Vol.65: e22220025, 2022
https://doi.org/10.1590/1678-4324-2022220025
ISSN 1678-4324 Online Edition

Article - Engineering, Technology and Techniques

Blockchain Based Adaptive Resource Allocation in Cloud Computing

Sumathi Muruganandam[1]
https://orcid.org/0000-0002-8039-0920

Vijayaraj Natarajan[2]
https://orcid.org/0000-0001-7450-4963

Raja Soosaimarian Peter Raj[3]
https://orcid.org/0000-0002-7216-2207

Venkatachalapathy Maharajan[4]
https://orcid.org/0000-0002-9693-8903

[1]SASTRA Deemed University, Department of School of Computing, Thanjavur, Tamil Nadu, India; [2]Vel Tech Rangarajan Dr. Sagunthala Research and Development Institute of Science and Technology, Department of Computer Science and Engineering, Chennai, Tamil Nadu, India; [3]Vellore Institute of Technology, School of Computer Science and Engineering, Vellore, Tamil Nadu, India; [4]K. Ramakrishnan College of Engineering, Department of Mathematics, Tiruchirappalli, Tamil Nadu, India.

Editor-in-Chief: Alexandre Rasi Aoki
Associate Editor: Alexandre Rasi Aoki

Received: 12-Jan-2022; Accepted: 13-Apr-2022.

*Correspondence: sumathi@it.sastra.edu; (S.M.).

HIGHLIGHTS

- The permissioned blockchain is used for resource allocation and monitoring.
- To reduce storage size, partial data is stored in the onchain blocks instead of entire data.
- The EU's based Fixed and variable size resource allocation techniques are proposed.

Abstract: Cloud computing is breakthrough technology with applications in education, industry and research. Implementing a cloud environment, however, depends on virtualization, parallel computing, resource management, service-oriented architecture, and distributed computing. The efficiency of cloud computing depends on effective resource allocation (R_A). Conventional R_A, from resource requests to price settlement, is handled entirely by cloud service providers (CSPs), which make it difficult for end users (EUs) to view details relating to resource availability and price. Hence, an alternative technique is required to provide an efficient R_A between EU's. In this work, the blockchain (BC) technique is used for R_A between the EU's without the intervention of CSP. The BC provides decentralized and secured communication between the EU's without any inconsistency. Based on demand, R_A is carried out in two different ways, fixed size and variable size. In a fixed size R_A, each user utilizes an equal quantity of resources at a particular time but in a variable size R_A each user utilizes the varying quantity of resources based on demand. Given that the R_A, chosen depends on resource availability and demand, the proposed work employs hybrid R_A (Adaptive Resource Allocation) schemes such as fixed-size adaptive R_A (FSARA) and variable-sized adaptive R_A (VSARA). The simulation results show the comparative analysis of the proposed and existing R_A techniques. When compared to existing RA techniques like optimal, greedy and iterative techniques, the proposed technique achieved more 90% customer satisfaction. Depends on priority the allocated resource price is varied in a proposed provides equal profit (50%) to R_R and R_P. When compared to existing random transmission technique, the proposed technique takes lesser than 8.5% of transmission time. The proposed technique reduces transmission latency, confirmation latency, and response time, while simultaneously increasing throughput and R_A scalability without CSP intervention.

Keywords: Resource allocation; Smart contract; Blockchain; Fixed and Variable size Resource Allocation.

INTRODUCTION

A standard cloud computing architecture encompasses the subsequent techniques like ubiquitous computing, on-demand network, and pervasive computing [1]. Cloud computing virtualization technique specifies the sharing of resources by using three deployment models (private, public and hybrid cloud), four service models (Software as a service, Plat-form as a services, Infrastructure as a Service and Storage as a services) and five characteristics (elasticity, measured services, broad network access, resource pooling and on-demand self-service) [2]. Five characteristics of cloud computing, incorporated in BC technology, include measured service, on-demand self-service, rapid elasticity, resource pooling, and broad network access. In Measured service, resource utilization is monitored, reported, and measured transparently between EU's with a pay-per-use model. On-demand self-service eliminates the need for human interaction. Cloud users have recourse to the cloud through a web-based self-service. Rapid elasticity involves the release of resources by cloud providers, based on demand. This particular characteristic extends resources anytime and anywhere, based on user needs. Resource pooling takes place when large numbers of cloud users access the same physical resources at any given time, with the resource pooling process separating resources logically. Broad network access facilitates the access of cloud computing resources over the network and supports heterogeneous user platforms in the form of mobile devices and workstations [3].

As the bedrock of decentralized crypto-currencies, the BC has numerous applications across fields like RA in cloud computing, the Internet of Things (IoT), and resource offloading [4]. The consensus protocol incorporates a basic process, called Proof-of-Work (PoW). The computational procedure is considered as mining, where the consensus nodes which contribute their computational capacity to mining are known as diggers [5]. Apart from the feature of public access, permissionless BC has the advantage in quickly establishing a self-organized data management platform to support various decentralized applications (DApps) [6]. To alleviate the computational bottleneck, the consensus nodes can access the cloud/fog computing service to offload their mining tasks, thus enabling BC-based DApps [7]. As the cloud/fog computing service generates additional consensus nodes to execute the mining process, it improves BC network robustness markedly. This raises DApp valuation and attracts other DApps users, forming a virtuous circle [8]. Since no prior approval is required, BC without authorization is particularly appropriate for decentralized self-governing information administration in numerous applications.

Traditionally, all RA has been entirely depended on CSPs to allocate resources to resource requesters (RR) and handle price settlements to resource providers (RP). Similarly, given that the resource availability (RAV) and price settlement details are overseen by public auditors, they and CSPs together exert complete control over the RA. The EU's are unable to control their resources and are unable to view the resource information. The proposed technique, therefore, is designed to view the RAV and resource prices through the BC technique, with little or no interference from CSPs and public auditors. Major issues with the RA and monitoring the cloud include difficulties with publishing or advertising the RAV to EUs, and verifying the publisher's authenticity. Occasionally same resources are allocated to more than one RR, affects the quality of services. Similarly, resources are allocated to RR without authorization verification leads to security issues. In a CSP-based RA, log records are fully CSP-dependent and changes to the records can only be instituted by the CSPs themselves. Hence, a BC based RA and management is proposed in this work without the intervention of CSP. The fact that the BC network is severely limited by storage means that large-sized chunks of information cannot be stored in a block. As a result, information is divided into two, transaction and resource. Transaction information is stored in the on-chain block and resource information in the off-chain block to overcome issues with BC storage constraints.

The rest of the paper is structured as follows: Section 2 discusses existing work related to the R_A in cloud computing, the use of BC in R_A and their merits and demerits. Section 3 outlines the proposed AR_A technique in cloud computing, using the BC with appropriate algorithms. Section 4 compares the experimental results of the proposed technique with existing techniques. Finally, the proposed technique is concluded with directions for future enhancements.

Related work

Hong bing Wang and coauthors [9], introduced the fitness enabled auction method for allocating cloud resources to R_R through CSP guarantees for fitness in terms of performance traits. The resource allocating algorithm takes into consideration the constraints of economic efficiency and system performance. The experimental results have shown that such allocation is far more efficient than it is with continuous double auction. Cloud R_A via the fitness-enabled auctions has introduced new measures like fitness, dynamic asking/bidding strategies, and an active bargaining model that computes the final dealing price and profits. Zhen Xiao and coauthors [10], proposed dynamic R_A based on application demands, applying virtualization technology in tandem with a green computing data center that uses an optimized number of servers. Multidimensional server R_A is introduced to deal with resource unevenness in terms of "skewness", which maximizes the performance of the server workload utilized and conserves energy. Physical resources are mapped with virtual machine and monitored by the virtual machine monitor (VMM). Data centers use different generation hardware for heterogeneous physical machines (PMs) to realize the twin goal of overload avoidance and green computing. Overload avoidance helps utilize the PM effectively to satisfy the needs of the cloud user and the CSP. Green computing discovers idle PMs and turns them off to minimize energy. The skewness algorithms periodically evaluate the R_A by using hot and cold spots methods. A skewness algorithm is implemented to achieve overall avoidance and monitor resource use.

Xingwei Wang and coauthors [11], introduced a scheme whereby multiple cloud users access multiple cloud providers by enabling a combinational double-auction protocol with a new intelligent and economical approach for dynamic R_A. Price matching algorithm and price prediction algorithm have been used for different mechanisms like price formation, bidding and multi round auction. This work helps participants from a cloud market overcome anomalies by applying the paddy field algorithm, which is implemented with the winner determination problem to facilitate the best possible use of resources. Chenhan and coauthors [12] and Christina Terese Joseph and coauthors [13], proposed the BC based intelligent resource management in the cloud datacenters. The request scheduler uses the BC technique for the R_A to EUs, and the reinforcement learning-based smart contract (SC) to minimize energy consumption and costs. The reinforcement technique uses prior knowledge to schedule the resources to the R_R to reduce data utilization costs.

Conventional CSP-based R_A and its use are fully dependent on CSPs. The EU's are unable to change the access policies and price of the resources. Similarly, when accessing resources, EUs need to obtain permission from the CSP to do so, resulting in delays. To this end, Zyskind and coauthors [14], proposed a BC-based data management scheme wherein BC blocks hold access to information on the data and metadata. CSP interaction is thus circumvented and processing speed increased as well. Varian and coauthors [15], proposed the BC based R_A scheme in which a SC is used for R_{AV} verification and R_R R_A. A sealed bid is used for mining process and the loser pays the winner a price is avail the resource needed, with the R_A depending on the winner. Yousafzai and coauthors [16] proposed a R_A life cycle process to advertise, allocate, monitor and freeze resources. An efficient R_A technique requires an effective monitoring system as well to track the R_A pricing, quality of service, and the R_{AV}.

