IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 18, 2022, accepted June 5, 2022, date of publication June 8, 2022, date of current version June 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3181546

Cone Model in Resource Provisioning for Service-Oriented Architecture System: An Effective Network Management to the Internet of Things

ROBIN SINGH BHADORIA[1], (Member, IEEE), HATIM GHAZI ZAINI[2], MD. MANZAR NEZAMI[3], MD. DANISH EQUBAL[4], ELFATIH ELMUBARAK MUSTAFA[5], AND SHERIF S. M. GHONEIM[6], (Senior Member, IEEE)

[1]Department of Computer Engineering and Applications, GLA University, Mathura 281406, India
[2]Computer Engineering Department, College of Computer and Information Technology, Taif University, Taif 21944, Saudi Arabia
[3]Department of Electronics and Communication Engineering, GLA University, Mathura 281406, India
[4]Department of Electrical Engineering, Galgotias College of Engineering and Technology, Greater Noida 201306, India
[5]Department of Computer Science, College of Science and Arts Balqarn, University of Bisha, Bisha 61985, Saudi Arabia
[6]Department of Electrical Engineering, College of Engineering, Taif University, Taif 21944, Saudi Arabia

Corresponding author: Sherif S. M. Ghoneim (s.ghoneim@tu.edu.sa)

This work was supported by Taif University Researchers Supporting Project TURSP-2020/345, Taif University, Taif, Saudi Arabia.

ABSTRACT The rapid growth of technologies like the Internet of Things (IoT) has thrown newer challenges to resource management, making it one of the crucial tasks in dynamic application scenarios of IoT. Service-Oriented Architecture (SOA) plays an important role in this respect by proffering reference architecture towards better positioning of infrastructural software and hardware components, mostly comprehending from a service-oriented perspective. In this direction, we propose Cone Model that deals with the performance and handling issues with SOA system. Service integration is an all-important feature of SOA that supports communication and interaction between multiple service components (Service Containers). In addition to discussing the related work for critical resource allocation algorithms, this paper proposes Cone Model for assigning an available set of resources to requested services, explained through mathematical expressions and algorithms of the model. These algorithms justify how different services from various devices in IoT can be provisioned on the adoption of this proposed cone model. It is illustrated through the estimated utilization of resources by the services. The proposed model aligns with the applications in multifarious domains, primarily in the IoT and in areas of resource sharing for cloud computing and massive data centers.

INDEX TERMS Cone model, critical resource allocation, Internet of Things, service computing, service-oriented architecture (SOA), SOA reference architecture.

I. INTRODUCTION

Recent years have seen a tremendous surge in domains like the Internet of Things (IoT). Apart from fast development in the hardware domain for technologies and platforms like IoT, the need for software solutions to provide logical infrastructure to the system on one hand and proffer suitable device abstraction and efficiency is inevitable. The necessary infrastructure for the variety of services provided by the IoT system is crucial and Service-Oriented Architecture (SOA) plays an important role in this respect. SOA defines the architectural paradigm for designing an efficient system, that facilitates providing services, including support for reusability, scalability, and flexibility in delivering end-to-end business solutions. It allows architectural developers to build robust guidelines and templates for better positioning of the architectural components in their infrastructural resources. SOA Reference Architecture (SOA-RA) does this task of offering better visibility to the system, rather than answering "what do SOA systems contain" [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaolong Li.

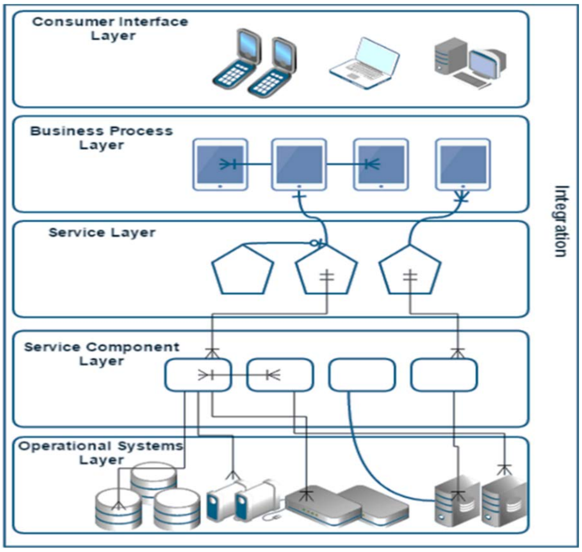

FIGURE 1. SOA reference architecture (logical view).

SOA for IoT is subtly different from its usual approach and focuses mostly on the reusability of software functionalities for different tasks [24]. Considering the model from a service-oriented perspective, the focus is on the design of the interactions of the IoT system with the outside world, which differs slightly from the traditional perspective of IoT. The introduction of SOA in IoT systems bolsters the security due to different levels of abstraction and interfaces on one hand and facilitates the communication between heterogeneous entities tied in the IoT system by virtualizing the communications and data distribution on the other hand.

SOA-RA is well defined and explained by many software industries and consortiums like OpenGroup Inc., OASIS, and IBM Inc. It consists of layered architecture for the design and implementation of the SOA system [3]. Fig. 1 illustrates a five-layer structure for the SOA-RA system.

It consists of five layers: 1) Operational system layer consists of key components like storage, processor units, etc. It is a position where the actual service provider resides. It is also responsible for running and deploying services over the web [2], [3]. 2) Service components layer manages and organizes service components (service modules), to facilitate the services in business process applications. It also supports the implementation of a particular service that is governed by the enterprise. It binds the service contract between its user and service suppliers by enabling loose coupling. 3) ***Service Layer*** consists of all logical services that have been supported by SOA. It consists of the service stack that may include interactive services, orchestration and business process services, data information services, application services, accessibility services, and service governance. This layer also defines the business-functionality support and capabilities for the services in SOA. 4) ***Business Process layer*** manages the interaction between services and business processes to retain the data and control flow. It also maintains the sequence of processes aligned with business goals. It acts as a central coordinator in aggregating loosely coupled services with SOA. It also helps in integrating web services in delivering end-to-end enterprise solutions. 5) ***Consumer layer*** is the position where a consumer resides. It can be a person, browser, or automation, for interaction with service in SOA.

A. CHALLENGES FOR RESOURCE ALLOCATION IN SOA SYSTEM

The major challenges for any SOA system with resource handling are linked to its service quality, data collection from heterogeneous sources, optimal methodology and algorithm for resource allocation, and satisfying quality of service (QoS) requirements to synchronize the workflows [16]. These challenges are directly related to the limitation of available resources for large-scale service integration over the grid and cloud computational resource handling [30]. In this respect, the following issues stand important:

Resource Scalability: When the computation is performed for large-scale service integration with the pool of resources, it needs a dynamic, scalable distributed system that works together as a powerful virtual machine over the internet. All such processes need to find an optimal resource allocation technique that supports the adaptability of resources in an efficient and scalable manner.

Dynamic Runtime Environment: A system must be developed and designed in such a manner that operates in an open and dynamic environment with an adaptive resource allocation methodology [8]. This vision combines the concept of Software-as-a-Service (SaaS) and Resource-as-a-Service (RaaS) on a common platform that supports dynamic runtime environments like self-management, and context-aware messaging and implements an efficient resource allocation algorithm. SOA system needs to be responsive and adaptable to the change in situations that arrive with large-scale service integration. It may include the rate at which service requests arrived, workflow synchronization, priorities for the task (workload), QoS milieu, resource availability, and various relevant environmental features that support dynamic runtime over the internet.

QoS Estimation: A SOA system must be capable of efficiently minimizing the gap between resource allocation and workflows. Analyzing the QoS feature need to be considered the issues of with Service-Level Agreement (SLA), loosely coupled and compositional behavior of the SOA system. Service monitoring is evaluated based on the hosting server, CPU time, network bandwidth, and memory used. In practice, a service generates a service instance called a service component that contains the actual service to deploy, would depending on workflow and services beneath. These components tangibly utilize the virtualized resource made available through the virtual machine (VM) concept. Overall, analysis of these components could produce a brief estimation of QoS in a dynamic runtime environment.

