Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

CLOUD BASED SIMULATION PLATFORM (CSP): A NOVEL WAY TO DEMOCRATIZE SIMULATION BASED EXPERIMENTATION

Rohan Vaidya[1], Abhineet Mittal[1], and Ganesh Nanaware[1]

[1]Worldwide Design and Engineering, Amazon.com, Seattle, WA, USA

ABSTRACT

Organizations today are integrating technologies such as cloud computing and digital twins in their manufacturing and logistical processes. In a capital-intensive logistics industry, discrete event simulation (DES) plays a crucial role in distribution center design, automation system performance analysis, optimization, and operational planning. Developing and deploying DES models demands proficiency in various simulation software and programming languages, imposing limitations on the widespread use of simulation for experimentation. This paper presents a cloud simulation platform (CSP) based on Amazon Web Services that is a secure and scalable solution for seamless execution and democratization of DES. CSP empowers simulation practitioners to execute simulation models and perform scientific experiments. The paper also provides details of the CSP architecture consisting of data import and export modules and a simulation integration module. As an example, a simulation-based staffing tool deployed through CSP is presented.

1 INTRODUCTION

Over the years, cloud computing as a technology has become progressively cheaper providing data storage capabilities and high performance computing at affordable prices. The emergence of competing cloud service providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform provide customers with multiple options for leveraging cloud computing. In manufacturing and logistical applications, the need for swift and precise decision-making pertaining to design and operations is critical. Simulation stands out as an essential tool empowering stakeholders to make optimal, data-driven decisions. Moreover, simulation plays a critical role in operations for resource planning and continuous improvement initiatives. Traditionally, executing complex simulation models has been computationally expensive, time-consuming, and required a simulation scientist for model development and sensitivity studies. The cloud simulation platform (CSP) enables developers to deploy sub-system simulation tools on the cloud, resulting in a significant reduction in computation costs and time investment.

2 LITERATURE REVIEW

Cloud-based simulation has been mentioned in the literature as early as Fishwick (1996), who talked about web-based simulation (WBS) as an idea. He describes the various advantages of WBS as (i) vast storage space is avaiable compared to local computers; (ii) simulation users can easily access the models through web browsers and can run multiple runs as needed; (iii) users across the world can execute simulation runs in parallel; and (iv) simulation models require sophisticated software and expensive hardware for high performance processing; these processing and licensing costs are handled by the cloud server instead of each individual user. Along these lines, Taylor et al. (2012) list various requirements for a cloud-based simulation solution. Some of these features include (i) accessibility on mobile devices, (ii) enabling multi user collaboration for shared experimentation, (iii) ability to connect to input data sources and incorporate live data, and (iv) enabling distributed and parallel simulation execution. The field of cloud-based simulation is still in its infancy and prior research in this field is scant. There are multiple open source and commercially available cloud simulation tools. As an example, Anylogic (2024) provide cloud-based enterprise simulation

solutions. Several researchers have developed cloud simulation platforms for running and executing discrete event simulations. Padilla et al. (2014) developed a cloud-based discrete event simulator primarily aimed towards introducing simulation concepts to students. They built an interactive user interface that can be accessed on mobile devices and through social media. Hofmann et al. (2022) presented a method for deploying DES models on AWS. They implemented a synchronous simulation architecture, where the processed data are fed in real time to a lambda function, which triggers the model build and run. Liu et al. (2012) explored the methodology to develop models in existing simulation software to run efficiently on a cloud. Heavey et al. (2014) developed an open source cloud platform for executing python-based simulation programs. They developed a custom web user interface to interact with the simulation server and a knowledge extraction module for processing simulation outputs. Several researchers have also talked about creating a digital twin (DT) using DES in logistics and manufacturing. Agalianos et al. (2020) performed a comprehensive review of the intersection of the current research in the space of DTs and DES. They concluded that the current state of art research is still in its embryonic stage and more research is needed in this space. Korth et al. (2018) presented a system architecture that combines a real-time DT of logistics systems with simulation logic in a single modularized model. Sakr et al. (2021) present an approach for a DT-based DES model. The proposed framework is applied to a semiconductor manufacturing system as a use case.

The current state of research has very few papers discussing the use of AWS for deploying cloud-based simulation. Additionally, to the best of the authors' knowledge, there is no research addressing scalability and security. For the application considered here, an architecture is required that could be scaled to worldwide operations in a secure manner. The solution should be easy to use for operation without high training effort. As there was no such solution addressing these requirements, a specific AWS architecture has been developed. In this paper, an AWS architecture is presented for executing simulation runs tailored towards manufacturing and operational users. It provides network security by authenticating users and scalability through the use of an application load balancer. It also enables users to upload simulation input data through the web and automatically exports simulation output.

