ISSN: 2366-1313

CLOUD COMPOSITION SERVICE USING RED FOX ALGORITHM

[1]Dr.M.SURESHBABU, [2]C.ABHINAV REDDY, [3]G.KAVYA, [4]B.PAVAN KUMAR.

[1](Professor) ,**CSE.** Teegala Krishna Reddy Engineering College Hyderabad

[234]B,tech scholar ,**CSE.** Teegala Krishna Reddy Engineering College Hyderabad

ABSTRACT

The abstract outlines the challenges faced in resource allocation within IoT enabled cloud environments, emphasizing the importance of low latency and high bandwidth. As high delay reduces the performance of the IoT enabled cloud platform, efficient utilization of task scheduling (TS) reduces the energy usage of the cloud infrastructure and increases the income of service provider via minimizing processing time of user job. It introduces a novel solution, the Oppositional Red Fox Optimization based Task Scheduling (ORFO-TSS) model, designed to enhance task scheduling efficiency and reduce energy consumption in cloud infrastructures. The model combines Oppositional Based Learning (OBL) with Red Fox Optimization (RFO) to optimize resource allocation for incoming tasks. The presented ORFO-TSS model resolves the problem of allocating resources from the IoT based cloud platform. It achieves the make span by performing optimum TS procedures with various aspects of incoming task. The designing of ORFO-TSS method includes the idea of oppositional based learning (OBL) as to traditional RFO approach in enhancing their efficiency. The experimental outcome highlighted the efficiency of the ORFO-TSS technique over existing approaches.

1. INTRODUCTION

Internet of Things (IoT) is the vital technique to form smart city because it enables objects or entities to deliver data and service to users by communicating and collaborating with others [1]. There has been a rapid progression that the multiple devices get interconnected to the system with the tremendous growth of the IoT. Once the device requests resource service from the cloud data centre simultaneously, it would take a massive network bandwidth, as well as information access and data transmission would be slow. Furthermore, This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. when some - sensitive requests namely emergency and medical are uploaded to the remote cloud to process, the delay created by bandwidth constraint and resource bottleneck of the cloud data centre affects the quality of service (QoS). In the meantime, cloud computing (CC), a novel computing structure, was extensively employed in the last few decades. CC is a technique which focuses on providing a flexible heterogeneous source pool via the system, and users rent distinct resources on demand [3,4]. User procures and releases computing resource that is generally virtual machine (VM) with distinct provisions, based on the particular requirements within a limited period. Since these techniques are highly dependent on the Internet, the CC and IoT are strongly associated with the role.

The IoT digitalizes various information and wisely manages equipment, and CC is utilized by a carrier for higher-speed information utilization, processing, and storage. CC provides the advantage of security, speed, and convenience that the lacks IoT, and the technique that makes intelligent analysis and the real time dynamic management of the IoT consistent. The symbiotic relationship between the Internet of Things (IoT) and cloud computing (CC) extends far beyond mere connectivity. As IoT continues to proliferate, CC serves as the backbone that supports the immense influx of data generated by IoT devices. 5 One significant advantage of CC in this context is its scalability. As the number of IoT devices within a smart city ecosystem continues to grow, CC provides the infrastructure needed to accommodate this expansion without compromising performance.

By dynamically allocating resources in response to fluctuating demand, CC ensures that essential services remain accessible even during peak usage periods, thereby mitigating the risk of network congestion and data transmission delays. Moreover, CC enhances the security and reliability of IoT deployments within smart cities. Through robust encryption protocols, intrusion detection systems, and multilayered authentication mechanisms, CC platforms safeguard sensitive data transmitted between IoT devices and cloud servers. This proactive approach to cybersecurity is

ISSN: 2366-1313

paramount, particularly in scenarios involving critical infrastructure, healthcare systems, and emergency response networks, where any compromise in data integrity could have severe consequences.

Furthermore, CC empowers smart cities to leverage advanced analytics and machine learning algorithms for real-time data analysis and decision-making. By harnessing the computational power of cloud-based resources, city administrators can derive actionable insights from the vast streams of IoT-generated data, enabling predictive maintenance, traffic optimization, energy management, and other proactive initiatives aimed at enhancing urban efficiency and sustainability. In essence, the integration of IoT and CC represents a transformative paradigm shift in the way cities are planned, managed, and experienced. By capitalizing on the synergies between these two technologies, smart cities can harness the full potential of interconnected ecosystems to create safer, more efficient, and more livable urban environments for residents and visitors alike.