Optimal Resource Algorithm: A SOA system must handle resource allocation at runtime to support the QoS requirements. Thus, an algorithm must be devised to manage multiple workflows with dynamic resource assignments. Such an algorithm should utilize the existing resources to their maximum and generate maximum throughput. This algorithm should also be adopted by service providers to facilitate their clients' resource allocation. Concurrently, multiple service components can utilize these existing sets of resources to federate optimal resource allocation methodology [16].

Usage Costing/Pricing of Allocated Resources: While in the utilization of resources, costing/pricing would be considered according to the pay-per-usage strategy for available resources. There have been several models that support resource renting policy which must be part of SLA at the time of contract signing by the client. The duration of service would be counted based on use and generated invoice against it. An SOA system must be capable of adapting to the change in resources according to the need of the client over runtime [8].

The motivation for this paper is related to the efficient handling and provisioning of resources over the cloud and how it directly affects the workload for IoT platforms. There have been several cases observed where sharing of resources became a critical issue for hosting various services over the cloud. The solution for such a challenge can be done by adopting the proposed ''Cone Model'' for assigning an available set of resources to requested services. It justifies how different services from various devices of IoT can be provisioned on the adoption of this proposed cone model. It also aligns with the applications in multifarious domains for resource sharing in massive data centers.

The major contributions of this paper can be mentioned below:

- This paper discusses the role of SOA with its logical view reference architecture and also points out the challenges for resource allocation in the SOA system.
- Also, presents the adapting changes for efficient resource provisioning according to the need of clients over runtime and different workloads with its extensive survey of the Internet of Things (IoT).
- This paper also proposed the cone model that helps in the efficient handling and provisioning of resources in computing over IoT. Also, justifies how many resources and their instances can be engaged by a particular service over the SOA system.
- Derive the mathematical implication of the cone model in resource provisioning and service accessibility with IoT adopted the system.
- Brief the algorithms for calculating throughput and formulating optimization methods to allocate critical resources in the service-based system for IoT devices.

Section II briefs about the background and related work in resource allocation methodology for the service-based system. This section also discusses different models and techniques used by several authors in critical resource allocation. Section III presents the proposed cone model in the SOA system, which illustrates the relationship between service components and their associated resource (server instances). Section IV provides the mathematical view for the cone model along with the proposed algorithm for critical resource allocation. Section V demonstrates the results concurred using the proposed algorithm based on the cone-model for critical resource allocation and utilization based on the SOA system. This paper is concluded with the facts of the cone model in the implication of better handling and allocation of resources during runtime for service computing in SOA systems and discusses the significance and relevance of this model in the bigger domain of IoT, cloud computing, and data sharing.

B. SOA WITH FEDERATE AND COMMUNICATE IN THE INTERNET OF THINGS

With the recent advances in the domain of IoT, the focus is not just on the hardware design but also on the enhancement of performance from a software perspective [17]. IoT resource allocation problem is one of the most critical issues in this respect, and this paper presents a perspective of SOA for resource allocation between multiple service components. The problem of resource allocation spans from small IoT devices to massive cloud-based systems as well. This work is aimed at providing a solution to the resource allocation problem in SOA. It explained the paradigm for SOA to build strong guidelines in better positioning of system components for critical resources on the one hand. These resources can dynamically be facilitated to achieve momentous economies of scale, illustrated in the SOA-RA. The SOA-RA sets up the strategy that helps in handling the existing set of resources. On the other hand, the proposed Cone Model in the SOA system helps to comprehend the core functionality in critical resource allocation [5]. This cone model is also applicable in different areas like space positioning, weather forecasting, rocket engine performance, etc. This model discussed and commented on the accessibility of services through its components–namely ''service containers'' to server instances for resources. Efficient utilization of resources can easily be managed with this model. This paper also demonstrated some existing algorithms for the allocation of critical resources based on priority, presenting optimal throughput.

Service integration is an important aspect of SOA to federate and communicate multiple services on a common platform [11]. The probability related to resource allocation and utilization via multiple service components has been determined through mathematical models. This model also comments on the issue related to a particular service that requires a specific number of resources to complete its execution. This service assessment can be done using this mathematical model. Estimating the number of resources also discussed the ''expectation or expected value'' concept for the SOA system. This gives an idea–up to what extent an organization should stock and hold resources of a particular type and what could happen if we increase the stock units.

TABLE 1. Literature for cloud resource provisioning in IoT based system.

Reference	Resource Handling	Load Balancing	Adopted Model	Comment
[17] Li et al., 2021	Discuss Enterprise Architecture in IoT resource handling	No attention to the load balancing factor	Brief several technologies used for IoT resource allocation	No emphasis was given on throughput for resource allocation
[23] Tsai, 2018	Discusses the IoT resource allocation problem (IRAP)	Uses concepts of a metaheuristic algorithm	Search Economics for IoT Resource Allocation (SEIRA)	Limited for the region-based collected information
[29] Zhan et al., 2020	Service-oriented IoT resources access and provisioning	Design & develop frame-based protocol stack for IoT system	IoT Context-Aware Environment	No focus was made on the throughput of maps resource operations
[35] Deng et al., 2020	Use the Markov decision process (MDP) for the allocation scheme	Use edge servers for handling IoT devices	Use the reinforcement learning (RL) method	No emphasis is given to system states segregation
[17] Li et al., 2021	Discuss Enterprise Architecture in IoT resource handling	No attention to the load balancing factor	Brief several technologies used for IoT resource allocation	No emphasis was given on throughput for resource allocation
[23] Tsai, 2018	Discusses the IoT resource allocation problem (IRAP)	Uses concepts of a metaheuristic algorithm	Search Economics for IoT Resource Allocation (SEIRA)	Limited for the region-based collected information
[29] Zhan et al., 2020	Service-oriented IoT resources access and provisioning	Design & develop frame-based protocol stack for IoT system	IoT Context-Aware Environment	No focus was made on the throughput of maps resource operations
[34] Deng et al., 2020	Use the Markov decision process (MDP) for the allocation scheme	Use edge servers for handling IoT devices	Use the reinforcement learning (RL) method	No emphasis is given to system states segregation
[17] Li et al., 2021	Discuss Enterprise Architecture in IoT resource handling	No attention to the load balancing factor	Brief several technologies used for IoT resource allocation	No emphasis was given on throughput for resource allocation
[23] Tsai, 2018	Discusses the IoT resource allocation problem (IRAP)	Uses concepts of a metaheuristic algorithm	Search Economics for IoT Resource Allocation (SEIRA)	Limited for a region-based collected information
[29] Zhan et al., 2020	Service-oriented IoT resources access and provisioning	Design & develop frame-based protocol stack for IoT system	IoT Context-Aware Environment	No focus was made on the throughput of maps resource operations
[34] Deng et al., 2020	Use the Markov decision process (MDP) for the allocation scheme	Use edge servers for handling IoT devices	Use the reinforcement learning (RL) method	No emphasis is given to system states segregation
[17] Li et al., 2021	Discuss Enterprise Architecture in IoT resource handling	No attention to a load balancing factor	Brief several technologies used for IoT resource allocation	No emphasis was given on throughput for resource allocation
[23] Tsai, 2018	Discusses the IoT resource allocation problem (IRAP)	Uses concepts of a metaheuristic algorithm	Search Economics for IoT Resource Allocation (SEIRA)	Limited for a region based collected information
[29] Zhan et al., 2020	Service-oriented IoT resources access and provisioning	Design & develop frame-based protocol stack for IoT system	IoT Context-Aware Environment	No focus was made on the throughput of maps resource operations

II. BACKGROUND AND LITERATURE WORK

In allocating and monitoring resources, workflow management has been a crucial issue for consideration. A system must be capable enough to handle multiple services. Quality of service (QoS) is a primary requirement to satisfy numerous application services over runtime due to loose coupling, reusability, and dynamic behavior of SOA systems. With the cloud computing aspect, a market-based autonomic resource management system has been proposed [19], which provides an interface based on a Service-Level Agreement (SLA) in managing and handling resource allocation between cloud service providers and clients.