3 CLOUD SIMULATION PLATFORM ARCHITECTURE

3.1 Overview

Figure 1 presents an overview of the CSP platform architecture which consists of three modules: Data Import Module, Simulation Integration Module (SIM), and Data Export Module. The simulation scientist develops and uploads the model onto the SIM. The SIM hosts the simulation models, the software license, as well as the data files for input and output. The user connects to the SIM through a web-based user interface (UI), which allows them to upload input files and start or stop simulation runs. The user can setup various simulation experiments by defining input variables through the input data file. The Data Import module feeds the input data file from the user into the SIM. The simulation model hosted in the SIM can access the input file for running simulation experiments. The output data file is created in the SIM and exposed through the Data Export module to the user. The following sections dive into the architecture of each module.

3.2 Simulation Integration Module

Figure 2 shows the simulation integration module architecture. The entire architecture is set up in a virtual private cloud (VPC). The simulation integration module consists of an EC2 cloud server instance, which hosts the simulation models and handles the simulation software licensing. The EC2 instance is hosted in a private subnet with restricted access through an application load balancer (ALB) and a windows bastion server. The ALB and the bastion server are hosted in a public subnet. The simulation models can be accessed through a UI, which is hosted on a secure domain. Incoming user web traffic is routed through the ALB. The ALB enables authentication of users through an identity provider (IdP) that is compliant

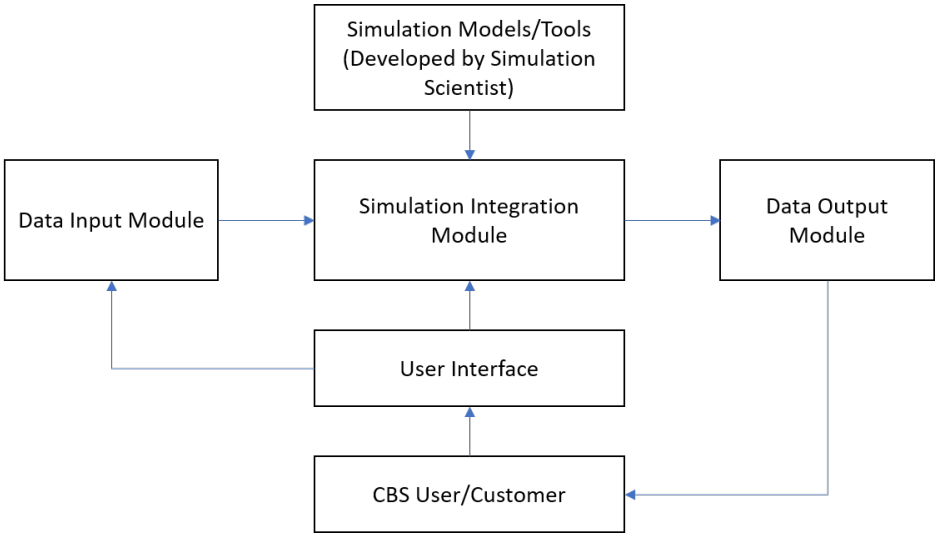

Figure 1: CSP overview.

with OpenID Connect (OIDC). An additional layer of security is added by creating authorized user groups for allowing access to the domain that hosts the UI. The ALB also provides the flexibility of launching multiple EC2 instances, each hosting a separate simulation server. This enables the architecture to be scaled to the customer's needs. The bastion server allows simulation developers and server admins to access the simulation server through the remote desktop protocol (RDP). The simulation UI allows the users to import data files into the model and run the model. The simulation model is set up to export the desired key performance indicators (KPIs) to a spreadsheet, which is stored in the cloud server.