Arpit Shukla and coauthors [17], proposed the peer to peer R_A in 6G environment with public BC. The peer to peer process takes lesser latency in an optimal bandwidth size. The Block-RSA technique provides highly reliable, trusted and transparent communication between EU's. Similarly, the interplanetary file system was used for optimal transactions. Yu Gong and coauthors [18], discussed deep reinforcement learning-based R_A in the BC network using a slice broker that collects requests and allocates resources, based on network slice tenants. A service-level agreement (SLA) is established between EUs using the deep reinforcement learning algorithm, with resource scheduling carried out by the SC. Depending on the SLA and the SC, resources are allocated to the R_R. The entire R_A process results in resource unavailability to other requesters. Yaodong Huang and coauthors [19], proposed the R_A and consensus of BC in pervasive edge computing environments. Optimal peer nodes are identified for transactions and for the storage of resources, especially the rapid retrieval of missing resources. The blocks are dynamically reallocated with low energy consumption.

Jianbo Du and coauthors [20], proposed the resource pricing and allocation in mobile network based BC along with reinforcement learning. The miner-based R_A, the R_R pays a high price to utilize resources, and the R_A depends on miners who pay more to avail much-needed resources. Hence, the actor critic based deep reinforcement learning was used to reduce the miner price and try to provide resource to requester with optimal price. Lejun Zhang and coauthors [21], proposed R_A and trust computing for BC-enabled edge computing. A new group agent strategy with trust computing has been used for ensuring the reliability of edge devices. The sorting and ranking process have been used for improving the R_A in each device. The Zipf distribution model was used for transaction of blocks between EU's. Similarly, the symmetric searchable encryption was used to ensure the secure transaction.

Amit Natthani and coauthors [22] proposed the police- based R_A in IaaS cloud. The dynamic scheduling based R_A is done based on new request. Whenever a new request is raised, the rescheduling process has been performed to allocate the resources to R_R. Due to rescheduling process, R_A time has been increased. Neeraj Kumar and coauthors [23] used the adaptive scheduling technique in cloud computing. The shared resource allocation technique has been used to allocate a resource to the requester. Depends on R_{AV}, scheduled the resources to R_R increases R_A time.

Summary of literature Review:

Table 1 shows the summary of the literature review.

Table 1. Summary of Literature Review

Sl.No	Key Parameters	Characteristics
1	BC Technique in R_A	Efficient R_A and monitoring through blocks without the interference of CSP
2	EU's control	EU's are have full control over their resources, their authenticity and authorization are established without difficulty.
3	BC network	Registered R_R's or R_P's are involved in the R_A, provides easy authenticity and authorization verification of members without any discrepancies.
4	Payment management	R_R and R_P communicate each other provides direct payment between them.
5	BC enabled Edge computing	Ranking based R_A improving R_A in each device.
6	Dynamic scheduling	Depends on resource availability and requested resource quantity, the rescheduling is performed frequently increases, R_A time.

Motivation and Justification of the Proposed Work:

The primary limitation of BC, firstly, is block size. It is impossible to store information in a single block, resulting in increased BC storage size. Secondly, the SC-based R_A increases latency during the R_A and scaling processes in the cloud. Further, the SC-based access policy and access control leak Sensitive Information (SI). Our motivation is to resolve these issues, the BC block size problem through on-chain and off-chain mode of storage. In a proposed technique, the block holds the resource and transaction information. Due to personal information along with resource information, the R_{AV} and price information need privacy than the transaction information. The transaction information is the frequent information and contains the resource type, price and quantity of resources are shared between the EU's. Hence, the transaction information doesn't require protection. To satisfy this constraint, the on-chain and off-chain storage representation is proposed in the work. Secondly, the SC issues are replaced by the EU's dependent priority based AR_A technique. This technique reduces latency and scaling process in the cloud environment. Finally, the access control (AC) updating and revocation is also a complicated task in the cloud environment. To avoid this issue, the registered authenticated user network is introduced in the proposed work.

Contributions:

The major contribution of the proposed work is listed as follows:

1. Currently R_A is managed and controlled by CSP's. The R_A and payment settlement depends on the CSP. The EU's are unaware of the R_{AV} and price of the resources. Likewise, unauthorized users involvements are high. To overcome these issues, the BC based R_A is proposed in this work.
2. The literature review clearly shows that the existing R_A techniques lead to higher response time, lesser throughput and profit to R_R and R_P. To overcome these issues, the priority based adaptive R_A technique is proposed in this work. There are two different R_A procedure is followed in this work such as fixed and variable sized R_A technique. Depending on the R_{Rq} and R_{AV}, either fixed or variable sized R_A technique is preferred to satisfy the R_R need and profit.
3. The major limitation of BC technique is storage size. The maximum size of each block is 1MB. Likewise, in a public BC everyone able access and view the information without any restriction. To overcome this

issue, the permissioned BC is used in the proposed work. The permissioned BC allows only the registered authorized users instead of all.

PROPOSED BLOCKCHAIN BASED ADAPTIVE RESOURCE ALLOCATION

The proposed BC based Adaptive Resource Allocation (AR_A) technique is developed in a permissioned BC network. Compared to permissionless network, a permissioned network provides the kind of secured and authorized grid that only permits registered user participation in the R_A. Hence, the proposed technique provides higher confidentiality, integrity and availability to EU's through onchain and offchain modes. The onchain block holds the transaction information, while offchain block holds the resource information like Resource Type (R_T), name of the R_P, Resource Availability (R_{AV}), Resource price (P_R) and duration of the R_{AV} (DR_{AV}). On-chain blocks are maintained in EU's local storage and off-chain blocks in cloud storage, which overcomes the limitations of BC storage size. Figure 1 shows the proposed system architecture.

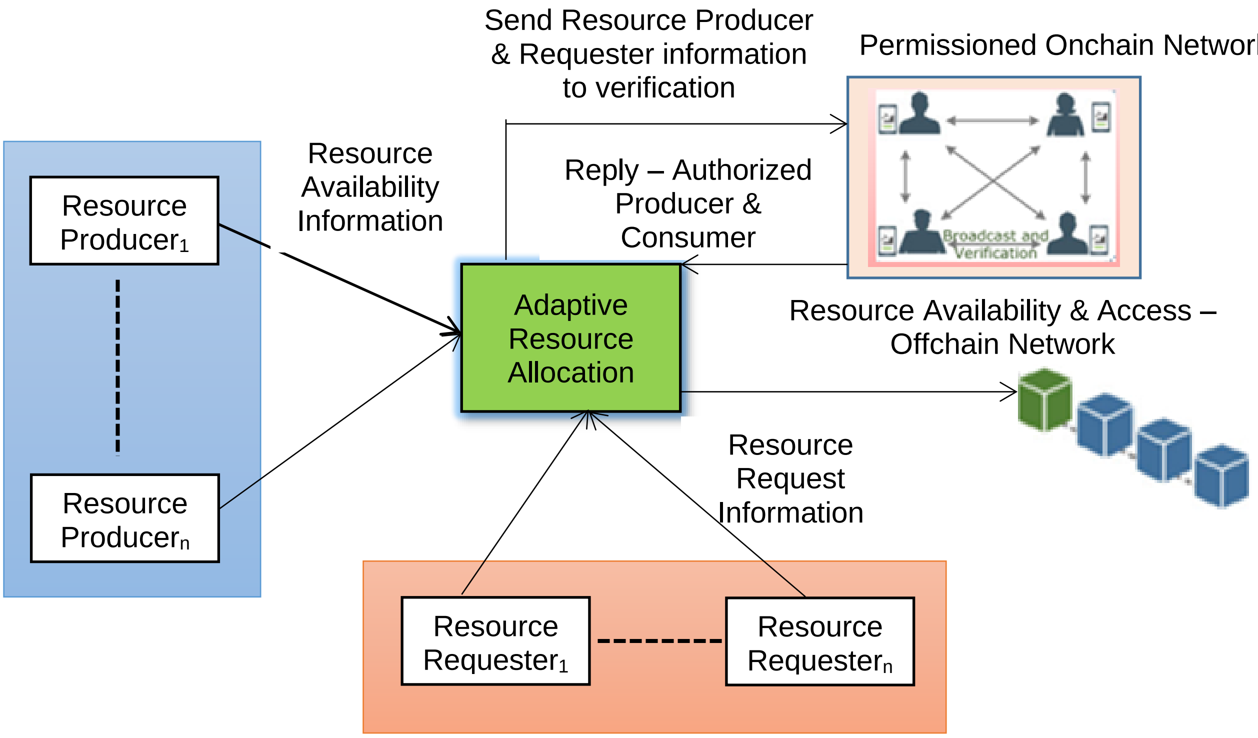

Figure 1. Outline of Proposed Blockchain Based Adaptive Resource Allocation in Cloud Computing.