For the advancement in middleware technology, a sophisticated system has been designed and developed to manage architecture based on dynamic resource management [34]. At runtime, this optimizes an available set of resources with the current demands and status of particular resources. This system supports QoS even for overloaded time and produces a maximum throughput during its critical operation. A workflow-based resource broker for grid computing systems has been depicted in [10]. The principal job of these resource brokers is to check the availability of resources and synchronize workflow. This requirement helps the resource provider to administer multiple domains of services of the same kind. These resource providers also support interfaces that could be accessible through the user's credentials for an available set of resources in grid computing [9].

For web-services applications, a local and decentralized greedy approach has been devised for dynamic resource allocation [19]. This proposed methodology involves a software agent that trades all the resources and network-related services between the service provider and its associated client. For cost estimation and pricing for the usage of resources in fog computing, an optimal methodology has been proposed for web service applications by [18]. With dynamic multi-service networks, this web-based application not only provides an optimal and well-organized resource allocation but also does pricing estimation for next-generation multiclass networks. In addition, it also formulates a nonlinear pricing model that provides a solution for network delay constraints.

For service-based systems like utility computing systems, cloud computing systems, and grid computing systems that effectively use various application services simultaneously in a distributed environment, the SOA system has been widely accepted as a distributed kind of system that handles and federates different services with different environmental support. A service that cannot decompose to further smaller components is called atomic service, and the rest of all other services are termed composite service. This supports the efficient organization of available resources for single (atomic) as well as group (composite) service(s), which has been done in [13]. This paper shows that each workflow is made up of multiple atomic services which are governed by its critical resources.

With the recent advances in the domain of IoT, the focus is not just on the hardware design but also on the enhancement of performance from a software perspective [17]. IoT resource allocation problem is one of the most critical issues in this respect, and this paper presents a perspective of SOA for resource allocation between multiple service components. The problem of resource allocation spans from small IoT devices to massive cloud-based systems as well. This work is aimed at providing a solution to the resource allocation problem in SOA. It explained the paradigm for SOA to build strong guidelines in better positioning of system components for critical resources on the one hand. These resources can dynamically be facilitated to achieve momentous economies of scale, illustrated in the SOA-RA. The SOA-RA sets up the strategy that helps in handling the existing set of resources. On the other hand, the proposed Cone Model in the SOA system helps to comprehend the core functionality in critical resource allocation [5]. This cone model is also applicable in different areas like space positioning, weather forecasting, rocket engine performance, etc. This model discussed and commented on the accessibility of services through its components–namely ''service containers'' to server instances for resources. Efficient utilization of resources can easily be managed with this model. This paper also demonstrated some existing algorithms for the allocation of critical resources based on priority, presenting optimal throughput.

Service integration is an important aspect of SOA to federate and communicate multiple services on a common platform [11]. The probability related to resource allocation and utilization via multiple service components has been determined through mathematical models. This model also comments on an issue related to a particular service that requires a specific number of resources to complete its execution. This service assessment can be done using this mathematical model. Estimating the number of resources also discussed the ''expectation or expected value'' concept for the SOA system. This gives an idea–up to what extent an organization should stock and hold resources of a particular type and what could happen if we increase the stock units.

III. PROPOSED CONE MODEL IN SERVICE COMPUTING FOR SERVICE ORIENTED ARCHITECTURE (SOA) SYSTEM

The cone model is derived from the geometrical cone sharp. It justifies the one-to-many mapping relationship between multiple objects or vice-versa. It provides a basic structure for forming resource utilization and support in establishing the connection between multiple service components in SOA systems. It also offers a path through which routing can be possible. The Cone model has also been applied to multiple areas like space weather forecasting, in which this model improves the accuracy of kinematical and geometrical properties that helps in identifying the exact location. Work done in [4] used a cone-based model with Coronal Mass Ejections (CME), in which several points tried to converge at the center, producing accurate results. This model is also adopted by [5] into Solar Energetic Particle (SEP) events analysis and forecasting. In Rocket Engine Performance, a small cone is formed in the engin's combustion chamber and gradually increases during the burning of propellant fuel. The implication of the cone model in the rocket engine depicts that the sharper the cone shape is, the larger would be thrust generated into the combustion chamber. The shape of a cone is used in predicting the amount of thrust generated and the performance of a rocket engine [6]. With this fundamental understanding, this work employed the cone model for the SOA-based Enterprise Service Computing Architecture to predict the service resource estimation. Each service needs

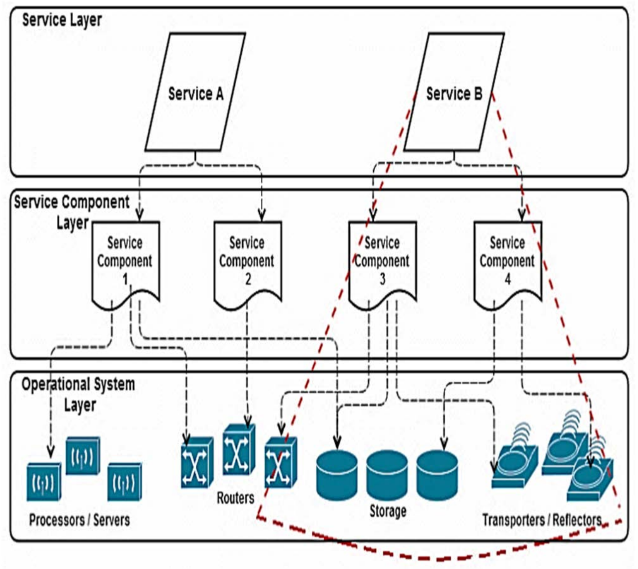

FIGURE 2. Logical view of the cone model in the SOA system.

resources to process its business logic–either it could be memory, processor, or bandwidth [11].

This model could best explain the critical resource allocation and utilization, in which the service requester (or simple service) is situated at the top of the cone and the rest of the infrastructure resources at the bottom. This model can be best viewed in SOA Reference Architecture with a layered approach, especially in the lower three layers, as shown in Fig. 2. Whenever any service is requested from the client-side for a particular purpose, it needs a lot of resources to hold for its processing. This enterprise service architecture is an allusion to SOA Reference Architecture that had been initiated by several developers & designers from different software industries [32]. Each service is incorporated by one or more service components, which is a 'service container'. These containers are the entities carrying data and associated information, which are plug-in and float with the service bus. This service bus helps in service integration between multiple services. This bus is known as the "Enterprise Service Bus (ESB)".

Fig. 2 depicts the logical view and understandability of the cone model in the SOA system. The operational layer consists of all types of infrastructural resources. Each service can be broken into one or more service components [3]. These components float or plug in with the service bus to access the critical resources. These resources must be utilized and assigned between multiple service components from the above layer. This methodology can be viewed in the shape of the cone as depicted by the red dash line above Fig. 2 and Fig.3.

ESB provides a common framework for developers to federate different environments on the common platform [35]. The ESB works in the Service Component Layer of SOA-RA (Fig. 1). The cone model provides a conceptual view for

FIGURE 3. Abstract view of cone model.

services and maps the resources held/required into a logical cone shape based on the relationship between a particular service and currently held resources by it. This also explains how services are being accessed and could be allotted with the available resources. Similarly, the reverse cone model provides a conceptual view of the available resources and recourses occupied by a set of services [36]. Such methodology could be best fitted into cloud-based computing systems.

For example, there are two services, namely service 'A' and service 'B'. Each service is decomposed into its respective components. Here, Service 'A' has two components i.e., service components 1 and 2. Similarly, Service 'B' also has two components i.e., service components 3 and 4. These service components have engaged these resources in the operating system layer as shown in Figures 2 and 4. This specifies the engagement of existing resources by service components and finally by its supervisory or parent service. The engagement of resources by service and its components creates a geometrical shape cone. It illustrates how many resources and instances can be engaged by a particular service. It also rectifies the possibility of deadlock occurrence for resources as it shows the details regarding the available set of resources and its associated services. It also provides the path through which routing can also possible for message handling through service bus (which is the job of ESB).