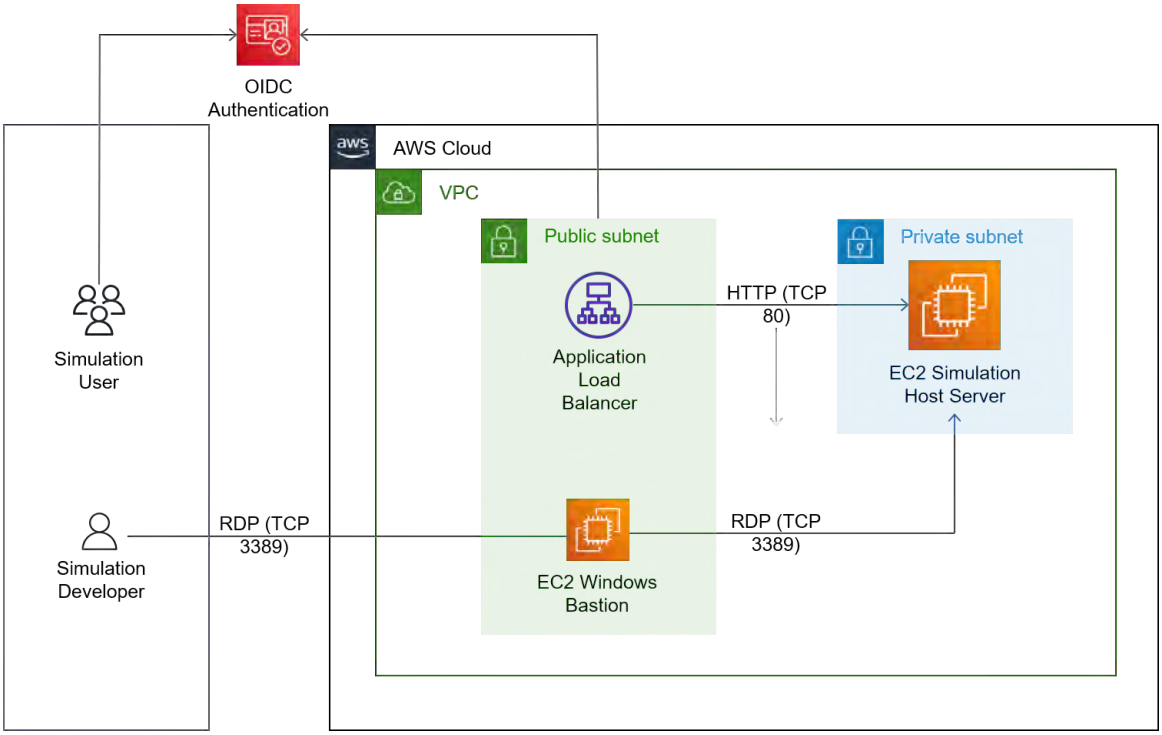

Figure 2: Simulation Integration Module architecture.

3.3 Data Import Module

The Data Import module enables the user to upload the simulation input file to the SIM. The file is unique to each simulation model hosted in the SIM. The file is initially created by the simulation scientist based on user requirements, allowing the user to modify certain input parameters and input data of the model. Figure 3 shows the architecture of the Data Import Module. It consists of an AWS lambda function, which generates an HTML form as a response to the user's HTTPS request. The user's HTTPS request is routed via an application load balancer (ALB). As stated in the previous section, the ALB enables authentication of users through an identity provider (IdP) that is compliant with OIDC. After authentication, the user can upload the input file through the UI. The file is uploaded to AWS S3 using HTTPS POST. HTTPS Post simplifies uploads and reduces upload latency where users upload data to store in Amazon S3. The S3 bucket is synced with the EC2 instance, which hosts the simulation models. All input files are uploaded to the cloud server.

