Effective Data Overflow Management Through Cloud Computing: Analysis Of Load Managment Technique

Dr. Suma T
Dept of CSE
Sri Venkateswara College of Engineering
Bengaluru,India

Ms Aishwarya S
Dept of CSE
Sri Venkateshwara College of Engineering.
Bengaluru, India

Ms Gowthami K
Dept of CSE
Sri Venkateshwara College of Engineering.
Bengaluru, India

Ms Anusha C N
Dept of CSE
Sri Venkateshwara College of Engineering.
Bengaluru, India

Ms Dhanuja K M
Dept of CSE
Sri Venkateshwara College of Engineering.
Bengaluru, India

Abstract— *The tremendous issues of handling the large flood of data have been brought about by the cloud computing field's fast development. By 2020, 247 EB will be achieved by data analytics and IoT-based applications, according to Cisco's Global Cloud Index. Load Managing strategies are critical in tackling cloud environments' data overflow. The study promotes Round Robin Load Managing and highlights how well it manages data overflow. Based on set policies, load managing techniques such IP hash, weighted round-robin, least connections, and round-robin allow the effective allocation of incoming traffic across active targets.*
One important remedy that is emphasized is cloud load management, which uses software-based techniques to keep an eye on the target's health inside a cluster. In order to avoid data loss and service outages, it foresees and manages unexpected increases in traffic, rerouting it to servers that aren't as busy. The book offers explanations on how the load distribution techniques are used, and a Python*example of the Round Robin algorithm.*

Keywords— *Cloud computing , Load Managing, Round-robin , Hardware-based load managing, Software-based Load Managing .*

I. .INTRODUCTION :

The cloud computing field is expanding quickly and is projected to do so in the future. This is because cloud computing may provide economies of scale, enhanced worldwide accessibility, and simplified, "outsourced" management and configuration at more affordable prices.
In figure 1 Block diagram of cloud computing is given.
According to Cisco's Global Cloud Index (GCI) study [1], data analytics and IoT-based applications will increase by a factor of 2.7 by 2020.

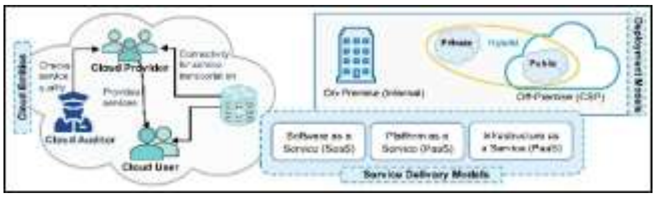


Figure 1: Block diagram of cloud computing

Each of the four essential Vs that comprise big data—volume, veracity, velocity, and variety—varies in the data generated by electronic devices. Moreover, this enormous volume of data will have grown tenfold by 2020, to a total of 247 EB [1]. As a result, the rapid rise in DC traffic may make it more challenging to control various network backbone resources, including load, bandwidth, and connection quality. As end users may choose to use a combination of public and private cloud services and are connected by public Internet or their own private WANs, major concerns like latency, bandwidth, data rate, fault tolerance, and security will have a significant impact on the performance of big data management in MCE. However, the sheer amount of flawlessly streamed data might overload the network infrastructure underneath, rendering any solution developed for this environment useless. Cloud computing has become a buzzword in today's IT industry, even though it's one of the major modern innovations that has revolutionized this industry. In a cloud computing environment, technology is introduced through services. With the use of different computer or smartphone models, it allows the secure and affordable utilization of servers, storage, and apps at any time.[3] One important topic of research in cloud computing systems is task scheduling. With the number of cloud clients increasing, service providers' main objectives are to maximize profit and offer enough access to remote resources. Task scheduling assigns tasks to customers and associates them with appropriate and reachable virtualized resources based on an efficient algorithm. [3] Cloud computing systems typically have three types of work scheduling algorithms: (1) heuristic algorithms, such as Min-Min and Max-Min algorithms; (2) traditional algorithms, such as round-robin (RR), shortest job first (SJF), largest job first (LJF), and first come first serve (FCFS); and (3) meta-heuristic algorithms, such as particle swarm optimization (PSO) and ant colony optimization algorithm (ACO).
In this paper we are going to use Round Roubin Load Managing technique for controlling the overflow of the data in cloud