Initially the R_P $\{R_{P_1}, R_{P_2} \dots\dots R_{P_N} : N \in Number\ of\ R_P\}$ sends the R_{AV} along with the R_T $\{R_{T_1}, R_{T_2}, \dots\dots, R_{T_N} : N \in Number\ of\ R_T\}$, Quantity of R_{AV}, Duration of Resource Availability (DR_{AV}), P_R and R_P authenticity information to AR_A module. Similarly, the R_R $\{R_{R_1}, R_{R_2}, \dots\dots, R_{R_N} : N \in Number\ of\ R_R\}$ sends the requested R_T, quantity of resources required, duration of the resource utilization, and expected P_R and authorization information to the AR_A Authorization and authentication information from the R_P and the R_R is sent to the on-chain network and verified, based on existing registration information and a confirmation from on-chain network members. Assuming the members accept the request, a reply will be sent to the AR_A module, which permits the R_P to create a block in the off-chain network for resource information storage.

Fixed Size Adaptive Resource Allocation (FSARA)

In the FSARA, let us assume that the R_P produces a quantity of resources that is equal to that of the R_A and R_R requests the same. The FSARA does the R_A depending on the R_{AV} and the demand. Algorithm 1 shows the FSARA based R_A. Initially, the R_{Ti} is set as 0. When a R_P sends $R_{AV}(R_{Ti})$, the corresponding R_{Ti} count is incremented and added to the block along with the R_T, P_R and DR_{AV} and R_P information of each resource. Each block contains information from a single R_P, equation 1 shows information from the blocks.

$$Block(B_i) = (R_{P_i}, R_{T_i}, QR_{AV}, P_R, DR_{AV}) \tag{1}$$

The collection of particular R_T generates a BC for maintaining separate resource information. In the FSARA, given that each R_P produces a fixed quantity of resources, all the blocks hold a fixed volume of data. Figure 2 shows the BC representation of FSARA.

The block information pertaining to the Previous Block hash value (HV) (PB_{HV}), R_T, R_{AV}, P_R, DR_{AV} and R_P is visible to every registered user in the permissioned BC. The information stored in a BC is maintained in the form of hash value (HV). Since a distinguishing feature of the HV is to produce fixed-size output to a variable-size input, each block holds fixed-size information. Another major benefit of BC is to provide data integrity to user data. This particular property ensures that the updated R_{AV} information in the block will remain unaltered, facilitating its rapid verification. Owning to collision-free property, the Secure Hash Algorithm 256 (SHA256) algorithm is preferred for generating the HV of blocks concerned, as shown in equation 2.

$$HV(B_i) = DSHA_{256}(PB_{HV} + R_T + R_{AV} + P_R + DR_{AV} + R_{PID}) \quad (2)$$

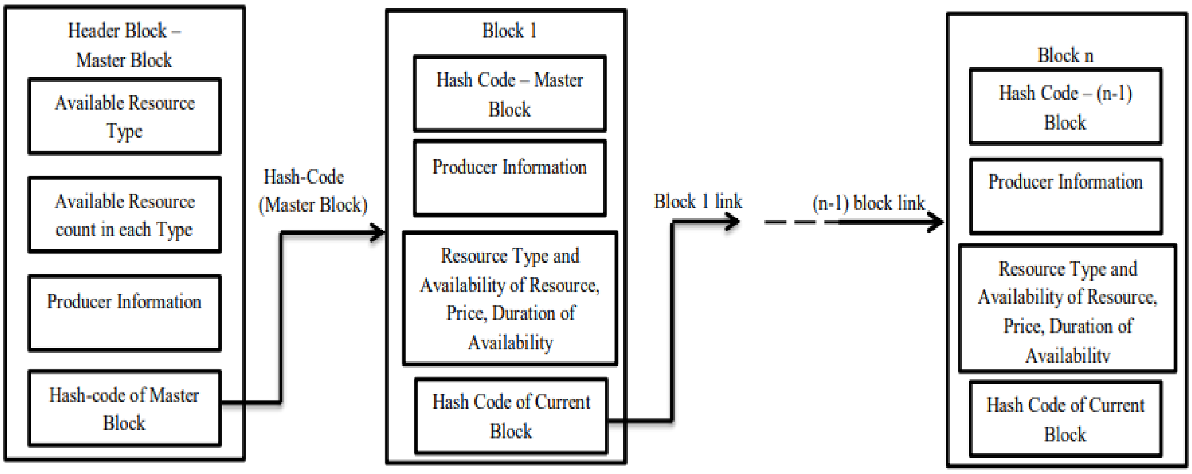

Figure 2. Fixed Size Resource Availability Maintenance in Blockchain

The FSARA handles the fixed quantity R_A and R_{Rq} only. When the R_{Rq} sends a request to the permissioned BC, the R_R authorization is verified by network members, following which the R_R gets a token from the R_P and submits it to the FSARA. Now, FSARA allows R_R and provides permission to access the block in the BC. The FSARA, which is implemented by the SC, permits the R_R to access the BC block. The SC is a tool used to control BC transactions and implemented by the programming language, Solidity.

Algorithm 1: Fixed Size Adaptive Resource Allocation

Input: R_{AV}, R_{Rq}
Output: Resource Allocation to R_R and Generate Block (B)

Begin Procedure
For all R_{Ti} do
Set $R_{Ti} \leftarrow 0$ // R_{AV} and R_{Rq} set as 0
If (R_P send $R_{AV}(R_{Ti})$ to FSARA)
$R_T \in i$,
$R_{Ti}{+}{+}$, *// Increment i^{th} R_T count*
Create a B_i
$Block(B_i) = (R_{P_i}, R_{T_i}, QR_{AV}, P_R, DR_{AV})$
End if
If (R_R send $R_{Rq}(R_{Ti})$ to FSARA)
Check R_{AV} in B_i
If ($R_{Rq}(R_{Ti}) \leq R_{AV}(R_{Ti})$)
Allocate resource to R_R
Else
Reject R_{Rq}
End if
End for
End Procedure

The R_{AV} based BC is created and stored in the off-chain storage location. Initially, the R_R sends an R_{Rq} (R_R ID, R_T, R_{AV}, duration of utilization) to the R_P which dispatches R_R information to the on-chain network to verify its authenticity and R_{AV}. If more than 50% of all members acknowledge the authenticity of R_R, their reply to the R_P carries an acceptance or a rejection. If the reply is accepted, the R_P sends a token to the R_R for access the off-chain blocks. Consecutively, the off-chain network verifies the R_{AV} based on request. If the requested resources are available, the corresponding R_R is able to access the resources according to their priority. If the reply is rejected, the particular R_{Rq} discontinuous the access transaction. If the R_R is verified and resources are available, the permitted R_R, sends the token to the off-chain storage for the blocks to be accessed. The off-line storage sends the token to the R_P for verification following which a verification message is sent back to the off-chain storage. If the verified message is received from the R_P, the off-chain storage sends the respective block to the R_R and an acknowledgement to the R_P. Similarly, the R_R sends the bill amount to the R_P. Finally, the R_P sends a verified message to the on-line network to generate the block, based on the R_R resource utilization. With this process, the R_R easily identifies the R_{AV} and the R_P has complete control over their resources. In this way, CSP interactions are completely eliminated and data integrity is verified without the need for third party public auditors.

Variable Size Adaptive Resource Allocation (VSARA)

In a FSARA, the R_P produces a fixed quantity of resources to R_A and the R_R request, likewise, is for the same fixed quantity of resources. Additionally, VS resources are produced and consumed with certain resources subject to high consumption and others not at all. Similarly, a particular R_T may be required long-term by the R_R and other resources over short-time period of time. Such cases need special care during the R_A and the monitoring. In a VSARA, the priority (P) is assigned to each request and allocates resources based on P. The starvation problem is overcome by an aging factor. The P calculation depends on amount and duration of the R_{Rq}. Based on threshold value (th), the resource request duration is classified as long-term (L_T), medium-Term (M_T) and short term (S_T). Threshold 'th' is evaluated as the average of total R_{AV} time. Equation 3 is used to classify the request based on time requirement. Table 2 shows the P calculation of R_{Rq}

$$Avg_{time}(th) = \frac{\sum_{i=1}^{N} DR_{AV}}{N}$$

$$L_T = T_R(R_{Rq}) > th \quad and \;\; M_T = L_T < T_R(R_{Rq}) > S_T \; (or) T_R(R_{Rq}) = th \quad and \;\; S_T = T_R(R_{Rq}) < th \qquad (3)$$

Where N represents total number of resources available and DR_{AV} the duration of the R_{AV}

Similarly, equation 4 is used to classify the quantity of the R_{Rq} as high, medium and low. The threshold 'th_1' is calculated as the average of R_{AV} of a particular R_T.

$$Avg(\, R_{AV}(th_1)) = \frac{\sum_{i=1}^{N} R_{AV}}{N}$$

$$High = Number(R_{Rq}) > th_1 \; and \; Medium = \; Number\;(R_{Rq}) = th_1 \text{and } Low = Number(R_{Rq}) < th_1 \; (4)$$

Based on resource utilization duration and quantity of the R_{Rq}, the R_{Rq} is classified as high, medium and low priority.

