iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** II **Month of publication:** February 2021

DOI: https://doi.org/10.22214/ijraset.2021.32965

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

Machine Learning based Hybrid Recurrent Data-Driven Flow Algorithm of Identify the Risk of Data Flow Error Detection in Cloud Computing

Mr.Senthil Kumar Paramsivapandi[1], Dr. S. Nagarajan[2]
[1]HCL Australia services pty Ltd, North Sydney, NSW.
[2]Department of Computer Applications Yadava College. Madurai Kamaraj University, Madurai, Tamilnadu, India.

***Abstract:** Cloud computing is transforming into a promising technology, as it integrates web mining analytics as well, especially when there is a high focus on ensuring the confidentiality and integrity of end-user data. These benefits, however, are restricted for the issue of confidentiality and data. When it comes to data transfer to the cloud services, cloud consumers lose control of their data and do not know how it will be processed and assigned to a service. Machine learning and business process integration are becoming more and more popular as companies cross organizational boundaries. Web mining is used to achieve business intelligence by finding data flow errors in cloud computing traffic. One of the important aspects of a workflow analysis of the data stream is the angle of view. Even when a proper processing sequence syntax is given, the workflow specification error still occurs due to incorrect data flow error detection based on workflow specification error execution. To overcome this setback, Hybrid Recurrent Data-Driven flow Algorithm (HRD-DFA) is proposed to monitor the cloud's data flow error. Whenever the user accesses the uploaded data, the HRD-DFA system checks any data flow errors in the reading and writing process and clarifies that error to evaluate the file size modification or file name change and data format varied. The HRD-DFA is evaluated based on the amount of stored data, and the data read and write time. The results show that the two amounts of sent data are optimized, and the data stream is cleared with high error reduction.*
***Keywords:** Cloud Computing, Data Flow Errors, Hybrid Recurrent Data-Driven Flow Algorithm (HRD-DFA), Machine Learning, Reading and Writing Process, Web Mining.*

I. INTRODUCTION

A data-centric security mechanism track or stream of execution information can improve cloud security in many ways. First, users are given the ability to coordinate with cloud platform cloud service providers and control data transmission. This encourages compliance with rules and regulations. Secondly, multi-tenancy is a practice of sharing services between cloud tenants, as cloud platforms can enforce inspections despite defects in the service itself to implement security policies. Third, tracking data from different services increases accountability and provides rigorous manipulation of cloud provider login tenant data. It is a difficult problem to detect effectively and clean up cloud data errors. New error detection methods are to be introduced to detect data errors from the network using large-scale storage, scalability, and cloud computing power.

Specifically, addresses run on multiple heterogeneous clouds use mathematical models and mixed-integer programming to optimize large-scale scientific data errors. This method allows us to describe the mathematical model and use the available optimization release set. On the other hand, this method can also use large workflows with various cloud resources for general planning problems. It becomes unrealistic. Due to the complexity of the problem, it is necessary to analyze the simplified model. In this study, the same strategy is applied to solve problems where the same or varied quantity changes tasks in smaller ranges. As observed, large scientific data often have the structure of a set of tasks, each with a stream consisting of multiple parallel stages or levels, i.e., tasks at each level. It has nothing to do with similarities to each other.

II. RELATED WORK

A DLBS (Dynamic Load-Balanced Scheduling) can formalize this problem. Effective Hash Scheduling Algorithm (EHSA) consists of data flow. Two open space-flow network models and subsequent time slots are being created. [1].It is proved that the method of calculating multiple frame data in a variable network bandwidth environment is more effective than the method of calculating the limit of a single frame data [2]. Data streaming for this application cannot be performed because it can cause inefficiencies, but it also has the requirement to limit access to data in selected geographic locations [3].

