OXFORD BROOKES UNIVERSITY

Faculty of Technology, Design and Environment

MASTER OF SCIENCE DISSERTATION

Dissertation Title:	Scalability and Performance Optimization in Cloud-based E-commerce and M-commerce Applications.
Surname:	SHARMA
First Name:	DEVASHISH
Student Number:	19224423
Supervisor:	Professor. Kashinath Basu
Dissertation Module:	**TECH7009**
Course Title:	MSc. Advanced Computer Science
Date Submitted:	02 October, 2023

Statement of Originality

Except for those parts in which it is explicitly stated to the contrary, this project is my own work. It has not been submitted for any degree at this or any other academic or professional institution.

Signature of Author		Date	02/10/2023

Regulations Governing the Deposit and Use of Master of Science Dissertations in the School of Engineering, Computing and Mathematics, Oxford Brookes University.

1. A copy of dissertation final reports submitted in fulfilment of Master of Science course requirements shall normally be kept by the School.

2. The author shall sign a declaration agreeing that, at the supervisor’s discretion, the dissertation will be submitted in electronic form to any plagiarism checking service or tool.

3. The author shall sign a declaration agreeing that the dissertation be available for reading and copying in any form at the discretion of either the dissertation supervisor or in their absence the Programme Lead of Postgraduate Programmes, in accordance with 5 below.

4. The project supervisor shall safeguard the interests of the author by requiring persons who consult the dissertation to sign a declaration acknowledging the author's copyright.

5. Permission for anyone other than the author to reproduce in any form or photocopy any part of the dissertation must be obtained from the project supervisor, or in their absence the Programme Lead of Postgraduate Programmes, who will give his/her permission for such reproduction only to the extent to which he/she considers to be fair and reasonable.

I agree that this dissertation may be submitted in electronic form to any plagiarism checking service or tool at the discretion of my project supervisor in accordance with regulation 2 above.

I agree that this dissertation may be available for reading and photocopying at the discretion of my project supervisor or the Programme Lead of Postgraduate Programmes in accordance with regulation 5 above.

Signature of Author		Date	02/10/2023

DISSERTATION REPORT: SCALABILITY AND PERFORMANCE OPTIMIZATION IN CLOUD-BASED E-COMMERCE AND M-COMMERCE APPLICATIONS

Table of Contents

List of Figures:

Abstract

In an era where online commerce dominates the digital landscape, the seamless performance and scalability of e-commerce and m-commerce applications are paramount. This dissertation project delves deep into the heart of this challenge, presenting a pioneering solution through the implementation of a Dynamic Resource Allocator (DRA). Rooted in a pragmatic research philosophy, this endeavor explores the amalgamation of cloud technologies and application architecture, resulting in a cloud-based e-commerce and m-commerce application of unparalleled efficiency. Rigorous performance analysis, encompassing standard usage, peak load, and stress conditions, unveils the transformative potential of the DRA. By dynamically adjusting resource allocation based on user traffic and system conditions, this framework not only optimizes performance but also charts a course towards cost-effectiveness. As the digital landscape continues to evolve, this project paves the way for future research in machine learning integration, real-time data processing, multi-cloud environments, and enhanced security measures. The path to redefining cloud-based commerce applications starts here, offering a glimpse into a future where performance knows no bounds.

Acknowledgements

I would like to extend my sincerest appreciation to Professor Kashinath Basu, who has served as my committed and invaluable supervisor throughout the entirety of this research endeavour. Professor Basu's unwavering backing, expert guidance, and mentorship have been integral in shaping the content and direction of this dissertation.

His profound expertise in the realm of cloud-based technologies and resource allocation mechanisms has illuminated my path and enhanced my comprehension. His perceptive feedback, meticulous evaluations, and constructive critique have played a pivotal part in refining the calibre and rigor of this dissertation.

I also wish to convey my gratitude to the entire faculty of Oxford Brookes University's School of Engineering, Computing, and Mathematics for fostering an environment that is concurrently motivational and intellectually stimulating.

Lastly, the successful completion of this dissertation would not have been possible without the unwavering support, encouragement, and patience of my family and friends, who have remained by my side throughout this challenging yet fulfilling journey.

I extend my heartfelt appreciation to all individuals involved for their invaluable contributions to the realization of this dissertation.

Chapter 1: Introduction

1.1. Background:

1.1.1. Transforming E-commerce Through Cloud Technology:

The online shopping has become a seamless part of our everyday lives in our fast-paced, technologically advanced society. The consistent expansion of e-commerce has revolutionized not just how businesses run but also how customers approach their buying experiences **(Naz, 2019)**. Online merchants must leverage the amazing potential of cloud-based solutions to deliver their goods and services precisely if they want to stay ahead of the competition **(Alshahrani, 2015).**

As with every invention, a combination of online businesses with cloud computing has its own set of advantages and disadvantages. Businesses get access to a big client base, the enticement of cost savings, and the flexibility to develop quickly **(Kavitha & Radha, 2021)**. To meet the constantly changing demands of today's affluent consumers, they must also manage their cloud resources properly, which is a complex job. To maintain operational efficiency while meeting or exceeding consumer expectations, firms must carefully analyse their resource allocation plans. Finding the proper balance is crucial. Those businesses must manage the complex process of incorporating cloud computing into their e-commerce platforms while keeping an eye on the always changing market environment **(Davids & Van Belle, 2017)**.

Online enterprises have the potential to drive transformation and steer the trajectory towards success in our contemporary digitally driven economy by implementing carefully considered strategies and leveraging the latest advancements in electronic commerce and cloud computing.

1.1.2. The Significance of Cloud-Based E-commerce

Cloud-based e-commerce and m-commerce applications have emerged as indispensable tools for thriving in the digital landscape. Leveraging cloud technology offers numerous advantages, including scalability, improved performance, cost-effectiveness, and global reach **(Mohammadpour & Tafte, 2016)**. Yet, the efficient utilization of cloud resources, especially during fluctuating demand, remains a formidable challenge **(Vijai & Nivetha, 2021).**

1.2. Aim & Objectives:

Enhancing Scalability and Performance through Resource Allocation.

The aim of this research endeavour is to enhance the scalability and performance of cloud-based electronic commerce and mobile commerce applications. Our main focus is on accomplishing this objective through the construction and execution of the Dynamic Resource Allocator (DRA) framework for intelligent resource allocation. This framework adjusts dynamically to user requirements and fluctuating circumstances.

The specific objectives of this study are designed to reflect the breadth and depth of our research efforts:

1.2.1. Architect and Develop:

Our investigation delves into the intricate interplay between cloud-based technologies and application architecture. The primary goal is to extract the most effective methods and innovative approaches that empower businesses to excel in the digital landscape. Building upon this research, we have undertaken the design and development of a cloud-based e-commerce and m-commerce application. This application prominently features the integration of the Dynamic Resource Allocation (DRA) framework, a mechanism crafted for intelligent resource allocation.

Sub-Objectives:

- **Architect Cloud-Based E-commerce and M-commerce Application:** We analyse and design the architecture of our application to harness the potential of cloud-based technologies.
- **Develop the Dynamic Resource Allocator (DRA) Framework:** Drawing inspiration from our research findings, we conceive and implement the DRA framework, leveraging a rule-based approach for intelligent resource allocation.

- **Integrate Real-Time Resource Allocation:** We ensure that the DRA framework dynamically allocates computational resources in response to real-time user demands and system conditions.
- **Enhance Performance and Scalability:** Through the DRA framework, we aim to optimize the application's performance and scalability, thereby providing an exceptional user experience even during peak loads.

1.2.2. Comprehensive Performance Analysis:

Evaluating the effectiveness of the DRA framework is pivotal. To achieve this, we have conducted rigorous performance analysis and testing of the developed application. This process entails subjecting the application to diverse scenarios, ranging from typical usage patterns to stress conditions, with the aim of assessing its performance and scalability.

Sub-Objectives:

- **Performance Metrics Evaluation:** We employ established performance metrics, including response time, throughput, and error rates, to comprehensively assess the application's behaviour under varying conditions.
- **Stress Testing:** We subject the application to stress tests, simulating scenarios of peak usage, to gauge its robustness and performance.

1.2.3. Comparison with Other Approaches:

In addition to conducting an analysis of performance, our aim is to offer a comparative assessment of the Dynamic Resource Allocator (DRA) in contrast to alternative strategies for resource allocation. This comparative evaluation emphasizes the distinctive benefits of our framework in enhancing the performance and scalability of applications.

Sub-Objectives:

- **Identify Alternative Resource Allocation Approaches:** We conduct a comprehensive review to identify various mechanisms for resource allocation from the existing body of studies.
- **Performance Comparison:** We evaluate and compare the performance of the DRA framework with these alternative approaches, highlighting the strengths and weaknesses of each.

