

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

Rohan Vaidya¹, Abhineet Mittal¹, and Ganesh Nanaware¹

¹Worldwide Design and Engineering, Amazon.com, 410 Terry Ave N, Seattle, WA 98109 USA

ABSTRACT

Organizations today are integrating technologies such as cloud computing, and digital twin 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 widespread use of simulation for experimentation. This paper presents an AWS based cloud simulation platform (CSP) which is a secure and scalable solution for seamless execution and democratization of discrete event simulation. 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/export modules and a simulation integration module. As an example, we present a simulation-based staffing tool deployed through CSP.

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 CSP platform 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 literature as back as (Fishwick 1996) who talked about web-based simulation (WBS) as an idea. They describe various advantages of WBS 1) Storage Space: The cloud infrastructures enable vast storage space as compared to local computers 2) Reuse and Accessibility: Simulation users can easily access the models through web browsers and can run multiple runs as needed. 3) Multi-user capability: Users across the world can execute simulation runs in parallel. 4) Reduced costs: 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) lists various features/requirements for a cloud based simulation solution. Some of these features include: 1) Accessibility on mobile devices, 2) Enabling multi user collaboration for shared experimentation, 3) Ability to connect to input data sources and incorporate live data. 4) 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 (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 which 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 is 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 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 digital twin of logistics systems with simulation logic in a single (modularized) model. (Sakr et al. 2021) present an approach for a digital twin based DES model. The proposed framework is applied to a semiconductor manufacturing system as an 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 our knowledge, we couldn't find research addressing scalability and security. For our application, we needed an architecture which could be scaled to worldwide operations in a secure manner. We were motivated to develop a solution which was easy to use for operations without a steep learning curve. As there was no such solution befitting our needs, we developed our own AWS architecture to address the need. In this paper, we present a AWS architecture 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 3 modules: Data Import Module, Simulation Integration Module (SIM), and Data Export Module. The simulation scientist develops and uploads the model onto simulation integration module. SIM hosts the simulation models, the software license and the input/output data files. The user connects to SIM through a web-based user interface (UI) which allows them to upload input files and start/stop simulation runs. The user can setup various simulation experiments by defining input variables through the input data file. The Data Import module intakes the input data file from the user and feeds it into SIM. The simulation model hosted in SIM can access the input file for running simulation experiments. The output data file is created in SIM and exposed through the Data export module to the user. The following section dives into the architecture of each module.

3.2 Simulation integration Module

Figure 2 shows the simulation integration module architecture. The entire architecture is setup in a virtual private cloud (VPC). The simulation integration module consists of an EC2 instance (cloud server) 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 OpenID Connect

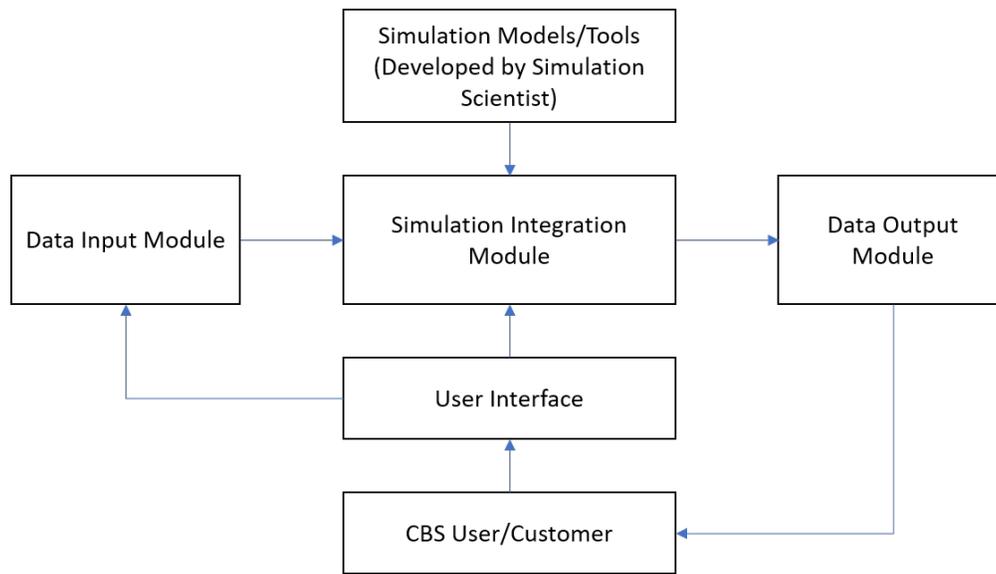


Figure 1: CSP Overview

(OIDC) compliant. An additional layer of security is added by creating authorized user groups for allowing access to the domain which 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 as per the customer needs. The bastion server allows simulation developers and server admin to access the simulation server through remote desktop protocol (RDP). The simulation UI allows the users to import data file into the model, and run the model. The simulation model is setup to export the desired key performance indicators (KPIs) to a spreadsheet which is stored in the cloud server.