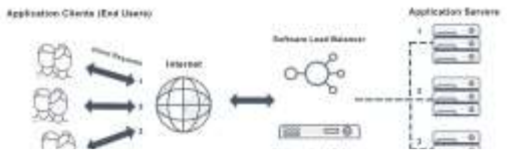


Figure 2:Round Roubin Load managing

II. LITERATURE SURVEY:

According to the literature review on load management strategies in cloud computing, these strategies are essential for allocating incoming traffic to active targets in accordance with a set strategy. A variety of methods are employed, such as hardware-based load managing, which is more prevalent in business data centers, and software-based cloud load managing. Numerous load managing techniques exist, including weighted round-robin, IP hash, least connections, and round-robin. Workloads can be distributed among several back-end systems or services with the use of services like Rackspace's Cloud Traffic Managers and those provided by other cloud service providers.

A review of the literature on data overflow control in cloud computing settings indicates that there are a number of difficulties, including cloud bursting's applicability for certain applications, security concerns during data transmission, and limited visibility across cloud providers. The necessity for further research and development in this field is highlighted by the proposals for security solutions and quality of service measures to solve these problems. Nonetheless, not every application may benefit from cloud bursting, particularly those that manage sensitive data or involve crucial business processes. Sensitive apps and data may be better kept in a regulated on-premises environment in such circumstances.

Organizations can apply a number of important tactics, according to a review of the literature on best practices and strategies for handling data overflow in cloud computing. These include utilizing cloud bursting to effectively distribute resources and eliminate disruptions, storing sensitive or low-latency applications on private clouds and on-premises infrastructure, closely monitoring resource use, and adopting a cloud-based data management architecture. In cloud computing environments, data deduplication is an essential technique for maximizing storage and reducing data overflow. Systems like Over Flow may drastically lower the amount of storage space needed to hold data by compressing and deduplicating it, freeing up important resources for other uses.

technique guarantees effective and safe data storage while lowering the quantity of network traffic needed to move data between sites. Organizations may efficiently control cloud computing data overflow and maintain dependable and efficient systems by putting these methods and solutions into practice.

"Transparent and Flexible Network Management for Big Data Processing in the Cloud" describes Flow Comb, a network management system for Big Data processing applications that uses software agents deployed on application servers[1] to forecast network transfers, sometimes even before they begin1. Using a centralized decision engine, Flow Comb is invisible to the application and gathers data movement information from agents to plan forthcoming flows on pathways to prevent network congestion.

The study titled "Cloud Computing: The Future of Big Data Management" delves at the development of Big Data analytics in cloud computing settings, charting significant turning points, advancements in technology, and influential contributions that have molded the contemporary terrain1. In addition to cutting-edge methods and models that take advantage of cloud computing technologies like Big Data-as-a-Service (BDaaS) and Analytics-as-a-Service (AaaS), it also examines applications, prospects, and limitations.

The study titled "Big Data Processing in Cloud Computing Environments" delves at the methods and difficulties involved in handling and analyzing large amounts of data inside cloud systems. It gives a summary of data storage and Cloud-centric Big Data placement solutions that highlight the connection between these two components for better Big Data management.

The article "Intelligent Data Management and Security in Cloud Computing" discusses methods for protecting and managing sensitive data with an emphasis on data services security. In order to protect data from unwanted acquisition, it presents linguistic threshold methods and cryptographic sharing algorithms. The methods covered here are applicable to cloud management procedures as well as different tiers of data management.

Security will have a huge influence on how well large data management works in MCE. But the enormous amount of flawlessly transmitted data might overwhelm the network infrastructure underneath, making any solution designed for this situation useless.

The pips Cloud platform, which integrates modern cloud computing with high-performance computing (HPC) approaches, is introduced in the paper "pips Cloud: High Performance Cloud Computing for Remote Sensing Big Data Management and Processing." It solves the problems of compute-intensive and data-intensive applications in environmental monitoring by enabling large-scale remote sensing data processing as on-demand real-time services.

A method called Information Flow Control (IFC) is introduced in the paper "Seeing through the clouds: Managing data flow and compliance in cloud computing." IFC allows for auditable, fine-grained data management as it flows throughout cloud platforms. By enhancing visibility and control over data transfers inside and across cloud services and applications, it satisfies legal and regulatory requirements.