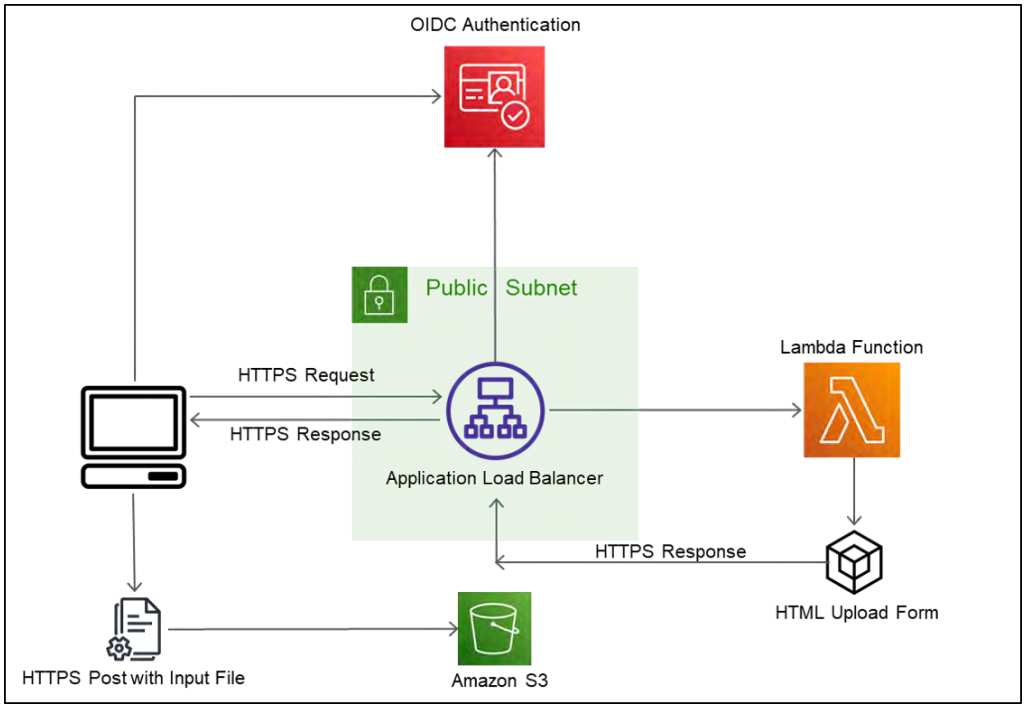

Figure 3: Data Import Module architecture.

3.4 Data Export Module

The Data Export Module facilitates exporting the simulation output KPIs to the end user. The output file provides the user with the input variables and corresponding output KPIs for the simulation run. The output file is uploaded to a desired location specified by the customer. Figure 4 shows the Data Export Module architecture. The simulation model creates an output file on the cloud server in the SIM. The server uploads the output file to a destination S3 folder. The upload of the file to S3 triggers a lambda function, which uploads the file to the user-specified destination.

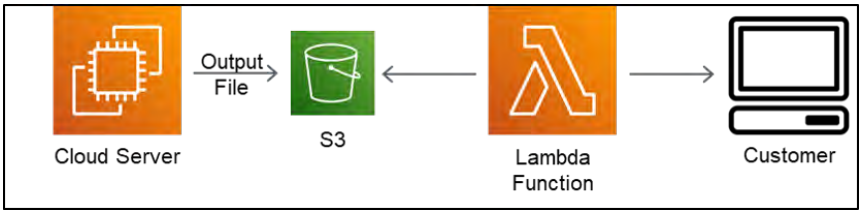

Figure 4: Data Export Module architecture.

4 APPLICATION OF CSP FOR DES-BASED TOOL DEPLOYMENT

Multiple DES-based tools have been developed by the authors for applications in design and operation. The users for CSP include a variety of teams including operations, design engineering, and technology partners. This section provides details on a staffing tool developed for operations.

4.1 Overview

The inbound staffing tool (IST) is a tool developed for inbound (IB) operations planning of the distribution center or warehouse. It is used by the operation manager (OM) at the start of each shift for optimal staff planning based on the IB freight mix. The tool simulates the process from IB dock to storage and evaluates KPIs such as system throughput, decanted units or SKUs per tote (UPT), operator utilization, and material handling equipment (MHE) performance. Based on the KPIs, the tool provides a forward-looking dynamic staff plan by accounting for freight mix and volume variation.

4.2 Problem Statement

Figure 5 shows an overview of the inbound process at the distribution center. IB totes are directly stored on the storage floors and case contents are decanted into totes. There are multiple challenges in an IB process at distribution centers that make it difficult to create optimal staff plans. Trailers arriving at the inbound area contain cases and totes of varying sizes and units per container, complicating the unloading process. Larger unit sizes result in fewer units per container in the decant area, leading to constraints on MHE and limiting storage floor capacity for processing the inbound volume. Similarly, an increased mix of cases in IB trailers may overwhelm decanters while starving downstream processes. Due to these complexities, it is imperative to optimize staffing levels and trailer sequencing based on freight mix, MHE capacity, and trailer contents. Traditional spreadsheet-based staffing tools fail to capture the variability in the freight mix and their impact on downstream processes. To address this gap, a simulation-based staffing tool generates an optimal staff plan based on the end-to-end system.

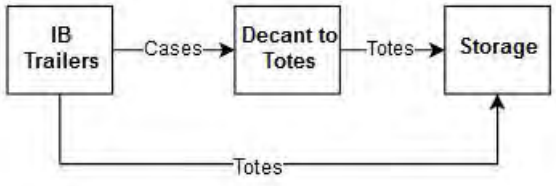

Figure 5: Inbound process at distribution center.

4.3 Simulation Model Development and Inputs

Prior to building the simulation, multiple stakeholders have been involved from operations, process engineering, maintenance, and vendors to validate simulation assumptions. The model is built in the simulation

software FlexSim (Nordgren 2002). Table 1 shows the critical parameters considered for model building. The simulation models accounts for all MHE details including speed, control logic. and conveyor types (accumulation and transportation). The model simulates the logic of the warehouse management system (WMS) for assigning destinations to flow items. Additionally, the simulation also models manual processes in detail accounting for process rates based on historical data.

Table 1: Simulation input parameters.

