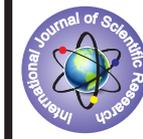


Multi Objective Tiger Hash Storing For Maximal Cloud Service Distribution in Green Environment



Computer Science

KEYWORDS: Data storage, Tiger Hash Tree, service rendering, green cloud, virtual cloud servers, cloud users

S.NIVAS

Assistant Professor, Department of Computer Science, Nift-Tea College of Knitwear Fashion, Tirupur, Tamilnadu, India

ABSTRACT

Data storage has become a significant issue with the extensive availability of cheap storage technology and the improvement of global data centers for green cloud storage. Recently, few research works has been designed for green cloud storage. But, there is a need for effective green cloud storage system for improving data storage capacity and reducing the data accessing time. In order to such limitation, Multi Objective Tiger Hash Storing (MOTHS) Technique is proposed. The MOTHS Technique improves the energy storage during the multiple communication between different virtual cloud servers in green cloud computing. In MOTHS Technique, Tiger Hash Tree is constructed to achieve efficient data dynamics for improving the storage models and reducing the user task service rendering complexity. Besides, the MOTHS Technique supports dynamic operation on cloud data while maintaining the same level of storage correctness and maximizes the service distribution according to green cloud environmental attributes. The user's scalability along with storage capability of communication tasks is analyzed to achieve better green cloud data storage. The performance of MOTHS Technique is measured in terms of data storage capacity, storage time, energy consumption and accessing time. The experimental analysis shows that the MOTHS Technique is able to improve the data storage capacity 16% and also reduces the data accessing time 42% when compared to the state-of-the-art works.

1. INTRODUCTION

Green computing is an environmentally sustainable computing. The key objective of green computing is to explore new computing systems, computing model and application with the low-cost and low power consumption and support the development of economy and society. In Green computing environment, enhancing the storage efficiency and power utilization of data storage equipment is a complex one with different factors to consider.

Recently, most of research works has been developed for cloud data storage. For example, Energy consumption analysis technique was presented in [1] to consider energy consumption with switching, transmission, data processing and data storage evaluated with three cloud computing services. Though, power consumption is reduced by cloud storage services. But, the data storage capacity remained unsolved. A space optimization technique called the proxy-based approach was designed in [2] for presenting space saving through database schema modification, particularly by dropping attributes from the schema under consideration. However, the data storage time was higher.

An environment-aware paradigm for virtual slices was designed in [3] for improving the energy efficient and handling intermittent power resources in green cloud environment. However, the global optimization problem and quality of services hosted by the GSN is remained unsolved. An Energy-aware performance analysis framework was designed in [4] to decrease the total energy consumption of cloud server's focus into different running modes and to achieve the maximum performance accuracy in green cloud computing. But, it lacks scalability.

Prediction based green scheduling algorithm was developed in [5] for improving the average resource utilization of the system and reducing energy consumption in green cloud computing. But, the power aware and thermal aware scheduling system for reduce energy savings from physical servers remained unaddressed. An energy efficient algorithm was presented in [6] to reduce power consumption and ensures the stochastic resources and probability to reduce the number of required servers in green cloud computing.

A Holistic resource management framework was introduced in [7] to determine the best tradeoff among maximizing revenue with minimum energy costs. Though, Greenhead provides near-optimal solution during maximum computational time to reduce acceptance ratio of VDCs partitioning. Virtualization security assurance architecture was designed in [8] that provide policy based trust management makes access control to the resources pool and keep privacy and cost using trust federation mechanism. However, the reliable and scalable NetApp operating system cannot construct by

the advanced virtualization technologies.

A genetic-based optimization algorithm for Chip Multiprocessor (CMP) provided with PCM memory was implemented in [9] for green clouds. The genetic-based algorithm schedules and allocates jobs to cores in the CMP system and presents a PCM MLC configuration which balances the performance of PCM memory and the efficiency. A novel architecture was designed in [10] to implements a resource monitor of machine status factors for example CPU, memory and network usage to optimize the virtualization system and data storage system.

2. RELATED WORKS

A high performance, load balancing and able-to-be-replicated system was developed in [11] to present data storage for private cloud users via a virtualization system. A storage optimization technique was introduced in [12] to address the file handling and security issues related to cloud storage which resulting in improved the cloud storage capacity and improved the speed of data deduplication. An Efficient Big Data Storage System was designed in [13] to present a single, highly scalable, proficient and performance-aware big data storage system.