The concept of resource allocation can also be viewed as particular access and assignment of the available instance of resources that could be provided from the server-side in the SOA system [21]. Whenever a particular resource is being engaged by multiple services, then it creates an inverted cone sharp geometry in accessing that particular resource. This can be termed as INVERTED CONE MODEL, as shown in Fig. 4 and Fig. 5.

The abstract view of the above-explained inverted Cone Model has been inherited from work in [5], [6] that justified "Identifying and characterizing the cone model". The model concept can be portrayed as shown in Fig. 5. This shows that multiple services are using a particular resource instance at a given time. This concept raises the issue of the capacity of a

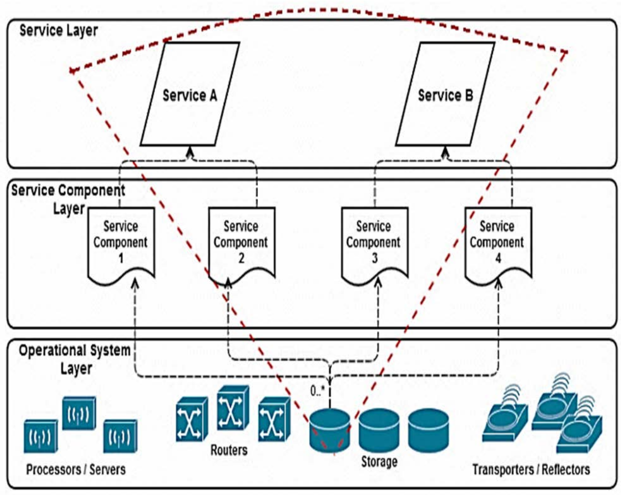

FIGURE 4. Logical view of inverted cone model in SOA system.

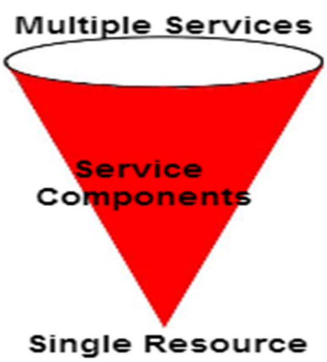

FIGURE 5. Abstract view for invert cone model.

particular resource with its efficiency in being accessed. The relationship between these two proposed models is based on multiple resources and published services from a particular domain of application. It could be a point of discussion in enterprise service computing architecture [22].

IV. THE IMPLICATION OF THE CONE MODEL IN RESOURCE PROVISIONING FOR SOA SYSTEM

This paper presents the mathematical idea for allocation and sharing of critical resources by multiple service components in an SOA system [30]. This mathematical expression illustrates how a particular service component can access and utilize a particular set of resources as depicted in cone model Figures 2 and 4.

This paper also provides the concept for handling and provisioning of resources over the cloud and how it directly affects the workload for IoT platforms. The role of cone mode and inverted cone model can be used in solving the problem of a critical issue in hosting various services over the cloud. The solution for such a challenge can be done by adopting the proposed "Cone Model" & "Inverted Cone Model" algorithms for assigning an available set of resources

TABLE 2. Various symbol justification.

Symbol	Justification
N	distinct resources
R	random variable
S_i	service components
k	number of successful or grant resources instances
P	probability mass function

to requested services. It justifies how different services from various devices of IoT can be provisioned on the adoption of this proposed cone model. It also aligns with the applications in multifarious domains for resource sharing in massive data center.

To overwhelm the functionality of service computing, a discrete random variable stands as a better option to point out the probability aspect of resource allocation and utilization by multiple service components [14]. This helps to calculate the probability of obtaining critical resources by these service components. Such a process continues until each service component obtains or utilizes a complete set of resources to fulfill its entire execution. It also provides an aggregated probability for supervisory or parent service.

Assuming that there are 'N' distinct resources and that one service process needs to obtains or utilizes for the completion of its assigned task. It should independent of any previous selection from such allocation. A random variable '*R*' is such a number from resource instances that are required to be collected until one obtains a complete set of available resource instances, where k is successful or grant instance [7].

It follows the probability mass function to discrete and for each service component $S_{i_{1,2,3,\ldots,N}}$ the cumulative distribution function could be expressed as:

$$
\begin{aligned}
&= P\left(\bigcup_{i=1}^{N} S_i\right) \\
&\quad = \sum_i P(S_i) - \sum_{i_1<i_2}\sum P\left(S_{i_1}S_{i_2}\right) + \cdots\cdots\cdots\cdots \\
&\quad + (-1)^{k+1} \sum_{i_1<i_2<\ldots<i_k} \sum\sum P\left(S_{i_1}S_{i_2}\ldots..S_{i_k}\right) \\
&\quad + \cdots\cdots\cdots\cdots + (-1)^{N+1} \sum_{i_1<i_2<\ldots<i_N} \\
&\quad \times \sum\sum P\left(S_{i_1}S_{i_2}\ldots..S_{i_N}\right) \qquad (1)
\end{aligned}
$$

Now, S_i will invoke '*n*' resources collectively and that can be with the probability of $\frac{(N-1)}{N}$

Now, S_i will consume/utilize '*n*' resources collectively and that can be with the probability of $\frac{(N-1)}{N}$

Therefore, $P(S_i) = \left(\frac{N-1}{N}\right)^n$

Also, for service components $S_{i_1}S_{i_2}$ will utilize avail set of resources and that can be with the probability of

$$P\left(S_{i_1}S_{i_2}\right) = \left(\frac{N-2}{N}\right)^n$$

Same could be possible for,

$$P\left(S_{i_1}S_{i_2}\ldots..S_{i_k}\right) = \left(\frac{N-k}{N}\right)^n$$

$$P\left(S_{i_1}S_{i_2}\ldots..S_{i_k}\right) = \left(\frac{N-N}{N}\right)^n = 0$$

For N^{th} term,

For $R > n$ After substituting the above values in Equation 1, we obtained

$$= P\left(\bigcup_{i=1}^{N} S_i\right)$$

$$= N\left(\frac{N-1}{N}\right)^n - \binom{N}{2}\left(\frac{N-2}{N}\right)^n + \binom{N}{3}\left(\frac{N-3}{N}\right)^n - \ldots\ldots\ldots + (-1)^N\binom{N}{N-1}\left(\frac{1}{N}\right)^n$$

$$= \sum_{i=1}^{N}\binom{N}{i}\left(\frac{N-i}{N}\right)^n(-1)^{i+1} \qquad (2)$$

Now, each service component S_i where $i = 1, 2, \ldots.., N$ can also integrate with other service components [15]. The above probability can help us to determine if 'n' several resources are being fixed and random variables 'R' for any given set of devices in the IoT Platform. This random variable could provide–how each service can utilize or hold distinct types of resources in IoT devices for which the available set of resources is supported by a cumulative distribution function. Since one must utilize at least N resources to obtain a complete set that must follow:

$$P\{R > n\} = 1; if \quad n > N$$

Therefore, from equation 2, we can obtain an interesting possible set of resources:

$$= \sum_{i=1}^{N}\binom{N}{i}\left(\frac{N-i}{N}\right)^n(-1)^{i+1} = 1$$

This can be re-written as:

$$= \sum_{i=0}^{N}\binom{N}{i}\left(\frac{N-i}{N}\right)^n(-1)^{i+1} = 0$$

That can be multiple of

$$(-1)^N N^n \; and \; j = N - i \qquad (3)$$

$$= \sum\binom{N}{i} j^n (-1)^{j-1} = 0; \quad for \; 1 \le n < N$$

NOTE: By Binomial theorem, the probability sum is 1, i.e.

$$= \sum_{i=0}^{N}(ni)(p)^i(1-p)^{n-i} = [p + (1-p)]^n = 1 \qquad (4)$$

If R is a random variable with (n, p) as parameters whereas $0 < p < 1$, then as 'i' goes from $0 \; to \; N, p\{R = i\}$, first increase monotonically and then decrease monotonically reaching its largest value in the case where 'i' largest integer $for \; i \le (N+1)\,p$. The above proposition has been proved by several mathematicians.