Table 2. Priority of Resource Request

Amount of R_{Rq}	Duration of R_{Rq}	Priority Level
High	L_T	High (1)
Medium	M_T	Medium (2)
Low	S_T	Low (3)

Equation 5 is used to measure the R_A to R_R based on P.

$$R_{R_i}(R_A) = P_i > P_j \qquad (5)$$

Resources are allocated to P_i and P_j must wait for P_i to release them. The block generation and block access procedure of VSARA similar to FSARA scheme. Through the P value, VSARA achieve better resource utilization. Algorithm 2 shows the VSARA based R_A. If the requested resource is allocated to P_i, the R_{AV} count is decremented from the total R_{AV}. Whenever the R_A is performed, the R_{AV} count is decremented. Equation 6 used to measure the current R_{AV} for a particular time.

$$R_{AV}(Time_T) = Total(R_{AV}) - Allocated_{Resource} \tag{6}$$

The current R_{AV} is updated in the block for further allocation. Such dynamic updation provides accurate information on the R_{AV} and enables a better transaction from the R_A to the R_R. The block construction is undertaken when the R_P produces the resources, or the R_R consumes them.

Algorithm 2: Variable Size Adaptive Resource Allocation

Input: R_{AV}, R_{Rq}
Output: Resource Allocation to R_R and Generate Block (B_i)

1. **Begin Procedure**
2. **For** all R_{Ti} do
3. Set $R_{Ti} \leftarrow 0$ // R_{AV} and R_{Rq} set as 0
4. **If** (R_P send $R_{AV}(R_{Ti})$ to VSARA)
5. $R_T \in i$,
6. $R_{Ti}++$, // Increment i^th R_T count
7. Create a B_i

$$\mathrm{Block}(B_i) = (R_{P_i}, R_{T_i}, QR_{AV}, P_R, DR_{AV})$$

8. **End if**
9. **If** (R_R send $R_{Rq}(R_{Ti})$ to VSARA)
10. Check R_{AV} in B_i
11. **If** ($R_{Rq}(R_{Ti}) \leq R_{AV}(R_{Ti})$ && $P_i > P_j$)
12. Allocate resource to R_R
13. **Else**
14. Reject the request
15. **End if**
16. **End for**
17. **End Procedure**

Price Estimation of Resource Utilization

P_R estimation plays a vital role in resource utilization. This work discusses two different R_A modes, FSARA and VSARA. In a FSARA, since each R_R utilizes a fixed quantity of resources during the course of a fixed duration, no price deviations occur between R_R. Since the R_R utilizes resources in a 'P' order in the VSARA, there is a price deviation that occurs. EUs looking for VS resources in the 'P' order are required to pay more than those based on the 'P' value. While high-priority request demands a higher payment, the inverse is true if the 'P' is low. The price deviation is to be mapped between the 'P' and the price. Table 3 shows the 'P' based price list of resources.

Table 3. Price Prediction of Resources

Resource Type	Priority Level	Price
Highly Demand	High	P_R * 20%
Moderately Demand	Medium	P_R * 10%
Low demand	Low	P_R

Table 3 shows the price allocation based on resource demand and 'P'. Normally, a particular R_T is frequently accessed is in great demand by the R_R, while other R_T are accessed sparingly, or almost never, by the R_R. In the proposed technique, the P level is fixed as 1 to 3. Priority level 1 is fixed as high, 2 are medium and 3 are fixed as low. Based on the demand and P level, the price is fixed for resource utilizations. The priority and price estimation based R_A is done by VSARA.

In a block, the fixed price is stored by R_P and the VS price details are calculated at the time of resource utilization. Consequently, the price utilization figure is stored in the online block mode, rather than the offline. The BC based R_A provides secure R_A between R_P and R_R without the intervention of CSP. Similarly, the BC provides EU's data integrity and averts data loss, thus warranting R_A efficiency.

The proposed R_A is compared to the existing R_A techniques like optimal, greedy and iterative. The existing optimal, greedy and iterative techniques are not considered the priority. In an optimal R_A technique, the resources are allocated on the basis of predefined minimal and maximum allocation. If the R_{Rq} is nearer to minimal size, the minimal size resources are allocated else if the R_{Rq} is above the predefined minimal size, the maximum size of resources is allocated for the request. In a greedy technique, the R_{Rq} are allocated to the R_R without considering the 'P'. Thus, the minimal size R_R needs to wait till the resources released. In an iterative R_A technique, the average value of the R_R is calculated and fixed the average value as the 'th' to resource distribution. Depends on the 'th', fixed size resource is allocated to R_R in an iterative manner. In these three cases, first come, first serve process is followed to allocate the resources to R_R.

PERFORMANCE ANALYSIS

This section analyses the performance of the proposed technique in terms of transmission latency, throughput, confirmation latency, throughput, confirmation latency, decentralization, security, response time, availability, reliability, resource utilization, and scalability.

Response Time

In a service-based system, response time is a key factor, and is defined as the duration of time between user requests for the R_A. Equation 7 is used for calculating the response time of the proposed technique.

$$Res_{Time}(R_A) = (P_T(Req) + \ D(R_R \rightarrow \ R_P) + Del_{Req}) \tag{7}$$

Response time ($Res_{Time}(R_A)$) depends on the time taken to process the request (P_T(Req)), distance between R_R and the R_P (D(R_R to R_P)), request processing delay (Del_{Req}), processing technique, and network bandwidth.

Availability

Efficient R_A demands that the R_{AV} is delivered to the authorized R_R. Depending on the availability of the requested resource, the R_R consumes it with the minimum delay or waits until it is made available. In the proposed technique, the R_{AV} time is also stored in the block. This means that when the R_R requests a resource from the R_P and fails to find an appropriate response, it may look for an alternate R_P. Equation 8 is used for measuring the R_{AV}.

$$R_{AV}(R_T) = \sum_{i=1}^{n} Total(R_{AV}) - \sum_{i=1}^{n} Total(R_A) \tag{8}$$

The $R_{AV}(R_T)$ is the difference between the sum of the total R_{AV}, during a particular time period and the total R_A. Based on equation 8, the $R_{AV}(R_T)$ for the particular time period is evaluated.

Reliability

Reliability is considered a quality of service parameter. In VM applications, given that resource failure and unavailability is a common issue, reliability is required to provide EU's continuous service. Reliability is defined as the probability of a successfully executed request. Equation 9 is used for find the reliability of the proposed technique.

$$Rep_{Req} = \left(\frac{Total_{Req} - Failure_{Req}}{Total_{Req}}\right) * 100 \tag{9}$$

In a proposed technique estimates the R_A and reliability based on user requests.

Resource Utilization

In R_A and resource utilization are common to the cloud. Resource utilization that is a hundred percent effective is impossibility, and calls for an analysis is measure customer satisfaction. Equation 10 is used for measure the resource utilization of the proposed technique.

$$Resource\_Utilization = \frac{\sum_{i=1}^{n}(Task\_compled\_time - Task\_starting\_Time)}{Total_{Utilization_{Tim}}} \quad (10)$$

Resource utilization time refers to the average of the task starting and completion time against the total utilization time. Effective resource utilization maximizes use and minimizes idle time.

Scalability

Scalability of the proposed technique is measured as by increasing or decreasing the number of R_R and R_P. Scaling in a cloud environment involves frequent up-and-down operations involving a scalability analysis. The auto-scaling process manages scalability such that it effectively sums up the resources available in different time periods. Depending on the 'th' value, the minimum requirement of each request is identified after the auto-scaling technique monitors the scaling up-and-down operations. A resource size that is smaller than the 'th' is to be allocated, based on the R_{AV}, to achieve resource scalability.

Transmission Latency

The average size of a single transaction is expressed by equation 11, where $\mathbb{\Gamma}$ represents the transmission latency,

$$\mathbb{\Gamma}_{req}^{tr} = \max_{u_m \in u}\{\frac{\mathbb{\Gamma}}{R_{U_i}, B_P}\} \quad (11)$$

$\mathbb{\Gamma}_{req}^{tr}$ the transmission latency based on the request, R_{U_i} the transmission rate for user 'i' and B_P the block generation time for the master node

Similarly, the computational delay (consisting of SC execution, HV generation, and verification) is also considered for latency calculation. Equation 12 is used to calculate the computational delay.

$$\mathbb{\Gamma}_{req}^{c} = \frac{\Lambda_{req}, B_n}{F_{B_n}} \quad (12)$$

Where $\mathbb{\Gamma}_{req}^{c}$ is the computational delay for the on-line mode, $F_{B_n}$the total number of blocks, and Λ_{req} the computational cost of the master node.

In the proposed technique, computational delay occurs only in the online transactions and block generation modes. Given the R_P carries out the offline block generation mode, computational delay is not considered here.

Throughput

Throughput represents the number of transactions successfully completed in a unit of time. In BC, throughput depends on block generation and reaching of consensus. Block creating depends on master node block size and processing capacity. Equation 13 is used to measure the throughput of the BC generation.

$$\daleth = \frac{(K_n - IB)}{K_n} \quad (13)$$

Where $\daleth$ represents the throughput, K_n the number of block generated continuously, and IB the number of blocks ignored.

Confirmation Latency

In the BC technique, block generation depends on a confirmation of the transaction by network members. The block, once generated in the BC, cannot be altered at any given point in time. Therefore, transaction confirmation is critical to the BC technique. The confirmation time includes two parts, time to computation (T_c) and time to propagation (T_p). Equation 14 shows how the total confirmation time calculated.

$$Total\ Confirmation\ Time = T_c + T_p \quad (14)$$

In a proposed system, confirmation latency occurs during the course of the R_R verification in the on-chain mode.