The Index Name Servers (INS) was presented in [14] to process the file compression, chunk matching, data de-duplication, real-time feedback control, IP information, busy level index monitoring and also file storage, optimized node selection and server load balancing. A review of different techniques designed for improving the file access efficiency in cloud data storage was discussed in [15]. An efficient data storage security system was intended in [16] for cloud service in which partition process allows storing of the data in easily and effective manner and also presents way for flexible access of data.

An Energy Aware Cloud Service Provisioning Approach was explained in [17] for service provisioning in green cloud environment and minimizing the unnecessary power utilization with aid of pre processed data by initiating automatic live migration. But, it has limited number of cloud computing entities such as virtual machine and datacenter in pre processing data for provisioning cloud service. A novel framework was designed in [18] to efficiently store the small files in cloud storage systems and to optimize file accessing in a cloud system.

A novel approach was developed in [19] to address the cloud storage issues and improving the performance of task scheduling, processing and storage of space optimization for DaaS on the cloud. However, the possibility of server failures remained unaddressed. An enhanced multi-objective optimization algorithm was employed in [20] that

integrates the generalized optimization based learning strategy and the smart time scheduling strategy for a multi-objective task scheduling problem. This multi-objective optimization algorithm improves the utilization of renewable energy and reduces the make span of tasks and the total energy consumption. But, energy consumption of file synchronization was not considered.

Based on the aforementioned techniques and methods presented, a novel framework called Multi Objective Tiger Hash Storing (MOTHS) Technique is designed for efficient green cloud data storage to rendering multiple cloud services in green cloud environment.

The rest of the paper organized as follows. In Section 2, a review of different techniques designed for efficient cloud data storage and service provisioning is presented. In Section 3, the proposed MOTHS Technique is explained with the help of neat architecture diagram. In Section 4, the experimental setting is discussed with exhaustive analysis of results described in Section 5. In Section 6, the concluding remarks are explained.

3. MULTI OBJECTIVE TIGER HASH STORING (MOTHS) TECHNIQUE

Cloud storage is a structure for networked storage resources in which data are stored in networked clusters of storage offered by third parties called Cloud Service Providers (CSPs). CSPs virtualizes their resources according to users' requirements and depict them as a universal storage platform where users can outsource their data in an on-demand way. Nowadays, the cloud storage infrastructure is based on highly virtualized architectures and has the benefits of agility, scalability, elasticity and multi-tenancy.

To improve the data storage capacity in green cloud environment, MOTHS technique is designed. The main objective of MOTHS technique is to achieve the efficient cloud data storage and to improve the speed of data accessibility. The MOTHS technique is used Tiger Hash Tree structure for storing the green cloud data where three tuple are used to define the node of the new Tiger tree that records the position of the corresponding node. Therefore, the cloud users can easily extract the required information from the Tiger tree by computing the root value directly without retrieving the whole Tiger tree. Hence, the speed of data accessibility using MOTHS technique is improved significantly. The architecture diagram of green cloud data storage is shown in below Figure 1.

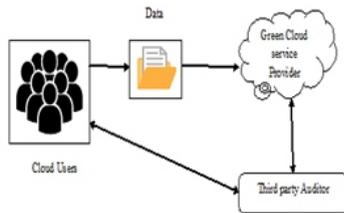


Figure 1 Architecture Diagram of Green Cloud Data Storage

As shown in Figure 1, the green cloud data storage architecture comprises of users, Cloud service Provider and Third party Auditor. Cloud service providers contain a cluster of the software and hardware resources and present flexible online computing and data storing services. Users outsource their data to the remote cloud storage to keep the storing cost and employ the computation resources provided by the service provider. The third-party auditor is a reliable, partially trusted and independent entity that can offer assess and arbitration if required. The cloud users interact with cloud server for accessing and updating stored data.

3.1 Tiger Tree Construction for Efficient Green Cloud Data Storage

The green cloud users stores data in cloud and also they maintain data its own locally in the green cloud data storage. Green Cloud data storage includes two different entities, the cloud user and the cloud

server. The Cloud User 'CU₁ = CU₁, CU₂, ..., CU_n' store their data in the cloud heavily depend on the cloud environment for storing heterogeneous data. Conversely, the Cloud Server 'CS' is administered by Cloud Service Provider 'CSP' with the purpose of providing services of data storage, managing computation resources and handling data storage in an efficient manner. The MOTHS technique is used Tiger Tree for achieving efficient green cloud data storage to provide a multiple cloud service with different user tasks. The overall architecture diagram of MOTHS technique is shown in below Figure 2.