Above mathematical expression could provide necessary information on–how a particular service S_i can obtain a complete set of resources/devices in IoT through its components $S_{i_1}S_{i_2}\ldots..S_{i_k}$. This brief is about the probability of a particular service can hold distinct and critical resources and how these resources could be allocated and utilized between multiple service components [24]. This gives the concept of the Cone Model that one service via its components could share or engage certain resources. Such value can be decided on a scale of zero to one with the selection of random variable R and parameter 'p' as shown in equation (4).

A. EXPECTATIONS IN RESOURCE HANDLING AND SERVICE ACCESSIBILITY

This mathematical expression helps us to calculate the estimated number of IoT resources/devices that could be stocked within an enterprise organization for smooth functioning [29], [31]. This could be done with help of ''Expectation or Expected Value'' in probability [7].

Definition: Suppose X is a discrete random variable with probability mass function $p(x)$, then expected value or expectation of X:

$$E[X] = \sum_{x:p(x)>0} xp(x)$$

It could be defined as the weighted average of the period value for X.

$$p(0) = \frac{1}{2} = p(1)$$

$$E[X] = 0\left(\frac{1}{2}\right) + 1\left(\frac{1}{2}\right) = \frac{1}{2}$$

Then,

We could formulate a mathematical model for deriving the estimated number of resources of a particular type that yield the highest performance probability. Here, several resource units are random variables having probability mass function $p(i)$; $i \ge 0$. This could also raise the demand for a particular resource in the purchase section for any organization in advanced. This helps us to determine the number of units of a particular resource that could be needed to better the functioning of the SOA system in service accessibility and expected profit.

Assume X denotes the number of units of a particular type of resource. It 'r' is available to set of resources in stock, then profit in terms of performance efficiency could be expressed as:

$$p(s) = aX - (r - X)b; \quad if \ \ X \leq r$$
$$= ar \quad if \ \ X > r$$

Therefore, the expected profit may be,

$$E[p(r)] = \sum_{i=0}^{r} [ai - (r-i)b] p(i) + \sum_{i=0}^{r} ra\, p(i)$$
$$= (a+b)\sum_{i=0}^{r} ip(i) - rb\sum_{i=0}^{r} p(i)$$
$$+ rb\left[1 - \sum_{i=0}^{r} p(i)\right]$$
$$= (a+b)\sum_{i=0}^{r} ip(i) - (a+b)\, r\sum_{i=0}^{r} p(i) + ra$$
$$= ra + (a+b)\sum_{i=0}^{r} (i-r)\, p(i) \tag{5}$$

It could analyze the situation whenever we increase the stock 'r' in the existing set of resources. This could also be done by replacing 'r' to '$(r+1)$'. This helps us to determine-what changes occurred in the performance efficiency of the overall SOA system [23]. We substitute $r = r+1$ in Equation 5, we obtain:

$$E[p(r+1)] = a(r+1) + (a+b)\sum_{i=0}^{r+1} (i-r-1)\, p(i)$$

Above equation would produce the same aspect for $i = 0\ to\ r$,

$$= a(r+1) + (a+b)\sum_{i=0}^{r} (i-r-1)\, p(i)$$

Therefore,

$$E[p(r+1)] - E[p(r)] = a - (a+b)\sum_{i=0}^{r} p(i) \tag{6}$$

Thus, an increase in the existing set of resources by one would have better results as compared to the current set of resources i.e. '$(r+1)$' would be Better than 'r' case.

B. Therefore, Equation 6, we could derive,

$$\sum_{i=0}^{r} p(i) < \frac{a}{(a+b)} \tag{7}$$

As seen from Equation 7; here, LHS is increasing in 'r', while RHS is constant [7]. So, this inequality would be satisfied for all values $r \leq r^*$, where r^* is a higher value of 'r' satisfying the above equation.

$$E[p(0)] < \cdots\cdots < E\left[p(r^*)\right]$$
$$< E\left[p(r^*+1)\right] > E\left[p(r^*+2)\right]$$
$$> \cdots\cdots \tag{8}$$

Algorithm 1 Calculating Throughput for Workflow in Service-Based Systems

S is a complete set of services such that $\{s|s$ is atomic service$\}$
For each $s \in S$
  α is the service cost
  R is the rate of service request
  A is % of allocated critical resources to s
  Let P_s be the throughput of s
  If $(R \leq A/\alpha)$ then $P_s = R$
  Else $P_s = A/\alpha$
  Throughput of workload $= Min\,(s \in S\,\{P_s\})$

It follows that an increase in the existing set of resource (r^*+1) items would generate maximize expected performance efficiency profit in the SOA system.

The above-mentioned algorithm (Algorithm 1) determines the throughput, especially for atomic service which cannot be split further into sub-services and treated as a single/isolated service [20]. These isolated services could form a composite service that is dependable on these atomic/isolated services [26]. In this algorithm, when an available set of critical resources is not bottleneck/exhaustive, then throughput P_s would be the rate of service request. Otherwise, throughput P_s would be allocated resources over service cost.

Algorithm 2 Formulating a Linear Programming to Allocate Critical Resources for Servers in a Service-Based System

W is the workflow in a service-based system such that $\{w|w$ is workload$\}$
Sv is available servers such that $\{sv \mid sv$ is server in service-based system$\}$
For each $w \in W$
Let throughput of w be TH_w and Priority of w be $\Pr_w$
  Linear Programming Objective function
  $Max\left(\sum_{w\in W} (TH_w \times \Pr_w)\right)$
  Let TR_w would be throughput-requirement of w
  Let SR_w is rate of service request
  For Task check $(TH_w \leq SR_w)$
    If $(TR_w \leq SR_w)$
    add Task $(TH_w \geq TR_w)$
For each $sv \in Sv$
  For each $w \in W$
  If w is atomic service directed from Sv
    Let C be ant constant, i.e. C = 0
    $C = C + \alpha \times TH_w$
AR_{sv} is the percentage (%) of the available set of critical resources of sv
    Add Task $(C \leq AR_{sv})$

The second algorithm presents the throughput relationship using Linear programming as done in [17]. This formulates the allocation of critical resources for server Sv. Here, the server makes avail the instance of resources $sv \in Sv$. These instances would be allocated to requested services based on priority $\Pr_w$ for workflow W. This gives new throughput based on the requirement TR_w that each instance sv could obtain with SR_w service request rate. For adding a task, throughput TH_w must be less than SR_w; otherwise, it would create a problem of a bottleneck for available resources to its critical point. To complete the overall workflow $w \in W$, each service s in sv used by w should be greater than the calculated constant C. This C is determined from a throughput TH_w based on service cost α as shown above.

Algorithm 3 Optimal Critical Resource Allocation Algorithm of Server in Service-Based System

In finding the optimal throughput for overall workflow in a service-based system
Let OT_w be optimal throughput of the overall workflow $w \in W$
For each $s \in S$
Let A_s be available critical resources allotted to s,
$A_s = 0$
For each workflow, w should be obtained s

$$A_s = A_s + \alpha \times OT_w$$

The third algorithm is for allocating critical resources optimally. This algorithm considered all given parameters that brief about workflow and server-related issues in the service-based system [27], [28]. This devises the optimal throughput OT_w for workflow which shows each service $s \in S$, must be completed by allocated A_s available critical resources.

The core algorithm is inherited from the work conducted in Ref [5] and Ref [6], which justified "Identifying and characterizing the cone model". In this work, the problem of modeling solar energetic particle (SEP) events is important to both space weather research and forecasting. Our proposed Cone Model/Inverted Cone Model also resolves the issue of service and resource provisioning for cloud-based IoT Platform by Calculating the throughput for workflow in service-based systems (Algo 1) and Allocating critical resources - Optimal (Algo 2 & Algo 3).