They have a parallel flow of a block for various applications through their distributed nature including the relevant tasks. Improved Flow Level Matrix Traditional techniques have led to poor application-level performance [4] and are realistic for work level needs. Equal-Cost Multi-Path (ECMP) routing does not consider traffic module or network status and it considers only current routing methods such as relying on cash flow. The guide in close proliferation is to be investigated. Multi commodity binary redirect the flow-optimized flow problem and present a solution for optimizing routing rules. A revised version of the problem is taken care of by using the binary multi commodity optimizing flow and hash collisions approach [5] is also reduced.

An optimized formulation is to be used to estimate the Markov decision process under cloud providers' flow service time for flow end time uncertainty and revenue in long-term operation. Since optimization prescriptions are mathematically stubborn, heuristic scheduling algorithms have been developed for these two scenarios: known and unknown streaming service times [6]. Considering changes into account, virtual machine performance and delays in instant acquisition at low cost are ensured [7] .The use of heuristic algorithms has just one advantage in calculating time schedules provided by clouds Deadline constraint scientific workflows..

A Profit Maximization Algorithm (PMA) is introduced to detect the global variability of mixed cloud prices. Temporary task scheduling provided by the PMA can schedule all arrivals tasks to run in dynamic private and public clouds. The sub problem has-induced stabilization of each Simulated Annealing Particle Swarm Optimization (SAPSO) [8] and the proposed hybrid formula optimization algorithm.

In most cases, large-scale scientific computing is expressed as a planning method and workflow run-time system. With this type of table, especially elastic pricing models (fixed and dynamic), assorted resource types, multiple service items, and imaginary cloud systems, cloud computing's dynamic properties have become a more complex issue. This sets workflow task mapping on structures, [9], NP-Complete is a perfect example of a common planning problem.

Compared to traditional high-performance computing systems such as power grids, the cloud can provide the software environment and customizable infrastructure required for key execution resources and Virtual Machines (VMs). Provides a scientific workflow for various challenges and especially the quality of service forecasts and optimal scheduling, remain unsolved [10].

Cloud computing has become one of the most widely used platforms. The computing, storage and analysis services are provided to end-users and companies are recognized based on rewards for high agility, availability, scalability and flexibility use. This allows individuals and companies to obtain a High Performance Computing (HPC) platform with high activation source and large swimming pools without building. This comparison reduces the individual VM's size [11] and has been extended in the estimation method of unbalanced loads.

The cloud is very fast, receives these tasks, and resource allocation needed to handle customer tasks intelligently is done. Through the hybrid bio processing algorithms proposed in this work, virtual machines within assigned tasks improve the particle swarm optimization algorithm and efficiently use resources [12].

Various workload displays, such as the individual H2O ((Hierarchical and Hybrid Online)) cloud, are highly scalable in the cloud platform configuration. They consider user demand information and detailed information on the dynamic pricing model. The hierarchy and hybridity of the framework, combined with its Deep Reinforcement Learning (DRL) engines enable H2O -Cloud to efficiently start on-the-go scheduling and learning in an unpredictable environment without pre-training and maintain Basic QoS [13] In terms of energy and cost, the experiments are proposed when comparing approaches to confirm the high efficiency of the H2O cloud.

Offline scheduling algorithm violates Oracle's new isolation of dual controls and concerts, including a random rounding technique that works to provide an almost optimal solution. The online scheduling algorithm uses a learning-oriented program and obtains a dual solution [14] to the online prototype dual structure. Scheduling work between nearby mobile devices has been identified as an important issue that needs to be resolved when node heterogeneity, the capabilities and vitality of unknown workers are identified. Wherever the edges can be broken, uneven adaptive participatory system is to be introduced [15] and work distribution, according to the independent work on various mobile nodes, is to be adopted to balance the need to use a well-known work-stealing mode.

DC (Data Center) scheduling includes the ability to ensure Quality of Service (QoS) to maintain such data migration costs, data transmission costs, and execution of user satisfaction application service delays for some networks. It is a constraint and a target. The task of quantifying the effective QoS of the SaaS type and determining the best decision rule select a specific DC (Data Center) [16].

