SIMATS SCHOOL OF ENGINEERING

ETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

CHENNAI-602105

Designing a Cloud-Based Inventory Management System for Azura's E-commerce Platform on AWS

A CAPSTONE PROJECT REPORT

Submitted in the partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

POOJA SRI V (192210152)

Under the Guidance of

Dr. Antony Joseph Rajan D

June 2024

DECLARATION

I am POOJA SRI V**,** student of **'Bachelor of Engineering in Computer Science Engineering**, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Designing a Cloud-Based Inventory Management System for Azura's E-commerce Platform on AWS** is the outcome of our own bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

(POOJA SRI V 192210152)

Date:

Place:

CERTIFICATE

This is to certify that the project entitled **Designing a Cloud-Based Inventory Management System for Azura's E-commerce Platform on AWS** submitted by **POOJA SRI V (192210152)**
has been carried out under our supervision. The project has been submitted as per the requirements for the award of degree.

Project Supervisor

Table of Contents

ABSTRACT:

In the rapidly evolving e-commerce landscape, efficient inventory management is critical for maintaining competitiveness, operational efficiency, and customer satisfaction. Azura's E-commerce Platform is embarking on the development of a cutting-edge Cloud-Based Inventory Management System (CIMS) leveraging the robust infrastructure and advanced services offered by Amazon Web Services (AWS). This abstract outline the design and implementation strategy for this system, emphasizing its architectural components, operational benefits, and strategic advantages.

The proposed CIMS is designed to streamline Azura's inventory processes by utilizing key AWS services. Amazon RDS (Relational Database Service) will be employed for its high availability, scalability, and robust performance in managing relational databases, which are crucial for storing and querying inventory data. AWS Lambda, a serverless computing service, will be used to run code in response to events, enabling real-time processing of inventory updates without the need for managing server infrastructure. Amazon S3 (Simple Storage Service) will provide secure, scalable storage for inventory data, ensuring durability and easy access.

The integration of these services ensures that Azura's inventory system can scale dynamically in response to varying loads, accommodating peak shopping seasons and promotional events without degradation in performance. Real-time inventory updates across multiple sales channels are facilitated by the use of Amazon API Gateway, which acts as a front door for applications to access data, business logic, or functionality from backend services.

One of the core components of the CIMS is the use of Amazon Quick Sight, AWS's business intelligence service, which provides advanced analytics and interactive dashboards. This enables Azura to gain actionable insights into inventory levels, turnover rates, and demand forecasts. The predictive analytics

capabilities of Quik Sight help in making informed decisions about stock replenishment, reducing the risk of overstocking or stockouts.

Furthermore, the CIMS incorporates AWS IoT (Internet of Things) services to enable real-time tracking of physical inventory across multiple warehouses. IoT devices can send real-time data to the cloud, where it is processed by AWS IoT Core and integrated into the inventory management system. This capability ensures accurate tracking and timely updates, enhancing operational efficiency and inventory accuracy.

Introduction:

Efficient inventory management is a cornerstone of success in the fast-paced e-commerce industry. Azura's E-commerce Platform aims to revolutionize its inventory processes by designing a Cloud-Based Inventory Management System (CIMS) leveraging Amazon Web Services (AWS). This system is poised to address the challenges of inventory tracking, operational scalability, and customer satisfaction through the adoption of cloud technologies.

Advanced analytics powered by Amazon Quick Sight will offer actionable insights into inventory levels and demand forecasting, enabling informed decision-making. Additionally, AWS IoT integration will facilitate real-time tracking of physical inventory across warehouses, enhancing visibility and control.

Security and compliance are integral to the design, with AWS's security services ensuring data protection and regulatory adherence. This cloud-based approach not only streamlines operations but also positions Azura to dynamically respond to market demands, reduce costs, and maintain a competitive edge in the e-commerce landscape.

Existing System:

Azura's existing inventory management system relies on manual processes and on-premises infrastructure, leading to inefficiencies, inaccuracies, and scalability issues. The lack of real-time updates and integration with sales channels hampers operational efficiency and customer satisfaction, necessitating the transition to a cloud-based solution on AWS.

I. Static Resource Allocation Models:

- Static resource allocation models allocate fixed computing resources regardless of actual demand, leading to inefficiencies in Azura's inventory management system. These models often result in over-provisioning during low-demand periods, wasting resources and increasing costs, or under-provisioning during peak times, causing performance bottlenecks and service disruptions.
- Such rigidity impedes the system's ability to scale dynamically in response to fluctuating workloads, ultimately affecting the accuracy of inventory tracking and customer satisfaction. Transitioning to AWS's cloud-based services will replace static models with dynamic resource allocation, ensuring optimal performance, cost-efficiency, and scalability to meet varying demand levels.