3.3 Data Import Module

The data Import module enables the user to upload the simulation input file to SIM. The file is unique to each simulation model hosted in 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 to 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 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 OpenID Connect (OIDC) compliant. 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 (cloud server) which hosts the simulation models and all input files are uploaded to the cloud server.

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 by the user. The output file is uploaded to a desired location specified by the customer. Figure 4 shows the Data Export Architecture. The simulation model creates an output file on the cloud server in 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.

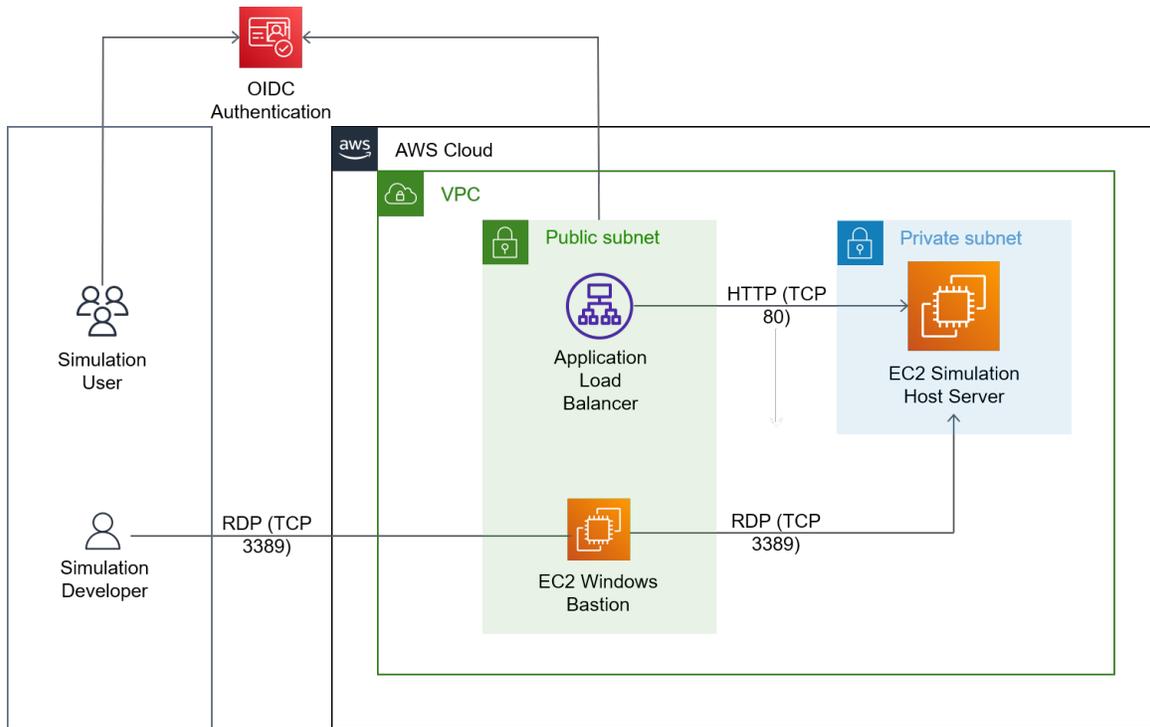


Figure 2: Simulation Integration Module Architecture

4 APPLICATION OF CSP FOR DES BASED TOOL DEPLOYMENT

As an application, we developed multiple design/operational DES-based tools. 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

Inbound (IB) staffing tool (IST) is a tool developed for distribution center/warehouse inbound (IB) operations planning. It is used by operations managers at the start of each shift for optimal staffing 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 tools provides a forward looking dynamic staffing 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 which it make it difficult to create optimal staffing 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 Material Handling Equipment (MHE) and limiting storage floor capacity for processing 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 its impact on downstream processes. To address this gap,