It serves multiple purposes simultaneously. In a dynamic environment like the cloud, computing is done to perform scientific workflows. It is a multipurpose scheduling for these applications cost and two main goals. Another important factor in large-scale scientific workflows is the transfer of a large amount of data at once. Therefore, in this study [17], Cloud data centers and network equipment significantly impact energy consumption including network usage data movement.

However, most data-intensive types of well-developed development do not enable container shutdown services when shutting down services in the big data era. For this reason, this article is aimed to enhance today's cloud services by enhancing a new generation of container shipping that takes utility performance into account. In particular, they have developed and analyzed a new product that analyzes loads performance and application performance at work. To avoid this kind of integrated optimization problems and application performance, existing studies have used statistical methods opinion and a balancing load mess to address it effectively. [18].

Cloud data center is a new cloud computing model used in predicting multiple data and processing a plurality of data center configurations. Data-centric computing depends largely on the cloud computing capability and the workload schedule of different cloud data centers is handed over to customer tasks. For the development of rules in cloud computing, high-performance workload scheduling technology rules are being explored in detail as they are the key challenges. It intends to reduce the completion time [19] of most past assignments to all assignments.

User Request (UR) and the service planning process significantly impact a cloud data center's performance. This is especially true because performance probability of essential Quality-of-Service (QoS) is the measure of the probability of UR, blocking a well-connected data center such as response time. This paper revolves around the proposal of a novel Deadline-Aware Scheduling Scheme (DASS) that aims to improve the QoS performance of the Matrix [20] time information center, mentioned above.

III. IMPLEMENTATION OF THE PROPOSED SYSTEM

Data streams are used to represent a variety of applications with high processing and storage requirements. As a solution to meet this need, the cloud computing model has become an on-demand resource provider. Data flow error detection is finding the correct allocation of task processors to optimize performance metrics such as public auditing level, link capability and minimum run time. Data errors lead to corruption of data values in the program, resulting in incorrect intermediate values and the final output file. Fault tolerance should be included to improve the reliability and integrity of the proposed system.