II. Dynamic Resource Allocation Models:

- Dynamic resource allocation models adapt to real-time demand, optimizing resource use and enhancing system performance for Azura's inventory management. Utilizing AWS's cloud capabilities, these models automatically scale resources up or down based on current workload requirements. AWS Auto Scaling and Elastic Load Balancing ensure that computing power, storage, and network capacity are precisely aligned with demand fluctuations, preventing over-provisioning and under-provisioning issues. This approach reduces costs by eliminating waste and improves system responsiveness during peak shopping periods.

- Dynamic allocation also supports real-time inventory updates across multiple sales channels, enhancing operational efficiency and customer satisfaction. With AWS services like Amazon RDS, Lambda, and S3, resources are provisioned dynamically, maintaining high availability and reliability. Additionally, the integration of advanced monitoring tools like Amazon CloudWatch provides insights and automated adjustments to resource allocation, ensuring seamless performance and scalability for Azura's cloud-based inventory management system.

III. **Energy-Aware Scheduling Algorithms**:

- Energy-aware scheduling algorithms optimize resource usage in Azura's cloud-based inventory management system by minimizing energy consumption while maintaining performance. Leveraging AWS's capabilities, these algorithms intelligently allocate tasks to computing resources based on their energy profiles, workload demands, and operational efficiency.
- Services like AWS Lambda, which only consumes power during execution, and EC2 instances with optimized energy settings, help reduce the overall carbon footprint.
- This approach not only supports sustainable business practices but also aligns with Azura's goals of operational efficiency and cost-effectiveness, creating a greener, more efficient cloud-based inventory management system

Proposed System:

Key Components

i. **Negotiation Agents**:

Negotiation agents in Azura's cloud-based inventory management system act as autonomous entities that manage resource allocation and task scheduling.

These agents negotiate resource usage based on current demand, energy efficiency, and operational priorities. By leveraging AWS’s AI and machine learning services, such as AWS SageMaker, these agents can make intelligent, data-driven decisions to optimize performance and reduce costs. Negotiation agents ensure that resources are dynamically allocated to balance workload and efficiency, enhancing the overall system responsiveness and reliability.

ii. **Energy-Aware Metrics**:

Energy-aware metrics are essential for monitoring and optimizing the energy consumption of Azura's inventory management system. These metrics include power usage effectiveness (PUE), carbon footprint, and energy efficiency ratio (EER). By integrating AWS CloudWatch and AWS Cost Explorer, Azura can continuously track these metrics, gaining insights into energy consumption patterns and identifying opportunities for optimization. Energy-aware metrics ensure that the

iii. **Negotiation Framework**:

The negotiation framework facilitates interactions between negotiation agents and other system components to achieve optimal resource allocation. This framework leverages AWS Step Functions to orchestrate complex workflows and manage negotiations in real time. It ensures that resource allocation decisions are made collaboratively, considering factors such as workload demand, energy efficiency, and service level agreements (SLAs). The negotiation framework enhances the system's adaptability and efficiency, ensuring seamless integration of new resources and technologies as needed.

iv. **Adaptive Algorithms**:

Adaptive algorithms are crucial for the dynamic and efficient operation of Azura's inventory management system. These algorithms, implemented using AWS Lambda and AWS Auto Scaling, enable the system to automatically adjust resource allocation in response to changing workloads and operational conditions.

System Workflow

1. **Initialization**:

   During the initialization phase, the cloud-based inventory management system is set up and configured. Key AWS services such as Amazon RDS, AWS Lambda, and Amazon S3 are provisioned. Initial inventory data is migrated to the cloud, and baseline configurations are established for dynamic resource allocation and energy-aware metrics. The system's negotiation agents are activated, and initial parameters for adaptive algorithms are defined. This phase ensures that the system is ready to handle inventory management tasks with the necessary infrastructure in place.

2. **Negotiation Phase**:

   In the negotiation phase, negotiation agents come into play. These agents continuously analyze real-time data on inventory levels, demand forecasts, and energy usage metrics. They communicate with each other and with the negotiation framework, implemented using AWS Step Functions, to determine the optimal resource allocation. This phase involves dynamic adjustments to workload distribution and energy consumption, ensuring that resources are allocated efficiently based on current needs and energy efficiency goals. The negotiation framework ensures that decisions are made collaboratively and in real time.