Decentralization

The major advantage of BC technique is that it provides EU's decentralized services, without loss of data, through replicas of the blocks produce every time. The Gini Coefficient-based replica generation technique produces perfect replicas in every instance, as shown in equation 15.

$$G(R) = \frac{\sum_{t\in T}\sum_{t'\in T}|R(t) - R(t')|}{2\ \sum_{t\in T}\sum_{t'\in r} R(t)} = \frac{\sum_{t\in T}\sum_{t'\in T}|R(t) - R(t')|}{2N\ \sum_{t\in T}\sum_{t'\in T} R(t)} \tag{15}$$

Here $G(R) \in \{0, 1\}$, if the G(R) with small values that decentralization is high in BC technique. If G(R)=0, perfect equality is achieved such that every replica produces an equal number of blocks and G(R)=1 indicates a higher level of inequality. Decentralization is ensured through the $G(R) \leq threshold$.

Finality (last irreversible block):

The transaction security is guaranteed by the prevention of transactions to be arbitrarily reversed or changed. The finality time is the time that the transaction cannot be revoked once the blocks are committed in the BC. Finality time (F^T) includes the propagation time (P^T) and computation time (C^T). Equation 16 is used to find the finality time of the proposed system.

$$F^T = P^T + C^T \tag{16}$$

Computational and Space Complexity Analysis:

Computational Complexity: The proposed technique computational complexity depends on number of R_P and R_R. When number of resources increases, the offchain block needs to update. Likewise, if the number of request is increased, the on-chain block construction is increased. The computational complexity of the proposed technique is $O(N^{(R_P+R_R)})$. The computational complexity is increased when number of block construction is increased.

Space Complexity: The proposed technique space complexity depends on the resource information (R_I) is stored in the off-chain network and space required to execute the algorithm (A_{ET}). The required space complexity of the proposed technique can be expressed as $O(N^{(R_I+A_{ET})})$.

EXPERIMENTAL RESULTS

This section discusses the experimental results of the proposed system when compared with existing R_A techniques such as the greedy algorithm based R_A, optimal solution based R_A, and iterative R_A techniques. Next, the price allocation of the proposed FSARA and VSARA techniques is compared to that of existing schemes. Further, the BC performance of the proposed system is analyzed in terms of random transmissions. In the proposed work, the VM is considered the resource, and different types of VMs and VM prices are collected from Cloudorado [24], which comprises VM configurations (Amazon, Microsoft and Google) and a price comparison. For the purpose of analysis in our experimental work, the service time for the R_R is considered the normal distribution, rather than rare long and short-term distribution cases. Hence, the normal distribution is considered for analysis. The CloudSim toolkit is used for cloud computing environments and evaluation of virtualization resources based on demand. The experiment is done with 10 to 75 nodes with 4 GB RAM, 100 GB storage and 200 to 1000 MIPS of each node. The BC implementation is done in Hyper-ledger fabric (HLF) platform. The HLF contains enterprise-ready BC features like scalability, performance, trust level etc. The HLF version 0.4.2 is used as the benchmark system. The parameters which are involved in the proposed system are listed in table 4.

Table 4. Parameters of Proposed System

Parameters	Range of Values	Parameters	Range of Values
Number of CPU	1-4	Block size space	0.5 & 1 MB
Performance of CPU (MIPS)	200-1000	Transaction time	2 sec
Storage Capacity (GB)	512-4096	'th' time to finality	4 sec
Bandwidth (Gb / S)	0.5 – 10		
Cost ($ / h)	1-4		

Resource Allocation Scheme Comparison

The proposed AR_A is compared with existing optimal, greedy and iterative R_A scheme. When compared to existing scheme, the proposed FSARA scheme provides higher R_{AV} to all the users. Further, the FSARA scheme equalizes the R_{AV} and the R_{Rq}, making it the preferred choice, while the VSARA is otherwise preferred by the R_R. Otherwise, the VSARA is preferred by the R_R. In the existing optimal and greedy technique, the requested resources are directly allocated to R_R without considering others request. New users looking to enter the R_{Rq} are required to wait until existing users release resources. The proposed technique eliminates such problems. Table 6 shows a comparison of the proposed R_A technique with the existing techniques. In the proposed technique, the quantity of the R_{Rq} is fixed as 20, based on R_{AV}. Requests that fall within this range choose either the FSARA or the VSARA. Existing optimal, greedy and iterative techniques are not considered in term of priority. Compared to the greedy, optimal and iterative R_A approaches, the proposed R_A technique allocate resources so efficiently that the optimal R_A and greedy algorithm obtained 60% and 80% customer satisfaction, respectively, while the iterative technique fell within the range of 60% to 80% range. The proposed technique R_A technique on the other hand, achieved 90% customer satisfaction, demonstrating that it offers the best R_A of all.

Price Allocation Comparison

The price settlement varies, depending on the types of R_A. Price allocation in the proposed FSARA scheme is identical to that of existing R_A price schemes, though the VSARA differs in this respect. In the VSARA technique, the price allocation depends on the 'P' of the R_{Rq}. When the R_{Rq} is high, the price allocation of 'P' increases. Thus, since the technique proves beneficial to the R_P and depends on the demand the R_R generates when it utilizes resources at a higher cost, the R_R is satisfied as well. Table 5 shows a price comparison of existing optimal, greedy and iterative methods with the proposed VSARA scheme. In the latter, since the average R_{Rq} is 20, requests greater than 20 are considered a high 'P', while 20 is medium, and anything less than 20 is a low 'P'. The price allocation is carried out depending on the 'P' values.

Table 7 shows how the price allocation of the proposed VSARA scheme varies from that of other schemes. The on-demand price of single instance VM in Google is $0.021811 is consider for price analysis. High and medium-priority prices are, respectively, 20% and 10% higher than the actual, while a low priority price is considered the actual cost. This calculation is applied to a price analysis of the proposed technique. When compared to the optimal, greedy and iterative R_A techniques, the price allocation of the proposed technique provides and equal percentage of profits (50%) both to the R_P and R_R. With other techniques, however, either the R_P or the R_R receives the profits, rather than both. The proposed technique is, as a result, preferred both by the R_P and the R_R. Table 5 shows the parameters are used in the proposed system.

Table 5. Parameters are used in Proposed System

Parameters	Value
Total Number of users	10
Total Number of R_{AV}	175
Average R_{RQ}	19.3
Optimal Allocation Min	20
Total Number of R_{Rq}	193
Optimal Allocation Max	45

Response Time

The response time of the proposed technique is compared with that of existing greedy, optimal and iterative techniques and found to be higher, on average. Figure 3 shows a comparison of the existing and proposed technique, with the latter responding quicker than existing technique.

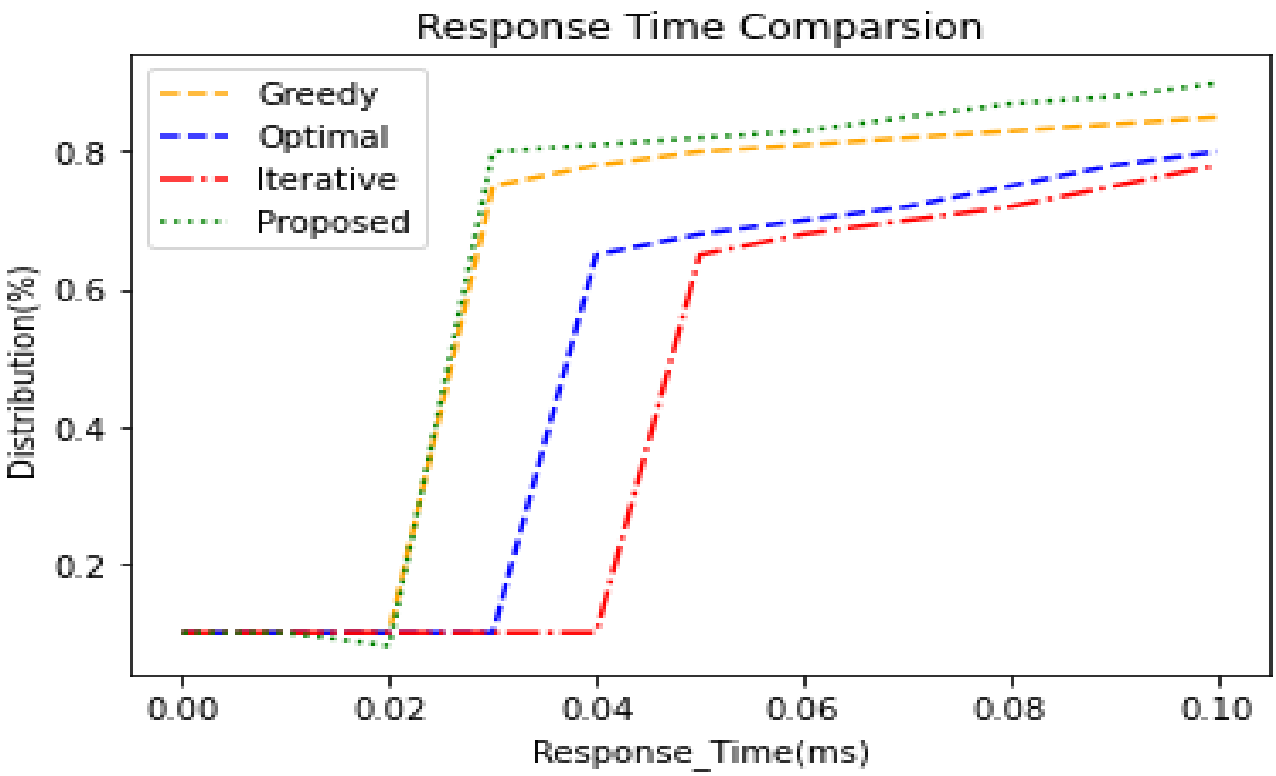

Figure 3. Response Time.