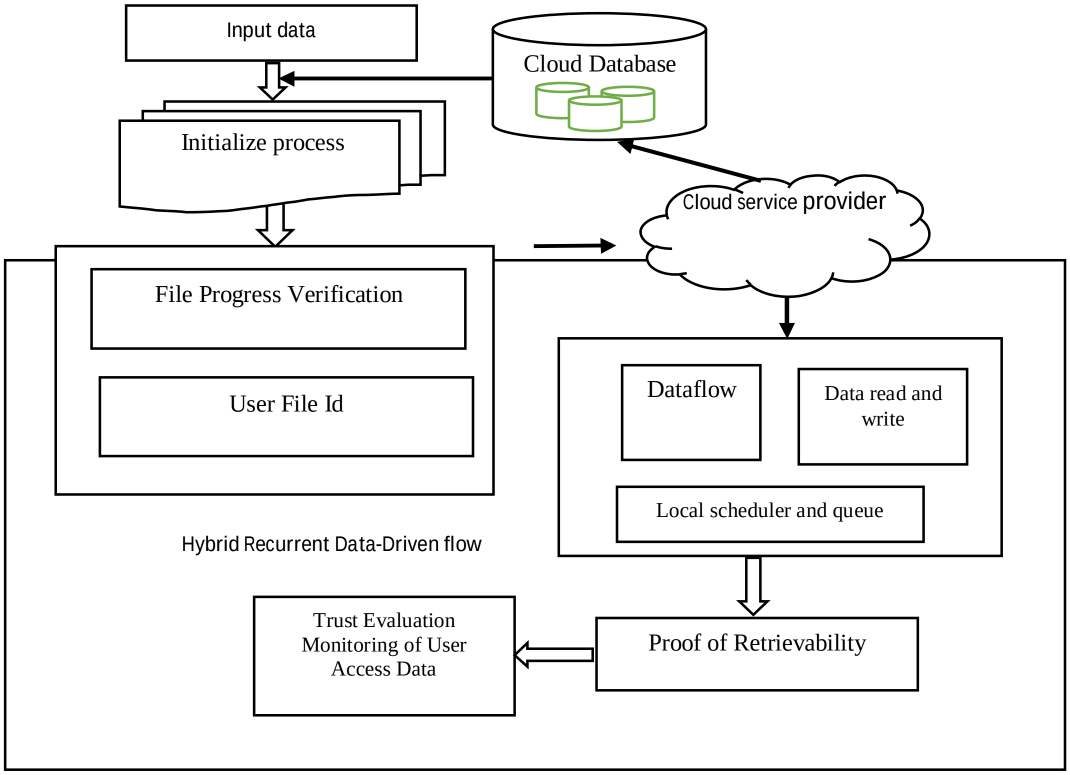


Figure 1 shows the proposed HRD-DFA to achieve data flow error detection works as given below. First, file progress verification is done to access the data and identify if any changes in accessing the data like file size change, file name modification. Then the cloud data flow management system model checks any data flow error when the user reads and writes on the received data and finally, trust evaluation of the user access data is performed to ascertain the time to read and write the data.

A. File Progress Verification

When a service provider uses a cloud provider's infrastructure for document validation services, it assesses cloud management risks in case of the file proposed by the service provider is sent to the cloud or moved to another version. In that case, the service provider and user's service-oriented features can deviate from the product and confuse. Non-performing characteristics such as performance, reliability, energy consumption and cloud economy have been verified based on the quality and service level contracts of managed services to temporary services.

File validation based on user ID is one way to verify the integrity of remote computing. Constraints on the operating time values of variables in the verification (personal cloud system) program created on the system are proposed here. At run time, it establishes a source for the calculation of file operation time values based on proverbial (public cloud system) variables. The proposed Hybrid Recurrent Data-Driven flow Algorithm used everything that should be checked by the service provider whenever the user uses the file.

B. Cloud Dataflow Management System Model

The steps involved in the execution of the dataflow are shown below.

Dataflow mapper: The executable is generated by an abstraction of the data stream provided by the client. The data stream is then reorganized for optimal performance.

Data read and write operation: Users cooperate reading which depends on the data given by their workflow, but if they wish, the person who uploaded the data can write the data. In this type of trust planning, the task is performed after the appropriate user completes and confirms it.

Local scheduler and queue: A proposed system for monitoring the processing of cluster tasks is introduced using local resources. A task is a local queue and the time it takes for a scheduled task to be submitted at a time called the queue delay. This delay depends on the access of local resources and is reflected in the performance of local scheduling.

1) *Input:* Number of tasks $t_1, t_2, \dots, t_n$ // task depends upon the users
2) *Output:* Analysis of Data flow error

Start

a) *Step1:* t_n is the total available tasks in the ready queue.
    For i = 1 to t_n
b) *Step2:* Assign priority to each task according to its t_{min} // assigning priority to the minimum task
    End for
c) *Step3:* For i = 1 to n
    if (file owner of f = = NULL)
        Find the file owner of f
    else (f == $read\ the\ file$)
        user read the data file
        do it for all the Unscheduled tasks
    End for
    End if
d) *Step4:* For each unscheduled tasks t_u
    Find the resources
    End for

C. Proof of Retrievability

This program uses some sentinel characters to check the read and write data integrity and data stored on cloud servers. The user's data check these sentinel characters to see if they are hidden in the original dataset. The character position is that the local data owner maintains the size of the file.

To verify the integrity of the file, send an inquiry message instructing the owner of the data. A text response corresponds to the server request. The server uses matching characters to determine the integrity of the data group.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue II Feb 2021- Available at www.ijraset.com

D. Trust Evaluation Monitoring Of User Access Data

In large distributed systems, data access to data failures caused by many individual events can flow into web mining applications. Therefore, it proposes a reliable service-oriented scheduling algorithm for the data flow's reliability to perform the service it refers. The general trust that combines direct trustand recommended trust is defined as follows:

$$Trust_i = W_i * D(U_i) + (1- W_i) * R(U_i) \quad (1)$$

Where $D(U_i)$ a direct trust of the i^{th} service by the experiences, which is based on the history of using the service by the users. $R(U_i)$Denoted recommendation trust of the ith service by other users. W_iDenote the weight of direct trust and recommendation trust for the i^{th}service, which can be computed as follows:

$$W_i = 1 - \frac{1}{k} \quad (2)$$

Where $k$denote the number of times that the i^{th} service. The credit rating is that transmission and data storage and data read/write times are the best, and data stream errors are very time-consuming.