2. LITERATURE SURVEY

The rapid advancement in Internet of Things (IoT) technology and cloud computing (CC) Infrastructure has significantly transformed the landscape of modern computing systems. This convergence has sparked considerable interest in leveraging the synergies between IoT and CC to meet the evolving demands of resource-constrained IoT applications. With IoT devices generating vast amounts of data that require efficient processing and analysis, cloud environments offer a promising solution by providing a flexible and scalable resource pool accessible over the network. One critical aspect in ensuring the efficiency of IoT-enabled cloud platforms is task scheduling (TS), which plays a crucial role in resource allocation and utilization. Efficient TS algorithms not only reduce latency and improve response times but also minimize energy consumption and enhance the overall revenue generation for service providers. Consequently, there is a growing emphasis on developing novel TS schemes tailored specifically for IoT-enabled cloud environments.

The proposed oppositional red fox optimization based task scheduling scheme (ORFO-TSS) presented in the article aims to tackle the resource allocation problem in IoT-based cloud platforms. By integrating the principles of oppositional-based learning (OBL) with the traditional red fox optimization (RFO) approach, the ORFO-TSS model enhances its efficiency and

ISSN: 2366-1313

effectiveness in optimizing task scheduling procedures. Through comprehensive experimental analysis conducted on the CloudSim platform, the efficacy of the ORFO-TSS technique is demonstrated, highlighting its superiority over existing approaches. Overall this underscores the importance of bridging the gap between IOT And computing through innovative solutions like ORFO-TSS Model. These advancements pave the way for more efficient and scalable computing paradigms tailored to the needs of modern digital ecosystem.

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

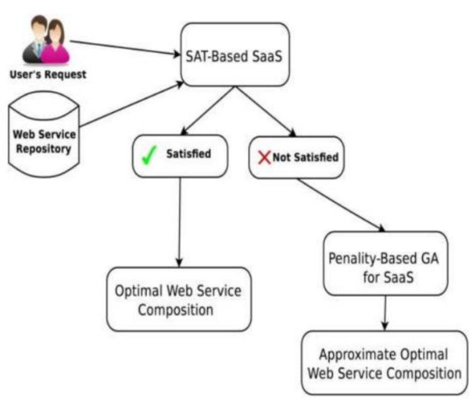


Fig 3.1 System Architecture

- **SAT-Based SaaS for Optimal Web Service Composition**

If the SAT-based SaaS approach satisfies the requirements, it will focus on optimizing web service composition. This approach uses a SAT (Satisfiability Testing) solver to find an optimal solution for web service composition based on given constraints and criteria. It ensures that the composition meets the desired objectives and requirements.

- **Penalty-Based GA for SaaS and Approximate Web Service Composition**

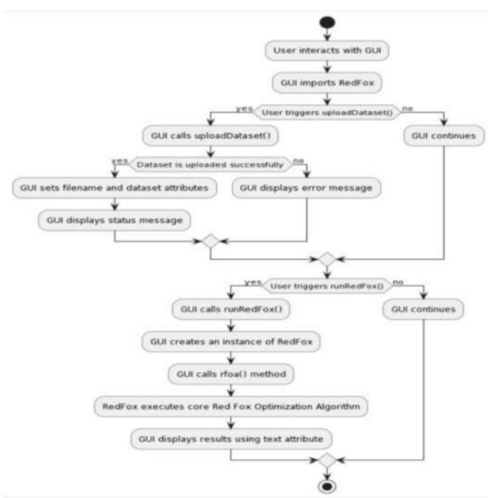
If the SAT-based SaaS approach does not meet the satisfaction criteria, a penalty based Genetic Algorithm (GA) is considered. 12 In this approach, a GA is used to search for an approximate solution for web service composition. While it may not guarantee an optimal solution, it attempts to find the best composition within given constraints and objectives. Penalties are applied to guide the search towards a suitable outcome.