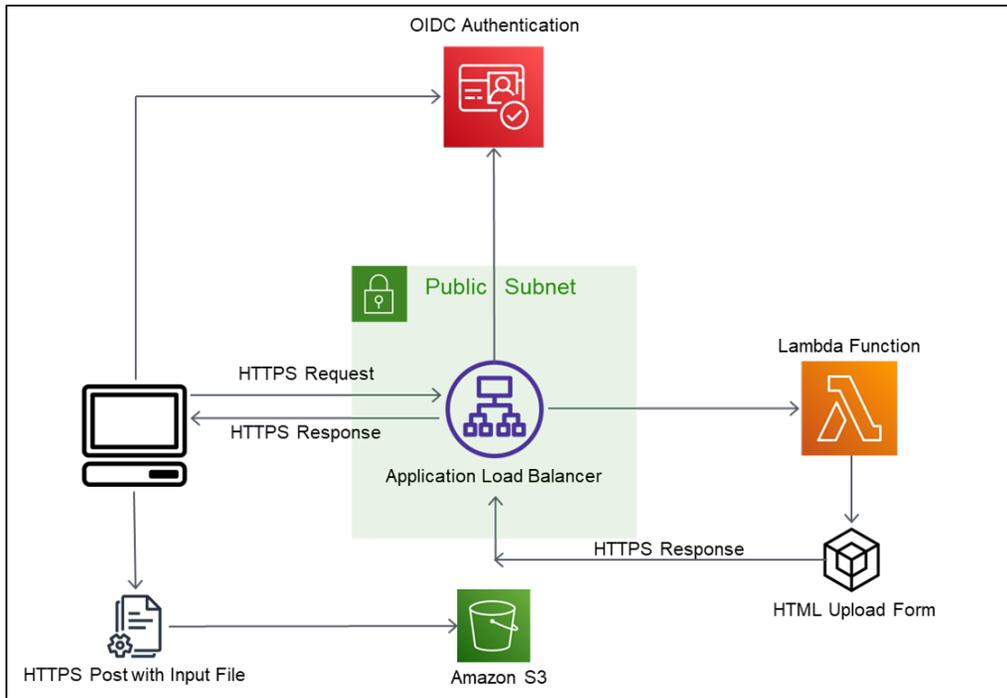


Figure 3: Data Import Module Architecture



Figure 4: Data Export Module Architecture

we developed a simulation-based staffing tool which generates an optimal staffing plan based on the end to end system.

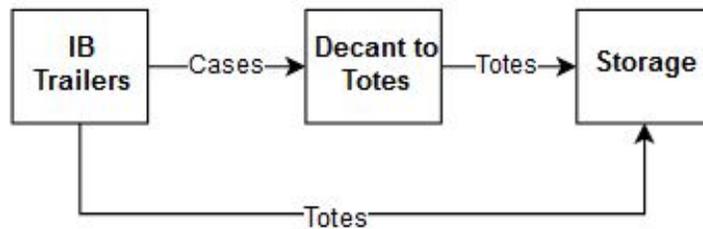


Figure 5: Inbound Process at Distribution Center

4.3 Simulation Model Development and Inputs

Prior to building the simulation, we worked with multiple stakeholders from operations, process engineering, maintenance and vendors to validate simulation assumptions. The model is built in FlexSim simulation software (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 vs transportation). The model simulates the warehouse management system (WMS) logic 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	Operations Plan
Planned Shift Headcount	Operations Plan
Decant Rate	Historical Observations
Stow Rate	Historical Observations

4.4 Trailer Selection Optimization

We developed an optimization function for level loading of trailers based on inbound freight mix. The optimization creates a subset of available trailers in the yard based on unload priority and volume requirement for specified time window. This subset of trailers is then sequenced and assigned an unload order. The optimization function tries to reduce consecutive trailers with high case percent mix and high unit size. The unit sizes are classified into 3 buckets: small, medium and large. By factoring in case percent mix and unit sizes for sequencing trailers, we minimize downstream process starvation and increase overall throughput. 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 Percent across selected trailers
Cs_avg	Average Case Percent across selected trailers
nTrailers	Number of Trailers
w	Weight

Table 3: Optimization Indices

Index	Description
I	Number of Trailers ($i = 1,2,3 \dots nTrailers$)
J	Number of Positions ($j = 1,2,3 \dots nPositions$)

The optimization objective function 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.

$$Sm_var(i) = abs(Sm(i) * \sum_j^J x_{ij} - Sm_avg) * w \text{ where } i = 1 \quad (1)$$

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

$$Sm_var(i) = abs(Sm(i) * \sum_j^J x_{ij} - Sm_avg) * w + abs(Sm(i-1) * \sum_j^J x_{ij} - Sm_avg) * w \quad (2)$$

where $i \geq 2$, for each $i \in I$

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

$$Case_var(i) = abs(Cs(i) * \sum_j^J x_{ij} - Cs_avg) * w, \text{ where } i = 1 \quad (3)$$