B. PROPOSED ALGORITHM FOR CRITICAL RESOURCE ALLOCATION IN SOA SYSTEM

Based on the mathematical view and previous work done in [25], [33], allocating available resources could be done with the help of this proposed cone model. This model not only determines the overload state of any SOA system but also rectifies the provision of available critical resources. Allocation of critical resource algorithm using the cone mode is provided below:

Algorithm 4 Allocation of Critical Resources Using Cone Model in SOA System

W is the workflow in SOA system such that $\{w|w$ is workload$\}$
Sv is available servers such that $\{sv \mid sv$ is server instances in SOA system$\}$
A_s is an available set of critical resources and O_s is obtained set of critical resources by $s \in S$
Let SR_w rate of service request
α is incurring service costs to make available resources
For each instance of server $sv \in Sv$
For adding new task, check $(SR_w \leq A_s)$
If w has already occupied sv instances of server
Then $A_s = A_s - O_s$ and increment the associated *flag for* that service s
Repeat until each $w \in W$
If $(O_s \geq A_s)$
Then, % of available critical resources gets overloaded
Else
Critical resources could be allotted to s for complete its execution
For each atomic service s in sv used by w
Rise value of *flag* per each allocation of instance of server sv to $s \in S$
Build formation of Cone with this *flag* indication F_c
If $(F_c \leq A_s)$
Then, SOA system would be efficient enough to handle service request SR_w
Else
SOA system may be overloaded at server side

The above proposed algorithm would be best to handle available critical resources in SOA system. With use of flag for each service $s \in S$, specifies the obtained set of resource O_s and available set of resources A_s. If percentage of available set of resources get drained out, i.e. $(O_s \geq A_s)$ then new task assignment must be in waited queue/postponed for some time. This shows that if flag counter for each atomic service s went up to a permissible limit which is bottleneck condition for particular resource, flag must be set to its critical point, which is critical resource valuation to those particular resources. This condition must be check $(F_c \leq A_s)$ for each time allotment of instance of server resources to $s \in S$.

V. RESULTS AND ANALYSIS

This section comprises the experimental study of services with their associated resources that are engaged by multiple service components. In this respect, the cloud computing system is best to experiment with resource utilization and sharing policies with client-server methodology [25]. This involves data centers (DC) in different regions of the world and a user base (UB). DC contains all infrastructural resources like several processors, memory, storage space, available bandwidth, and virtualization policy. UB consists of fields like several requester/clients, peak hours, etc. DC contains Linux-based OS and "Xen" as Virtual Machine Model, which helps to establish virtualization in resources for being shared.

This involves the auto-loading and load balancing between multiple DC. The corresponding plots (as Fig.6–Fig.11) justify the increase in the demand from specific UB on different DCs could be standard by adopting the cone model.

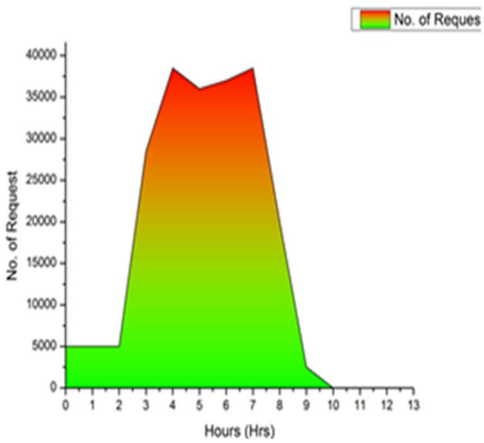

FIGURE 6. Loading for data center for #DC01.

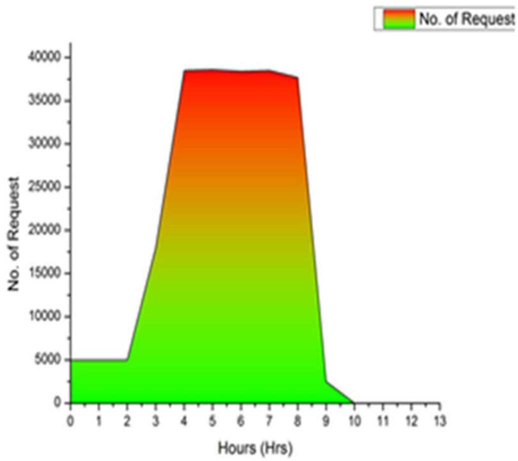

FIGURE 7. Loading for data center #DC02.

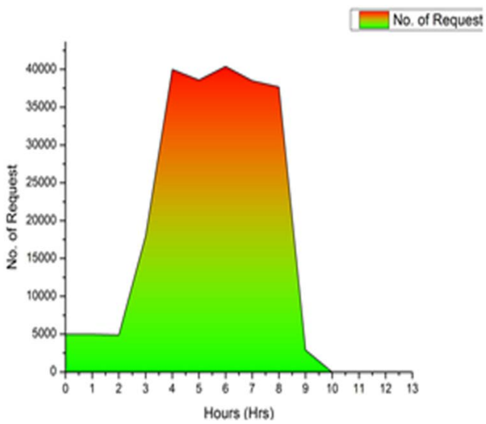

FIGURE 8. Loading for data center #DC03.

Figs. 6, 7 & 8 show almost the same number of requests per hour as it was placed in the Asia region and all these DCs have a similar number of resources. The maximum number of requests per hour is 40,000 in all three DCs. These requests have been generated by its corresponding UBs in that region, leading to these DCs having similar loading metrics. The increase in the demand from specific UB on different DCs could be standard by adopting the cone model. The shape

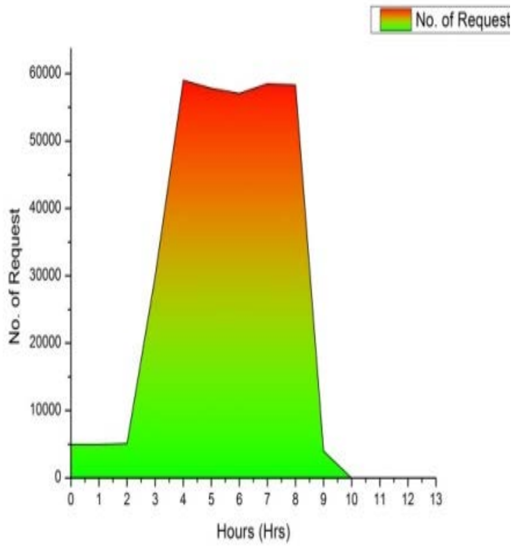

FIGURE 9. Loading for data center #DC04.

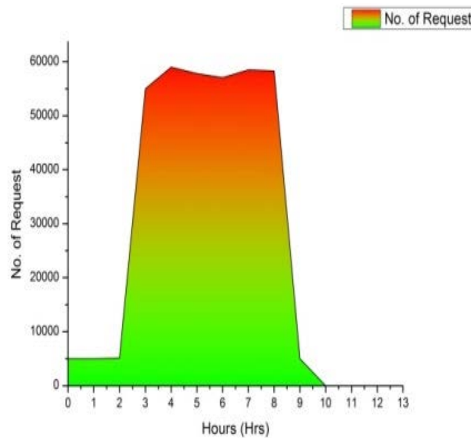

FIGURE 10. Loading for data center #DC07.

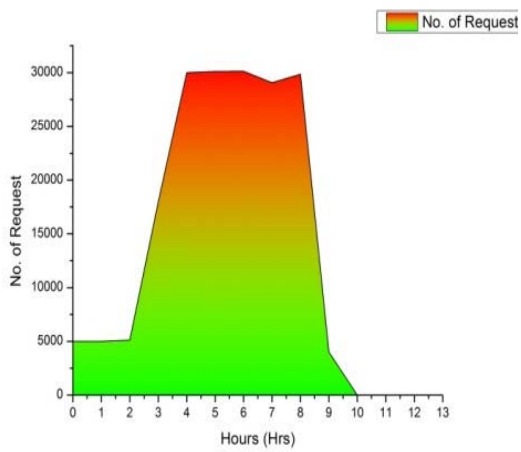

FIGURE 11. Loading for data center #DC06.

of the cone in the above figure justifies the requirement that each resource had been handled with the proposed algorithms discussed in sub-section 4 (B).