Resource Utilization

The resource utilization of the proposed technique is higher than that of the existing technique, producing higher scheduling results than the existing R_A. Figure 4 shows a comparison of resource utilization with existing techniques.

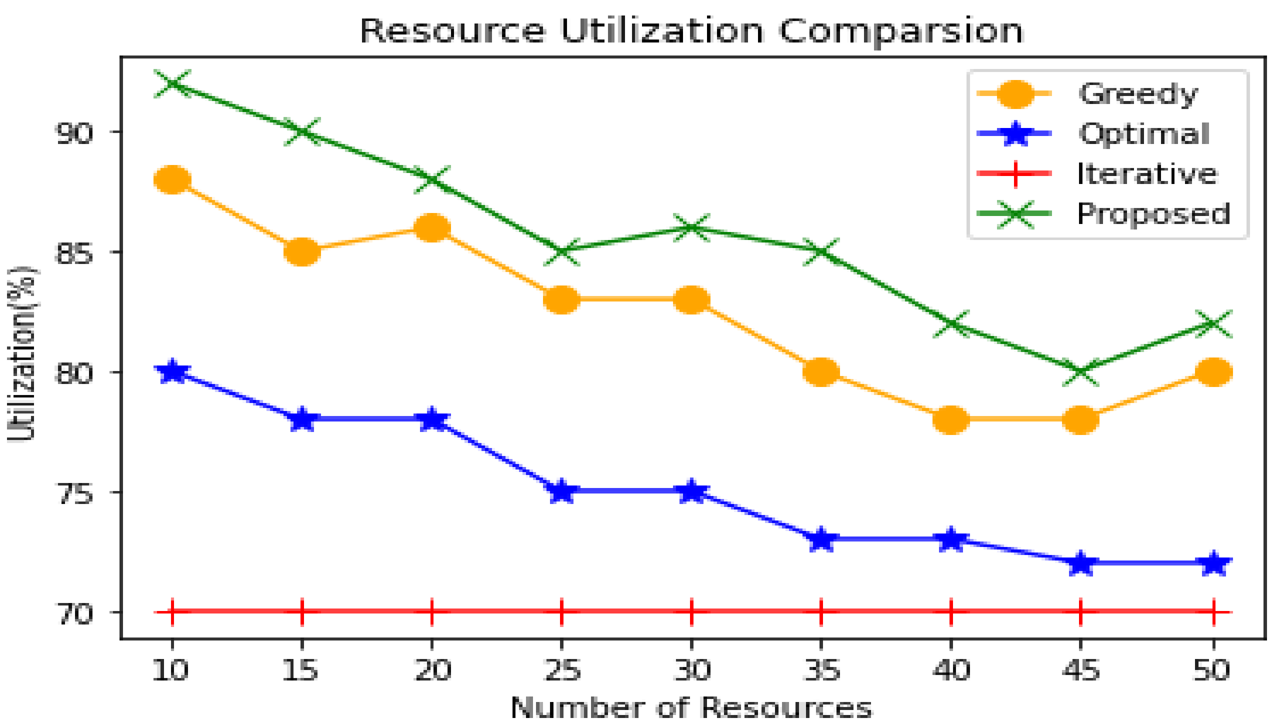

Figure 4. Resource Utilization

Reliability

Resource reliability, as a primary requirement of cloud R_A is measured by the number of users expressing satisfaction with their requests being attended to within a particular duration. In the proposed VSARA technique, the varied R_A satisfied more requests on a particular time period. The experimental results have demonstrated that the proposed technique provides better reliability than existing R_A technique. Thus, the proposed technique has proved that the R_A is carried out precisely and with the highest reliability. Figure 5 shows a comparison of the proposed and existing techniques.

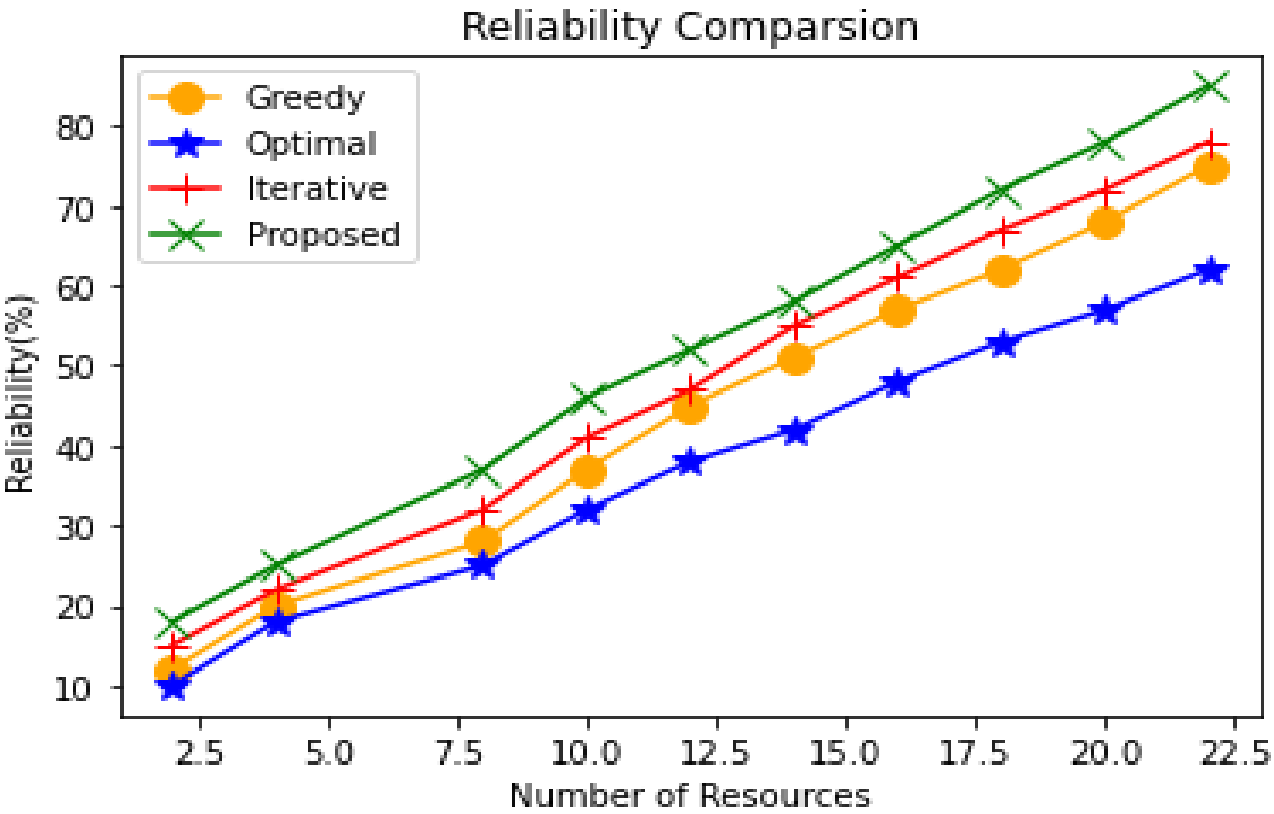

Figure 5. Reliability Comparison

Table 6. Resource Allocation Comparison of Proposed and Existing Techniques

User ID	Resource Requested	Priority of Request	Optimal Allocation	Greedy Allocation	Iterative Allocation	Proposed FSARA	Proposed VSARA	Proposed Technique R_A order
1	15	3	20	10	20		15	4
2	30	4	45	30	20		30	6
3	8	2	20	8	20		8	2
4	10	1	20	10	20		10	1
5	20	3	20	20	20	20		5
6	15	5	20	15	20		15	9
7	25	4	Wait	25	20		25	7
8	15	2	Wait	15	20		15	3
9	45	5	Wait	Wait	Wait		Wait	
10	10	4	Wait	Wait	Wait		10	8

Table 7. Price Allocation Comparison Proposed Technique with Existing Technique

User ID	Resource Requested	Priority of Request	Actual Price	Price Allocation				Price Deviation Comparison			
				Optimal Allocation	Greedy Allocation	Iterative Technique	Proposed Technique	Optimal Allocation	Greedy Allocation	Iterative Allocation	Proposed Technique
1	15	3	0.3271	0.4362	0.3271	0.4362	0.3598	0.1091	0	0.1091	0.0327
2	30	4	0.6543	0.9814	0.6543	0.4362	0.6543	0.3271	0	0.2181	0
3	8	2	0.1744	0.4362	0.1744	0.4362	0.2092	0.2618	0	0.2618	0.0348
4	10	1	0.2181	0.4362	0.2181	0.4362	0.2617	0.2181	0	0.2181	0.0436
5	20	3	0.4362	0.4362	0.4362	0.4362	0.4798	0	0	0	0.0436
6	15	5	0.3271	0.4362	0.3271	0.4362	0.3271	0.1091	0	0.1091	0
7	25	4	0.5452	0.9814	0.5452	0.4362	0.5452	0.4362	0	0.1090	0
8	15	2	0.3271	0.4362	0.3271	0.4362	0.3925	0.1091	0	0.1091	0.0654
9	45	5	0.9814	0.9814	0.9814	0.4362	0.9814	0	0	0.5452	0
10	10	4	0.2181	0.4362	0.2181	0.4362	0.2181	0.2181	0	0.2181	0

Blockchain Based Resource Allocation Performance Analysis

In a proposed work, the permissioned BC is used for block construction and transaction. The R_{AV} information is stored in the offchain network and transaction information is stored in the onchain network by a R_P. The number of R_P's and R_R's is varying from 10 to 75 in a permissioned network and as an average 1 to 5 blocks are generated per minute. Resources are randomly requested by a R_R and the size of the block is considered to be a maximum of 1MB. All the block generation and transmission times are recorded for the analysis of transmission overhead and block delivery time.