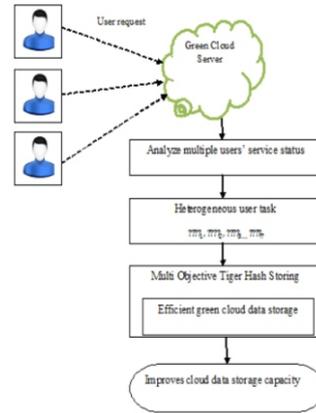


Figure 2 Architecture Diagram of MOTHS Technique

As shown in Figure 2, initially the request is sent from the cloud users to Green Cloud Server. Then, MOTHS technique analyzes the multiple users' service status and their heterogeneous user tasks for rendering the services. After that, MOTHS technique is construct the tiger hash tree for efficiently storing the c UT₁, UT₂, UT₃...UT_n d data with aiming at improving the data storage capacity and reducing the storage time.

The MOTHS technique is used a tiger hash tree for efficient data storage in green cloud environment. Tiger hash tree is a tree where each non-leaf node is labeled as the hash value of its children nodes and every leaf node is labeled as the hash value of a data block. There is a root on the top of the tiger tree. The key objective of employing tiger hash tree in MOTHS technique is to improve the green cloud data storage capacity with minimal storage time. A typical tiger hash tree structure that consisting of n leaf nodes is mathematically formulated as follows,

$$THT = \{N_i | N_i = h(x_i), 1 \leq i \leq n \quad (1)$$

From the equation (1), represents the number of nodes in tree and h () denotes a hash function where indicates the different number of data stored in tiger tN_ie. Then, the value of the non-leaf node N_i is mathematically expressed as

$$N_i = h(N_i^l | N_i^r) \quad (2)$$

From the equation (2), and characterizes N_i' s left child and right child nodes respectively. The root node of THT is represented as N_{root}. A tiger hash tree strN_i ure fN_i efficient green cloud data storage is shown in below Figure 3.

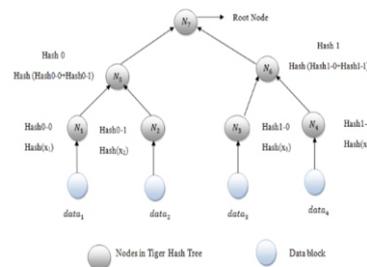


Figure 3 Tiger Hash tree for Green Cloud Data Storage

As shown in Figure 3, all non-leaf nodes are labeled with the hash value of its children nodes and all leaf nodes are labeled with the hash value of a data block for improving the data storage capacity. The user requested the data is retrieved from the tiger hash tree based on the hash value of nodes. The tiger hash tree structure also allows dynamic operations such as insert, delete etc. The algorithmic process of tiger hash tree construction for storing the green cloud data is shown in below.

```
//Tiger Hash Tree Based Green Cloud Data Storage Algorithm
Input: Nodes  $N_1=N_1, N_2, N_3, N_4, \dots, N_n$ 
Output: Improved Data Storage capacity with minimum storage time
Step 1: Begin
Step 2: Construct tiger hash tree using (1)
// Insertion Operation
Step 3: If the tiger hash tree is not full, then insert the data objects
Step 4: Allocate new leaf node and then insert the newly provided cloud user data objects
Step 5: insert operation=THT[add new( $N_1, N_2, N_3, N_4, \dots, N_n$ )]
Step 6: If the root node splits, then the root node has one key and two intermediate nodes
Step 7: Repeat step until found no iteration
Step 8: End if
// Removal Operation
Step 8: Root node on tiger hash tree where it belongs
Step 9: Remove the unwanted leaf node as
Removal operation=THT[remove old( $dN_1, N_2, N_3, N_4, \dots, N_n$ )]
Step 10: If the leaf node removed, then the leaf node has two intermediate nodes
Step 11: Repeat step until found no iteration
Step 12: End
```

Algorithm 1 Tiger Tree Based Green Cloud Data Storage

By the construction of tiger tree, MOTHS technique makes efficient green cloud storage system that allows storing of data and permits the user to work with the data without any difficulty, trouble of the resources. Therefore, MOTHS technique improves the data storage capacity of green cloud environment in an effective manner.

3.2 Data Access from Tiger Hash Tree

Let us consider green cloud environment consisting of many users and servers. Data are retrieved from the green cloud storage as per the cloud users request or demand. Data access process from the green cloud storage is shown in below Figure 4.