III. PROPOSED SYSTEM:

The proposed system for managing data overflow in cloud computing environments involves implementing a strategy that uses quality of service parameters such as Resource Scheduling Efficiency, Energy uses, Response Time, and Average Success Rate to handle overflow traffic. Additionally, it emphasizes the importance of strong encryption and security protocols to ensure data security during cloud bursting, while also advocating for measures to account for inconsistencies across different cloud providers' protocols and frameworks.

Implementing a strategy that uses various load managing techniques, such as round-robin, least connections, IP hash, and weighted round-robin. These techniques help determine in advance which servers are more likely to become overburdened, allowing traffic to be diverted to other regions that can handle more traffic. Additionally, the system leverages autoscaling capabilities, which do not require pre-warming, enabling faster resource allocation and deployment to effectively manage data overflow.

Implementing a strategy that uses key techniques such as data deduplication, data compression, and geo-replication. These techniques, combined with a uniform data management system like Over Flow, can optimize storage and reduce data overflow by minimizing transfer cost and time. Additionally, the system leverages cloud-based data management infrastructure for automated updates, freeing up users to focus on other tasks while the necessary updates are implemented into the cloud system.

IV. HOW DO THESE TECHNIQUES HELP IN MANAGING DATA OVERFLOW IN CLOUD ENVIRONMENTS?

Although cloud systems are renowned for their capacity to manage enormous volumes of data, controlling the overflow of this kind of data is still difficult. The quick scaling of traffic, which may go from empty to full in a matter of seconds, is one of the primary problems [3]. However, because it makes use of load managing algorithms to help identify ahead of time which servers are most likely to get overloaded, cloud load management has been built to withstand massive and unplanned surges in traffic [3][5]. This lessens the ability of data loss and services going down by allowing traffic to be redirected to other areas that can manage higher load [3][5]. Furthermore, autoscaling enables quicker resource use because it does not require pre-warming.
distribution and implementation [3]. Cloud load managers that use planning based on load managing algorithms can also more quickly reroute server traffic to healthier nodes, resulting in quicker response times and better overall performance [5]. These methods are vital for controlling data overflow in cloud environments and for guaranteeing that cloud services continue to function dependably and be reachable even during times of heavy traffic.

V. IMPLEMENTATION:

In cloud computing, load management is a crucial procedure that distributes incoming traffic between active targets in accordance with a set strategy. Cloud load management uses a software-based approach to monitor the health of each target inside the cluster [4]. In contrast, hardware-based load management—which is more common in enterprise data centers—uses specialized equipment. Before rerouting traffic in the event of a failure, load- managed networks may evaluate the geographical distance or server load [5]. Different methodologies are used in cloud load management implementations, and they differ in how they distribute and control network traffic [4]. There are eight widely used load management algorithms in cloud computing, and each one selects which servers to accept client requests in a different way [5]. For instance, Rackspace offers a solution called Cloud Load Controllers that lets users split up workloads among several back-end systems or services via a RESTful web service interface [4]. In addition to major cloud platforms, cloud service providers also provide load managing capabilities [4].

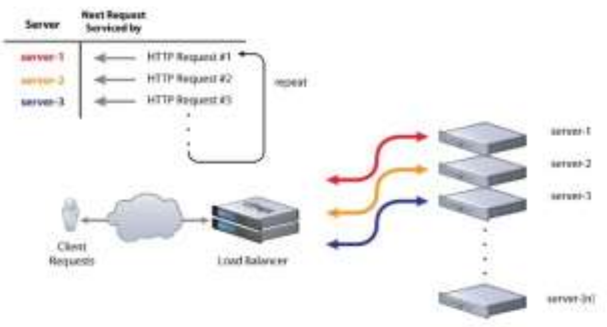


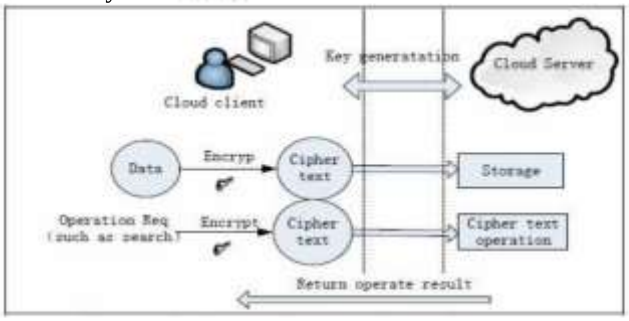
Figure 3: Block Diagram of Round Robin Technique