Transmission overhead depends on the number of EU's in the BC network, which usually averages between 30 and 50. Minimum of 30 users in the network leads to high transmission overhead with a great distance between users. Consequently, the transmission time is nearly 75.8m and drops to 61.5m if the user count is 50. When the number of users exceeds 50, the average transmission time decreases. Figure 6 (a) and 6 (b) shows transmission latency and throughput of proposed BC based R_A.

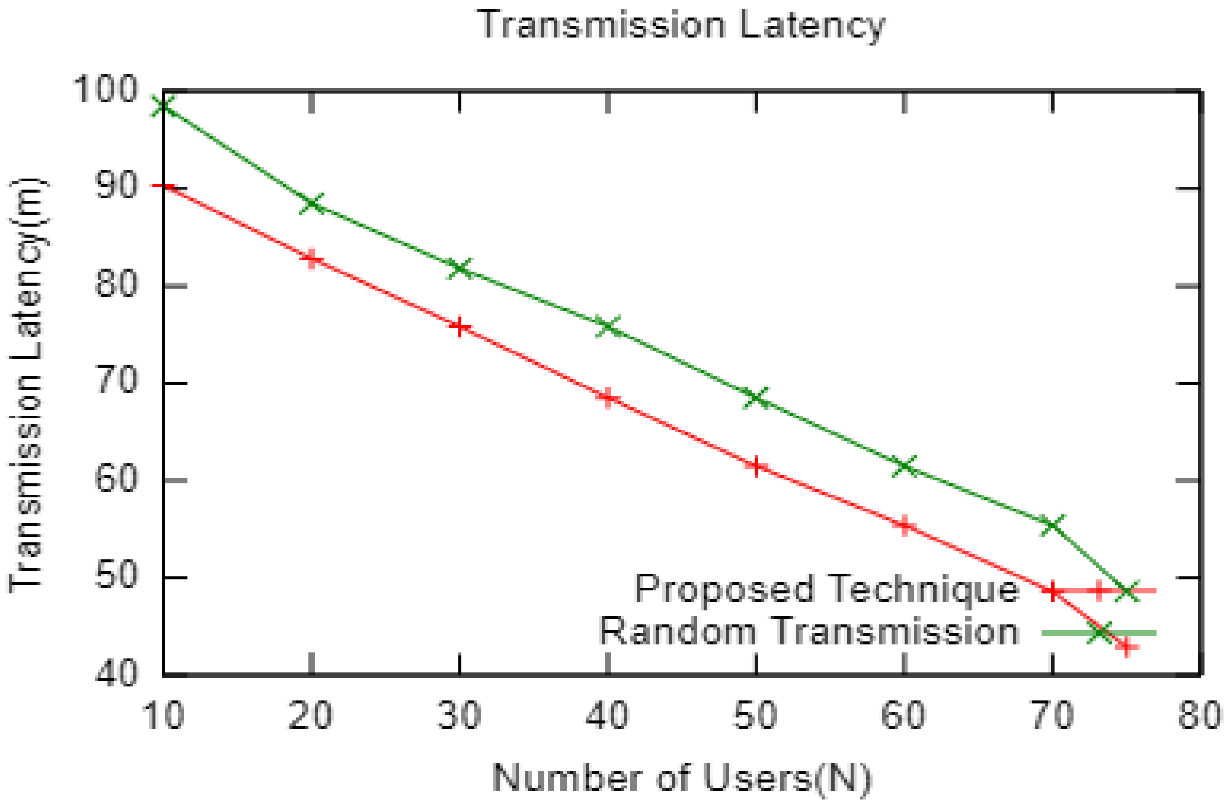

Figure 6. (a) Transmission Latency

The proposed AR_A technique has proved that it takes 8.5% less transmission time than existing random transmission techniques. When transmission latency is decreased, throughput is increased. Further, the proposed RA technique produces and processes more blocks within the time period stipulated than the random generation technique, as seen in figure 6(b).

Block confirmation latency in the proposed technique is far less than in the permissionless network-based block confirmation technique. In a proposed technique, block construction is carried out in parallel in two operation modes, online and offline, to reduce confirmation latency. When number of users are present in the network is high, the confirmation latency is increased and vice versa. Figure 6 (c) shows the confirmation latency analysis of the proposed technique with permissionless BC technique.