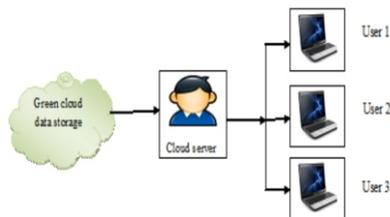


Figure 4 Data Access from Green Cloud Data Storage

As shown in Figure 4, the data can be retrieved or restored from the server ensuring the data correction. Data accessed from cloud server provides the services in secured manner. In tiger tree, each node is aware of its relative position to its parent node and its position information is bound with the value of each node. The nodes in tiger hash tree are denoted as $N_i = N_i, N_p, N_r, N_v, \dots, N_n$ in which Node N_i has three tuples like $N_i = (N_i, p, r, v)$. Here, N_i, p denotes node N_i 's relative position to its parent node, N_i, r indicates the number of N_i 's leaf nodes and N_i, v signifies the value of the node N_i . The node N_i 's relative position to its parent node are estimated by using following mathematical formula,

$$N_i, p = \begin{cases} 0, & \text{if } N_i \in \text{left subtree} \\ 1, & \text{if } N_i \in \text{right subtree} \\ \text{null}, & \text{if } N_i \text{ is root} \end{cases} \quad (3)$$

Besides, the number of N_i 's leaf nodes is estimated by using following mathematical formula,

$$N_i, r = \begin{cases} N_i^l, r + N_i^r, r, & \text{if } N_i \text{ is non_leaf node} \\ 1, & \text{if } N_i \text{ is leaf node} \end{cases} \quad (4)$$

In addition, the value of the node N_i is estimated by using following mathematical formula,

$$N_i, v = \begin{cases} h(N_i, p || N_i^l, v || N_i^r, v || N_i, r), & \text{if } N_i \text{ is non_leaf node} \\ h(N_i, p || x_i || 1), & \text{if } N_i \text{ is leaf node} \end{cases} \quad (5)$$

From the Figure 2, N_1 is a leaf node that indexes the data block x_1 and N_{-1} is in the left sub tree of its parent node N_2 . By using the formula (3), (4), (5) for $N_i, p=0$ and $N_i, r=1$ and $N_i, v=h(0 || x_1 || 1)$. Similarly, we can obtain all the nodes information of above tiger tree (Figure 2) which is listed in below table.

Table 1 Nodes Information of Tiger hash Tree

Node number	N_i, p	N_i, r	N_i, v
N_1	0	1	$h(0 x_1 1)$
N_2	1	1	$h(1 x_2 1)$
N_3	0	1	$h(0 x_3 1)$
N_4	1	1	$h(1 x_4 1)$
N_5	0	2	$h(0 N_1, v N_2, v 2)$
N_6	1	2	$h(1 N_3, v N_4, v 2)$
N_7	Null	4	$h(\text{Null } N_5, v N_6, v 4)$

The key benefit of tiger hash tree is that each node of the tree is aware of the relative location to its parent nodes. Therefore, for accessing data, MOTHS technique does not need to retrieve the complete tiger tree. To achieve this, MOTHS technique adopt a 3-tuple to define the node of the tiger tree. The 3- tuple vector records the location of the corresponding node and hence green cloud user can easily access the data stored in tiger tree via computing the root value directly (no need to get the whole tiger tree back). For that reason, the data accessing time using MOTHS technique is significantly reduced. The service rendering process using MOTHS technique is shown in below Figure 5.

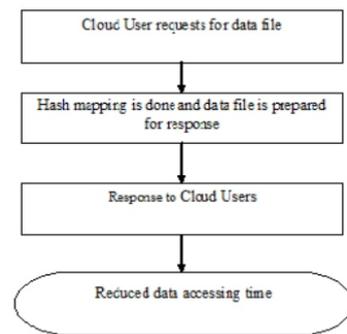


Figure 5 Service Rendering Process Using MOTHS Technique

As shown in Figure 5, when the cloud user request for data, MOTHS technique performs hash mapping (by using node information in table 1) and data is prepared for response to cloud users which resulting in reduced data accessing time. Therefore, MOTHS technique maximizes the service distribution and also reduced user task service rendering complexity in green cloud environment.

4. EXPERIMENTAL SETTINGS

The Multi Objective Tiger Hash Storing (MOTHS) Technique is implemented using java language with CloudSim simulator environment. The CloudSim simulator employs Amazon Dataset for conducting the experimental work. The CloudSim simulator performed on green Cloud environment presents varied rendering services with the available resources. The particular toolkit is chosen as a simulation platform in green Cloud environments. The Cloudsim necessitates minimal time to perform user requested service provisioning task. The emerging green cloud computing

infrastructures and application services allow the users to focus on precise green cloud system design with different type of processors. A green cloud data center comprises of many processors with varying range.