1.3. Problem Overview:

1.3.1. The Resource Allocation Dilemma

At the heart of cloud-based e-commerce and m-commerce lies a pivotal challenge: intelligent resource allocation. Balancing resource allocation to optimize both performance and scalability while managing costs is the linchpin of success in these applications **(Fard et al., 2020)**. Achieving this equilibrium requires an intricate understanding of resource allocation mechanisms and their profound impact on the entire system.

1.3.2. Navigating the Resource Allocation Challenge

The central focus of this field concerns the efficient distribution of cloud resources. Although cloud platforms provide the ability to adjust resources in response to changing needs, the fundamental question remains: How can resources be intelligently allocated based on user requirements and evolving circumstances? Which method is the most appropriate for dynamically assigning resources to enhance performance and scalability in the realm of e-commerce and m-commerce applications, and how does it compare to alternative approaches to resource allocation?

1.4. Product Overview:

1.4.1. Prototype description:

The core of our research project is a foundational prototype web application developed to showcase the intersection of cloud-based technologies and the online shopping domain, strategically designed as a base for our investigation into the Dynamic Resource Allocator (DRA). While the prototype may not be fully functional as a commercial e-commerce platform, its primary purpose is to serve as a controlled environment for us to rigorously assess and demonstrate the capabilities of the DRA framework and performance of e-commerce and m-commerce applications.

Key Characteristics:

- **Research-Centric Design:** The prototype is intentionally streamlined to prioritize research objectives. It may not encompass the full scope of a commercial e-commerce platform, but its structure allows us to isolate and scrutinize the resource allocation mechanisms powered by the DRA.
- **Resource Allocation Sandbox:** Within this prototype, we create a controlled environment to showcase the DRA's effectiveness in allocating computational resources. By utilizing simplified scenarios, we can meticulously analyse how the DRA optimally manages processing capacity and memory based on varying user demands and system conditions.

1.4.2. A Spotlight on Resource Allocation Mechanisms:

The focal point of our research endeavour undeniably rests upon the Dynamic Resource Allocator (DRA) framework. Within the prototype, the DRA is not merely an ancillary element but rather the central point of interest, facilitating a meticulous investigation of its influence on resource allocation within the realm of cloud-based e-commerce and m-commerce applications.

The DRA as a Beacon of Research:

- **Dynamism and Adaptability:** The DRA framework, even within the confines of our prototype, exhibits its capacity to dynamically allocate computational resources in response to real-time user requirements and shifting system circumstances. This adaptability serves as a testament to its potential for optimizing application performance.

- **Performance under Scrutiny**: Through the utilization of the DRA within this simplified environment, we can closely scrutinize its performance under diverse conditions, including heavy traffic loads. This scrutiny grants us profound insights into how the DRA tackles fluctuations in user demand, ensuring consistent application responsiveness.

1.5. Significance of Study:

Navigating the Future of E-commerce:

This research project assumes significant relevance in the ever-evolving landscape of e-commerce and m-commerce. By tackling the challenges of scalability and performance optimization through intelligent resource allocation, we aim to contribute valuable insights and pragmatic solutions to the industry. Our findings have the potential to guide businesses in elevating their online platforms, benefiting both enterprises and consumers alike.

1.6. Scope of the study:

The scope of this dissertation is to research and develop an ecommerce application which offer various functionalities, including user authentication, product catalogue management and shopping cart features. The study involve deployment of the prototype application on AWS cloud environment. This dissertation mainly focuses on enhancing the scalability and performance of cloud-based e-commerce application. By effectively distributing computing power, storage capacity, and network resources, the application will be able to handle varying loads and user demands efficiently. The research delves into the architectural design of this application prototype, with a particular emphasis on intelligent resource allocation mechanism, exemplified by the Dynamic Resource Allocator (DRA) framework, leveraging a rule-based approach for intelligent resource allocation which will optimize the application's performance and scalability by intelligently assigning resources based user demands and conditions. The study includes rigorous performance analysis of the application with DRA integration and comparative assessments of rule-based approach with other resource allocation approaches.

1.7. Limitation:

While study provides a comprehensive understanding of resource allocation mechanisms in cloud-based e-commerce and m-commerce applications, it is essential to acknowledge certain limitations:

- **Prototype Complexity:** The developed prototype serves as a representation of a simplified e-commerce and m-commerce application. While it demonstrates the principles of resource allocation, real-world applications may have more intricate features and functionalities.
- **Limited Real-World Testing:** The performance analysis and testing conducted in this research are based on simulated scenarios. Real-world testing with live users and diverse conditions may provide additional insights.
- **Resource Allocation Specificity:** Since the rule-based Dynamic Resource Allocator (DRA) framework is a central focus, other resource allocation techniques might exist. However, this research does not aim to explore all possible mechanisms exhaustively.
- **Scope of Comparative Analysis:** The comparative analysis focuses on the rule-based DRA framework and a select number of alternative resource allocation approaches. It may not comprehensively cover all existing mechanisms.

Despite these drawbacks, this study offers insightful information on how to allocate resources in cloud-based e-commerce and mobile commerce apps, with a focus on improving scalability and performance using clever approaches.

Chapter 2: Background Review:

2.1. Scalability in E-commerce:

E-commerce's rapid growth necessitates the ability of online platforms to handle increasing user loads, transactions, and data while maintaining peak performance. Scalability is important factor in modern e-commerce due to several key reasons:

- **Catering to Rapid Growth:** E-commerce businesses often experience surges in traffic, especially during seasonal sales or promotions. Scalability ensures the platform can seamlessly accommodate these spikes in users and transactions.
- **Enhancing User Experience:** Instant access to products and seamless shopping experiences have become the norm in the digital landscape. Scalable e-commerce platforms provide rapid response times and uninterrupted services, enhancing user satisfaction.
- **Global Reach:** E-commerce operates without geographical boundaries, serving a global audience. Scalability ensures that the platform can effectively meet the needs of users worldwide.
- **Cost Efficiency**: Scalability can lead to cost savings by dynamically allocating resources based on actual demand, optimizing operational costs.

2.1.1. Definitions and Importance:

Scalability in e-commerce refers to an online platform's ability to handle increasing loads of users, transactions, and data without compromising performance or user experience **(Li et al., 2018).** It is a vital attribute because e-commerce platforms often experience unpredictable traffic spikes, especially during promotions or seasonal events. Academic research by **(Gunther, 2001)** and **(Lehrig et al., 2018)** provides comprehensive definitions and mathematical models to assess scalability, emphasizing its multifaceted nature. These definitions serve as a foundation for understanding scalability's significance in e-commerce.

(Zhao et al., 2020) (Kim & Kim, 2014) research identifies scalability strategies such as content delivery network (CDN) integration and advanced caching mechanisms. These studies underscore the importance of a holistic approach to scalability, considering both hardware and software elements.

2.1.2. User Experience and Scalability:

Scalability is not solely about technical enhancements but also about ensuring a seamless and enjoyable shopping experience. **(Tang et al., 2021)** presents a method of user experience design that identifies the scalability of experience proposes different dimensions for vertical and horizontal scalability of experience, allowing for a comprehensive understanding of scalability's impact on user experience.

The multifaceted nature of scalability in e-commerce, its definitions, strategies, and its direct impact on user experience lay the foundation for a comprehensive exploration of performance optimization.

2.2. Performance Optimization:

Performance optimization is essential for drawing in and keeping customers in the fiercely competitive world of e-commerce. In order to ensure that compute resources, memory, and network bandwidth are used efficiently, performance optimization strives to increase the application's resource consumption efficiency **(Uzayr, 2022)**. This includes a range of measures including increasing throughput, decreasing errors, and ensuring a seamless user experience.

2.2.1. Importance of Performance Optimization

Performance optimization is of paramount importance in e-commerce for several compelling reasons:

- **Competitive Advantage:** E-commerce is fiercely competitive. Websites that load quickly and respond promptly tend to attract and retain customers. **(Pavic & Brkic, 2021)** reveals a direct correlation between website speed and conversion rates, highlighting the competitive advantage of optimized performance.

