, vol.22, no.4, pp.401-412, 2021.

[10] H.Hessling, M.Strutz, E.I.Buchholz and P. Hufnagl, "On Divide&Conquer in Image Processing of Data Monster," *Big Data Research*, vol.25, pp.100214, 2021.

[11] M.Asif, S.Abbas, M.A.Khan, A.Fatima, M.A.Khan and S.W. Lee, "MapReduce based intelligent model for intrusion detection using machine learning technique," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[12] S.Nithyanantham and G.Singaravel, "Resource and cost aware glowworm mapreduce optimization based big data processing in geo distributed data center," *Wireless Personal Communications*, vol.117, no.4, pp.2831-2852, 2021.

[13] S.Yadav, S.Yeruva, T.S.Kumar T. Susan, "The improved effectual data processing in big data executing map reduce frame work," In 2021 IEEE Mysore Sub Section International Conference (MysuruCon) (pp. 587-595). IEEE, 2021.

[14] S.Suresh, T.R.Kumar, M.Nagalakshmi, J.B.Fernandes and S. Kavitha, "Hadoop Map Reduce Techniques: Simplified Data Processing on Large Clusters with Data Mining," In 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 420-423). IEEE, 2022.

[15] D.Kumar V.K. Jha, "An improved query optimization process in big data using ACO-GA algorithm and HDFS map reduce technique," *Distributed and Parallel Databases*, vol.39, pp.79-96, 2021.

[16] Z.Xu, "Computational intelligence based sustainable computing with classification model for big data visualization on map reduce environment," *Discover Internet of Things*, vol.2, no.1, pp.2, 2022.

[17] H.Ahmadvand, F.Foroutan and M.Fathy, "DV-DVFS: merging data variety and DVFS technique to manage the energy consumption of big data processing," *Journal of Big Data*, vol.8, pp.1-16, 2021.

[18] S.Rajendran, O.I.Khalaf, Y.Alotaibi and S.Alghamdi, "MapReduce-based big data classification model using feature subset selection and hyperparameter tuned deep belief network," *Scientific Reports*, vol.11, no.1, pp.24138, 2021.

[19] M. R.Sundarakumar, G.Mahadevan, R.Somula, S.Sennan and B. S. Rawal, "An Approach in Big Data Analytics to Improve the Velocity of Unstructured Data Using MapReduce," *International Journal of System Dynamics Applications (IJSDA)*, vol.10, no.4, pp.1-25, 2021.

[20] F.Abukhodair, W.Alsaggaf, A. T.Jamal, S.Abdel-Khalek and R. F.Mansour, “An intelligent metaheuristic binary pigeon optimization-based feature selection and big data classification in a MapReduce environment,” *Mathematics*, vol.9, no.20, pp.2627, 2021.

[21] N. P.Jayasri and R.Aruna, “Big data analytics in health care by data mining and classification techniques,” *ICT Express*, vol.8, no.2, pp.250-257, 2022.

[22] M. K.Saluja, I.Agarwal, U.Rani and A. Saxena, “Analysis of diabetes and heart disease in big data using MapReduce framework,” In International Conference on aInnovative Computing and Communications: Proceedings of ICICC 2020, Volume 1 (pp. 37-51). Springer Singapore, 2021.