IV. RESULT AND DISCUSSION

The resulting HR-DFA data is given as a standard processing cloud computing data flow error detection method. Visual Studio Framework is implemented with SQL Server Authorization powered by 4.0. The proposed safety performance programs demonstrate high efficiency as a result of the following.

Table I
Implementation Parameters Used In The Proposed Method

Processed Parameter	Value Processed
Service levels	5
Type of data	Data files
Number of users	300
Service provider	CSP

Table I shows the defined values and safety inspection parameters for the proposed operation. The proposed Hybrid Recurrent Data-Driven flow Algorithm (HRD-DFA) is compared with the previous Profit Maximization Algorithm (PMA) and Effective Hash Scheduling Algorithm (EHSA).

Table II
Public Auditing Analysis

Methods/users	public auditing efficiency in%		
Users	EHSA	PMA	HRD-DFA
100	76.54	78.65	85.43
200	79.76	82.87	87.65
300	81.67	84.76	89.76

Above table II shows the public auditing level analysis which depends on the number of users accessing the web mining file.

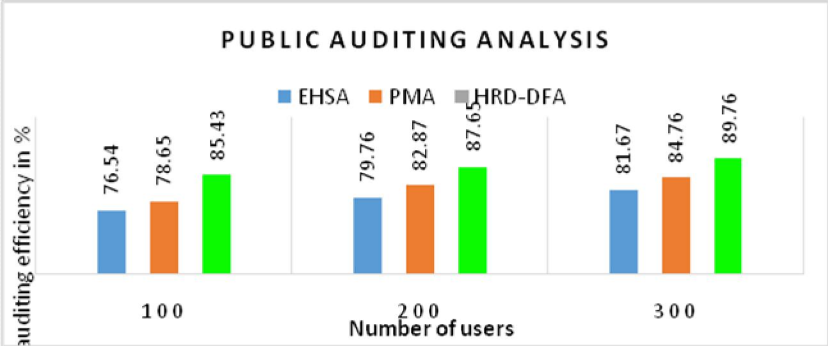


Fig. 2 Public Auditing Analysis

The above figure 2 defines the file verification of public auditing proficiency with different methods. It is understood that the proposed HRD-DFA system has produced 89.76 % public auditing analysis which is better than the previous EHSA 81.67%, PMA 84.76%.

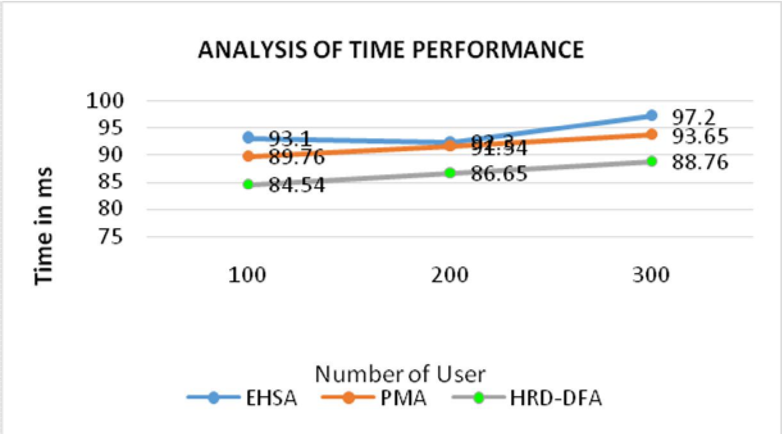


Fig. 3 Analysis of Time Performance

Above figure 3 shows the time performance analysis of proposed HRD-DFA which is 88.76 sec and the previous EHSA is 97.2 ms, PMA is 93.65 in ms.