Metric	**Source**
Spatial layout	AutoCAD design layout
Hourly trailer data	Live database
MHE specifications	Vendor
Sorter control logic	WMS
Planned shift volume	Operation plan
Planned shift headcount	Operation Plan
Decant rate	Historical observations
Stow rate	Historical observations

4.4 Trailer Selection Optimization

An optimization function has been developed for level loading of trailers based on the inbound freight mix. The optimization creates a subset of available trailers in the yard based on unload priority and volume requirement for a specified time window. This subset of trailers is sequenced and an unload order is assigned. The optimization function tries to reduce consecutive trailers with high case mix and unit size. The unit sizes are classified into three buckets: small, medium, and large. By factoring in case mix and unit sizes for sequencing trailers, downstream process starvation is minimized and overall throughput increased. The mathematical formulation of the optimization is detailed below. Table 2 shows the parameters used in the optimization function and Table 3 shows the indices used for number of trailers and number of trailer positions

Table 2: Optimization parameters.

Parameter	**Description**
nCase	Cases per trailer
nTotal	Cases + totes in a trailer
Sm	Percentage of small sized ASINs per trailer
Cs	Percentage of cases per Trailer
Sm_avg	Average Smalls across selected trailers (%)
Cs_avg	Average case across selected trailers (%)
nTrailers	Number of trailers
w	Weight

Table 3: Optimization indices.

Index	**Description**
I	Number of trailers (i = 1,2,3… nTrailers)
J	Number of positions (j = 1,2,3… nPositions)

The objective function for the optimization is defined using two variables, Sm_var and Cs _var. For each position in the sequence, the deviation of the smalls percentage and case percentage from the average are calculated. The difference in deviation from the average for consecutive trailers in sequence is defined as Sm_var and Cs _var. Equation 1 shows the Sm_var calculation for the first trailer in the sequence.

$$Sm\_var(i) \;=\; abs(Sm(i) * \sum_{j}^{J} x_{ij} - Sm\_avg) * w \;\; where \; i \;=\; 1 \tag{1}$$

Equation 2 shows the Sm_var calculation for all the subsequent trailers in the sequence.

$$Sm\_var(i) \;=\; abs(Sm(i) * \sum_{j}^{J} x_{ij} - Sm\_avg) * w + abs(Sm(i-1) * \sum_{j}^{J} x_{ij} - Sm\_avg) * w \tag{2}$$

$$where \; i \geq 2, \; for \; each \; i \; \varepsilon \; I$$

Equation 3 shows the Case_var calculation for the first trailer in the sequence.

$$Case\_var(i) \;=\; abs(Cs(i) * \sum_{j}^{J} x_{ij} - Cs\_avg) * w, \; where \; i \;=\; 1 \tag{3}$$

Equation 4 shows the Case_var calculation for all the subsequent trailers in the sequence.

$$Cs\_var(i) \;=\; abs(Cs(i) * \sum_{j}^{J} x_{ij} - Cs\_avg) * w + abs(Cs(i-1) * \sum_{j}^{J} x_{ij} - Cs\_avg) * w, \tag{4}$$

$$where \; i \geq 2, \; for \; each \; i \; \varepsilon \; I$$

Equation 5 and Equation 6 ensure that there is only one trailer assigned for each position in the sequence and only one position assigned per trailer.

$$\sum_{j}^{J} x_{ij} = 1, for \; each \; i \; \varepsilon \; I \tag{5}$$

$$\sum_{i}^{I} x_{ij} = 1, for \; each \; j \; \varepsilon \; J \tag{6}$$

On this basis, the Objective Function 7 follows:

$$\text{Minimize: } \sum_{i}^{I} (Sm\_var(i) + Case\_var(i)) \tag{7}$$

4.5 Deployment of the Tool on the Cloud Simulation Platform

This section describes the implementation of the staffing tool on the CSP. Figure 6 presents an overview of the implementation process. The OM for the shift is trained to create the input file and execute the simulation run. The trailer sequence optimization is performed outside of the simulation model and serves as input to the simulation. The input file and the trailer sequence are uploaded to the cloud through the UI of the Data Import Module. When the file is uploaded, the username and user team name are automatically tagged to the file. This information is maintained throughout the process allowing the data export module to share the output file to the specific user or team. This enables a varied set of users to use the UI to upload input files concurrently. The simulation model is uploaded by the developer and is hosted in the SIM. After uploading the input file, the simulation runs are executed through a UI generated by the SIM.

The UI allows the user to run multiple simulation runs simultaneously. After a run is completed, the data export module sends an automated email to a predefined user group with the output file. Table 4 shows the output provided by the simulation model. It provides the OM with required hourly staffing for decant and stow based on the IB trailer mix for the day. The recommended hourly staffing aims to maximize operator utilization and provides guidance to the OM for resource planning during the shift. After receiving the output file, the OM can choose to run multiple scenarios by varying input parameters to evaluate their impact on shift performance.