3. **Resource Allocation**:

   The resource allocation phase involves the actual distribution of cloud resources based on the decisions made during the negotiation phase. AWS Auto Scaling adjusts the number of EC2 instances and other resources to match the current workload, while AWS Lambda scales function execution to handle event-driven processes. The adaptive algorithms, implemented using machine learning models on AWS SageMaker, predict demand patterns and adjust resource allocation proactively. This phase ensures that the system can handle varying workloads efficiently, maintaining high performance and minimizing energy consumption.

4. **Performance and Energy Monitoring**:

   In the performance and energy monitoring phase, the system continuously tracks key metrics to ensure optimal operation. AWS CloudWatch and AWS Cost Explorer provide real-time monitoring of performance metrics such as response times, throughput, and resource utilization, as well as energy-aware metrics like power usage effectiveness (PUE) and carbon footprint. The monitoring tools alert the system to any anomalies or inefficiencies, allowing for immediate adjustments. This phase ensures that the system operates efficiently and sustainably, with ongoing optimization of both performance and energy usage.

Implementation:

I. **Infrastructure Setup**

The first step in implementing the cloud-based inventory management system is to set up the necessary infrastructure on AWS. This involves:

- ➢ Provisioning Amazon RDS for relational database management to store inventory data.
- ➢ Setting up AWS Lambda for serverless computing to handle event-driven processes.
- ➢ Configuring Amazon S3 for secure, scalable data storage.
- ➢ Establishing Amazon API Gateway for enabling API interactions.
- ➢ Configuring AWS IoT Core for real-time tracking of physical inventory.

II. **Development of Negotiation Agents**

- ➢ Negotiation agents are developed to manage resource allocation and task scheduling. This involves:
- ➢ Utilizing AWS Sage Maker to build and train machine learning models that enable agents to make intelligent, data-driven decisions.

- Implementing communication protocols between agents using AWS Step Functions to ensure efficient negotiation and coordination.
- Coding the agents to respond to real-time data and adjust resource allocation based on current inventory levels, demand, and energy efficiency metrics.

III. Energy-Aware Metrics and Monitoring

Setting up energy-aware metrics involves:

- Integrating AWS CloudWatch and AWS Cost Explorer to monitor and track energy consumption and performance metrics.
- Defining key energy metrics such as power usage effectiveness (PUE), carbon footprint, and energy efficiency ratio (EER).
- Implementing real-time alerts and dashboards to provide visibility into energy usage patterns and opportunities for optimization.

IV. Negotiation Framework and Protocols

- The negotiation framework facilitates the interactions between negotiation agents and other system components. This involves:
- Using AWS Step Functions to orchestrate complex workflows and manage negotiations in real-time.
- Establishing protocols for how negotiation agents interact with the system to ensure collaborative and efficient resource allocation.
- Integrating the negotiation framework with other AWS services to ensure seamless operation and communication.

V. Adaptive Algorithms

Developing adaptive algorithms includes:

- Implementing machine learning models using AWS SageMaker that predict demand patterns based on historical data.
- Coding algorithms that automatically adjust resource allocation in response to changing workloads and operational conditions.
- Utilizing AWS Lambda and AWS Auto Scaling to enable real-time adjustments and scalability.

VI. System Integration and Testing

1. Integration Planning

2. Identify Integration Points: List all the components that need to be integrated (e.g., web application, database, third-party services).
3. Define Integration Workflow: Create a workflow diagram that shows how data flows between different components.
4. Integration Environment Setup

5. Development Environment: Set up a development environment using AWS Cloud9 or a similar IDE.
6. Staging Environment: Set up a staging environment that mimics the production environment as closely as possible.
7. Component Integration

VII. Performance Evaluation

1. Manual Testing

2. Exploratory Testing: Perform exploratory testing to uncover any issues that automated tests might miss.
3. User Acceptance Testing (UAT): Conduct UAT with a group of end-users to ensure the system meets their needs and expectations.
4. VII. Performance Evaluation
5. Define Performance Metrics

6. Response Time: Measure the time taken to respond to user requests.
7. Throughput: Measure the number of transactions processed per second.
8. Scalability: Assess how well the system scales with increasing load.
9. Availability: Measure the system's uptime and reliability.
10. Performance Testing Tools

Future scope:

Future enhancements could include implementing machine learning algorithms for demand forecasting, further automation of supplier interactions, and enhanced analytics using AWS services like Amazon Redshift. Integrating IoT devices for real-time inventory tracking and expanding multi-warehouse management capabilities would also add significant value.

Conclusion:

The cloud-based inventory management system designed for Azura will streamline operations, improve accuracy, and enhance scalability. Leveraging AWS's robust infrastructure ensures high availability and security, positioning Azura to efficiently manage inventory and meet growing customer demands.