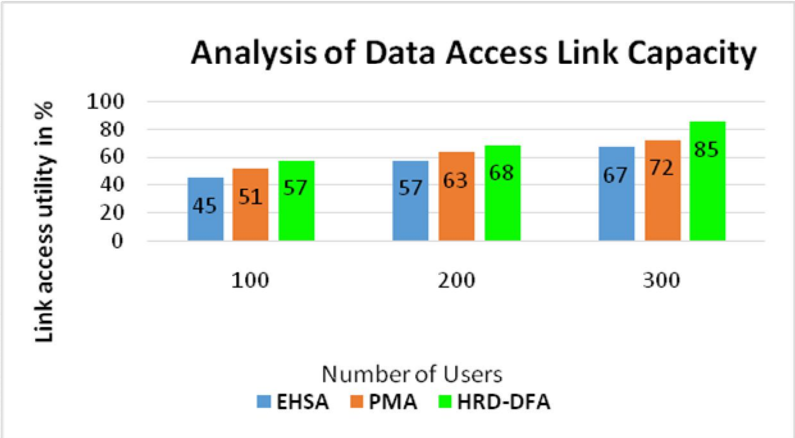


Fig. 4 Analysis of Data Access Link Capacity

The above figure 4 shows the analysis of data accesses link capability level. The proposed HRD-DFA offers data access link capability level up to 85%, and previous EHSA offers only 57%, PMA offers only 68%. Hence, the proposed system gives better link capability.

V. CONCLUSION

The HRD-DFA proposed by current cloud innovations provides unlimited access to the security of cloud server links. The data based on web mining flow and the data flow error detection are presented by the system proposed here. An efficient task scheduling mechanism has improved the cloud computing environment's overall performance and improved resource utilization of data flow error detection. However, scheduling web mining tasks often depends on the size of data and files, which are often very resource-intensive for static task requirements. The proposed HRD-DFA is used to identify the data flow error detection during data uploading or data retrieving. The proposed HRD-DFA ensures the public auditing analysis level of 89.76 %, the time complexity of 88.76 ms, and the data access link capacity is analyzed as 85%. In future work, users can revoke the public revocation list without updating other users' private keys, which enables users and new users to decrypt files directly from their participating cloud.

REFERENCES

[1] Tang, Feilong; Yang, Laurence T.; Tang, Cang; Li, Jie; Guo, Minyi (2016). A Dynamical and Load-Balanced Flow Scheduling Approach for Big Data Centers in Clouds. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2016.2543722.

[2] Li, Jianqiang; Huang, Luxiang; Zhou, Yaoming; He, Suiqiang; Ming, Zhong (2017). Computation partitioning for mobile cloud computing in big data environment. IEEE Transactions on Industrial Informatics, 1–1. doi:10.1109/tii.2017.2651880.

[3] Esposito, Christian; Castiglione, Aniello; Frattini, Flavio; Cinque, Marcello; Yang, Yanjiang; Choo, Kim-Kwang Raymond (2018). On Data Sovereignty in Cloud-based Computation Offloading for Smart Cities Applications. IEEE Internet of Things Journal, 1–1. doi:10.1109/JIOT.2018.2886410.

[4] Dong, Fang; Guo, Xiaolin; Zhou, Pengcheng; Shen, Dian (2019). Task-aware flow scheduling with heterogeneous utility characteristics for data center networks. Tsinghua Science and Technology, 24(4), 400–411. doi:10.26599/TST.2018.9010122.

[5] Sehery, Wile; Clancy, Charles (2017). Flow Optimization in Data Centers with Clos Networks in Support of Cloud Applications. IEEE Transactions on Network and Service Management, 14(4), 847–859. doi:10.1109/tnsm.2017.2761321.