- **User Retention:** A slow or unresponsive website can frustrate users, leading to high bounce rates and cart abandonment **(SNG, 2016).** According to a feedback survey from 1048 online shoppers, 47% of consumers expect a web page to load in two seconds or less **(Anderson, 2023).** This highlights the significance of performance optimization in user retention.
- **Search Engine Ranking:** Search engines like Google factor in website speed when determining search rankings. A fast-loading website can improve search engine visibility, potentially increasing organic traffic **(Barbar & Ismail, 2019).** A study by **(Jambhale, 2019)** highlights the importance of page speed on search engine optimization (SEO).
- **Mobile optimization:** With the rise of mobile devices, it is crucial to enhance performance on tablets and smartphones. According to research by **(Statista, 2023)**, mobile e-commerce traffic is overwhelmingly dominant. In the UK, during the second quarter of 2023, 75% of visits to online retail sites were made using a smartphone, and 70% of online orders were made using a smartphone. This is because effective performance optimization guarantees a flawless mobile shopping experience.

2.2.2. Performance Optimization Strategies

A multitude of strategies are employed to optimize e-commerce performance, ensuring fast load times and responsive user interfaces:

- **Content Delivery Networks (CDNs):** CDNs distribute website content across a network of servers or multiple data centres, reducing latency, ensuring that content is served from a location geographically closer to the user. This reduces latency and enhances load times **(Puzhavakath Narayanan et al., 2016).** Studies by leading CDN providers, Akamai and Cloudflare delve into the impact of their CDNs on website performance and emphasizes the importance of measuring the performance of their network and benchmarking themselves against industry players to ensure they provide the best and fastest service to their clients **(Hearne, 2021) (Graham-Cumming, 2023).**

- **Caching Mechanisms:** Caching involves storing frequently accessed data or web pages in a temporary storage layer, such as in-memory caches or content delivery networks (CDNs) which reduces the need for repeated data retrieval. Caching can significantly reduce the load on backend servers and improve response times **(Zulfa et al., 2020)**. Some studies explores advanced caching mechanisms and their effectiveness in performance optimization. **(Ramu, 2023)** acknowledges the effectiveness of caching mechanisms, such as in-memory caches, in improving performance in microservices-based systems. It highlights the importance of implementing caching mechanisms to reduce database queries, improve response times, and minimize network overhead. Another research **(Vu et al., 2018)** on caching analyses two notable caching strategies: uncoded and coded caching, in edge-caching wireless networks. The author proposes optimization problems to minimize content delivery time for the caching strategies and presents numerical results to verify their effectiveness as well as calculates backhaul and access throughputs for these strategies and derives closed-form expressions for system energy efficiency (EE).
- **Content Compression:** web content, such as images and scripts, can slow down websites. Content compression techniques, such as lazy loading and image compression, are essential for fast-loading product pages and reduces data transfer times and speeds up page loading, contributing to improved user experience **(França et al., 2020) (Boopathiraja et al., 2019).**
- **Code Optimization:** Efficient coding practices, including eliminating redundant code, reducing database queries, and optimizing algorithms, can lead to faster execution times and improved performance of the web application **(Bajwa et al., 2016).**

2.2.3. Impact of Performance Optimization

Achieving optimal performance is crucial for e-commerce success. This requires daily monitoring of changes, regular site rebranding and redesign to ensure faster load times and responsive design. Ultimately, this results in increased customer satisfaction and higher conversion rates **(Uzayr, 2022)**. These findings highlight the tangible benefits of implementing performance optimization strategies.

Performance optimization holds a considerable influence over gaining a competitive edge, maintaining user engagement, elevating search engine rankings, and keeping pace with mobile trends. The strategies discussed here provide a foundation for understanding how performance optimization enhances e-commerce operations.

2.3. Overview of Cloud-Based Technologies and Their Relevance to E-commerce and M-commerce Applications:

The integration of cloud-based technologies into e-commerce and m-commerce applications has reshaped the landscape of online businesses **(Rotkar & Butey, 2021).** In this chapter, we delve into the world of cloud computing and explore its significance in the context of digital commerce.

2.3.1. Cloud Computing: A Primer

Cloud computing, at its core, is a paradigm that provides on-demand access to computing resources over the internet **(Singhal & Ahuja, 2023).** These resources encompass a wide spectrum, including but not limited to servers, storage, databases, networking, software, analytics, and intelligence. One of the hallmark features of cloud computing is its ability to offer these services with pay-as-you-go pricing, scalability, and flexibility **(Zhang et al., 2017).**

2.3.2. Relevance to E-commerce and M-commerce

The adoption of cloud computing within the e-commerce and m-commerce domains is driven by several factors:

- **Scalability:** Cloud platforms enable businesses to scale their infrastructure up or down as needed. This scalability aligns with the fluctuating demands often encountered in e-commerce and m-commerce, ensuring that applications remain responsive even during peak traffic periods **(Vijai & Nivetha, 2021).**

- **Cost Efficiency:** Cloud services provide a cost-effective alternative to traditional on-premises data centres. Businesses can optimize resource utilization and reduce operational expenses, contributing to improved profit **(AWS & Gribbin, 2023)**
- **Global Reach:** Cloud infrastructure spans the globe, allowing e-commerce and m-commerce applications to reach customers worldwide. This global presence enhances accessibility and ensures low-latency experiences for users across different regions **(Google Cloud, n.d.).**
- **Innovation and Agility:** Cloud providers continually introduce innovative services and features in various areas, including API management, serverless computing, content delivery networks, storage services, datacentre technologies, all-flash arrays, and digital twin technologies. Businesses can harness these innovations to stay competitive and rapidly deploy new features or updates **(Devare, 2019).**
- **Data Management:** Cloud-based storage and databases simplify data management. E-commerce and m-commerce applications generate vast amounts of data, and cloud solutions provide robust tools for data storage, retrieval, and analysis. Cloud storage offers data redundancy and backup, ensuring that data is protected and can be recovered in case of hardware failure or other disasters. **(Oracle Cloud, n.d.).**

2.3.3. Key Considerations

While the benefits of cloud computing are undeniable, it's essential for businesses to consider certain aspects when migrating or building e-commerce and m-commerce applications in the cloud:

- **Security:** With the digital nature of these applications, security is paramount. Businesses must implement robust security measures to protect customer data, transactions and other sensitive information **(Rajendranath et al., 2022).**
- **Compliance:** E-commerce and m-commerce applications often deal with sensitive customer information. Compliance with data protection regulations is only a legal obligation but also crucial for maintaining trust with customers and stakeholders. **(Rivis & Ying, 2013)**.

- **Vendor Selection:** Choosing the right cloud service provider is a critical decision. Factors like reliability, performance, and cost should be carefully evaluated. It is also important to evaluate the track record and reputation of the cloud service provider, as well as their ability to meet specific business requirements **(Mahrishi et al., 2020).** Cloud brokers can be especially helpful for small businesses and organizations that do not have the expertise or resources to manage cloud computing on their own **(Fard et al., 2020)**.

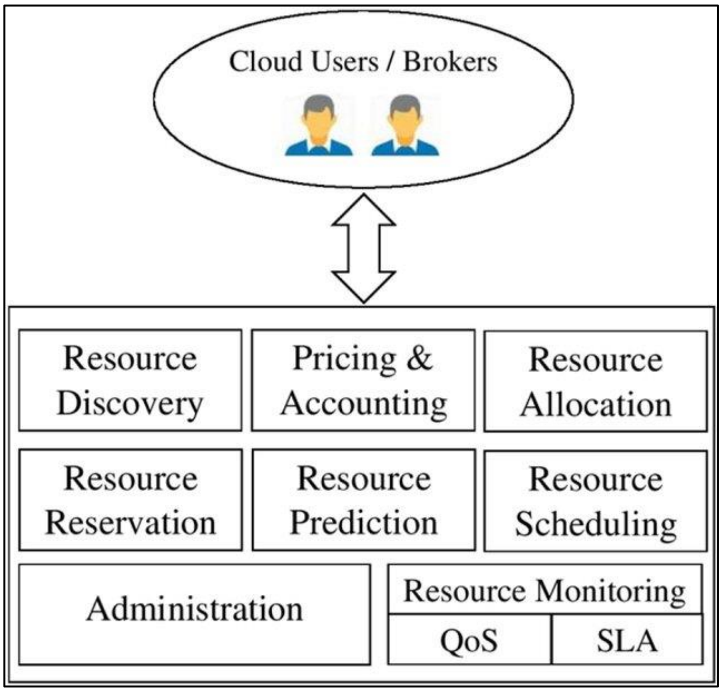

FIGURE 1: RMS MODULES (FARD ET AL., 2020)

- **Scalability Planning:** While cloud platforms offer scalability, businesses need a well-thought-out scalability plan to ensure efficient resource allocation. The cloud brokers can help users to assess their needs and develop a cloud strategy **(Fard et al., 2020)**.

By providing a versatile infrastructure, cloud-based technologies empower e-commerce and m-commerce applications to meet the evolving demands of modern consumers.