Fig. 9 & 10 demonstrate the loading for data centers #DC04 and #DC07 respectively that are placed in a region, where the number of users is high but has a smaller number of resources. In this scenario, DC is highly loaded and as a consequence, has a large number of requests per hour (reaching a maximum of 60,000 requests per hour).

Fig. 11 illustrates the loading scenario for DC#06, which is the place in a region, where users are fewer and have a higher number of resources. This would generate a smaller number of requests per hour (reaching a maximum of 30,000 requests per hour). This is just the reverse of the previous case explained in Fig. 9 & 10.

The above-explained result shows the compatibility of the cone model on resource allocation and utilization perspectives. As the number of user requests increases, the load on data centers also got increased, and that in turn would also raise the situation critical resources allocation as discussed in algorithms 2 and 3. It brings out an important aspect of service integration along with a critical resource allocation algorithm in the SOA system. This would create a logical cone shape in the perception of handling and allocation of resources by multiple services components as depicted in Fig. 2.

Here, the shape of a cone, i.e., logical & virtualized, could brief about the throughput of the SOA system and also discuss the allocation scheme with resources through multiple service components. Therefore, the changes in the shape of the cone depend on the rate of service requests. This relationship can also deduce as the sharper the cone shape, the larger would be its performance, and on the other hand, the larger the cone shape, the lower its performance. Hence, if the service requester rate is more, then it would be more requirements related to resource allocation. This allocation may result in adverse or better performance of the system according to the situation.

There have been certain limitations that can be considered while adopting this cone model for resource provisioning. While handling different types of resources from different data centers, there has been a situation that arises due to lack of availability or denial of services associated with resource instance allocation or assignment. This cone model has such limitations based on the availability of resource instances.

VI. CONCLUSION

This paper discusses the role of SOA with its logical view reference architecture and presents the various challenges for resource allocation in SOA systems, especially for IoT platforms. Also, focuses on the need of clients over runtime and different workloads with its extensive survey of the Internet of Things (IoT). This paper proposes the cone model that helps in the efficient handling and provisioning of resources in computing over IoT. Also, deduce the mathematical implication of the cone model in resource provisioning and service accessibility with IoT adopted the system. It also discusses the algorithms for calculating throughput and formulating optimization ways to allocate critical resources in the service-based system for IoT devices.

For simulation purposes, this paper experiments with the proposed algorithm based on the Cone Model on different data center resources with a variable number of service requests at a variable rate. With an increase in the number of resources using its instances with particular DC and client requests, there would be moderate growth in requests per hour using the proposed model (which is 32,000 as the average service request rate in existing work). This shows system performance could be above moderate. Similarly, in situations where the number of resources within a particular DC is large and generated client requests are less in number, system performance could be high with fewer resources allocation and utilization. In a condition where client requests are higher in number along with a lesser number of resources, system performance is degradable with excess utilization of resources.

ACKNOWLEDGMENT

The authors appreciate Taif University Researchers Supporting Project TURSP 2020/345, Taif University, Saudi Arabia for supporting this work.

REFERENCES

[1] G. Schmalzhaf, "Resource management in SOA environments with adaptive computing virtualization," SAP Community Netw., SAP, Walldorf, Germany, Tech. Rep., 2008.

[2] N. Niknejad, W. Ismail, I. Ghani, B. Nazari, M. Bahari, and A. R. B. C. Hussin, "Understanding service-oriented architecture (SOA): A systematic literature review and directions for further investigation," *Inf. Syst.*, vol. 91, Jul. 2020, Art. no. 101491.

[3] A. Bamhdi, "Requirements capture and comparative analysis of open source versus proprietary service oriented architecture," *Comput. Standards Interfaces*, vol. 74, Feb. 2021, Art. no. 103468.

[4] H. Xie, L. Ofman, and G. Lawrence, "Cone model for halo CMEs: Application to space weather forecasting," *J. Geophys. Res.*, vol. 109, pp. 1–13, 2004, doi: 10.1029/2003JA010226.

[5] J. G. Luhmann, S. A. Ledvina, D. Odstrcil, M. J. Owens, X.-P. Zhao, Y. Liu, and P. Riley, "Cone model-based SEP event calculations for applications to multipoint observations," *Adv. Space Res.*, vol. 46, no. 1, pp. 1–21, Jul. 2010.

[6] The National Aeronautics and Space Administration (NASA), USA. (May 16, 2022). *Model Rocket Engine Performance*. [Online]. Available: http://www.grc.nasa.gov/WWW/k-12/rocket/rktengperf.html

[7] S. Ross, *A First Course in Probability*. Upper Saddle River, NJ, USA: Pearson, 2002.

[8] X. Liu, S. Wang, Y. Ma, Y. Zhang, Q. Mei, Y. Liu, and G. Huang, "Operating systems for resource-adaptive intelligent software: Challenges and opportunities," *ACM Trans. Internet Technol.*, vol. 21, no. 2, pp. 1–19, Mar. 2021.

[9] O. Ascigil, A. Tasiopoulos, T. K. Phan, V. Sourlas, I. Psaras, and G. Pavlou, "Resource provisioning and allocation in function-as-a-service edge-clouds," *IEEE Trans. Services Comput.*, early access, Jan. 18, 2021, doi: 10.1109/TSC.2021.3052139.

[10] A. Y. Nikravesh, S. A. Ajila, and C.-H. Lung, "An autonomic prediction suite for cloud resource provisioning," *J. Cloud Comput.*, vol. 6, no. 1, pp. 1–20, Dec. 2017.

[11] N. D. Nguyen, L.-A. Phan, D.-H. Park, S. Kim, and T. Kim, "ElasticFog: Elastic resource provisioning in container-based fog computing," *IEEE Access*, vol. 8, pp. 183879–183890, 2020.

[12] F. H. Khoso, A. Lakhan, A. A. Arain, M. A. Soomro, S. Z. Nizamani, and K. Kanwar, "A microservice-based system for industrial Internet of Things in fog-cloud assisted network," *Eng., Technol. Appl. Sci. Res.*, vol. 11, no. 2, pp. 7029–7032, Apr. 2021.

[13] X. Cao, G. Tang, D. Guo, Y. Li, and W. Zhang, "Edge federation: Towards an integrated service provisioning model," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1116–1129, Jun. 2020.

[14] X. Liu and R. Buyya, "Resource management and scheduling in distributed stream processing systems: A taxonomy, review, and future directions," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–41, May 2021.

[15] R. Singh, M. Singh, S. Garg, I. Perl, O. Kalyonova, and A. Penskoi, "Multicriteria decision making based optimum virtual machine selection technique for smart cloud environment," *J. Ambient Intell. Smart Environ.*, vol. 13, no. 3, pp. 185–199, 2021.

[16] M. M. Badawy, Z. H. Ali, and H. A. Ali, "QoS provisioning framework for service-oriented Internet of Things (IoT)," *Cluster Comput.*, vol. 23, no. 2, pp. 575–591, Jun. 2020.

[17] X. Li and L. D. Xu, "A review of Internet of Things—Resource allocation," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8657–8666, Jun. 2021.

[18] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Towards end-to-end resource provisioning in fog computing over low power wide area networks," *J. Netw. Comput. Appl.*, vol. 175, Feb. 2021, Art. no. 102915.

[19] S. Venugopal, X. Chu, and R. Buyya, "A negotiation mechanism for advance resource reservation using the alternate offers protocol," in *Proc. 16th Int. Workshop Quality Service (IWQoS)*, 2008, pp. 40–49.

[20] B. Wang, C. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, "A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds," *Cluster Comput.*, vol. 23, no. 4, pp. 2809–2834, Dec. 2020.