ACTIVITY DIAGRAM

Activity diagrams are the graphical depiction of workflows of the stepwise activities and actions with iteration. In the UML, activity diagrams are also used to describe the operational step-by-step workflows of the components in a system. Activity Diagrams in UML serve to visually

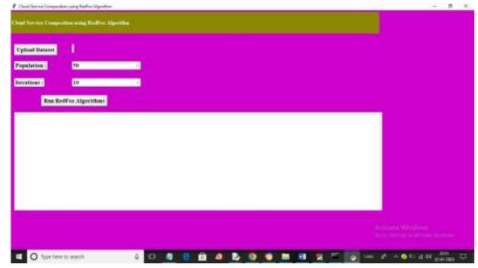
ISSN: 2366-1313

represent dynamic workflows showcasing the sequence and conditions of activities within a system or business process. The key components include nodes, representing actions or decisions, and transitions, illustrating the flow between these nodes. Initial and final nodes mark the activity's start and end. Control flows connect actions, specifying the order of execution, while decision nodes enable branching based on conditions. Forks and joins manage parallel flows, and swim lanes partition activities among different entities for clarity.

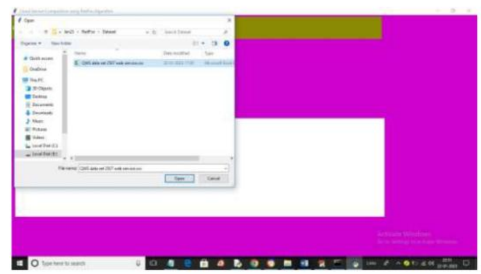


4. OUTPUT SCREENS

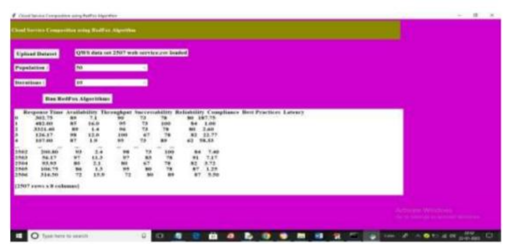
To run project double click on 'run.bat' file to get below screen



In above screen click on 'Upload Dataset' button to upload dataset and get below output.

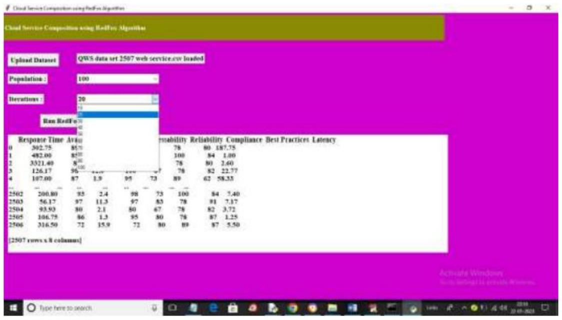


In above screen selecting and uploading dataset file and then click on 'Open' button to load dataset and get below output.

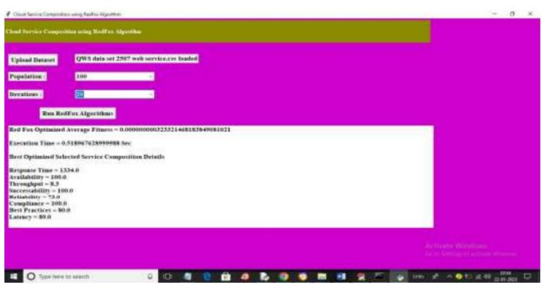


In above screen dataset loaded and now from drop down boxes you can select Population and Iterations and then click on 'Run RedFox Algorithm' button to get below service values

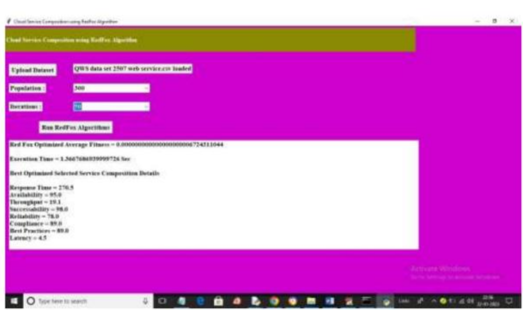
ISSN: 2366-1313



In above screen I selected number of population as 100 and then selecting iterations as 20 and then press button to get below output



In above screen in text area we can see fitness value and execution time as output and then can see optimized service values such as Response Time, throughput etc. Similarly you can select any number of population and iterations and get best service values as the output



In above screen for selected population and iteration we got above output.

5. CONCLUSION

In this ,a novel ORFO-TSS algorithm has been developed to resolve the problem of allocating resources from the IoT based cloud platform. It fulfils the make span by performing optimum TS procedures with various aspects of incoming tasks. The designing of ORFO-TSS method includes the idea of OBL as to typical RFO algorithm in enhancing its efficiency. The experimental outcome highlighted the efficiency of the ORFO-TSS technique over existing approaches. Thus, the ORFOTSS technique can be exploited for optimizing the efficiency of the IoT enabled cloud environment. In future, hybrid deep learning models can be employed to schedule the sources that exist in the IoT enabled cloud environment.