2.4. Dynamic Resource Allocation:

The keystone of our effort to maximize the functionality and scalability of e-commerce and m-commerce apps in cloud systems is resource allocation. The DRA in the cloud refers to automatically adjusting resources based on user needs, aiming to optimize performance and scalability while managing costs **(Belgacem, 2022).**

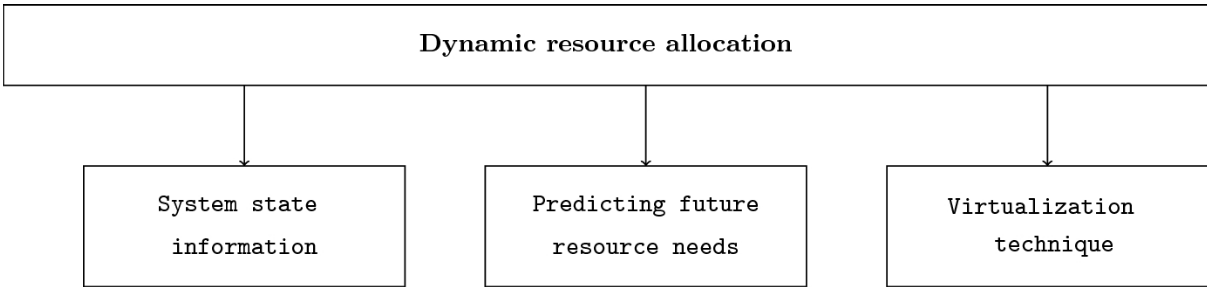

FIGURE 2: DYNAMIC RESOURCE ALLOCATION IN CLOUD (BELGACEM, 2022)

The Dynamic Resource Allocator (DRA) framework, at the core of our research, is highlighted in great detail as we conduct a comprehensive exploration of resource allocation mechanisms in this section.

2.4.1. Different Resource Allocation Mechanisms:

2.4.1.1. Static Resource Allocation:

Static resource allocation involves the allocation of computing resources to applications or services based on predetermined configurations without frequent adjustments or adaptations to changing conditions **(Xu et al., 2019).** These allocations typically remain fixed until manually modified. For example, a static rule-based approach might allocate a fixed amount of CPU and memory to an application based on predefined criteria, regardless of variations in user traffic or system conditions. Research by **(Pawar & Wagh, 2012)** identifies the challenges associated to static allocation, particularly in handling unexpected traffic surges. It highlights that this approach tends to lead to underutilized resources during periods of low demand and performance bottlenecks during spikes.

2.4.1.2. Rule-Based Dynamic Resource Allocation:

When Distributing computing resources in real-time to improve the speed and scalability of applications, rule-based resource allocation is a dynamic technique used in resource management and cloud computing systems. These rules have been carefully designed to adjust to changing conditions, user requirements, and performance measures. Various criteria, including user-specific requirements, performance-based criteria, time-based patterns, and threshold-based triggers, are included in rule-based allocation. Administrators or system operators often create and oversee these regulations. However, they also agree that rule-based processes face difficulties in quickly responding to rapidly changing environments **(Chen et al., 2023).** While rules can be effective in specific contexts, they may struggle to keep pace with the dynamic nature of modern e-commerce and m-commerce applications. **(Kolinski & Zieliński, 2023)** evaluates the rule-based approach presenting the importance of policy-based management in Cloud-native applications, enabling flexible system behaviour and compliance with CI/CD objectives. It proposes the AMoCNA framework, which stands for Autonomic Management of Cloud-native Applications as an extension of current orchestrator implementations, providing enhanced resource management and support for policy specification, processing, and runtime reconfiguration.

2.4.1.3. Load Balancing:

Load balancing evenly distributes incoming network traffic across multiple servers, preventing any single server from becoming overwhelmed. This enhances application performance and ensures fault tolerance **(Saadat & Masehian, 2019).** In the context of load balancing servers in the network, a dynamic scheduling algorithm is used to assign tasks to servers based on their load, ensuring that tasks are evenly distributed and no server becomes a single point of failure. It establishes connectivity between front-end and back-end servers, calculates the load on each sub server, and assigns tasks to servers with lesser load, thereby optimizing task distribution **(Pendke et al., 2017).** Work by **(Mohmmed & Abdalrahman, 2021)** discusses the significance of load balancers in enhancing fault tolerance and improving performance. It identifies Replication as a major fault tolerance technique that provides better reliability and improved response time for failure tasks in cloud computing.

2.4.1.4. Auto-Scaling:

Auto-scaling is a cloud scaling approach that automatically adjusts the number of computing resources allocated to an application based on its workload **(Bellenger et al., 2011).** Studies conducted by **(Fallah & Arani, 2015)** provides valuable insights by combining the threshold-based algorithm and learning automata to determine the best scaling up and scaling down of virtual machines. Effective scalability strategies are critical for e-commerce success. A case study by Amazon Web Services **(Barr, 2016)** offers insight into how AWS Elastic Load Balancing and Auto Scaling empowered Amazon.com to handle enormous traffic during Prime Day. This real-world example showcases the practical implementation of scalability strategies by one of the world's largest e-commerce platforms.

2.4.1.5. Machine Learning-Based Allocation:

Machine learning-based resource allocation, also known as Adaptive resource allocation (ARA) methods have gained prominence due to their ability to adapt to dynamic workloads and complex patterns. In the domain of e-commerce and m-commerce applications, their significance is notable for the following reasons:

- **Adaptive Resource Allocation**: Machine learning models excel at learning and adapting to patterns in data. In cloud-based e-commerce and m-commerce applications, where workloads can vary greatly, machine learning-based resource allocation offers adaptability and efficiency **(D & Nagineni, 2018).**
- **Predictive Resource Management:** Machine learning models can predict future resource needs based on historical data and real-time information, enabling proactive resource allocation **(Yadav & Yadav, 2021**)
- **Complex Decision-Making:** E-commerce and m-commerce applications often face multifaceted resource allocation challenges. Machine learning methods, such as

reinforcement learning and deep learning, are capable of handling intricate decision-making processes **(Kougioumtzidis et al., 2022).**

A wide array of strategies and techniques have been developed to leverage machine learning for resource allocation in cloud-based applications:

- **Reinforcement Learning:** Reinforcement learning algorithms is used to optimize the allocation of resources in a cloud computing environment. It involves training an agent to make sequential decisions on how to allocate resources based on the current state of the system and the desired goals using Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) algorithms. The agent learns through trial and error, receiving feedback in the form of rewards or penalties based on the quality of its decisions **(Pendyala et al., 2023)**. This approach allows for dynamic and adaptive resource allocation, taking into account factors such as workload demands, resource availability, and cost optimization.

- **Predictive Models:** Predictive models, including time series analysis and regression, are employed to forecast resource requirements. **(Yadav & Yadav, 2021**) proposes the use of the traditional classical model called Autoregressive Integrated Moving Average (ARIMA) for time series forecasting to predict the real-time fluctuating workload over the network. It is trained on historical time series data to capture patterns and trends, and then used to forecast future values based on these patterns. Another study on predictive models by **(Adegboyega, 2017)** introduces the Autoregressive Conditional Score (ACS) model, an econometric modeling technique for cloud workload prediction

.

2.5. Comparative Analysis:

While various resource allocation mechanisms exist, we've opted to focus our research primarily on the Dynamic Resource Allocator (DRA) framework for several compelling reasons. We meticulously considered various factors as mentioned below in Table 1:

Factor	Dynamic Resource Allocator (DRA)	Static Allocation	Load Balancing	Auto-Scaling	Machine Learning-Based Allocation
Cost Efficiency	Real-time optimization, cost savings	Fixed resource allocation	Load balancing can reduce costs	Scalable, cost-effective	Optimizes based on historical data
Scalability	Excellent, responds to varying workloads	Limited scalability, fixed resources	Scalable, distributes traffic	Scales resources as needed	Scalable with predictive allocation
Complexity	Requires setup and rule tuning	Simple setup but lacks adaptability	Moderate complexity, configuration needed	Configuration and threshold setting	Complexity in model training
Adaptability	Highly adaptable to changing traffic	Static allocation remains constant	Adaptable to traffic distribution	Adapts to workload fluctuations	Adapts to historical and current patterns
Fault Tolerance	Swift resource redistribution in case of failures	No inherent fault tolerance	Enhances fault tolerance	Ensures fault tolerance	Adapts to ensure reliability
Alignment with Objectives	Aligns with optimizing cloud resource usage	Limited adaptability to changing demands	Enhances performance but lacks adaptability	Aligns with dynamic workloads	Tailored allocation for application goals

TABLE 1: COMPARATIVE ANALYSIS OF RULE-BASED DRA WITH OTHER APPROACHES (AUTHOR, 2023)

The table demonstrates DRA's advantages in a number of crucial areas that are directly aligned to the objectives of our study, making it a compelling choice for research and development.