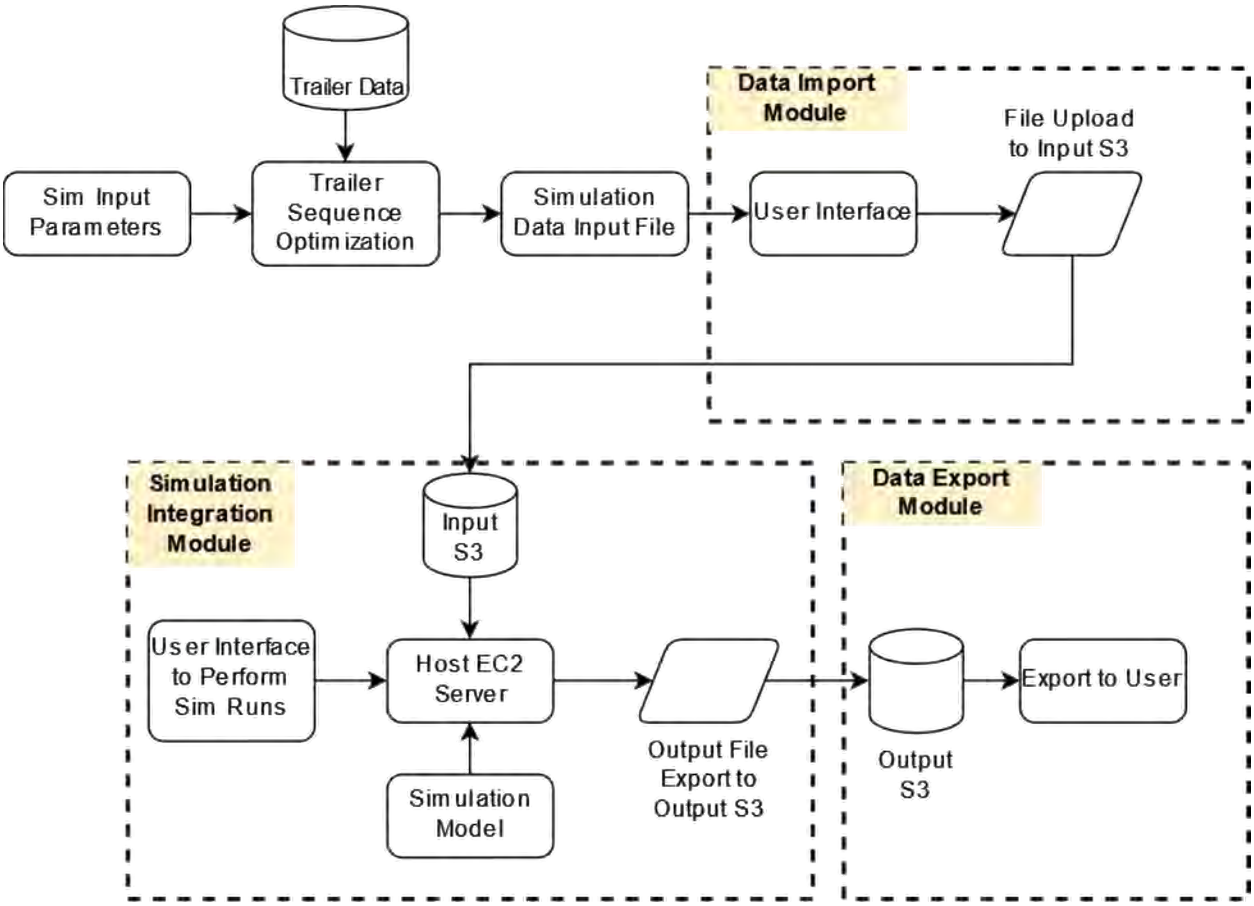

Figure 6: Implementation of tool on the cloud platform.

Table 4: Staffing tool output.