[6] Truong-Huu, Tram; Gurusamy, Mohan; Girisankar, Sharmila Tranquebar (2017). Dynamic Flow Scheduling With Uncertain Flow Duration in Optical Data Centers. IEEE Access, 5 11200–11214. doi:10.1109/access.2017.2716345.

[7] Sahni, Jyoti; Vidyarthi, Deo (2015). A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2015.2451649.

[8] Yuan, Haitao; Bi, Jing; Tan, Wei; Li, Bo Hu (2016). Temporal Task Scheduling With Constrained Service Delay for Profit Maximization in Hybrid Clouds. IEEE Transactions on Automation Science and Engineering, 1–12. doi:10.1109/tase.2016.2526781.

[9] Arabnejad, Vahid; Bubendorfer, Kris; Ng, Bryan (2018). Budget and Deadline Aware e-Science Workflow Scheduling in Clouds. IEEE Transactions on Parallel and Distributed Systems, 1–1. doi:10.1109/TPDS.2018.2849396.

[10] Li, Weiling; Xia, Yunni; Zhou, Mengchu; Sun, Xiaoning; Zhu, Qingsheng (2018). Fluctuation-Aware and Predictive Workflow Scheduling in Cost-Effective Infrastructure-as-a-Service Clouds. IEEE Access, 6 61488–61502. doi:10.1109/ACCESS.2018.2869827.

[11] Ibrahim, Muhammad; Nabi, Said; Baz, Abdullah; Alhakami, Hosam; Raza, Muhammad Summair; Hussain, Altaf; Salah, Khaled; Djemame, Karim (2020). An in-depth Empirical Investigation of state-of-the-art Scheduling Approaches for Cloud Computing. IEEE Access, 1–1. doi:10.1109/ACCESS.2020.3007201.

[12] domanal, shridhar; Guddeti, Ram Mohana; Buyya, Rajkumar (2017). A Hybrid Bio-Inspired Algorithm for Scheduling and Resource Management in Cloud Environment. IEEE Transactions on Services Computing, 1–1. doi:10.1109/TSC.2017.2679738.

[13] Cheng, Mingxi; Li, Ji; Bogdan, Paul; Nazarian, Shahin (2019). Resource and Quality of Service-Aware Task Scheduling for Warehouse-Scale Data Centers: A Hierarchical and Hybrid Online Deep Reinforcement Learning-Based Framework. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1–1. doi:10.1109/TCAD.2019.2930575.

[14] Zhou, Ruiting; Li, Zongpeng; Wu, Chuan (2018). Scheduling Frameworks for Cloud Container Services. IEEE/ACM Transactions on Networking, 26(1), 436–450. doi:10.1109/TNET.2017.2781200.

[15] Fernando, Niroshinie; Loke, Seng W.; Rahayu, Wenny (2016). Computing with Nearby Mobile Devices: a Work Sharing Algorithm for Mobile Edge-Clouds. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2016.2560163.