[21] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, "Resource provisioning for IoT services in the fog computing environment: An autonomic approach," *Comput. Commun.*, vol. 161, pp. 109–131, Sep. 2020.

[22] S. S. Yau and H. G. An, "Adaptive resource alocation for service-based systems," *Int. J. Softw. Informat.*, vol. 3, no. 4, pp. 483–499, 2009.

[23] C.-W. Tsai, "SEIRA: An effective algorithm for IoT resource allocation problem," *Comput. Commun.*, vol. 119, pp. 156–166, Apr. 2018.

[24] M. Hammoudeh, G. Epiphaniou, S. Belguith, D. Unal, B. Adebisi, T. Baker, A. S. M. Kayes, and P. Watters, "A service-oriented approach for sensing in the Internet of Things: Intelligent transportation systems and privacy use cases," *IEEE Sensors J.*, vol. 21, no. 14, pp. 15753–15761, Jul. 2021.

[25] B. Fei, X. Zhu, D. Liu, J. Chen, W. Bao, and L. Liu, "Elastic resource provisioning using data clustering in cloud service platform," *IEEE Trans. Services Comput.*, early access, Jun. 16, 2020, doi: 10.1109/TSC.2020.3002755.

[26] M. P. Yadav, N. Pal, and D. K. Yadav, "Resource provisioning for containerized applications," *Cluster Comput.*, vol. 24, pp. 1–22, May 2021, doi: 10.1007/s10586-021-03293-5.2021.

[27] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, "Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification," *Sensors*, vol. 21, no. 5, p. 1832, Mar. 2021.

[28] S. Challita, F. Korte, J. Erbel, F. Zalila, J. Grabowski, and P. Merle, "Model-based cloud resource provisioning with TOSCA and OCCI," *Softw. Syst. Model.*, vol. 20, no. 5, pp. 1609–1631, 2021.

[29] L. Zhao, B. Cheng, and J. Chen, "Service-oriented IoT resources access and provisioning framework for IoT context-aware environment," in *Proc. IEEE World Congr. Services (SERVICES)*, Oct. 2020, pp. 245–251.

[30] V. Conti, C. Militello, L. Rundo, and S. Vitabile, "A novel bio-inspired approach for high-performance management in service-oriented networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 1709–1722, Oct. 2021, doi: 10.1109/TETC.2020.3018312.

[31] S. Bansal and D. Kumar, "IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 3, pp. 340–364, Sep. 2020.

[32] A. K. Tripathi, S. Agrawal, and R. D. Gupta, "Cloud enabled SDI architecture: A review," *Earth Sci. Informat.*, vol. 13, no. 2, pp. 211–231, Jun. 2020.

[33] Z. Zhou, S. Yu, W. Chen, and X. Chen, "CE-IoT: Cost-effective cloud-edge resource provisioning for heterogeneous IoT applications," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8600–8614, Sep. 2020.

[34] S. Deng, Z. Xiang, P. Zhao, J. Taheri, H. Gao, J. Yin, and A. Y. Zomaya, "Dynamical resource allocation in edge for trustable Internet-of-Things systems: A reinforcement learning method," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6103–6113, Sep. 2020.

[35] R. S. Bhadoria, N. S. Chaudhari, and G. S. Tomar, "The performance metric for enterprise service bus (ESB) in SOA system: Theoretical underpinnings and empirical illustrations for information processing," *Inf. Syst.*, vol. 65, pp. 158–171, Apr. 2017.

[36] R. Bavaresco, J. Barbosa, H. Vianna, P. Büttenbender, and L. Dias, "Design and evaluation of a context-aware model based on psychophysiology," *Comput. Methods Programs Biomed.*, vol. 105299, p. 189, Jun. 2020.

[37] S. Chauhan, M. Mittal, M. Woźniak, S. Gupta, and R. P. De Prado, "A technology acceptance model-based analytics for online mobile games using machine learning techniques," *Symmetry*, vol. 13, no. 8, p. 1545, Aug. 2021.

ROBIN SINGH BHADORIA (Member, IEEE) received the Bachelor of Engineering (B.Eng.) and Master of Technology (M.Tech.) degrees in computer science and engineering from Rajiv Gandhi Technological University (RGTU), Madhya Pradesh, India, in 2008 and 2011, respectively, and the Ph.D. degree from the Discipline of Computer Science and Engineering, Indian Institute of Technology (IIT) Indore, Madhya Pradesh, in January 2018. He is currently working as an Assistant Professor with the Department of Computer Engineering and Applications, GLA University, Mathura, India. His current research interests include big data analytics, service-oriented architecture, the Internet of Things, and wireless sensor networks. He has been awarded with Gold Medal in M.Tech. degree program at state level.

HATIM GHAZI ZAINI received the bachelor's degree in electrical and computer engineering from the Faculty of Engineering, Umm Al-Qura University, Mecca, Saudi Arabia, in 1993, and the master's and Ph.D. degrees from the Florida Institute of Technology, Melbourne, FL, USA, in 1998 and 2003, respectively. In 2005, he joined the Faculty of Applied Computer Sciences, Umm Al-Qura University, as an Assistant Professor. He became a Supervisor of Cisco Networking Academy, in 2007, and a member of the College Committee of Quality Control and Quality Assurance, the College Higher Committee of Consultation, and the Technical College at Makkah, in 2008, and the Department of Computer Technology, Saudi Arabia, in 2009. From June 2011 to January 2013, he became an Assistant Professor and an Associate Professor with the Department of Electrical Engineering, Taif University, Saudi Arabia. His research interests include programming, computation algorithms, and real time control systems.

MD. MANZAR NEZAMI was born in Samastipur, Bihar, India, in 1984. He received the B.Sc. (Engineering) degree in electronics and communication from Magadh University, Bodh Gaya, India, in 2009, the M.E. degree in instrumentation and control engineering from the National Institute of Technical Teacher's Training and Research, Chandigarh, in 2014, and the Ph.D. degree in instrumentation and control engineering from Jamia Millia Islamia (A Central University), New Delhi, India, in 2020. He was with the Department of Electronics and Communication Engineering, Siwan Engineering and Technical Institute, J. P. University, Chapra, from 2009 to 2011. Currently, he is working with the Department of Electronics and Communication Engineering, GLA University, Mathura, India. His research interests include condition monitoring of transformer's insulation, thin film sensor fabrication with their signal conditioning circuits, and application of soft-computing techniques.

MD. DANISH EQUBAL received the B.Tech. degree in electrical engineering, the M.Tech. degree in power system, and the Ph.D. degree in electrical engineering from Jamia Millia Islamia, New Delhi, India, in 2006, 2009, and 2019, respectively. Currently, he is an Associate Professor with the Electrical Engineering Department, Galgotias College of Engineering and Technology, Greater Noida, India. His research interests include insulation condition monitoring using dissolved gas analysis (DGA) of transformer oil and application of soft-computing techniques.

ELFATIH ELMUBARAK MUSTAFA received the bachelor's degree from the Sudan University of Science and Technology, in 2003, the master's degree from the College of Computer Science and Information Technology, Al-Neelain University, in 2008, and the Ph.D. degree from the College of Graduate Studies, Al-Neelain University, in 2013. His research interests include E-government's applications, artificial intelligence, and big data.

SHERIF S. M. GHONEIM (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Faculty of Engineering at Shoubra, Zagazig University, Egypt, in 1994 and 2000, respectively, and the Ph.D. degree in electrical power and machines from the Faculty of Engineering, Cairo University, in 2008. From 2005 to 2007, he was a Guest Researcher with the Institute of Energy Transport and Storage (ETS), University of Duisburg, Essen, Germany. After that, he joined the Department of Electrical Engineering, Faculty of Engineering, Taif University, as an Associate Professor. Since 1996, he has been teaching with the Faculty of Industrial Education, Suez Canal University, Egypt. His research interests include grounding systems, dissolved gas analysis, breakdown in SF6 gas, and AI technique applications.