2.6. Summary and Gaps:

2.6.1. Key Findings:

Scalability in E-commerce:

- Scalability is crucial for accommodating rapid traffic surges and enhancing user experiences in e-commerce.
- Global reach and cost efficiency are additional benefits of scalability in the e-commerce domain.

Performance Optimization:

- Performance optimization is essential for competitive advantage, user retention, search engine ranking, and mobile optimization.
- Strategies such as CDNs, caching mechanisms, content compression, and code optimization play pivotal roles in performance optimization.

Cloud-Based Technologies:

- Cloud computing offers scalability, cost efficiency, global reach, innovation, and robust data management.
- Considerations include security, compliance, vendor selection, and scalability planning.

Resource Allocation Approaches:

- Different resource allocation mechanisms include Static Allocation, Dynamic Rule-Based Resource Allocation, Load Balancing, Auto-Scaling, and Machine Learning-Based Allocation.
- Each approach has unique characteristics related to cost efficiency, scalability, complexity, adaptability, and fault tolerance.

Dynamic Resource Allocator (DRA):

- DRA is a rule-based framework that dynamically allocates resources in real-time, optimizing cost efficiency, scalability, and adaptability.

- It aligns well with the objectives of optimizing cloud resource usage and adapting to dynamic workloads.

2.6.2. Bridging the identified gaps:

Enhancing Scalability:

Gap: Limited research focuses on the practical implementation of scalability strategies, especially in real-world e-commerce scenarios with dynamic workloads.

Addressing this gap will contribute to achieving the objective of enhancing scalability in e-commerce and m-commerce applications by providing actionable insights into the real-world deployment of scalability strategies.

Optimizing Performance:

Gap: Emerging performance optimization techniques need further exploration in the context of e-commerce and m-commerce applications.

Bridging this gap is essential to achieve the objective of optimizing performance, as it will involve researching and implementing cutting-edge performance optimization techniques tailored to digital commerce.

Efficient Resource Allocation:

Gap: Research on hybrid resource allocation approaches that combine multiple mechanisms for optimized resource management is limited.

Addressing this gap directly contributes to the objective of achieving efficient resource allocation. Developing and studying hybrid resource allocation methods will enhance resource utilization in digital commerce.

Chapter 3: Research and Development Methodology:

3.1. Type of research:

This research is primarily based on existing information and research, supplemented by the development of a prototype application to test and validate the proposed resource allocation approach. The study involves a two-fold approach:

- **Literature Review:** Gathering existing knowledge and research from online publications, libraries, research papers, and scholarly articles. This phase serve as the foundation for understanding cloud-concepts and resource allocation mechanisms in the context of e-commerce and m-commerce applications.
- **Prototype Development:** Designing, developing, and deploying a prototype application that incorporates the proposed resource allocation mechanism. This phase allows us to practically implement and evaluate the effectiveness of the resource allocation approach.

3.2. Data Collection and sources:

To collect and analyse relevant information, we will utilize the following tools and sources:

- **Online Publications and Libraries:** Accessing digital libraries and online publications to gather existing research papers, scholarly articles, and case studies related to performance, scalability, resource allocation, cloud computing, and e-commerce/m-commerce applications.
- **Cloud Service Providers Documentation:** Examining documentation and resources provided by leading cloud service providers, such as Google Cloud Platform (GCP), Microsoft Azure and mainly focusing Amazon Web Services (AWS). These documents will offer insights into cloud-based resource allocation strategies and best practices.

3.2.1. Source Selection Criteria:

The foundation of our data collection process is the establishment of stringent source selection criteria. These criteria serve as the gatekeepers for determining which sources are eligible for inclusion in our study. The primary criteria considered include:

- **Relevance:** Sources were meticulously assessed for their relevance to the topic of cloud-based e-commerce and m-commerce applications, specifically focusing on resource allocation and performance optimization.
- **Credibility:** The credibility and authority of the publication source were paramount. Academic journals, reputable conference proceedings, and renowned industry reports were prioritized.
- **Recency:** Given the rapidly evolving nature of cloud technology, sources were evaluated for recency to ensure that our study draws from the latest insights.

3.2.2. Search Strategy:

The process of source selection commenced with the formulation of a comprehensive search strategy. A diverse set of resources was utilized, including academic databases, search engines, and specialized repositories. The search strategy involved employing a combination of carefully chosen keywords and Boolean operators to enhance the precision of our searches. Keywords such as "cloud computing," "e-commerce," "m-commerce," "resource allocation," and "performance optimization" were instrumental in identifying relevant literature.

3.3. Prototype Development:

3.3.1. Building the E-commerce application:

The prototype application is developed using Python and the Django framework. Django, known for its robustness and scalability, offers a solid foundation for our e-commerce platform. It simplifies the development process, allowing us to focus on implementing the dynamic resource allocation logic and application functionality. The rule-based DRA logic was integrated into the application's backend, ensuring that resources were allocated dynamically based on user requirements and system conditions.

3.3.2. Designing the Application Architecture:

The architecture of our e-commerce application is characterised as a microservice architecture, deployed on AWS cloud using Amazon Elastic Beanstalk (EB) which offers simplified integration on cloud environment as shown in above Figure 3:

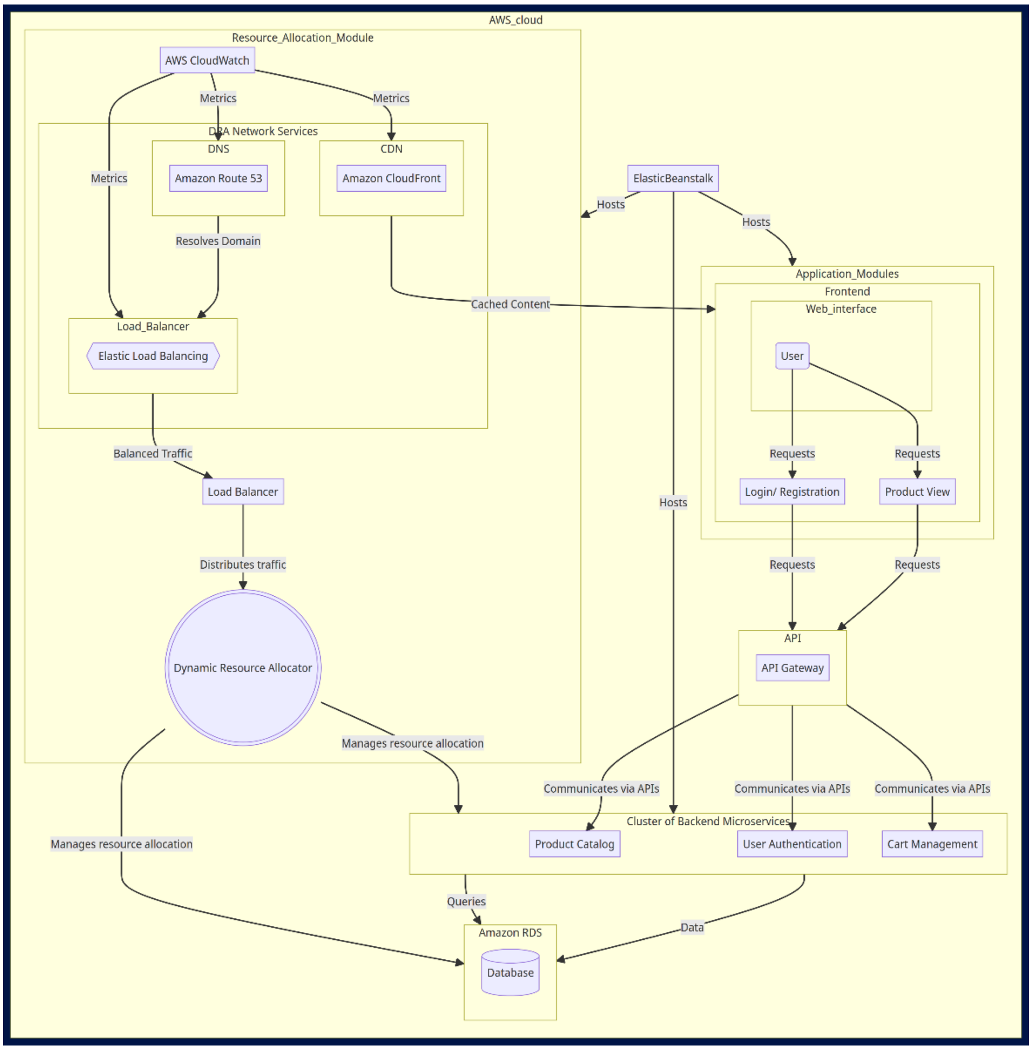

FIGURE 3: SYSTEM ARCHITECTURE ON CLOUD ENVIRONMENT (AUTHOR, 2023)

This architecture promotes modularity, scalability, and agility, making it suitable for complex and dynamic applications like e-commerce and m-commerce platforms. We have carefully structured the system to ensure efficient resource allocation based on user requirements and dynamic system conditions. The key components of our architecture include:

- **Frontend:** This component is responsible for rendering the user interface and handling user interactions. It consists of two frontend components, denoted as Login/ Registration and Product View in our architecture diagram. Users interact with the application through a web interface provided by the frontend. Each microservice is responsible for specific functionalities such as product catalog management, user authentication, and cart management.