Parameter	**Description**
Decant Staffing	Hourly headcount required for decant
Stow Staffing	Hourly headcount required for decant
Decant Rate	Units per hour decanted
Stow Rate	Units per hour stowed
Trailers Unloaded	List of trailers unloaded

4.6 Tool Adaptation and Validation

To ensure the effectiveness and reliability of the inbound staffing tool (IST), a 15-day pilot study has been conducted at an existing distribution center. The goal was to compare the staffing plan generated by the IST against the traditional spreadsheet-based staffing approach used by the operations team. The simulation model developed for the IST accurately replicated the end-to-end inbound process, from dock unloading to inventory stowing on the storage floors. The model captured critical details such as processing rates, MHE constraints, and WMS control logic. By leveraging the trailer sequence as an input, the simulation could generate a detailed hourly staffing plan, which accounted for the variability in the inbound freight mix. During the pilot, ae close collaboration was installed with the operation team to gather their feedback and address any concerns. One of the key challenges was earning the trust of the operation personnel in using the predicted headcount from the simulation-based tool, as they were accustomed to the traditional spreadsheet-based approach. To overcome this challenge, the operation team has been actively involved in the pilot and provided with a detailed walk-through of the tool's functionality. They have also been provided with comprehensive user guides and training videos to ensure a smooth on-boarding process for teams with varying technical expertise.

Figure 7 illustrates the decrease in IST-recommended staffing compared to the existing spreadsheet-based staffing plan. On average, the IST achieved a 7 % reduction in staffing requirements while maintaining the same throughput level. The successful pilot demonstrated the effectiveness of the simulation-based staffing tool in generating optimal staffing plans that account for the dynamic inbound freight mix. The operations team appreciated the tool's ability to provide data-driven insights and improve resource utilization, leading to increased confidence in adopting the solution. Moving forward, the tool will be continuously refined based on the feedback from the operation team and opportunities explored to further enhance its capabilities. Additionally, the authors plan to investigate ways to integrate the simulation model with the actual warehouse management system, enabling real-time data exchange and synchronization for even more accurate and responsive decision support.

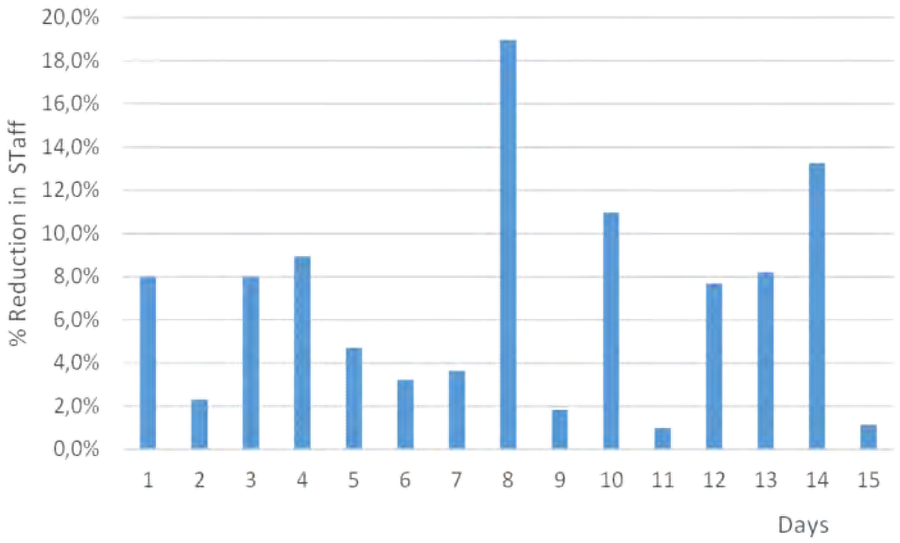

Figure 7: Reduction in staff through IST implementation.

5 CONCLUSION AND FUTURE SCOPE

This paper has introduced the Cloud Simulation Platform (CSP), a secure and scalable solution designed to democratize simulation for design and operation applications. By leveraging the power of cloud computing, the CSP enables the seamless deployment of simulation-based tools, expediting problem-solving and promoting data-driven decision-making across various industries. As an illustrative example, the successful implementation of a simulation-based staffing tool for inbound operations planning at distribution centers has been wpresented, deployed through the CSP. The tool leverages discrete event simulation to generate optimal staffing plans based on inbound freight mix, maximizing throughput and operator utilization while minimizing staffing requirements.

Looking ahead, a key aspect of future research involves the incorporation of cloud-based emulation capabilities into the CSP. This entails establishing bidirectional connections between simulation models and warehouse management systems (WMS), paving the way for a more integrated and comprehensive solution. By enabling real-time data exchange and synchronization between the simulation environment and the actual operational systems, the CSP will unlock new possibilities for advanced decision support, what-if scenario analysis, and continuous process improvement.