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

$$Cs_var(i) = abs(Cs(i) * \sum_j^J x_{ij} - Cs_avg) * w + abs(Cs(i-1) * \sum_j^J x_{ij} - Cs_avg) * w, \quad (4)$$

where $i \geq 2$, for each $i \in I$

Equation 5 and Equation 7 ensure that there is only one trailer assigned for each position in the sequence and only one position assigned per trailer.

$$\sum_j^J x_{ij} = 1, \text{ for each } i \in I \quad (5)$$

$$\sum_i^I x_{ij} = 1, \text{ for each } j \in J \quad (6)$$

Objective Function:

$$\text{Minimize: } \sum_i^I (Sm_var(i) + Case_var(i)) \quad (7)$$

4.5 Deployment of Tool on Cloud Simulation Platform

This section describes the implementation of the staffing tool on the cloud simulation platform (CSP). Figure 6 presents an overview of the implementation process. The operations manager (OM) for the shift is trained to create the inputs file and execute the simulation run. The trailer sequence optimization is performed outside of the simulation model and serves as one of the inputs to the simulation. The input file and the trailer sequence is uploaded to the cloud through data import module UI. 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/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 simulation integration module (SIM). After uploading the input file, the simulation runs are executed through a UI generated by SIM. The UI allows the user to run multiple simulation runs simultaneously. After the 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 impact on shift performance.