- **Independent and Decentralized Backend Microservices**: Each backend component is implemented as a separate microservice which allows independent development, deployment, and scaling of each service. These microservices are represented as "Product Catalog," "User Authentication," and "Cart Management" in the architecture diagram. They communicate with the API Gateway to handle API requests. This decentralization enables flexibility and avoids tight coupling between services.
- **Continuous Integration and Deployment (CI/CD)**: Microservices are conducive to CI/CD practices, enabling rapid development, testing, and deployment of new features or updates.
- **Database (Amazon RDS):** Data storage is crucial for our e-commerce application. Considering the flexibility, performance, and compatibility with Django, we utilize Amazon RDS for PostgreSQL to store product information, user profiles, and transaction records. The backend microservices interact with the database to retrieve and store data as needed. PostgreSQL is known for its reliability and support for complex data structures, making it a good choice for e-commerce applications.
- **Elastic Beanstalk (EB):** The entire application, including frontend, backend microservices, the Resource Allocation Module, and the database, is deployed on AWS Elastic Beanstalk (EB). This platform simplifies application deployment and automatically manages resources, making it an ideal choice for our architecture.
- **Resource Allocation Module (DRA):** The heart of our dynamic resource allocation approach, this module is responsible for analysing user traffic and system conditions to make real-time decisions on resource allocation. It includes components such as Load Balancer, Dynamic Resource Allocator (DRA), DNS (Amazon Route 53), CDN (Amazon CloudFront), and CloudWatch for monitoring.

3.3.3. The rule-based Dynamic Resource Allocation (DRA) logic:

The Rule-Based DRA logic is designed to dynamically allocate computing resources to different application modules based on predefined hypothetical rules and thresholds. It optimizes resource utilization, responding to user requirements and system conditions. The rule-based DRA uses a deterministic algorithm to make real-time decisions regarding resource allocation based on the following factors:

User Requirements (UR): The DRA takes into account user requirements, which can be expressed as UR.

System Conditions (SC): The DRA continuously monitors system conditions, which can be expressed as SC.

Allocation Decision (AD): The DRA determines the allocation decision (AD) based on these factors and predefined rules. AD is expressed as a function of UR, SC, and the defined rules (R1, R2, ... Rn).

Mathematically, the allocation decision (AD) can be represented as:

$$\boldsymbol{AD = f\,(UR, SC, R1, R2, \ldots Rn)}$$

Resource Allocation (RA): Once the allocation decision (AD) is made, the DRA updates the resource allocation (RA) for each application module based on the decision. RA is expressed as:

$$\boldsymbol{RA = CA + \Delta CA}$$

Where:

CA: Current CPU allocation for the application module.

ΔCA: Change in CPU allocation based on the allocation decision (AD) and predefined rules.

Cost Efficiency (CE): The DRA aims to optimize cost efficiency by ensuring that allocated resources are used effectively. Cost efficiency (CE) can be expressed as:

$$\boldsymbol{CE = Cost\,/\,Performance}$$

Where:

Cost: The cost associated with CPU allocation.

Performance: The performance achieved by the allocated resources.

Scalability (S): The DRA ensures that the application can scale resources up or down based on user demand and system load. Scalability can be expressed as:

$$\boldsymbol{S = \Delta CA \,/\, CA}$$

Where:

ΔCA: Change in CPU allocation.

CA: Current CPU allocation.

Fault Tolerance (FT): The DRA enhances fault tolerance by reallocating resources in case of failures or high error rates. Fault tolerance (FT) can be expressed as:

$$\boldsymbol{FT = 1 - (ER \,/\, Total\ Requests)}$$

Where:

ER: Error rate.

Total Requests: Total incoming requests.

3.3.4. Implementing Rule-Based Resource Allocation

The essence of our project lies in the Dynamic Resource Allocator (DRA), a core module meticulously designed to enable intelligent resource allocation within our prototype application. In this section, we delve into the practical aspects of implementing rule-based resource allocation logic for the DRA.

Integration into the Prototype Application

The rule-based DRA logic is seamlessly integrated into the backend of our prototype application. This integration forms the heart of our resource allocation strategy, enabling the application to dynamically adapt to varying user requirements (UR) and system conditions

(SC). The DRA operates continuously, making real-time allocation decisions (AD) based on predefined rules (R1, R2, ... Rn).

Rule Definitions

Let's explore the key rules that define our resource allocation strategy:

Rule 1: High User Traffic

When the user traffic surges, exceeding a predefined threshold (e.g., more than 1000 concurrent users), the DRA springs into action. Its objective is to ensure the application handles the increased load efficiently. It makes resource allocation decisions as follows:

- If CPU utilization exceeds 70% for a sustained period and user traffic remains high, the DRA allocates additional CPU resources.
- If memory usage surpasses 80% for a sustained period and user traffic remains high, the DRA allocates more memory resources.

Rule 2: Low User Traffic

Conversely, when user traffic dwindles, falling below a set threshold (e.g., fewer than 100 concurrent users), the DRA strives for cost-efficiency. It reduces resource allocation as follows:

- If CPU utilization consistently remains below 30% and user traffic is low, the DRA decreases CPU allocation.
- If memory usage consistently remains below 40% and user traffic is low, the DRA reduces memory allocation.

Rule 3: Normal User Traffic

For user traffic within a normal range (typically between 100 and 1000 concurrent users), the DRA maintains the current resource allocation without making significant adjustments. It adopts a more subtle approach:

- It monitors CPU and memory utilization, making minor adjustments within a predefined range (e.g., between 30% and 70%) based on the application's workload. However, it avoids drastic changes to ensure stability.

Implementation and Scripting:

Defining Key Thresholds

To ensure effective resource allocation, we have defined thresholds for both user traffic and resource utilization as in table 2 below. These thresholds serve as essential reference points for our allocation rules. Below, we elaborate on these thresholds:

- **High Traffic Thresholds:** During high user traffic periods (exceeding 1000 concurrent users), we trigger Rule 1 when CPU or memory utilization surpasses defined thresholds.
- **Low Traffic Thresholds:** Conversely, during low user traffic (fewer than 100 concurrent users), Rule 2 is applied when CPU or memory utilization falls below specified levels.
- **Normal Traffic Ranges:** Rule 3 guides resource allocation within defined CPU and memory utilization bounds for user traffic levels typically between 100 and 1000 concurrent users

Metric	**Threshold Value**
High Traffic Users Threshold	1000
High Traffic CPU Threshold	70
High Traffic Memory Threshold	80
Low Traffic Users Threshold	100
Low Traffic CPU Threshold	30
Low Traffic Memory Threshold	40
Normal Traffic CPU Lower Bound	30
Normal Traffic CPU Upper Bound	70
Normal Traffic Memory Lower Bound	30
Normal Traffic Memory Upper Bound	70

TABLE 2 CLOUDWATCH METRICS THRESHOLD

Gathering Metrics:

Our script periodically collects CPU and memory utilization metrics from AWS CloudWatch. These metrics provide valuable insights into the application's performance and resource consumption. The **get_metrics ()** function plays a pivotal role in this data retrieval process as shown in Snippet 1.

```python
def get_metrics():
    response_cpu = cloudwatch_client.get_metric_statistics(
        Namespace="AWS/EC2",
        MetricName="CPUUtilization",
        Dimensions=[
            {
                "Name": "AutoScalingGroupName",
                "Value": auto_scaling_group_name,
            },
        ],
        StartTime="2023-09-01T00:00:00Z",
        EndTime="2023-09-02T00:00:00Z",
        Period=300,
        Statistics=["Average"],
    )

    response_memory = cloudwatch_client.get_metric_statistics(
        Namespace="AWS/EC2",
        MetricName="MemoryUtilization",
        Dimensions=[
            {
                "Name": "AutoScalingGroupName",
                "Value": auto_scaling_group_name,
            },
        ],
        StartTime="2023-09-01T00:00:00Z",
        EndTime="2023-09-02T00:00:00Z",
        Period=300,
        Statistics=["Average"],
    )

    # Extract CPU and memory utilization data
    cpu_metrics = response_cpu["Datapoints"]
    memory_metrics = response_memory["Datapoints"]
```