The performance of MOTHS Technique is compared against with existing two methods namely Energy consumption analysis technique [1] and proxy- based approach [2]. The performance of MOTHS Technique is measured in terms of data storage capacity, storage time, energy consumption and data accessing time.

5. RESULTS AND DISCUSSIONS

In this section, the result analysis of MOTHS Technique is estimated. The efficiency of MOTHS Technique is compared against with Energy consumption analysis technique [1] and Proxy- Based Approach [2] respectively. The performance of MOTHS Technique is evaluated along with the following metrics with the help of tables and graphs.

5.1 Measurement of Data Storage Capacity

In MOTHS Technique, data storage capacity refers to the data storage performed in green cloud environment based on the number of transaction threads (i.e. users' tasks), data to be stored and the time taken for data storage process. The data storage capacity is measured in terms of kilo bits per second (kbps) and is mathematically formulated as,

$$\text{Data Storage Capacity} = T * d * \text{time} \quad (6)$$

From the equation (6), the green cloud data storage capacity is obtained where 'd' states the data to be stored with respect to time 'time' respectively. While the data storage capacity is higher, the method is said to be more efficient.

Table 2 Tabulation for Data Storage Capacity

Number of User tasks	Data Storage Capacity (kbps)		
	Energy consumption analysis technique	Proxy- Based Approach	MOTHS Technique
5	63.48	75.26	84.56
10	65.68	76.98	86.47
15	66.92	78.14	87.16
20	69.47	79.54	88.66
25	71.26	81.26	91.47
30	74.59	83.14	93.14
35	75.35	86.47	95.61
40	77.66	87.98	96.78
45	79.14	89.47	98.21
50	82.14	91.58	99.14

Table 2 demonstrates the comparative result analysis of data storage capacity using three methods based on the different number of user tasks in the range of 5-50. From the table value, it clear that the data storage capacity using proposed MOTHS Technique is higher as compared to Energy consumption analysis technique [1] and Proxy-Based Approach [2]. Based on the table value, the graph is drawn in below Figure 6.

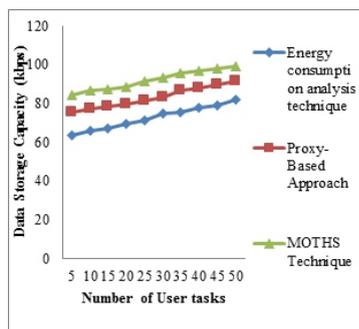


Figure 6 Measurement of Data Storage Capacity

Figure 6 explains the impact of data storage capacity versus diverse of user tasks in the range of 5-50. As illustrated in figure, the proposed MOTHS technique provides better data storage capacity as compared to Energy consumption analysis technique [1] and Proxy-Based Approach [2]. Besides, while increasing the number of user task for rendering the multiple services, data storage capacity is also gets increased using all the three methods. But comparatively, the data storage capacity using proposed MOTHS Technique is higher as compared to other existing works. This is because of tiger tree construction in MOTHS Technique. The proposed MOTHS technique employed tiger tree for efficiently storing user requested green cloud data. This in turn assists for improving the data storage capacity in an effective manner. As a result, the proposed MOTHS Technique is improved the cloud storage capacity by 21% as compared to Energy consumption analysis technique [1] and 10% as compared to Proxy- Based Approach [2] respectively.

5.2 Measurement of Storage Time

In MOTHS Technique, storage time measures the amount of time taken for storing the data. Therefore the storage time is defined as the product of transaction data size 'TDS' and time taken for each data storage 'Time (DS)'. The storage time is measured in terms of milliseconds (ms) and mathematically formulated as

$$\text{Storage Time} = \text{TDS} * \text{Time (DS)} \quad (7)$$

From the equation (7), the amount of time taken for storing cloud data is obtained. While the storage time is lower, the method is said to be more efficient.