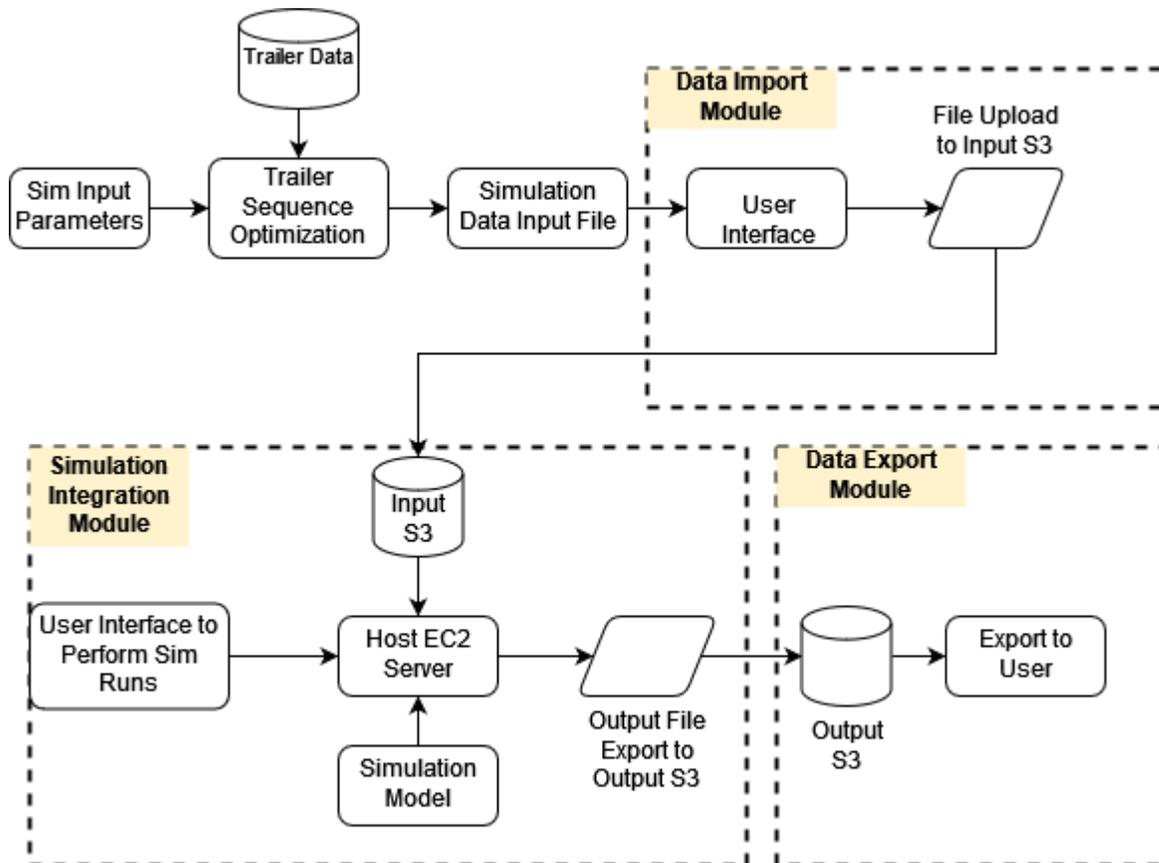


Figure 6: Implementation of Tool on the Cloud Platform

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), we conducted a 15-day pilot study at an existing distribution center. The goal was to compare the staffing plan generated by IST against the traditional spreadsheet-based staffing approach used by the operations team. The simulation model developed for 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, material handling equipment (MHE) constraints, and warehouse management system (WMS) control logic. By leveraging the trailer sequence as an input, the simulation could generate a detailed hourly staffing plan that accounted for the variability in inbound freight mix. During the pilot, we closely collaborated with the operations team to gather their feedback and address any concerns. One of the key challenges was earning the trust of the operations 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, we actively involved the operations team in the pilot and provided them with a detailed walk-through of the tool’s functionality. We also developed comprehensive user guides and training videos to ensure a smooth on-boarding process for teams with varying technical expertise.

Figure 7 illustrates the percent 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, we will continue to refine the tool based on feedback from the operations team and explore opportunities to further enhance its capabilities. Additionally, we 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.

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 operations 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, we presented the successful implementation of a simulation-based staffing tool for inbound operations planning at distribution centers, 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 our future scope 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

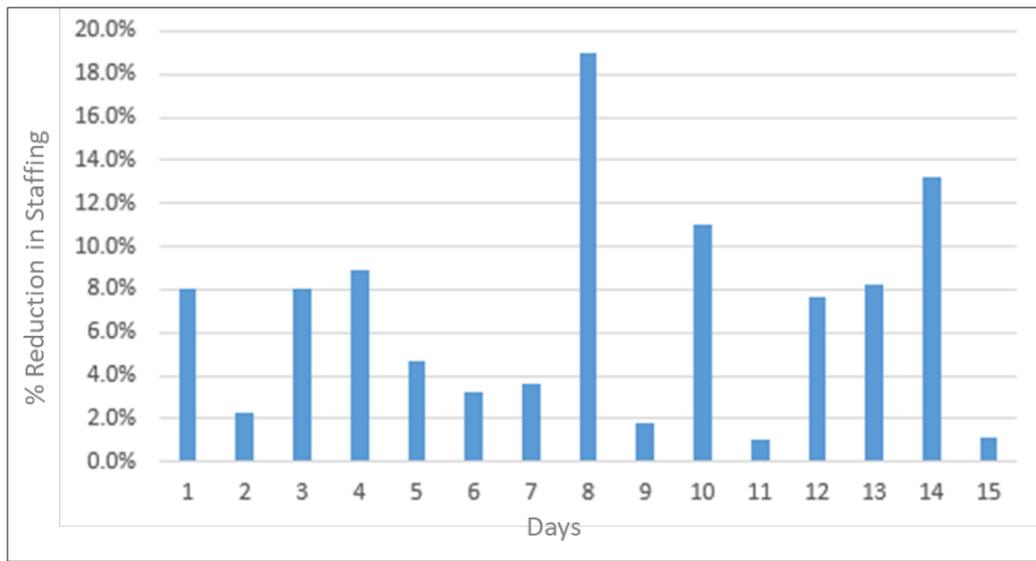


Figure 7: Percent Reduction in Staffing through IST Implementation

and the actual operational systems, the CSP will unlock new possibilities for advanced decision support, what-if scenario analysis, and continuous process improvement.

Furthermore, our ongoing efforts include scaling the CSP to accommodate different simulation software. The CSP architecture we developed is using Flexsim, but it can be expanded or scaled to various other software including physics-based simulation software. Physics simulations are 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 operations 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(2351-9789):1636–1641.
- Anylogic 2024. "AnyLogic simulation software". [Online; accessed 27-June-2024].
- 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 vol.1 <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, Vasterås, Sweden*, 01–08.
- 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)*, 1–15 <https://doi.org/10.1109/WSC.2012.6465310>.

AUTHOR BIOGRAPHIES

ROHAN VAIDYA is a Senior Simulation Scientist with Amazon. He received his Master’s in Mechanical Engineering from 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 Worldwide design and engineering team. He obtained his Master’s in Industrial Engineering from Arizona State University in 2014. He has over 10 years of industry experience with 2 years in Fiat Chrysler Automobiles and 8 years at Amazon Simulation Science. His research interest 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 his Master’s in Mechanical Engineering, a Master’s in Business Administration, MIT Machine Learning and PMP certification. He has 20+ years of 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