SNIPPET 1: GET_METRICS () FUNCTION

Resource Allocation Adjustment

The Python script achieves these resource allocation adjustments by interfacing with AWS Elastic Beanstalk. It updates the desired capacity of the application's Auto Scaling group and environment settings based on the defined rules. The script adjusts resource allocation, and the Load Balancer distributes traffic to the Dynamic Resource Allocator (DRA).

```python
# Adjust resource allocation
if desired_capacity != current_capacity:
    response = elasticbeanstalk_client.update_environment(
        EnvironmentName=environment_name,
        OptionSettings=[
            {
                "Namespace": "aws:autoscaling:asg",
                "OptionName": "MinSize",
                "Value": str(desired_capacity),
            },
            {
                "Namespace": "aws:autoscaling:asg",
                "OptionName": "MaxSize",
                "Value": str(desired_capacity),
            },
            {
                "Namespace": "aws:elasticbeanstalk:environment:process:default",
                "OptionName": "MinProcesses",
                "Value": str(desired_capacity),
            },
        ],
    )

    print(f"Resource allocation adjusted. Desired Capacity: {desired_capacity}")
```

SNIPPET 2 ALLOCATION ADJUSTMENT

Chapter 4: Professional and Ethical Considerations

4.1. Professional Codes

Adherence to recognized professional codes of conduct is of utmost importance when undertaking this research and its associated project. Such professional standards or codes serve as ethical benchmarks for researchers and practitioners in the field of information technology. The British Computer Society (BCS) and the Association for Computing Machinery (ACM) are two widely known professional organizations that have established comprehensive standards of conduct that are highly relevant and applicable to our project.

4.1.1. BCS Code of Conduct

The BCS, The Chartered Institute for IT, delineates a Code of Conduct that encompasses principles such as integrity, competence, and professionalism. Within the context of our project, the following principles are particularly relevant **(BCS, 2022):**

- **Public Interest:** As computing professionals, we are dedicated to utilizing our skills and knowledge for the betterment of society. Our dissertation project aligns with this principle as it aims to optimize cloud-based e-commerce applications, ultimately enhancing user experiences and contributing to the improvement of online commerce.
- **Professional Competence and Integrity:** The BCS places a strong emphasis on maintaining professional competence and integrity. Over the course our project, we have consistently strived to uphold these principles by carrying out thorough research, adhering to ethical guidelines, and ensuring the integrity of our data and analysis.
- **Duty to Relevant Authority:** Our project recognizes the importance of complying with relevant legal and regulatory authorities. This is reflected in our ethical considerations, where we ensure that our research and prototype development satisfy all legal requirements.

4.1.2. ACM Code of Ethics and Professional Conduct:

- The ACM provides an all-encompassing Code of Ethics and Professional Conduct that provides guidance to individuals in the computing profession. **(ACM, 2018)** Key principles that align with our project include:
- **Contribute to Society:** Our dissertation project aligns with this principle by contributing to the advancement of cloud-based e-commerce, which has become an integral part of modern society.
- **Respect Privacy:** Privacy is a fundamental ethical consideration in our project. We have implemented data protection measures and ensured that user privacy is upheld in our application. Legal, Social, Ethical, and Environmental Issues

The implications of our dissertation project extend into various dimensions, including legal, social, ethical, and environmental aspects.

4.2.1 Legal Considerations

In the legal context, we have meticulously adhered to copyright regulations and intellectual property rights throughout our research. Proper citations and references have been maintained to respect the work of other researchers and organizations.

4.2.2. Social Impact

Our project's social impact is significant. By enhancing the performance and scalability of cloud-based e-commerce applications, we contribute to providing users with smoother and more efficient online shopping experiences. This can have a positive effect on society by simplifying access to goods and services.

4.2.3. Ethical Aspects

Ethical considerations have been woven into the fabric of our project. We have upheld ethical research practices, ensuring that our data collection, analysis, and reporting are conducted with integrity and respect for individuals' rights and privacy.

4.2.5. Environmental Implications

Although not a primary focus, our project indirectly impacts the environment by promoting efficient resource allocation. By optimizing the performance of cloud-based applications, we reduce the energy consumption associated with data centres and servers, aligning with broader environmental sustainability goals.

Chapter 5: Conclusion and Future Work

In this dissertation research, we embarked on a scholarly exploration to enhance the scalability and performance of cloud-based e-commerce and m-commerce applications by implementing a Dynamic Resource Allocator (DRA). Our journey commenced with a thorough examination of the existing academic literature, delving into the complexities of cloud technologies, and formulating a design for a cloud-based application seamlessly integrated with the DRA.

Our achievements consist of:

Thorough Investigation: We carried out a comprehensive examination of the literature to acquire profound insights into the challenges related to the allocation of resources faced by cloud-based e-commerce and m-commerce applications.

Development of a Prototype: We conceived and created a cloud-based e-commerce and m-commerce application, successfully incorporating the DRA framework. This framework serves as a central module for the intelligent allocation of resources.

Allocation of Resources Based on Rules: We implemented resource allocation strategies guided by rules within the DRA, enabling dynamic adjustment of resource allocation according to user traffic and system conditions.

Future Directions:

Despite the considerable progress we have made in enhancing resource allocation for cloud-based e-commerce and m-commerce applications, there exist a number of opportunities for future research and development:

Integration of Machine Learning: Prospective endeavors can involve the amalgamation of machine learning algorithms to construct more adaptive and predictive models for resource allocation. Machine learning has the capability to scrutinize historical data and user behavior in order to make more precise allocation decisions.

Real-Time Data Processing: The implementation of real-time data processing capabilities within the Dynamic Resource Allocator (DRA) can provide swifter responses to fluctuating traffic patterns, thereby augmenting performance during unforeseen surges in traffic.

Optimization of Costs: Further investigations can focus on optimizing resource allocation not only for performance purposes but also for cost-effectiveness. This entails dynamically adjusting resources to minimize operational costs while simultaneously meeting performance targets.

Multi-Cloud Environments: As organizations progressively embrace multi-cloud strategies, research pertaining to resource allocation and load balancing across multiple cloud providers becomes indispensable.

Considerations for Security: Future endeavours should also delve into the integration of security measures within the resource allocation process to safeguard applications from potential threats.

In conclusion, this project signifies a momentous advancement in addressing the challenges associated with resource allocation encountered by cloud-based e-commerce and m-commerce applications. Through the design and implementation of a rule-based Dynamic Resource Allocator, as well as the undertaking of comprehensive performance analysis, we have laid the groundwork for more efficient and responsive cloud applications. The future holds promising prospects for further research and innovation in this domain, ultimately benefiting businesses and users alike.

Appendices:

GitHub link to the program code:

https://github.com/devSharma31/vamos-new

Bibliography:

List of citations and references following university guidelines or IEEE standards.

Alshahrani, M. (2015) 'Cloud commerce: A drift from E-commerce to cloud', International Journal of Computer Applications, 114(6), pp. 24–26. doi:10.5120/19984-1933.

Bajwa, M.S., Agarwal, A.P. and Gupta, N. (2016) Code optimization as a tool for testing software | IEEE conference ., 3rd International Conference on Computing for Sustainable Global Development (INDIACom). Available at: https://ieeexplore.ieee.org/document/7724405/ (Accessed: 15 August 2023).

Boopathiraja, S., Kalavathi, P. and Dhanalakshmi, C. (2019) 'Significance of image compression and its upshots - A survey', International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 1203–1208. doi:10.32628/cseit1952321.

Davids, F. and Van Belle, J.-P. (2017) 'Understanding the business strategy factors that drive the business impacts of cloud computing', 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence [Preprint]. doi:10.1109/confluence.2017.7943163.

Fard, M.V. et al. (2020) 'Resource allocation mechanisms in cloud computing: A systematic literature review', IET Software, 14(6), pp. 638–653. doi:10.1049/iet-sen.2019.0338.

França, R.P. et al. (2020) 'A review on the technological and literary background of multimedia compression', Handbook of Research on Multimedia Cyber Security, pp. 1–20. doi:10.4018/978-1-7998-2701-6.ch001.

Gunther, N.J. (2001) 'Performance and scalability models for a hypergrowth e-commerce web site', Lecture Notes in Computer Science, pp. 267–282. doi:10.1007/3-540-45156-0_17.