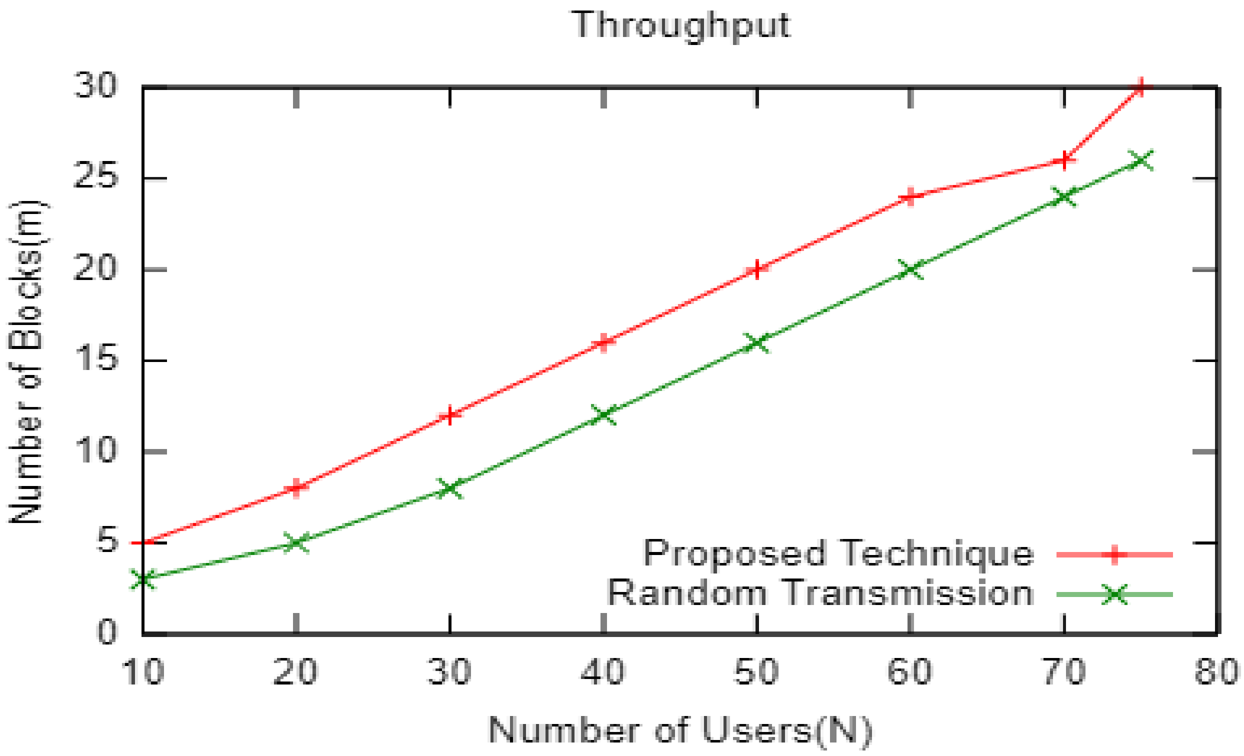

Figure 6. (b) Throughput

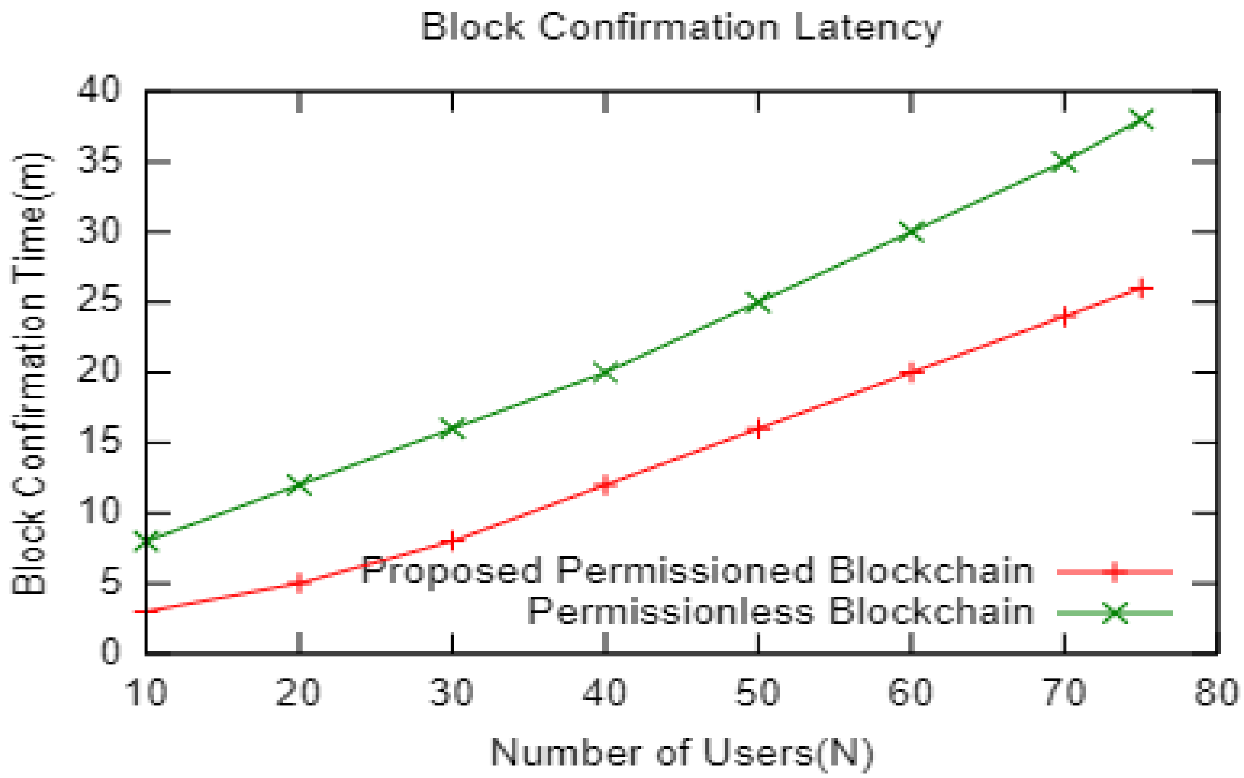

Figure 6. (c) Block Confirmation Latency

Transaction per second (TPS) and Inbound/outbound traffic network

In a proposed technique, the transactions are stored in the BC per second is measured by throughput. When the number of transactions per second increases, the throughput also increases such as the successful transactions are stored in the BC. Generally, single R_R takes 125ms to respond and the multiple R_R is calculated by number of R_R = 1250ms / 125ms = 10TPS. Depends on number of R_R and R_{AV}, the throughput is 10TPS with the maximum latency of 150ms. The authentication and R_{AV} verification is done in advance so not included in the calculation. If the number of requests takes more than 125 per millisecond, the R_R need not wait more than 125ms for confirmation. If the number of R_R increases to 100 the average TPS is 40 and elapse time is 100 milliseconds. Likewise the R_R increases, the TPS is also increasing. In the proposed technique, permissioned blockchain is used. Hence, limited authorized users are only participated in the process. Hence, inbound-outbound traffic volume is negligible.

Performance Analysis Discussion:

The conventional CSP based R_A depends on third party CSPs not to EU's. The EU's unaware of the resource availability and pricing. To resolve this issue, the proposed BC-based adaptive R_A scheme provides efficient R_A to R_R based on the demand, in two different ways, through the FSARA and the VSARA. The VSARA increases R_P profits and R_R utilization, satisfying the expectations of both. Additionally, the on-chain and off-chain-based storage representation reduces transaction and block confirmation latency, and increases throughput as well. The BC-based RA scheme provides EU's secured and confidential record maintenance. As a result, resource security, availability, and integrity are maintained without the intervention of CSPs and public auditors. To prove the efficiency of the proposed technique, the existing techniques like optimal, greedy and iterative techniques performances are compared. When compared to these techniques, the proposed technique provides improved results and satisfies the R_P and R_R in an efficient way. The off-chain storage technique provides overcomes the storage constrain issues of the BC and onchain blocks stores 1MB of data at the maximum. Thus, the BC storage restrictions are avoided in the proposed technique. The EU's are directly communicating with each other in an efficient way without the intervention of CSP.

CONCLUSION AND FUTURE ENHANCEMENT

A blockchain-based adaptive resource allocation scheme has been developed in the proposed work for efficient cloud resource allocation and to provide secured communication between resource providers and requesters. Resource allocation of fixed and variable sizes is proposed in this work to facilitate efficiency. The secured price settlement between the producer and requester is effectively achieved through blockchain technique. The results demonstrated by the proposed resource allocation scheme obtained more than 90% satisfaction between end users. By using the permissioned blockchain-based record maintenance avoids centralized control and record loss in a decentralized storage system, in addition to providing much-needed confidentiality to user records. The proposed price allocation satisfied both producer and consumer price restrictions by more than 50%. The on-chain and off-chain based storage representation reduces transaction and computational latency, and improves throughput as well, better than the entire data-based storage. The proposed adaptive resource allocation technique takes less response time and offers more resource utilization and reliability than existing greedy, optimal and iteration resource allocation techniques. Our proposed work is implemented in a static environment in our laboratory. It is anticipated, in future, that the proposed work will be applied to different resource types (Infrastructure, software and platform based resources) with a dynamic resource allocation and pricing technique. Similarly, implement this technique in the irregular information with improved quality of service.

REFERENCES

1. Kumar N, Saxena S. A Preference-based resource allocation in cloud computing systems. 3rd International conference on rec. trends in comp. 2015. Vol.57:104-11.
2. Sumathi M, Sangeetha S. Enhanced Elliptic Curve Cryptographic Technique for protecting sensitive attributes in cloud storage. IEEE Int. Conf. Computar. Intel. and Computar. Res. 2018 Dec:1-5.
3. Sumathi M, Sangeetha S. Blockchain Based Sensitive Attribute Storage and Access Monitoring in Banking System. Int. J. Cloud App Comp. 2020 April; 10(2):77-92.
4. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. Avaliable from: https://bitcoin.org/bitcoin.pdf
5. Garay J, Kiayias A, Leonardos N. The Bitcoin backbone protocol: Analysis and Applications. In Proc. Annu. Int. Conf, Theory Appl. Crypt. Techn. 2014 August: 281-310.
6. Pradip K, Sharma PK, Chen M.Y. A software defined fog node based distributed blockchain cloud architecture for IoT. IEEE Access.2017 Sep:308-15.
7. Devi BM, Agrawal S, Srivinvaslu C. An efficient resource allocation technique for uniprocessor system. Int. J Adv. Eng Technol. 2013:1-8.
8. Guo F, Yu FR, Zhang H, Ji H, Liu M, Leung VCM. Adaptive Resource Allocation in Future Wireless networks with blockchain and mobile edge computing. IEEE Trans. Wirel. Commun. 2020 March; 19(3):1689-704.

9. Wang H, Kang Z, Wang L. Performance-Aware Cloud Resource Allocation via Fitness-Enabled Auction. IEEE Trans. Parallel Distrib. Syst. 2016; 27(4): 1-15.
10. Xiao Z, Song W, Chen Q. Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. IEEE Trans. Parallel Distrib. Syst. 2013; 24(6): 1-11.
11. Xingwei W, Xueyi W, Che H, Li K, Huang M, Gao C. An intelligent economic approach for dynamic resource allocation in cloud services. IEEE Trans. Cloud Comput. 2015; 3(3): 275-89.
12. Xu C, Wang K, Guo M. Intelligent resource management in blockchain based cloud datacenter. IEEE Trans. Cloud Comput. 2017; 4(6): 1-13.
13. Joseph CT, Chandrasekaran K. IntMA: Dynamic Interaction aware resource allocation for containerized micro-services in cloud environments. J. Syst. Archit. 2020: 1-15.
14. Zyskind G, Nathan O, Pentland AS. Decentralizing privacy: Using blockchain to protect personal data. IEEE Secur. Priv. Workshops. 2015:180–4.
15. Varian HR, Harris C. The VCG auction in theory and practice. Amer. Econ. Rev. 2014; 104 (5): 442–5.
16. Yousafzai, Gani, Noor M. Cloud resource allocation schemes: review, taxonomy, and opportunities. Knowledge and information systems. 2017; 5: 347-81.
17. Shukla A, Gupta R, Tanwar S, Kumar N, Rodrigues JJPC. Block-RAS: A P2P resource allocation scheme in 6G environment with public blockchains. IEEE Xplore. 2021: 1-6.
18. Gong Y, Sun S, Wei Y, Song M. Deep reinforcement learning for edge computing resource allocation in blockchain network slicing broker framework. IEEE 93RD Conference. 2021: 1-6.
19. Huang Y, Zhang J, Duan J, Xiao B, Ye F, Yang Y. Resource allocation and consensus of blockchains in Pervasive Edge Computing Environments. IEEE Trans. Mob. Comput. 2021;21: 1-14.
20. Du J, Cheng W, Lu G, Lu H, Cao H, Chu X, et al. Resource pricing and allocation in MEC enabled blockchain systems: An A3C Deep reinforcement learning approach. IEEE Trans. Netw. Sci. Eng. 2021;9:1–12.
21. Zhang L, Zou Y, Wang W, Jin Z, Su Y, Chen H. Resource allocation and trust computing for blockchain enabled edge computing system. Comput Secur. 2021:1-13.
22. Nathani A, Chaudhary S, Somani G. Policy based resource allocation in IaaS Cloud. Future Gener. Comput. Syst. 2012: 94-103.
23. Kumar N, Chilamkurti N, Zeadally S, Jeong YS. Achieving Quality of Service using resource allocation and adaptive scheduling in cloud computing with grid support. Comput. J. 2014; 57 (2): 1-10.
24. Cloud Computing price comparison. 2021. Cloudorado, Available from: https://www. Cloudorado.com/.