Furthermore, ongoing efforts include scaling the CSP to accommodate different kinds of simulation software. The CSP architecture has been developed using Flexsim, but it can be expanded or scaled to various other tools including physics-based simulation software. Physics simulation is particularly relevant in product design applications in logistics chute design, cart design and robotic end of arm tools. The platform could be expanded to support a wider range of simulation paradigms, such as agent-based modeling, system dynamics, and hybrid simulations, catering to diverse application domains beyond manufacturing and logistics. As the adoption of digital technologies continues to accelerate across industries, the demand for sophisticated simulation tools and platforms is poised to grow. The CSP positions itself as a powerful enabler, bridging the gap between cutting-edge simulation capabilities and the needs of diverse stakeholders, from operation managers to design engineers. By democratizing access to simulation and fostering collaboration, the CSP holds the potential to drive innovation, enhance operational efficiency, and unlock new frontiers in data-driven decision-making.

REFERENCES

Agalianos, K., S. Ponis, E. Aretoulaki, G. Plakas, and O. Efthymiou. 2020. "Discrete Event Simulation and Digital Twins: Review and Challenges for Logistics". *Procedia Manufacturing* 51:1636–1641.

Anylogic. 2024. AnyLogic Simulation Software. https://www.anylogic.com/, accessed 27th June.

Fishwick, P. 1996. "Web-based Simulation: Some Personal Observations". In *1996 Winter Simulation Conference (WSC)*, 772–779 https://doi.org/10.1145/256562.256807.

Heavey, C., G. Dagkakis, P. Barlas, I. Papagiannopoulos, S. Robin, M. Mariani, *et al*. 2014. "Development of an Open-Source Discrete Event Simulation Cloud Enabled Platform". In *2014 Winter Simulation Conference (WSC)*, 2824–2835 https://doi.org/10.1109/WSC.2014.7020124.

Hofmann, W., S. Lang, P. Reichardt, and T. Reggelin. 2022. "A Brief Introduction to Deploy Amazon Web Services for Online Discrete-event Simulation". *Procedia Computer Science* 200:386–393.

Korth, B., C. Schwede, and M. Zajac. 2018. "Simulation-ready Digital Twin for Realtime Management of Logistics Systems". In *2018 IEEE International Conference on Big Data (Big Data), December 10th–13th, Washington DC, USA*, 4194–4201.

Liu, X., Q. He, X. Qiu, B. Chen, and K. Huang. 2012. "Cloud-based Computer Simulation: Towards Planting Existing Simulation Software into the Cloud". *Simulation Modelling Practice and Theory* 26:135–150.

Nordgren, W. 2002. "Flexsim Simulation Environment". In *2002 Winter Simulation Conference (WSC)*, Volume 1, 250–252 https://doi.org/10.1109/WSC.2002.1172892.

Padilla, J. J., S. Y. Diallo, A. Barraco, C. J. Lynch and H. Kavak. 2014. "Cloud-based Simulators: Making Simulations Accessible to Non-experts and Experts Alike". In *2014 Winter Simulation Conference (WSC)*, 3630–3639 https://doi.org/10.1109/WSC.2014.7020192.

Sakr, A. H., A. Aboelhassan, S. Yacout, and S. Bassetto. 2021. "Building Discrete-Event Simulation for Digital Twin Applications in Production Systems". In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), September 7th–10th, Vasters, Sweden* https://doi.org/10.1109/ETFA45728.2021.9613425.

Taylor, S. J. E., R. Fujimoto, E. H. Page, P. A. Fishwick, A. M. Uhrmacher, and G. Wainer. 2012. "Panel on Grand Challenges for Modeling and Simulation". In *2012 Winter Simulation Conference (WSC)*, 2614–2628 https://doi.org/10.1109/WSC.2012.6465310.

AUTHOR BIOGRAPHIES

ROHAN VAIDYA is a Senior Simulation Scientist with Amazon. He received his Master degree in Mechanical Engineering from the University of Cincinnati in 2017. His research interests include discrete event simulation, cloud based simulation, process optimization for operational excellence, and machine learning. His email address is rohanvai@amazon.com.

ABHINEET MITTAL is a Senior Science Lead for Core Distribution Simulation within the worldwide design and engineering team. He obtained his Master degree in Industrial Engineering from Arizona State University in 2014. He has over ten years of industry experience with two years in Fiat Chrysler Automobiles and eight years at Amazon Simulation Science. His research interests are simulation optimization, robotics, and machine learning. His email address is abhineem@amazon.com.

GANESH NANAWARE is a Senior Manager of Research Science with Amazon's Worldwide Design Engineering. He received a Master in Mechanical Engineering, a Master in Business Administration, MIT Machine Learning, and PMP certification. He has more than 20 years professional experience in leading the research and simulation teams in various industry sectors. His research interest includes science-driven process and product optimization, AI-based simulation, machine learning, distribution process simulation to drive innovation, and research-based strategy development. His email address is nanawg@amazon.com.