Table 3 Tabulation for Storage Time

Transaction data size (KB)	Storage Time (ms)		
	Energy consumption analysis technique	Proxy- Based Approach	MOTHS Technique
50	21.8	18.6	11.2
100	34.5	27.1	23.5
150	46.4	35.6	34.1
200	55.6	46.2	41.6
250	62.7	54.8	49.8
300	69.8	63.6	56.8
350	75.1	72.1	62.3
400	84.3	81.6	70.7
450	90.5	89.5	78.6
500	99.4	96.4	85.2

The result analysis of storage time using three methods based on the different number of transaction data size in the range of 50-500KB is depicted in Table 3. While the 100KB data is taken for storage, MOTHS Technique takes 23.5ms storage time whereas the Energy consumption analysis technique [1] and Proxy- Based Approach [2] takes 34.5 ms and 27.1 ms respectively. Therefore, the storage time using proposed MOTHS Technique is lower as compared to other exiting methods. Based on the table value, the graph is designed in below Figure 7.

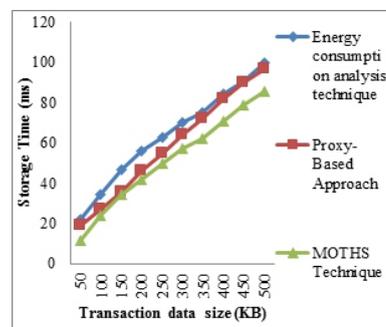


Figure 7 Measurement of Storage Time

Figure 7 portrays the impact of storage time versus dissimilar of user

tasks in the range of 50-500. As shown in figure, the proposed MOTHS Technique provides better storage time as compared to Energy consumption analysis technique [1] and Proxy- Based Approach [2]. Besides, while increasing the number of transaction data size, storage time is also gets increased using all the three methods. But comparatively, the storage time using proposed MOTHS Technique is lower as compared to other existing works. This is due to tiger tree construction in MOTHS Technique. With the support of tiger tree structure, MOTHS technique constructs efficient green cloud storage system which allows storing of data and permits the user to work with the data without any complexity, trouble of the resources. This in turn helps for reducing the storage time in a significant manner. Therefore, the proposed MOTHS Technique is reduced the storage time by 33% as compared to Energy consumption analysis technique [1] and 18% as compared to Proxy- Based Approach [2] respectively.

5.3 Measurement of Energy Consumption

In MOTHS Technique, the energy consumption measures the total amount of energy taken for providing the different services in the form of user requested tasks. The energy consumption is measured in terms of Joules (J) and mathematically formulated as below,

$$\text{(Average energy consumed for each task)} = \frac{\text{Total energy}}{\text{Number of user tasks}} \tag{8}$$

From the equation (8), the amount of energy taken for providing the different services is obtained. While the storage time is lower, the method is said to be more efficient.

Table 4 Tabulation for Energy Consumption

Number of user tasks	Energy Consumption (J)		
	Energy consumption analysis technique	Proxy- Based Approach	MOTHS Technique
5	46	38	29
10	49	41	32
15	52	43	36
20	56	46	41
25	59	49	43
30	62	53	46
35	64	55	48
40	69	58	52
45	71	61	55
50	75	64	59

The result analysis of energy consumption using three methods based on the varied number of user tasks in the range of 5-50 is illustrated in Table 4. While the number of user task is 25, MOTHS Technique consumes 43J energy for rendering the services to the corresponding cloud users in green cloud environment whereas the Energy consumption analysis technique [1] and Proxy- Based Approach [2] consumes 59J and 49J respectively. Therefore, the energy consumption using proposed MOTHS Technique is lower as compared to other exiting methods. Based on the table value, the graph is plotted in below Figure 8.

Figure 8 describes the impact of amount of energy taken for rendering the multiple services versus diverse of user tasks in the range of 5-10. As exposed in figure, the proposed MOTHS Technique provides better energy consumption as compared to Energy consumption analysis technique [1] and Proxy- Based Approach [2].

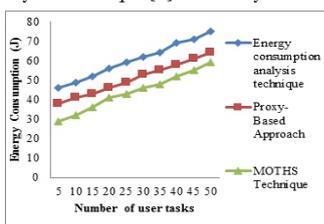


Figure 8 Measurement of Energy Consumption

Besides, while increasing the number of user tasks, the amount of energy taken for rendering the multiple services is also gets increased using all the three methods. But comparatively, the energy consumption using proposed MOTHS Technique is lower as compared to other existing works. This is because of Tiger Tree structure employed for storing the green cloud data in which three tuple are utilized to describe the node that records the location of the corresponding node. Hence, the cloud users can easily access the necessary information from the Tiger tree through computing the root value directly without retrieving the whole Tiger tree. This in turn supports for improving the data storage capacity and reducing the storage time which results in reduced energy consumption for providing multiple services in green cloud environment. As a result, the proposed MOTHS Technique is reduced the energy consumption by 39% as compared to Energy consumption analysis technique [1] and 17% as compared to Proxy- Based Approach [2] respectively.