VI. METHODOLOGY

A basic Python implementation of the Round Robin algorithm is provided, showcasing its practical application in managing server workloads.

```
class RoundRobinLoadManager:
    def __init__(self, servers):
        self.servers = servers
        self.current_server_index = 0
    def manage_request(self):
        # Get the next server in the list
        next_server =
self.servers[self.current_server_index]

        # Update the index for the next request
        self.current_server_index =
(self.current_server_index + 1) % len(self.servers)

        return next_server

# Example usage
servers_list = ["Server1", "Server2", "Server3"]
load_manager = RoundRobinLoadManager (servers_list)

# Simulating incoming requests
for _ in range(10):
    next_server = load_manager.manage_request()
    print(f"Request sent to {next_server}")
```

software-based cloud load managing and hardware-based alternatives is highlighted, with a focus on the former's flexibility in monitoring individual target health within a cluster. Additionally, the discussion touches upon factors like geographical distance and server load evaluation, which contribute to effective traffic redirection in case of failover.
The text goes on to enumerate various load managing algorithms employed in cloud computing, emphasizing their differences in determining which servers should receive client requests. Examples include round-robin, least connections, IP hash, and weighted round-robin. Practical applications from service providers, such as Rackspace's Cloud Load managment services, further illustrate the diverse techniques used in distributing workloads across multiple back-end systems.



A critical analysis of static load managing follows, acknowledging its simplicity but pointing out drawbacks such as reliance on a single device and potential delays in workload redistribution during traffic spikes. Despite these drawbacks, the text recognizes the utility of static load managing algorithms in specific scenarios where a straightforward and easy-to-implement solution is essential.
The final section focuses on the resilience and scalability of Cloud Load managing. It discusses the design features aimed at handling unexpected spikes in traffic, utilizing load managing algorithms to predict and preemptively divert traffic to less burdened servers. The significance of autoscaling, faster resource allocation, and deployment is emphasized, along with the role of load managing algorithms

in redirecting server traffic rapidly to healthier nodes for improved overall performance.

VII. ADVANTAGES AND DISADVANTAGES OF LOAD MANAGING TECHNIQUE:

Static load management is one of the easiest methods to use out of all the load managing strategies available. It does, however, have a unique set of disadvantages. Static load managing largely depends on a single device or server to spread the load evenly among all servers in the network, which makes it unsuitable for implementation on other devices. Furthermore, load shifting may not always be the most effective technique to distribute the load evenly across the network because it is independent of the system's present condition [6]. The fact that load managing tasks are produced and only function after they are is another major disadvantage of the static load maintaining method. When a large surge in traffic happens, this may cause a delay in dispersing the burden, which might result in longer response times and even user outages [6]. Static load management algorithms are nonetheless helpful in certain situations when a straightforward and simple-to-implement solution is needed, despite these disadvantages.

Additionally, the use of encryption before sending data to the cloud, such as Bring Your Own Key (BYOK) encryption, could become more prevalent[2]. These advancements would aim to enhance data security and control in the face of increasing data volumes and complexity in cloud computing environments[2].

V11 RESULT

The Round Robin Load Managing technique, emphasizing its simplicity and effectiveness. This technique distributes incoming requests sequentially to servers in a cyclical manner, ensuring that no single server becomes overloaded. A basic Python implementation of the Round Robin algorithm is Implemented, showcasing its practical application in managing server workloads.

The next part explores load management in cloud computing in more detail. It emphasizes how important load balancing is in allocating incoming traffic to active targets in accordance with set regulations. The differences between hardware-based and software-based cloud load managing solutions are emphasized, with an emphasis on the former's adaptability in tracking the health of individual targets within a cluster. Furthermore, the discourse addresses variables such as geographic separation and server load assessment that support efficient traffic rerouting in the event of a failure.
We took into consideration the case where one data center has four hosts and eight virtual machines (VMs) in order to understand the operation and comparative analysis of the suggested technique.
1) The reaction time of Cloudlets
2) Processing time in data centers

The VM Configuration Used for Simulation.