Kavitha, M.G. and Radha, D. (2021) 'Quality, security issues, and challenges in Multi-cloud environment: A comprehensive review', Operationalizing Multi-Cloud Environments, pp. 269–285. doi:10.1007/978-3-030-74402-1_15.

Kosińska, J. and Zieliński, K. (2023) 'Experimental evaluation of rule-based Autonomic Computing Management Framework for cloud-native applications', IEEE Transactions on Services Computing, 16(2), pp. 1172–1183. doi:10.1109/tsc.2022.3159001.

Li, J., Pisano, G. and Zhu, F. (2018) 'Marketplace scalability and strategic use of platform investment', SSRN Electronic Journal [Preprint]. doi:10.2139/ssrn.3310484.

Mohammadpour, S. and Tafte, F. (2016) 'M-commerce: The state of the art, Challenges, and cloud-based solutions', 2016 Eighth International Conference on Information and Knowledge Technology (IKT) [Preprint]. doi:10.1109/ikt.2016.7777789.

Naz, T. (2019) 'Online shopping behaviour technology advancement: A great change in consumer behaviour', SSRN Electronic Journal [Preprint]. doi:10.2139/ssrn.3333079.

Ramu, V.B. (2023) 'Performance impact of microservices architecture', The Review of Contemporary Scientific and Academic Studies, 3(6). doi:10.55454/rcsas.3.06.2023.010.

Vijai, C. and Nivetha, P. (2021) E-commerce on cloud: Opportunities and challenges, SSRN. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3758721 (Accessed: 22 August 2023).

Vu, T.X., Chatzinotas, S. and Ottersten, B. (2018) 'Edge-caching wireless networks: Performance Analysis and Optimization', IEEE Transactions on Wireless Communications, 17(4), pp. 2827–2839. doi:10.1109/twc.2018.2803816.

Anderson, S. (2023) How fast should a website load in 2023?, Hobo. Available at: https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/ (Accessed: 10 September 2023).

Andry, J.F., Christianto, K. and Wilujeng, F.R. (2019) 'Using webqual 4.0 and importance performance analysis to evaluate e-commerce website', Journal of Information Systems Engineering and Business Intelligence, 5(1), p. 23. doi:10.20473/jisebi.5.1.23-31.

Barbar, A. and Ismail, A. (2019) 'Search engine optimization (SEO) for websites', Proceedings of the 2019 5th International Conference on Computer and Technology Applications [Preprint]. doi:10.1145/3323933.3324072.

Barr, J. (2016) Blogs, Amazon. Available at: https://aws.amazon.com/blogs/aws/how-aws-powered-amazons-biggest-day-ever/ (Accessed: 10 August 2023).

Graham-Cumming, J. (2023) Benchmarking Edge Network Performance: Akamai, Cloudflare, Amazon Cloudfront, fastly, and google, The Cloudflare Blog. Available at: https://blog.cloudflare.com/benchmarking-edge-network-performance/ (Accessed: 20 August 2023).

Hearne, S. (2021) Stacking cdns for improved performance and security - akamai, Akamai. Available at: https://www.akamai.com/blog/performance/stacking-cdns-for-improved-performance-and-security (Accessed: 20 August 2023).

Jambhale, P.C. (2019) 'Search engine optimization with google-web', International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 227–230. doi:10.32628/cseit195632.

Kim, T. and Kim, E.-J. (2014) 'Hybrid storage-based caching strategy for Content Delivery Network Services', Multimedia Tools and Applications, 74(5), pp. 1697–1709. doi:10.1007/s11042-014-2215-8.

Lehrig, S. et al. (2018) 'CloudStore — towards scalability, elasticity, and efficiency benchmarking and analysis in cloud computing', Future Generation Computer Systems, 78, pp. 115–126. doi:10.1016/j.future.2017.04.018.

Pavic, F. and Brkic, L. (2021) 'Methods of improving and optimizing react web-applications', 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO) [Preprint]. doi:10.23919/mipro52101.2021.9596762.

Puzhavakath Narayanan, S. et al. (2016) 'Reducing latency through page-aware management of web objects by Content Delivery Networks', Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science [Preprint]. doi:10.1145/2896377.2901472.

SNG, Y.F. (2016) 'Study on factors associated with bounce rates on consumer product websites', Business Analytics, pp. 526–546. doi:10.1142/9789813149311_0019.

Statista, S.R.D. (2023) E-commerce in the United Kingdom (UK), Statista. Available at: https://www.statista.com/topics/2333/e-commerce-in-the-united-kingdom/#topicOverview (Accessed: 15 August 2023).

Zhao, J. et al. (2020) 'Recent developments in Content Delivery Network: A survey', Parallel Architectures, Algorithms and Programming, pp. 98–106. doi:10.1007/978-981-15-2767-8_9.

Zulfa, M.I., Hartanto, R. and Permanasari, A.E. (2020) 'Caching strategy for web application – A systematic literature review', International Journal of Web Information Systems, 16(5), pp. 545–569. doi:10.1108/ijwis-06-2020-0032.

ACM, E.C. (2018) ACM Code of Ethics and Professional Conduct. Available at: https://www.acm.org/binaries/content/assets/about/acm-code-of-ethics-and-professional-conduct.pdf (Accessed: 01 June 2023).

Adegboyega, A. (2017) 'Time-series models for Cloud Workload Prediction: A comparison', 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) [Preprint]. doi:10.23919/inm.2017.7987292.

BCS, T.B. (2022) BCS, The Chartered Institute for IT CODE OF CONDUCT FOR BCS MEMBERS, Code of conduct - BCS, the Chartered Institute for IT | BCS. Available at: https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf (Accessed: September 2023).

Belgacem, A. (2022) 'Dynamic Resource Allocation in cloud computing: Analysis and Taxonomies', Computing, 104(3), pp. 681–710. doi:10.1007/s00607-021-01045-2.

Bellenger, D. et al. (2011) Scaling in cloud environments: Proceedings of the 15th WSEAS International Conference on Computers, Guide Proceedings. Available at: https://dl.acm.org/doi/abs/10.5555/2028299.2028329 (Accessed: 15 July 2023).

D, M. and Nagineni, S.K. (2018) A review on machine learning based Resource Allocation in cloud computing, International Journal of Research . Available at: https://journals.pen2print.org/index.php/ijr/article/download/13476/12720 (Accessed: 15 August 2023).

Fallah, M. and Arani, M.G. (2015) 'Astaw: Auto-scaling threshold-based approach for web application in cloud computing environment', International Journal of u- and e-Service, Science and Technology, 8(3), pp. 221–230. doi:10.14257/ijunesst.2015.8.3.21.

Kosińska, J. and Zieliński, K. (2023) 'Experimental evaluation of rule-based Autonomic Computing Management Framework for cloud-native applications', IEEE Transactions on Services Computing, 16(2), pp. 1172–1183. doi:10.1109/tsc.2022.3159001.

Kougioumtzidis, G. et al. (2022) 'A survey on multimedia services QoE assessment and machine learning-based prediction', IEEE Access, 10, pp. 19507–19538. doi:10.1109/access.2022.3149592.

Mohmmed, T. and Abdalrahman, N. (2021) 'A load balancing with fault tolerance algorithm for cloud computing', 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) [Preprint]. doi:10.1109/iccceee49695.2021.9429597.

Pawar, C.S. and Wagh, R.B. (2012) A review of Resource Allocation Policies in cloud computing - researchgate, 2012 International Symposium on Cloud and Services Computing. Available at: https://www.researchgate.net/profile/Chandrashekhar-Pawar-2/publication/338477941_A_review_of_resource_allocation_policies_in_cloud_computing/links/5e16bac7a6fdcc28376336f6/A-review-of-resource-allocation-policies-in-cloud-computing.pdf (Accessed: 15 August 2023).

Pendyala, A. et al. (2023) ContainerGym: A real-world reinforcement learning benchmark for Resource Allocation, arXiv.org. Available at: https://arxiv.org/abs/2307.02991v1 (Accessed: 10 August 2023).

Raghuvanshi, S. and Dubey, P. (2018) 'Resource scheduling techniques in cloud computing environment : A survey', International Journal of Scientific Research in Science, Engineering and Technology, pp. 64–68. doi:10.32628/ijsrset11841119.

Tang, X. et al. (2021) 'User experience design method from the perspective of scalability', Lecture Notes in Networks and Systems, pp. 640–648. doi:10.1007/978-3-030-80091-8_76.

Yadav, M.P. and Yadav, D.K. (2021) 'Workload prediction for cloud resource provisioning using time series data', Advances in Intelligent Systems and Computing, pp. 447–459. doi:10.1007/978-981-16-2712-5_37.