5.4 Measurement of Data Accessing Time

In MOTHS Technique, the data accessing time measures the amount of time taken to respond to a user's request for data. The data accessing time is measured in terms of milliseconds and mathematically formulated as follows,

$$\text{Data accessing time} = \text{No. of user requests} \times \text{time (responding one user request)} \tag{9}$$

From the equation (9), the data accessing time is obtained. While the data accessing time is lower, the method is said to be more efficient.

Table 5 Tabulation for Data Accessing Time

Number of user requests	Data Accessing Time (ms)		
	Energy consumption analysis technique	Proxy- Based Approach	MOTHS Technique
5	14.9	11.6	6.2
10	20.4	15.8	10.6
15	28.6	22.4	17.4
20	32.6	25.6	20.7
25	40.7	32.5	28.3
30	46.4	40.9	31.5
35	48.6	42.7	39.7
40	55.6	46.1	41.2
45	60.2	50.7	46.8
50	68.9	59.6	55.6

The data accessing time for rendering the multiple services using three methods based on the different number of user requests in the range of 5-50 is demonstrated in Table 3. While the number of user request is 15, MOTHS Technique acquires 17.4ms data accessing time whereas the Energy consumption analysis technique [1] and Proxy- Based Approach [2] acquires 28.6ms and 22.4ms respectively. Thus, the data accessing time using proposed MOTHS Technique is higher as compared to other exiting methods. Based on the table value, the graph is plotted in below Figure 9.

Figure 9 depicts the impact of data accessing time versus different of user request in the range of 5-10. As shown in figure, the proposed MOTHS Technique provides better data accessing time as compared to Energy consumption analysis technique [1] and Proxy- Based Approach [2].

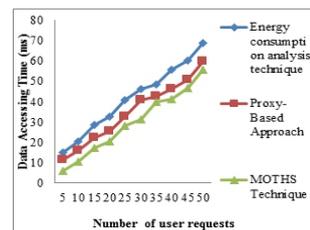


Figure 9 Measurements of Data Accessing Time

Besides, while increasing the number of user request, data accessing time for rendering the different services is also gets increased using all the three methods. But comparatively, the data accessing time using proposed MOTHS Technique is lower as compared to other existing works. This is owing to tiger tree construction in MOTHS Technique in which all node of the tree is aware of the relative location to its parent nodes and 3- tuple vector records the location of the corresponding node. Therefore, for accessing data, MOTHS technique does not need to retrieve the complete tiger tree. This in turn helps to reduce the data accessing time in an efficient manner. As a result, the proposed MOTHS Technique is reduced the data accessing time by 56% as compared to Energy consumption analysis technique [1] and 27% as compared to Proxy- Based Approach [2] respectively.

6. CONCLUSION

A Multi Objective Tiger Hash Storing (MOTHS) Technique is devolved for improving the energy storage during the multiple communication between diverse virtual cloud servers in green cloud computing. The main goal of MOTHS Technique is to enhance the cloud data storage capacity with minimal data accessing time in green cloud environment. This objective is achieved by constructing the tiger hash tree for efficient green cloud data storage. With the aid of tiger hash tree, MOTHS Technique builds effective green cloud storage system that allows storing of data and permits the user to work with the data without any complexity, trouble of the resources. The effectiveness of MOTHS Technique is tested with the metrics such as data storage capacity, storage time, energy consumption and data accessing time. With the experiments conducted for MOTHS Technique, it is observed that the data storage capacity and data storage time is provided more accurate results as compared to state-of-the-art works. The experimental results demonstrates that MOTHS Technique is provides better performance with an improvement of data storage capacity by 16% and the reduction of data accessing time by 42% when compared to the state-of-the-art works.