Parameters	Values
Image size of VM	10000
The Memory of VM	512 MB
The Bandwidth of VM	1000

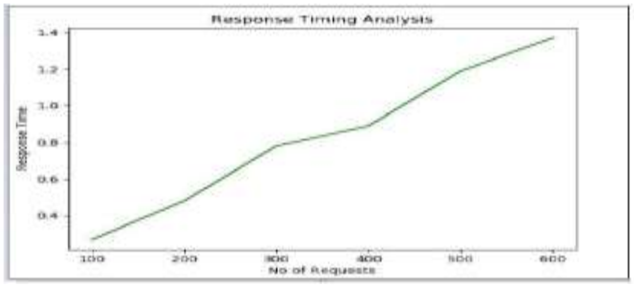


Figure 4:Graph between Response time and Response time analysis

We learned from the prior graph that response times increase in tandem with an increase in requests at each given node. The graph's slope is very straight when the number of queries is between 300 and 400; this means that the algorithm is delivering more responses faster. We may readily comprehend these variances by looking at graph .

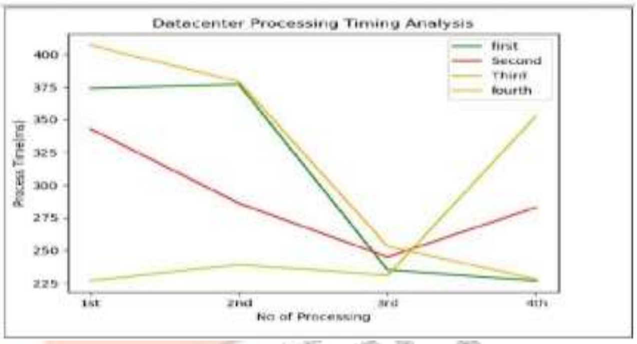


Figure 5:Graph between processing time lines and Datacenter Processing time analysis

VII.FUTURE SCOPE

The future scope of data overflow control in cloud computing environments could involve the development of more advanced and secure data sharing schemes in the cloud environment[1]. This could include the implementation of new standards, rules, and regulations to protect crucial customer data[2]. Additionally, the use of encryption before sending data to the cloud, such as Bring Your Own Key (BYOK) encryption, could become more prevalent. These advancements would aim to enhance data security and controlin the face of increasing data volumes and complexity in cloud computing environments. The future scope of load managing techniques in cloud computing could involve the development of more advanced algorithms and methods that can efficiently handle the increasing complexity and size of cloud workloads. There is also potential for the integration of machine learning and AI techniques to enhance the prediction and management of load distribution, thereby improving the overall performance and efficiency of cloud systems.
The future scope of data overflow management in cloud computing could involve the development of more advanced and secure data sharing schemes in the cloud environment[1]. This could include the implementation of new standards, rules, and regulations to protect crucial customer data. Additionally, the use of encryption before sending data to the cloud, such as Bring Your Own Key (BYOK) encryption, could become more prevalent. These advancements would

aim to enhance data security and control in the face of increasing data volumes and complexity in cloud computing environments².

IX. CONCLUSION:

Load managing - A Crucial Tool for Managing Data Overflow in Cloud Environments

Despite the ever-expanding capabilities of cloud computing, handling massive data influx and preventing overflow remains a key challenge. This is particularly true due to the rapid and unpredictable nature of traffic spikes. The good news is, numerous load managing techniques exist, offering effective solutions to mitigate overflow and ensure service reliability.

Key Techniques and Benefits:

- Cloud Load managing: This software-based approach utilizes intelligent algorithms to predict server overload and divert traffic accordingly. This not only prevents data loss and service disruption but also facilitates faster response times through autoscaling and optimized server assignments.
- Round Robin Algorithm**:** This simple but effective technique distributes incoming requests equally among servers, ensuring no single server gets overwhelmed. While basic, it offers a reliable foundation for load managing, especially in scenarios with predictable workloads.
- Other Load managing Algorithms**:** Beyond Round Robin, various algorithms cater to specific needs, such as Least Connections which prioritizes servers with fewer active connections, and Weighted Round Robin which assigns more requests to more powerful servers.

Addressing Static Load managing Limitations:

While acknowledging the simplicity and ease of implementation of static load managing, the conclusion also sheds light on its limitations. This includes reliance on a single device and potential delays in workload redistribution during traffic spikes. However, it emphasizes the continued usefulness of static algorithms in specific scenarios where simplicity and straightforward implementation are prioritized.

Overall Impact:

In conclusion, load managing techniques, particularly cloud-based implementations with advanced algorithms, play a crucial role in managing data overflow and ensuring reliable, accessible cloud services even under heavy traffic loads. These

techniques offer agility, scalability, and efficient resource allocation, ultimately contributing to a seamless and positive user experience within the cloud environment.