6. FUTURE ENHANCEMENTS

The ORFO-TSS algorithm represents a significant advancement in addressing resource allocation challenges within IoT-enabled cloud environments. As technology

ISSN: 2366-1313

continues to evolve and demands on cloud infrastructure grow, there are several avenues for extending and enhancing the proposed solution: Integration of Machine Learning Techniques: Incorporating machine learning algorithms, such as reinforcement learning or deep learning, can enhance the adaptability and intelligence of the ORFO-TSS model. By continuously learning from past scheduling decisions and adapting to changing workload patterns, the algorithm can further optimize resource allocation and improve efficiency. Dynamic Resource Provisioning: Future enhancements could focus on enabling dynamic resource provisioning based on real-time demand and workload fluctuations. By dynamically adjusting resource allocation in response to changing environmental conditions and user requirements, the ORFO-TSS algorithm can better accommodate varying levels of demand while minimizing energy consumption and maintaining low latency.

Multi-Objective Optimization: Extending the ORFO-TSS model to support multi-objective optimization can enable the simultaneous optimization of multiple performance metrics, such as make span, energy consumption, and resource utilization. By considering trade-offs between competing objectives, the algorithm can provide more flexible and customizable solutions that better align with the diverse needs of IoT applications. Edge Computing Integration: With the proliferation of edge computing devices in IoT environments, future enhancements could explore the integration of edge resources into the task scheduling process. By leveraging edge computing capabilities for offloading and processing tasks closer to the data source, the ORFOTSS algorithm can reduce latency and bandwidth requirements while improving overall system performance.

7.REFERENCES

[1] E. H. Houssein, A. G. Gad, Y. M. Wazery and P. N. Suganthan, “Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends,” Swarm and Evolutionary Computation, vol. 62, no. 3, pp. 100841, 2021.

[2] S. E. Shukri, R. A. Sayyed, A. Hudaib and S. Mirjalili, “Enhanced multi-verse optimizer for task scheduling in cloud computing environments,” Expert Systems with Applications, vol. 168, no. 4, pp. 114230, 2021.

ISSN: 2366-1313

[3] S. Velliangiri, P. Karthikeyan, V. M. A. Xavier and D. Baswaraj, "Hybrid electro search with genetic algorithm for task scheduling in cloud computing," Ain Shams Engineering Journal, vol. 12, no. 1, pp. 631–639, 2021.

[4] M. A. Elaziz and I. Attiya, "An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing," Artificial Intelligence Review, vol. 54, no. 5, pp. 3599–3637, 2021.

[5] K. Karthikeyan, R. Sunder, K. Shankar, S. K. Lakshmanaprabu, V. Vijayakumar et al., "Energy consumption analysis of virtual machine migration in cloud using hybrid swarm optimization (ABC-BA)," Journal of Supercomputing, vol. 76, no. 5, pp. 3374–3390, 2020.

[6] P. Bal, S. Mohapatra, T. Das, K. Srinivasan and Y. Hu, "A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques," Sensors, vol. 22, no. 3, pp. 1242, 2022.

[7] W. Jing, C. Zhao, Q. Miao, H. Song and G. Chen, "QoS-DPSO: QoSaware task scheduling for cloud computing system," Journal of Network and Systems Management, vol. 29, no. 1, pp. 5, 2021. 36

[8] M. S. Sanaj and P. M. J. Prathap, "An efficient approach to the mapreduce framework and genetic algorithm based whale optimization algorithm for task scheduling in cloud computing environment," Materials Today: Proceedings, vol. 37, pp. 3199–3208, 2021.

[9] H. B. Alla, S. B. Alla, A. Ezzati and A. Touhafi, "A novel multiclass priority algorithm for task scheduling in cloud computing," Journal of Supercomputing, vol. 77, no. 10, pp. 11514–11555, 2021.

[10] R. Masadeh, N. Alsharman, A. Sharieh, B. A. Mahafzah and A. Abdulrahman, "Task scheduling on cloud computing based on sea lion optimization algorithm," International Journal of Web Information Systems, vol. 17, no. 2, pp. 99–116, 2021.