REFERENCES

- Jayant Baliga, Robert W. A. Ayre, Kerry Hinton, and Rodney S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport", Proceedings of the IEEE, Volume 99, Issue 1, Pages 149 – 167, 2011
- Nurul A. Emran, Noraswaliza Abdullah, Mohd Noor Mat Isa, "Storage Space Optimisation for Green Data Center", Procedia Engineering, Elsevier, Volume 53, Pages 483-490, 2013
- Kim Khoa Nguyen, Mohamed Cheriet, "Environment-Aware Virtual Slice Provisioning in Green Cloud Environment", IEEE Transactions on Services Computing, Volume 8, Issue 3, Pages 507 – 519, 2015
- Dario Bruneo, Audric Lhoas, Francesco Longo and Antonio Puliafito, "Modeling and Evaluation of Energy Policies in Green Clouds", IEEE Transactions on Parallel and Distributed Systems, Volume 26, Issue 11, Pages 3052 – 3065, 2015
- Supriya Kingler, Rajesh Kumar, Anju Sharma, "Prediction Based Proactive Thermal Virtual Machine Scheduling in Green Clouds", The Scientific World Journal, Hindawi Publishing Corporation, Volume (2014), Pages 1-12, 2014
- Wenyng Yue and Qiushuang Chen, "Dynamic Placement of Virtual Machines with Both Deterministic and Stochastic Demands for Green Cloud Computing", Mathematical Problems in Engineering, Hindawi Publishing Corporation, Volume 2014, Pages 1-11, 2014
- Ahmed Amokrane, Mohamed Faten Zhani, Rami Langar, Raouf Boutaba, and Guy Pujolle, "Greenhead: Virtual Data Center Embedding across Distributed Infrastructures", IEEE Transactions on Cloud Computing, Volume 1, Issue 1, Pages 36 – 49, 2013
- Jianxin Li , Bo Li , Tianyu Woa, Chunming Hua, Jinpeng Huai , Lu Liu , K.P. Lamc, "CyberGuarder: A virtualization security assurance architecture for green cloud computing " ,Future Generation Computer Systems , Elsevier, Volume 28, Pages 379–390, 2012
- Meikang Qiu, Zhong Ming, Jiayin Li, Keke Gai, Ziliang Zong, "Phase-Change Memory Optimization for Green Cloud with Genetic Algorithm" IEEE Transactions on Computers, Volume 64, Issue 12, Pages 3528 – 3540, 2015
- Chao-Tung Yang, Wen-Chung Shih, Chih-Lin Huang, "Implementation of a Distributed Data Storage System with Resource Monitoring on Cloud Computing", Advances in Grid and Pervasive Computing, Springer, Pages 64-73, 2012
- Chao-Tung Yang, Wen-Chung Shih, Chih-Lin Huang, Fu-Cheng Jiang, William Cheng-Chung Chu, "On construction of a distributed data storage system in cloud", Computing, Springer, Volume 98, Issue 1, Pages 93–118, January 2016.
- Prajakta B. Mane, N. B. Pokale, "Privacy Preserving and Storage Optimization on Cloud", International Journal of Science and Research (IJSR), Volume 4, Issue 9, Pages 41-44, September 2015
- Yang Li, Li Guo, Yike Guo, "An Efficient and Performance-Aware Big Data Storage System", Cloud Computing and Services Science, Springer, Pages 102-116, 2013
- Tin-Yu Wu, Jeng-Shyang Pan and Chia-Fan Lin, "Improving Accessing Efficiency of Cloud Storage Using De-Duplication and Feedback Schemes", IEEE Systems Journal, Volume 8, Issue 1, Pages 208 – 218, March 2014
- Vidhya N. Gavali, Rahul P. More, Suvarna D. Potdukhe, "A Review on Cloud Storage

- Performance to Improve File Accessing Efficiency", International Journal of Engineering Research & Technology (IJERT), Volume 3, Issue 11, Pages 989-992, November-2014
- A.R.Zade, Shaikh Umar, Potghan Rahul, Rale Sagar and Borade Sagar, "Improving Cloud Data Storage Using Data Partition and Recovery", International Journal of Engineering and Computer Science, Volume 4, Issue 1, Page No. 10144-10249, January 2015
- T. Arthi, H. Shahul Hameed, "Energy Aware Cloud Service Provisioning Approach For Green Computing Environment", IEEE 2013 International Conference on Energy Efficient Technologies for Sustainability (ICEETS), Pages 139-144, 2013
- Xiong Fu, Wenjie Liu, Yeliang Cang, Xiaojie Gong and Song Deng, "Optimized Data Replication for Small Files in Cloud Storage Systems", Hindawi Publishing Corporation, Mathematical Problems in Engineering, Volume 2016, Article ID 4837894, Pages 1-8, 2016
- Klathem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi, "Partial Storage Optimization and Load Control Strategy of Cloud Data Centers", Hindawi Publishing Corporation, The Scientific World Journal, Volume 2015, Article ID 836561, Pages 1-14, 2015
- Hongtao Lei, Rui Wang, Tao Zhang, Yajie Liu, Yabing Zha, "A multi-objective evolutionary algorithm for energy-efficient scheduling on a green data center", Computers & Operations Research, Elsevier, Volume 75, Pages 103–117, 2016.