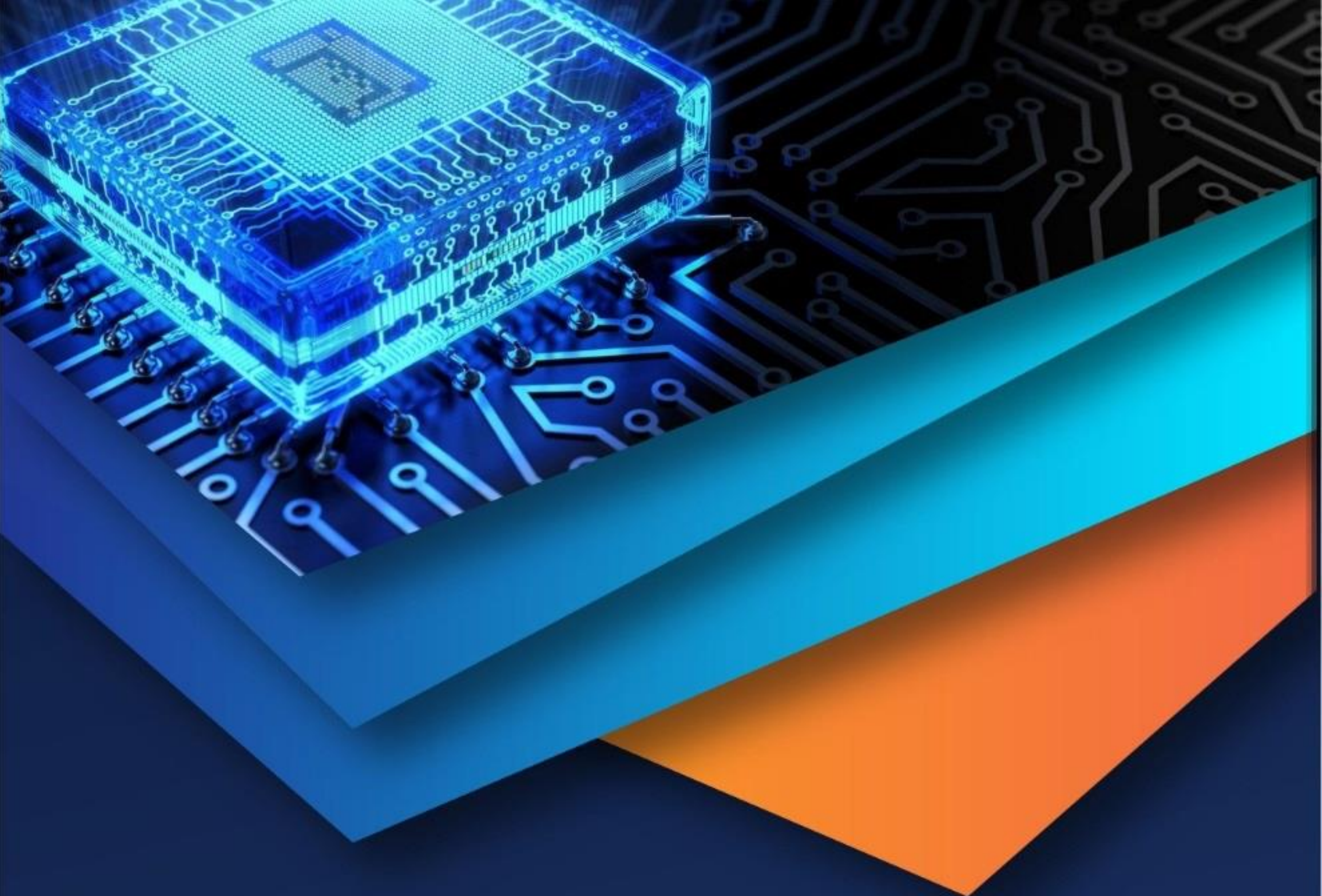
[16] Cha terjee, Subarna; Misra, Sudip; Khan, Samee (2015). Optimal Data Center Scheduling for Quality of Service Management in Sensor-cloud. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2015.2487973.

[17] Wangsom, Peerasak; Lavangnananda, Kittichai; Bouvry, Pascal (2019). Multi-Objective Scientific-Workflow Scheduling With Data Movement Awareness in Cloud. IEEE Access, 7, 177063–177081. doi:10.1109/access.2019.2957998.

[18] Zhao, Dongfang; Mohamed, Mohamed; Ludwig, Heiko (2018). Locality-aware Scheduling for Containers in Cloud Computing. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2018.2794344.

[19] Chen, Yunliang; Wang, Lizhe; Chen, Xiaodao; Ranjan, Rajiv; Zomaya, Albert; Zhou, Yuchen; Hu, Shiyan (2016). Stochastic Workload Scheduling for Uncoordinated Datacenter Clouds with Multiple QoS Constraints. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2016.2586048.

[20] Khabbaz, Maurice; Assi, Chadi (2015). Modelling and Analysis of A Novel Deadline-Aware Scheduling Scheme For Cloud Computing Data Centers. IEEE Transactions on Cloud Computing, 1–1. doi:10.1109/TCC.2015.2481429.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY